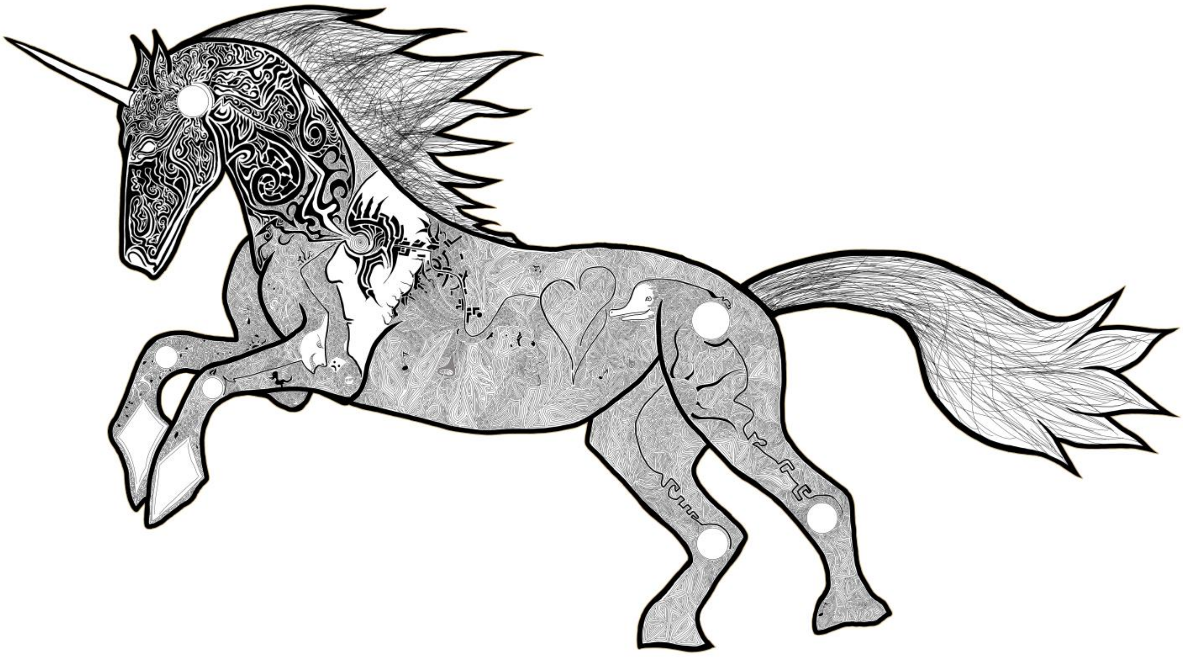
Annals of Computer Science and Information Systems
Volume 35

Proceedings of the 18th Conference on Computer Science and Intelligence Systems

September 17–20, 2023. Warsaw, Poland



Maria Ganzha, Leszek Maciaszek, Marcin Paprzycki,
Dominik Ślęzak (eds.)

Annals of Computer Science and Information Systems, Volume 35

Proceedings of the 18th Conference on Computer Science and Intelligence Systems

Maria Ganzha, Leszek Maciaszek, Marcin Paprzycki, Dominik Ślęzak (eds.)

2023, Warszawa,
Polskie Towarzystwo
Informatyczne

2023, New York City,
Institute of Electrical and
Electronics Engineers

Annals of Computer Science and Information Systems, Volume 35

Proceedings of the 18th Conference on Computer Science and Intelligence Systems

ART: ISBN 978-83-969601-0-8, IEEE Catalog Number CFP2385N-ART
USB: ISBN 978-83-967447-9-1, IEEE Catalog Number CFP2385N-USB
WEB: ISBN 978-83-967447-8-4

ISSN 2300-5963
DOI 10.15439/978-83-967447-8-4

Ul. Solec 38/103
00-394 Warsaw, Poland

Contact: secretariat@fedcsis.org
http://annals-csis.org/

Cover art: Jednorożec (unicorn)
Karol Jaworski,
Elbląg, Poland

Also in this series:

Volume 37: Communication Papers of the 18th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-969601-3-9, ISBN USB: 978-83-969601-4-6**
Volume 36: Position Papers of the 18th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-969601-1-5, ISBN USB: 978-83-969601-2-2**
Volume 34: Proceedings of the Third International Conference on Research in Management and Technovation **ISBN 978-83-965897-8-1**
Volume 33: Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering, **ISBN WEB: 978-83-965897-6-7, ISBN USB: 978-83-965897-7-4**
Volume 32: Communication Papers of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-965897-4-3, ISBN USB: 978-83-965897-5-0**
Volume 31: Position Papers of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-965897-2-9, ISBN USB: 978-83-965897-3-6**
Volume 30: Proceedings of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-962423-9-6, ISBN USB: 978-83-965897-0-5**
Volume 29: Recent Advances in Business Analytics. Selected papers of the 2021 KNOWCON-NSAIS workshop on Business Analytics **ISBN WEB: 978-83-962423-7-2, ISBN USB: 978-83-962423-6-5**
Volume 28: Proceedings of the 2021 International Conference on Research in Management & Technovation, **ISBN WEB: 978-83-962423-4-1, ISBN USB: 978-83-962423-5-8**
Volume 27: Proceedings of the Sixth International Conference on Research in Intelligent and Computing in Engineering, **ISBN WEB: 978-83-962423-2-7, ISBN USB: 978-83-962423-3-4**
Volume 26: Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-959183-9-1, ISBN USB: 978-83-962423-0-3**
Volume 25: Proceedings of the 16th Conference on Computer Science and Intelligence Systems, **ISBN Web 978-83-959183-6-0, ISBN USB 978-83-959183-7-7, ISBN ART 978-83-959183-8-4**
Volume 24: Proceedings of the International Conference on Research in Management & Technovation 2020, **ISBN WEB: 978-83-959183-5-3, ISBN USB: 978-83-959183-4-6**

DEAR Reader, it is our pleasure to present to you Proceedings of the 18th Conference on Computer Science and Intelligence Systems (FedCSIS 2023), which took place on September 17-20, 2023, in Warsaw, Poland.

FedCSIS 2023 was chaired by Jarosław Arabas and Sławomir Zadrożny, while Przemysław Biecek acted as the Chair of the Organizing Committee. This year, FedCSIS was organized by the Polish Information Processing Society (Mazovia Chapter), IEEE Poland Section Computer Society Chapter, Systems Research Institute of Polish Academy of Sciences, as well as Faculty of Electronics and Information Technology and Faculty of Mathematics and Information Sciences of Warsaw University of Technology.

FedCSIS 2023 was technically co-sponsored by IEEE Poland Section, IEEE Czechoslovakia Section Computer Society Chapter, IEEE Poland Section Systems, Man, and Cybernetics Society Chapter, IEEE Poland Section Computational Intelligence Society Chapter, Committee of Computer Science of Polish Academy of Sciences, and Mazovia Cluster ICT. Moreover, two years ago, the FedCSIS conference series formed strategic alliance with QED Software, a Polish software company developing AI-based products, and this collaboration has been continued.

FedCSIS 2023 was sponsored by QED Software, Samsung, Hewlett Packard Enterprise, Łukasiewicz Research Network – Institute of Innovative Technologies EMAG, MDPI, Sages, Efigo, and CloudFerro.

This year, another round of evolutionary adaptations, to the FedCSIS conference series, took place. Specifically, they concerned the conference structure. Post conference publications (Proceedings, Position Papers and Communication Papers volumes) illustrate the direction of this evolutionary process. In short, starting from 2023, FedCSIS conferences have Main Track with five Topical Areas, Thematic Tracks and, possibly, Competitions. The new structure emphasizes the integrity of the conference. For all five Topical Areas, situated within a general domain of Computer Science, the continually emerging topic of Intelligence Systems, stands as the common denominator. All Thematic Tracks refer to Intelligence Systems as well, from different perspectives. Even Competitions, having strong roots in the realm of AI, data science, machine learning, computer vision, and natural language processing, are regarded as a path toward introducing more Intelligence into Computer Science and IT.

In this context, these Proceedings consist of six parts. Part 1 contains Invited Contributions. Part 2 collects Main Track full contributions (arranged alphabetically, according to the last name of the first author, with Topical Area represented in the metadata). Part 3 contains Main Track short contributions. Part 4 contains full contributions, originating from Thematic Tracks. (Again, texts are arranged alphabetically, according to the last name of the first author, with the name of Thematic Track stated in the metadata.) Part 5 collects short papers from all Thematic Tracks. Finally, Part 6 is devoted to Competitions that run within the context of FedCSIS conferences.

Keeping this in mind, let us now introduce Keynote Speakers, the remaining Invited Contributions, and the five Topical Areas of FedCSIS 2023 Main Track.

I. INVITED CONTRIBUTIONS

FedCSIS 2023 invited four keynote lecturers to deliver lectures matching Topical Areas and thus providing a broader context for the conference participants. Moreover, two past FedCSIS keynote speakers have been invited to prepare contributions, which refer to the core focus of the conference series. There are also two contributions corresponding to the additional invited talks and three contributions corresponding to tutorials.

The aforementioned common denominator of all FedCSIS Topical Areas is clearly visible in this part. First, we can see here the core AI works on trustworthiness and robustness of neuro-symbolic and neural network models. Second, there are contributions related to the foundations of intelligent decision making and uncertainty modeling, using tools taken e.g. from soft computing and information theory. On the other hand, we have also contributions referring to machine learning and data science software, as well as examples of applications of AI methods in practical domains. This generally reflects our understanding of the place of Intelligence Systems in the realm of Computer Science. Namely, according to our vision, it is to develop and adjust the AI-related methods to let them work efficiently as the essential component of modern software systems and solutions.

II. ADVANCED ARTIFICIAL INTELLIGENCE IN APPLICATIONS

This Topical Area is a conceptual continuation of a series of international AAIA Symposiums, which have been held since 2006. It aims at covering wide range of core aspects of AI. Nowadays, AI is usually perceived as closely related to the data, therefore, the scope of this Topical Area includes elements of machine learning, data science, and big data processing, with important emerging aspects such as interactive learning and human-centered AI, as well as interpretable learning, explainable AI, and the aforementioned topic of trustworthiness. Furthermore, since the realm of AI is far richer, the ultimate goal of this Topical Area is to show relationships between all of AI subareas, emphasizing a cross-disciplinary nature of various research branches. In 2023, the collection of papers accepted to this Topical Area has reflected this cross-disciplinary nature particularly well. We can see here various areas of AI (also outside so-called “core AI”), as well as a mix of theoretical and practical contributions. From the perspective of the general scope of FedCSIS, this Topical Area embraces particularly AI methods and examples of their applications in different practical fields.

This Topical Area is curated by:

+ Corizzo, Roberto, American University, USA
+ Sosnowski, Łukasz, Systems Research Institute of Polish Academy of Sciences, Poland
+ Szczuka, Marcin, University of Warsaw, Poland
+ Zdravevski, Eftim, Ss. Cyril and Methodius University, Macedonia

III. Computer Science & Systems

This Topical Area aims at integrating and creating synergy between Computer Science and related disciplines, with AI being of the core interest. The area's scope spans themes ranging from hardware issues close to computer engineering via software issues tackled by the theory and applications of Computer Science. When compared to the previously-discussed Topical Area on "Advanced Artificial Intelligence in Applications", herein we are interested more in software system realizations and computational aspects. Therefore, we make a step from AI regarded as the set of methods towards Intelligence Systems understood as software systems with the elements of AI. As an example, the domains such as reinforcement learning or AI-based games and simulations are studied here not only from the perspective of the quality of obtained results but also taking into account their performance, resource consumption, and scalability.

This Topical Area is curated by:

+ Casalino, Gabriella, University of Bari "Aldo Moro", Italy
+ Ducange, Pietro, University of Pisa, Italy
+ Pawłowski, Wiesław, University of Gdańsk, Poland
+ Świechowski, Maciej, QED Software, Poland
+ Wasielewska-Michniewska, Katarzyna, Systems Research Institute of Polish Academy of Sciences, Poland

IV. Network Systems & Applications

Modern network systems encompass a wide range of solutions and technologies, including wireless and wired networks, network systems, services, and applications. On the one hand, network technologies are used in majority of areas that make human life easier and more comfortable. On the other hand, the rapid need for network deployment brings new challenges in network management and network design, which are reflected in hardware, software, services, and security-related problems. Going back to the main scope of FedCSIS, it is obvious that appropriate network solutions are one of the crucial layers of scalable modern software systems, including those with the elements of AI. On the other hand, equally obviously, AI methods can be useful to make network systems and their applications more efficient. Accordingly, the aim of this Topical Area is to bring more Intelligence into network systems. Moreover, besides network systems, one should think also about network models, network algorithms, etc. Therefore, this Topical Area covers not only the technological side, but also the societal and social impacts of network developments.

This Topical Area is curated by:

+ Armando, Alessandro, University of Genova, Italy
+ Awad, Ali Ismail, United Arab Emirates University, United Arab Emirates
+ Furtak, Janusz, Military University of Technology, Poland
+ Hodoň, Michal, University of Žilina, Slovakia
+ Suri, Niranjan, Institute of Human and Machine Cognition, United States

V. Information Technology for Business & Society

The aim this Topical Area is to integrate and create synergy between disciplines of IT, Intelligence Systems, and social sciences. Collected contributions address issues relevant to IT and necessary for practical, everyday needs of business, other organizations and society at large. Moreover, they take a socio-technical view on Intelligence Systems and, at the same time, relate to ethical, social and political issues that they raise. Thus, from the viewpoint of the FedCSIS as a whole, this Topical Area goes beyond Computer Science itself. It refers to the fact that every software system or solution, and especially a system or solution with some flavors of Intelligence, needs to be carefully deployed in real life. In other words, it is not only about machines – it is also about humans. Accordingly, this Topical Area embraces particularly research on methods and processes of adoption of AI and Intelligence Systems in society and particular markets of business applications. Going back to one of the aforementioned invited contributions, the means for trustworthiness can be regarded as an important tool in such processes too.

This Topical Area is curated by:

+ Cano, Alberto, Virginia Commonwealth University, Richmond, United States
+ Dias, Gonçalo, University of Aveiro, Portugal
+ Miller, Gloria, Maxmetrics, Germany
+ Naldi, Maurizio, LUMSA University, Italy
+ Wątróbski, Jarosław, University of Szczecin, Poland
+ Ziemba, Ewa, University of Economics in Katowice, Poland

VI. Software, System & Service Engineering

For decades, an open question in the software industry remains, how to provide fast and effective software process and software services, and how to come to software systems, embedded systems, autonomous systems, or cyber-physical systems that will address the open issue of supporting information management process in many, particularly complex organization systems. Even more, it is a hot issue how to provide a synergy between systems in common, and software services as a mandatory component of each modern organization, particularly in terms of IoT, big data, and Industry 4.0 paradigms. Therefore, the main goal of this Topical Area is to address open questions and real potentials for various applications of modern approaches and technologies to develop and implement effective software services in a support of information management and system engineering. We can see here a clear linkage to AI and Intelligence Systems as well. On the one hand (going back to one of invited talks reported in Invited Contributions), AI tools can be applied to improve the quality of software and to optimize the performance of computer systems. While on the other hand (going back again to one of the papers associated with the FedCSIS 2023 keynote lectures), AI-based models and algorithms need to be tested, maintained and monitored just like any other components of complex software systems.

This Topical Area is curated by:

+ Luković, Ivan, University of Belgrade, Serbia

- + Kolukısa Tarhan, Ayça, Hacettepe University, Turkey
- + Mernik, Marjan, University of Maribor, Slovenia
- + Popović, Aleksandar, University of Montenegro, Podgorica, Montenegro

VII. Zdzisław Pawlak Awards

The above-described five Topical Areas of FedCSIS Main Track reflect five fundamental aspects of understanding, developing, and applying Intelligence Systems. This topical integrity is emphasized by the Professor Zdzisław Pawlak award, considered in four categories: Best Paper, Young Researcher, Industry Cooperation, and International Cooperation. Over time, this award has gone through significant evolution together with the FedCSIS conference series. Originally, it was granted only to papers published within the series of AAIA Symposiums. However, although Professor Zdzisław Pawlak has been often recognized as "the father of Polish AI", his research achievements have gone far beyond AI itself, in particular, toward AI applications and Intelligence Systems as we mean them. Accordingly, over time, we decided to expand this award to the whole conference – not only all Main Track Topical Areas but also all Thematic Tracks.

This year, Award Committee (a part of FedCSIS Senior Program Committee) had a particularly hard task to select a single winner of Best Paper Award. Therefore, after discussion, additional paper was distinguished with Distinction Award. The following contributions have been awarded:

- In the category **Best Paper:** Julian Premm, Hagen Peukert, Dennis Rössel, Mareike Silber, for the paper „Analysis of a GPT-3 chatbot with respect to its input in a sales dialogue"
- Additional **Distinction Award:** Giovanna Castellano, Pasquale De Marinis, Gennaro Vessio, for the paper „Applying Knowledge Distillation to Improve Weed Mapping With Drones"
- In the category **Young Researcher:** Anastasiya Danilenka, for the paper „Mitigating the effects of non-IID data in federated learning with a self-adversarial balancing method"
- In the category **Industry Cooperation:** Mikołaj Pudo, Mateusz Wosik, Artur Janicki, for the paper „Open Vocabulary Keyword Spotting with Small-Footprint ASR-based Architecture and Language Models"
- In the category **International Cooperation Award:** Samaneh Mohammadi, Mohammadreza Mohammadi, Sima Sinaei, Ehsan Nowroozi, Francesco Flammini, Mauro Conti, for the paper „Balancing Privacy and Accuracy in Federated Learning for Speech Emotion Recognition"

Industry Cooperation Award was sponsored by QED Software, International Cooperation Award – by MDPI, while the remaining awards were sponsored by Mazovia Branch of Polish Information Processing Society.

Here, it is also worth noting that each of those five papers comes from a different Topical Area or Thematic Track. Yet, they are all aligned with what we described before as the FedCSIS common denominator.

VIII. Statistics

Each contribution, found in this volume, was refereed by at least two referees and the acceptance rate of regular full papers was approximately 19% (68 accepted contributions, out of 358 general submissions).

IX. Committees

The Senior Program Committee of FedCSIS 2023 consisted of:

- van der Aalst, Wil, RWTH Aachen University, Germany
- Alba, Enrique, University of Málaga, Spain
- Aiello, Marco, University of Stuttgart, Germany
- Armando, Alessandro, University of Genova, Italy
- Atiquzzaman, Mohammed, University of Oklahoma, Norman, USA
- Awad, Ali Ismail, United Arab Emirates University, United Arab Emirates
- Blum, Christian, Artificial Intelligence Research Institute (IIIA-CSIC), Spain
- Bosch, Jan, Chalmers University of Technology, Sweden
- Boustras, George, European University, Cyprus
- Bryant, Barrett, University of North Texas, USA
- Buyya, Rajkumar, University of Melbourne, Australia
- Cano, Alberto, Virginia Commonwealth University, United States
- Casalino, Gabriella, University of Bari "Aldo Moro", Italy
- Corizzo, Roberto, American University, USA
- Cornelis, Chris, Ghent University, Belgium
- Dias, Gonçalo, University of Aveiro, Portugal
- Djidjev, Hristo, Los Alamos National Laboratory, USA and Institute of Information and Communication Technologies, Bulgaria
- Ducange, Pietro, University of Pisa, Italy
- Duch, Włodzisław, Nicolaus Copernicus University, Poland
- Fill, Hans-George, University of Fribourg, Switzerland
- Fred, Ana, Instituto Superior Técnico (IST—Technical University of Lisbon), Portugal
- Furtak, Janusz, Military University of Technology, Poland
- Giancarlo Guizzardi, Free University of Bolzano-Bozen, Italy
- Herrera, Francisco, University of Granada, Spain
- Hinchey, Mike, Lero, University of Limerick, Ireland
- Hodoň, Michal, University of Žilina, Slovakia
- Kacprzyk, Janusz, Systems Research Institute, Polish Academy of Sciences, Poland
- King, Irwin, The Chinese University of Hong Kong, China
- Kolukısa Tarhan, Ayça, Hacettepe University, Turkey
- Komorowski, Jan, Uppsala University, Sweden
- Kwaśnicka, Halina, Wrocław University of Science and Technology, Poland
- Luck, Michael, King's College London, United Kingdom
- Luković, Ivan, University of Belgrade, Serbia

- Matwin, Stan, Dalhousie University, University of Ottawa, Canada and Institute of Computer Science, Polish Academy of Science, Poland
- Mernik, Marjan, University of Maribor, Slovenia
- Michalewicz, Zbigniew, University of Adelaide, Australia
- Miller, Gloria, maxmetrics, Germany
- Naldi, Maurizio, LUMSA University, Italy
- Pawłowski, Wiesław, University of Gdańsk and Systems Research Institute, Polish Academy of Sciences, Poland
- Pedrycz, Witold, University of Alberta, Canada
- Popović, Aleksandar, University of Montenegro, Montenegro
- Raś, Zbigniew, University of North Carolina, United States
- Segal, Michael, Ben-Gurion University of the Negev, Israel
- Skowron, Andrzej, Systems Research Institute, Polish Academy of Sciences, Poland
- Słowiński, Roman, Poznań University of Technology, Poland
- Sosnowski, Łukasz, Systems Research Institute, Polish Academy of Sciences, Poland
- Sowa, John F., VivoMind Research, LLC, USA
- Spanoudakis George, University of London, United Kingdom
- Suri, Niranjan, Institute of Human and Machine Cognition, United States
- Świechowski, Maciej, QED Software, Poland
- Szczuka, Marcin, University of Warsaw, Poland
- Wasielewska-Michniewska, Katarzyna, Systems Research Institute, Polish Academy of Sciences, Poland
- Wątróbski, Jarosław, University of Szczecin, Poland
- Zdravevski, Eftim, Ss. Cyril and Methodius University, Macedonia
- Ziemba, Ewa, University of Econmics in Katowice, Poland

The FedCSIS 2023 Program Committee consisted of:

- Abramowicz, Witold, Poznań University of Economics and Business, Poland
- Ahad, Mohd Abdul, Jamia Hamdard, India
- Ahmad, Muhammad Ovais, Karlstad University, Sweden
- Al-Naday, Mays, University of Essex, United Kingdom
- Almeida, Luis, University of Porto, Portugal
- Alshayeb, Mohammad, King Fahd University of Petroleum & Minerals, Saudi Arabia
- Anastassi, Zacharias, ASPETE School of Pedagogical and Technological Education, Greece
- Andres, Frederic, National Institute of Informatics, Japan
- Aneta Poniszewska-Maranda, Lodz University of Technology, Poland
- Arabas, Jaroslaw, Warsaw University of Technology, Poland
- Arruda Filho, Emílio José, University FUMEC, Brasil
- Atanassov, Krassimir T., Bulgarian Academy of Sciences, Bulgaria
- Atasever, Mesut, Uşak University, Turkey
- Azad, Mohammad, Jouf University, Saudi Arabia
- Aziz, Shariq, University of Lahore, Pakistan
- Babur, Önder, Wageningen University & Research, the Netherlands
- Bacco, Manlio, Institute of Information Science and Technologies, National Research Council, Italy
- Bachan, Jolanta, Adam Mickiewicz University, Poland
- Badica, Amelia, University of Craiova, Romania
- Badica, Costin, University of Craiova, Romania
- Bajdor, Paula, Czestochowa University of Technology, Poland
- Balazs, Krisztian, Budapest University of Technology and Economics, Hungary
- Baldán Lozano, Francisco Javier, University of Granada, Spain
- Ballas, Rüdiger G., Mobile University of Technology, Germany
- Banach, Richard, University of Manchester, United Kingdom
- Banaszak, Zbigniew, Warsaw University of Technology, Poland
- Barisic, Ankica, Université Côte d'Azur, France
- Barreiro, Anabela, Universidade de Lisboa, Portugal
- Bartosz Walter, Poznań University of Technology, Poland
- Bauer, Markus, InfAI, Germany
- Belciug, Smaranda, University of Craiova, Romania
- Bellinger, Colin, National Research Council of Canada, Canada
- Ben-Assuli, Ofir, Ono Academic College, Israel
- Białas, Andrzej, Institute of Innovative Technologies EMAG, Poland
- Bicevskis, Janis, University of Latvia, Riga
- Bielecki, Wlodzimierz, ZUT Szczecin, Poland
- Bigi, Brigitte, Laboratoire Parole et Langage, CNRS, France
- Binnewitt, Johanna, BIBB, and University of Cologne, Germany
- Biro, M, Software Competence Center Hagenberg, Austria
- Bjeladinovic, Srdja, University of Belgrade, Serbia
- Blachnik, Marcin, Silesian University of Technology, Poland
- Blasband, Darius, RainCode, Belgium
- Bluemke, Ilona, Warsaw University of Technology, Poland
- Bodyanskiy, Yevgeniy, Kharkiv National University of Radio Electronics, NURE, Ukraine
- Boeva, Veselka, Blekinge Institute of Technology, Sweden
- Bogumiła Hnatkowska, Wrocław University of Science and Technology, Poland

- Boiński, Tomasz, Gdańsk University of Technology, Poland
- Bolanowski, Marek, Rzeszow University of Technology, Poland
- Borkowski, Bolesław, University of Warsaw, Poland
- Borzemski, Leszek, Wroclaw University of Technology, Poland
- Brezovan, Marius, University of Craiova, Romania
- Bridova, Ivana, University of Zilina, Slovakia
- Bronselaer, Antoon, Ghent University, Belgium
- Brugnano, Luigi, Università di Firenze, Italy
- Brzoza-Woch, Ada, AGH University of Science and Technology, Poland
- Bubak, Marian, AGH Krakow, Poland and University of Amsterdam, the Netherlands
- Buchalcevova, Alena, University of Economics, Czech Republic
- Burczynski, Tadeusz, Polish Academy of Sciences, Poland
- Byrski, Aleksander, AGH University Science and Technology, Poland
- Cabri, Giacomo, Università di Modena e Reggio Emilia, Italy
- Camilli, Matteo, Politecnico di Milano, Italy
- Cano, Alberto, Virginia Commonwealth University, USA
- Caraffini, Fabio, Swansea University, United Kingdom
- Carbone, Roberto, Security & Trust Unit, FBK, Italy
- Carchiolo, Vincenza, Universita di Catania, Italy
- Casalino, Gabriella, Università degli studi di Bari "A.Moro", Italy
- Castrillon-Santana, Modesto, University of Las Palmas de Gran Canaria, Spain
- Ceci, Michelangelo, University of Bari "A. Moro", Italy
- Charytanowicz, Malgorzata, Catholic University of Lublin, Poland
- Chelly, Zaineb, Université Paris-Saclay, UVSQ, DAVID, France
- Cherukuri, Aswani Kumar, VIT University, India
- Chomiak-Orsa, Iwona, Wroclaw University of Economics and Business, Poland
- Chren, Stanislav, Aalto University, Finland
- Christozov, Dimitar, American University in Bulgaria, Bulgaria
- Chudán, David, Prague University of Economics and Business, Czech Republic
- Cicirelli, Franco, Dimes - Unical, Italy
- Ciucci, Davide, Università di Milano-Bicocca, Italy
- Clarke, Nathan, University of Plymouth, United Kingdom
- Colantonio, Sara, ISTI-CNR, Italy
- Corpetti, Thomas, University of Rennes, France
- Courty, Nicolas, University of Bretagne Sud, France
- Coviello, Giuseppe, Politecnico di Bari, Italy
- Cyganek, Bogusław, AGH University of Science and Technology, Poland
- Czarnacka-Chrobot, Beata, Warsaw School of Economics, Poland
- D'Ambra, Pasqua, IAC-CNR, Italy
- da Silva, Marcelino Silva, Federal University of Pará, Brasil
- Dabrowski, Wlodzimierz, Warsaw University of Technology, Poland
- Dahyot, Rozenn, Maynooth University, Dublin, Ireland
- Dajda, Jacek, AGH University Of Science And Technology, Poland
- Damasevicius, Robertas, Silesian University of Technology, Poland
- Daszczuk, Wiktor, Warsaw University of Technology, Poland
- De Juana-Espinosa, Susana, Universidad de Alicante, Spain
- de Souza, Efren Lopes, Universidade Federal do Oeste do Pará, Brasil
- De Tré, Guy, Ghent University, Belgium
- Derezinska, Anna, Warsaw University of Technology, Poland
- Dettmer, Sandra, Swansea University, United Kingdom
- Dey, Lipika, Innovation Labs, TCS, India
- Dimitrieski, Vladimir, Faculty Of Technical Sciences, Serbia
- Domanska, Joanna, Institute of Theoretical and Applied Informatics, Poland
- Drag, Pawel, Wroclaw University, Poland
- Drezewski, Rafal, AGH University of Science and Technology, Poland
- Düntsch, Ivo, Brock University, Canada
- Duarte Salinas, Diana, Emory University, USA
- Dupas, Remy, Université de Bordeaux, France
- Durillo, Juan J., Leibniz Supercomputing Centre (LRZ), Germany
- Dutta, Arpita, National University of Singapore, Singapore
- Dutta, Arpita, National University of Singapore
- Dutta, Soma, University of Warmia and Mazury in Olsztyn, Poland
- Eisenbardt, Monika, Univeristy of Economics Katowice, Poland
- Ekpenyong, Moses, University of Uyo, Nigeria
- El-Halim, Essam H. Houssein, Minia University, Egypt
- Engelbrecht, Andries, University of Stellenbosch, South Africa
- Erata, Ferhat, Yale University, USA
- Escalona, M. J., University of Seville, Spain
- Fafoutis, Xenofon, Technical University of Denmark
- Fareh, Messa, University Blida 1, Algeria
- Farooq, Ali, University of Turku, Finland

- Fechner, Richard, University of Tübingen, Germany
- Felkner, Anna, NASK – Research and Academic Computer Network, Poland
- Fialko, Sergiy, Cracow University of Technology, Poland
- Filipe, Vitor, INESC TEC / UTAD, Portugal
- Flasinski, Mariusz, Jagiellonian University, Poland
- Fonseca, José Manuel, UNINOVA, Portugal
- Fourneau, Jean-Michel, DAVID, Universite de Versailles St Quentin, France
- Fournier-Viger, Philippe, University of Moncton, Canada
- Fuchs, Christoph, University of Bonn, Germany
- Fujita, Hamido, Iwate Prefectural University, Japan
- Furnell, Steven, University of Nottingham, United Kingdom
- G. Barbosa, Jorge, University of Porto, Portugal
- G.-Tóth, Boglárka, University of Szeged, Hungary
- Gabryelczyk, Renata, University of Warsaw, Poland
- Ganea, Eugen, University of Craiova, Romania
- García-Mireles, Gabriel, Universidad de Sonora, Mexico
- Gawin, Bartłomiej, University of Gdańsk, Poland
- Gawkowski, Piotr, Warsaw University of Technology, Poland
- Gburzyński, Paweł, University of Alberta, Canada; Vistula University, Poland
- Ge, Mouzhi, Deggendorf Institute of Technology, Germany
- Georgiev, Krassimir, Bulgarian Academy of Sciences, Institute of Information and Communication Technologies, Bulgaria
- Gepner, Pawel, PAWEŁ GEPNER AI, Poland
- Geri, Nitza, The Open University of Israel, Israel
- Gheisari, Mehdi, Islamic Azad University, Iran
- Giannoutakis, Konstantinos, University of Macedonia, Greece
- Girardi, Rosario, UFMA, Brazil
- Gjoreski, Hristijan, Ss. Cyril and Methodius University in Skopje, North Macedonia
- Gobov, Denys, NTUU KPI, Ukraine
- Goczyła, Krzysztof, Gdańsk University of Technology, Poland
- Godboley, Sangharatna, NIT Nagpur, India
- Göknil, Arda, SINTEF Digital, Norway
- Gomes, Luis, Universidade NOVA de Lisboa, Portugal
- Gomolinska, Anna, University of Bialystok, Poland
- González-Deleito, Nicolás, Sirris, Belgium
- Gora, Paweł, University of Warsaw, Poland
- Grabara, Dariusz, University of Economics in Katowice, Poland
- Grabowski, Mariusz, Cracow University of Economics, Poland
- Gracanin, Denis, Virginia Tech, United States
- Graliński, Filip, Adam Mickiewicz University, Poznań, Poland
- Gravvanis, George, Democritus University of Thrace, Greece
- Grochla, Krzysztof, Institute of Theoretical and Applied Informatics of PAS, Poland
- Grönman, Jere, Tampere University, Finland
- Gunasekaran, Karthick, University of Massachusetts / Amazon, United States
- Habela, Piotr, Polish-Japanese Institute of Information Technology, Poland
- Hadj Salem, Khadija, INESC TEC, Portugal
- Hakius, Bettina, BTA Wiedenest, Germany
- Halasz, David, Masaryk University, Czech Republic
- Halawi, Leila, Embry-Riddle Aeronautical University, USA
- Hamel, Oussama, University Blida 1, Algeria
- Hammoudeh, Mohammad, Manchester Metropolitan University, United Kingdom
- Hanslo, Ridewaan, University of Pretoria, South Africa
- Haręźlak, Katarzyna, Silesian University of Technology, Poland
- Hasso, Hussein, Fraunhofer FKIE, Wachtberg, Germany
- Heil, Sebastian, Technische Universität Chemnitz, Germany
- Hein, Kristine, BIBB, Bonn, Germany
- Helsingius, Mika, Finnish Defence Research Agency, Finland
- Hernes, Marcin, Wroclaw University of Economics and Business, Poland
- Herold, Sebastian, Karlstad University, Sweden
- Herrera Viedma, Enrique, University of Granada, Spain
- Hnatkowska, Bogumila, Wroclaw University of Technology, Poland
- Horváth, Zoltán, Eötvös Loránd University, Hungary
- Hosobe, Hiroshi, Hosei University, Japan
- Hrach, Christian, InfAI, Germany
- Hsiao, Michael, Virginia Tech, United States
- Hu, Bao-Gang, Institute of Automation, Chinese Academy of Sciences, China
- Hübenthal, Tobias, University of Cologne, Germany
- Hullam, Gabor, Budapest University of Technology and Economics, Hungary
- Hussain, Shahid, Institute of Business Administration, Pakistan
- Huzar, Zbigniew, Wroclaw University of Technology, Poland
- Ienco, Dino, Territories, Environment, Remote Sensing and Spatial Information, France
- Ignaciuk, Przemyslaw, Lodz University of Technology, Poland
- Inayat, Irum, National University of Computers and Emerging Sciences, Pakistan

- Iserte, Sergio, Universitat Jaume I, Spain
- Iwanowski, Marcin, Warsaw University of Technology, Poland
- Jakovljevic, Niksa, University of Novi Sad, Serbia
- Jana, Purbita, The Institute of Mathematical Sciences (IMSc.), India
- Janicki, Artur, Warsaw University of Technology, Poland
- Janicki, Ryszard, McMaster University, Canada
- Janousek, Jan, Czech Technical University Prague, Czechia
- Jarzabek, Stanislaw, Bialystok University of Technology, Poland
- Jassem, Krzysztof, Adam Mickiewicz University, Poland
- Jaworski, Rafał, Adam Mickiewicz University, Poland
- Jelonek, Dorota, Czestochowa University of Technology, Poland
- Jensen, Richard, Aberystwyth University, United Kingdom
- Johnsen, Frank, Norwegian Defence Research Establishment, FFI, Norway
- Jovanovik, Milos, Ss. Cyril and Methodius University in Skopje, North Macedonia
- Kacprzyk, Janusz, Systems Research Institute, Polish Academy of Sciences, Poland
- Kaczmarek, Katarzyna, University of Strathclyde, United Kingdom
- Kaloyanova, Kalinka, University of Sofia, Bulgaria
- Kanciak, Krzysztof, Military University of Technology, Poland
- Kania, Krzysztof, University of Economics in Katowice, Poland
- Kapczyński, Adrian, Silesian University of Technology, Poland
- Karaduman, Burak, University of Antwerp, Belgium
- Kasprzak, Wlodzimierz, Politechnika Warszawska, Poland
- Katic, Marija, University of London, United Kingdom
- Keir, Paul, University of the West of Scotland, Scotland
- Kelner, Jan, Military University of Technology, Poland
- Keswani, Bright, Suresh Gyan Vihar University, Jaipur, India
- Khan, Md. Aquil, Indian Institute of Technology Indore, India
- Khlif, Wiem, FSEGS, Tunisia
- Kieraś, Witold, Institute of Computer Science, Polish Academy of Sciences, Poland
- Kimovski, Dragi, University of Klagenfurt, Austria
- Kitchenham, Barbara, Keele University, United Kingdom
- Klapp, Iftach, Institute of Agricultural Engineering, Volcani Institute, Agricultural Research Organization Bet Dagan, Israel
- Klein, Sarah, Sirris, Belgium
- Kliegr, Tomáš, Prague University of Economics and Business, Czech Republic
- Kluza, Krzysztof, AGH University of Science and Technology, Poland
- Kobylinski, Andrzej, Warsaw School of Economics, Poland
- Koczy, Laszlo, Szechenyi Istvan University, Hungary
- Kokosinski, Zbigniew, Cracow University of Technology, Poland
- Kolog, Emmanuel Awuni, University of Ghana, Ghana
- Kononova, Anna, LIACS, Leiden University, the Netherlands
- Kopczyńska, Sylwia, Poznan University of Technology, Poland
- Koržinek, Danijel, Polish-Japanese Academy of Information Technology, Poland
- Kosar, Tomaz, University of Maribor, Slovenia
- Kosiuczenko, Piotr, ISI, WAT, Poland
- Koumaras, Harilaos, National Centre For Scientific Research Demokritos, Greece
- Kovatcheva, Eugenia, University of Library Studies and Information Technologies, Bulgaria
- Kozak, Jan, University of Economics in Katowice, Poland
- Kozielski, Stanislaw, Silesian University of Technology, Poland
- Kozłowski, Artur, Łukasiewicz Research Network, Poland
- Krajsic, Philippe, University Leipzig, Germany
- Krawczyk, Henryk, Gdańsk University of Technology, Poland
- Krawiec, Krzysztof, Poznan University of Technology, Poland
- Krdžavac, Nenad, Technische Informationsbibliothek (TIB), Germany
- Kretowski, Marek, Bialystok University of Technology, Poland
- Kropp, Martin, University of Applied Sciences and Arts Northwestern Switzerland, Switzerland
- Krüger, Kai, BIBB, Bonn, Germany
- Kryvinska, Natalia, Comenius University in Bratislava, Slovakia
- Kryvyi, Serhii, Taras Shevchenko National University of Kyiv, Ukraine
- Kuchanskyy, Vladislav, National Academy of Sciences in Ukraine, Institute of Electrodynamics of the National Academy of Sciences of Ukraine
- Kulakov, Andrea, University "Ss.Cyril and Methodius", North Macedonia
- Kulczycki, Piotr, Systems Research Institute, Polish Academy of Sciences, Poland
- Kurasova, Olga, Institute of Mathematics and Informatics, Bulgaria
- Kusy, Maciej, Rzeszow University of Technology, Poland

- Kwasnicka, Halina, Politechnika Wroclawska, Poland
- Kwater, Tadeusz, Państwowa Wyższa Szkoła Techniczno-Ekonomiczna, Poland
- Kwolek, Bogdan, AGH University of Science and Technology, Poland
- Lacalle Úbeda, Ignacio, Universitat Politècnica de València, Spain
- Laccetti, Giuliano, University of Naples Federico II and INFN, Italy
- Lameski, Petre, Ss. Cyril and Methodius University in Skopje, North Macedonia
- Lano, Kevin, King's College London, United Kingdom
- Lasek, Piotr, University of Rzeszów, Poland
- Laskov, Lasko, New Bulgarian University, Bulgaria
- Lastovetsky, Alexey, University College Dublin, Ireland
- Lech Madeyski, Wrocław University of Science and Technology, Poland
- Lencastre, Maria, Escola Politécnica de Pernambuco – UPE, Brasil
- Lerga, Jonatan, University of Rijeka, Croatia
- Leszczyna, Rafał, Gdańsk University of Technology, Poland
- Lewowski, Tomasz, Wrocław University of Science and Technology, Poland
- Li, Tianrui, Southwest Jiaotong University, China
- Ligeza, Antoni, AGH University of Science and Technology, Poland
- Lilik, Ferenc, Szechenyi Istvan University, Hungary
- Lin, Zhe, Department of Philosophy Xiamen University Xiamen, China
- Lirkov, Ivan, Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Bulgaria
- Ljubić, Sandi, University of Rijeka, Croatia
- Lloret, Elena, University of Alicante, Spain
- Lobato, Fábio, Federal University of Western Pará, Brasil
- Lovassy, Rita, Obuda University, Hungary
- Ltaief, Hatem, King Abdullah University of Science and Technology, Saudi Arabia
- Luna, Jose, University of Cordoba, Spain
- Lunesu, Ilaria, University of Cagliari, Italy
- Luque, Gabriel, University of Málaga, Spain
- Luszczek, Piotr, University of Tennessee Knoxville, USA
- Machado, José, Universidade do Minho, Portugal
- Madeyski, Lech, Wroclaw University of Science and Technology, Poland
- Majdik, Andras, Institute for Computer Science and Control, Hungary
- Malecki, Piotr, Institute of Nuclear Physics PAN, Poland
- Mangioni, Giuseppe, University of Catania, Italy
- Manso, Marco, PARTICLE LTD., Portugal
- Mansurova, Madina, al-Farabi Kazakh National University, Kazakhstan
- Maravilla, Javier Calpe, University of Valencia, Spain
- Marchiori, Massimo, UNIPD and EISMD, Italy
- Marcińczuk, Michał, Wroclaw University of Science and Technology, Poland
- Marciniak, Jacek, Adam Mickiewicz University, Poland
- Marcinkowski, Bartosz, University of Gdansk, Poland
- Marek Bolanowski, Rzeszow University of Technology, Poland
- Marghitu, Daniela, Auburn University, USA
- Martínez López, Pablo E., UNQ, Argentina
- Maślankowski, Jacek, University of Gdańsk, Poland
- Mathà, Roland, Distributed and Parallel Systems Group, Austria
- Matson, Eric, Purdue University, USA
- Matthies, Christoph, Hasso Plattner Institute, University of Potsdam, Germany
- Melzer, Sylvia, Universität Hamburg, Germany
- Mele, Valeria, University of Naples Federico II, Italy
- Meneses, Claudio, Pontificia Universidad Católica de Chile, Chile
- Mercier-Laurent, Eunika, Jean Moulin Lyon 3 University, France
- Mernik, Marjan, University of Maribor, Slovenia
- Mesiar, Radko, Slovak University of Technology, Slovakia
- Michał Smialek, Warsaw University of Technology, Poland
- Michalik, Krzysztof, University of Economics in Katowice, Poland
- Micota, Flavia, West University of Timisoara, Romania
- Mignone, Paolo, Bari University, Italy
- Mihaescu, Marian Cristian, University of Craiova, Romania
- Mihajlov, Martin, Jozef Stefan Institute, Slovenia
- Mihálydeák, Tamás, University of Debrecen, Hungary
- Milašinović, Boris, University of Zagreb, Croatia
- Mildorf, Tomas, University of West Bohemia, Czech Republic
- Milella, Annalisa, Institute of Intelligent and Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, Italy
- Miler, Jakub, Gdansk University of Technology, Poland
- Miller, Gloria, maxmetrics, Germany
- Millham, Richard, Durban University of Technology, South Africa
- Milosavljevic, Gordana, Faculty of Technical Sciences, Serbia
- Mirosław Ochodek, Poznań University of Technology, Poland

- Misra, Sanjay, Østfold University, Norway
- Mocanu, Mihai, University of Craiova, Romania
- Modoni, Gianfranco, STIIMA-CNR, Italy
- Mohapatra, Durga Prasad, NIT, India
- Mongay Batalla, Jordi, National Institute of Technology, Poland
- Mora, André Damas, UNINOVA, Portugal
- Morales Trujillo, Miguel Ehécatl, University of Canterbury, New Zealand
- Moshkov, Mikhail, King Abdullah University of Science and Technology, Saudi Arabia
- Mozgovoy, Maxim, The University of Aizu, Japan
- Mullins, Roisin, University of Wales Trinity Saint David, United Kingdom
- Muñoz, Andres, Cadiz University, Spain
- Muszyńska, Karolina, University of Szczecin, Poland
- Myszkowski, Pawel, Wrocław University of Science and Technology, Poland
- Nawrocki, Jerzy, Poznan University of Technology, Poland
- Nazaruka, Erika, Riga Technical University, Latvia
- Neumann, Michael, Hochschule Hannover, Germany
- Ng, Yen Ying, Nicolaus Copernicus University, Poland
- Niewiadomska-Szynkiewicz, Ewa, Warsaw University of Technology, Poland
- Noguera i Clofent, Carles, Institute of Information Theory and Automation (UTIA), Academy of Sciences of the Czech Republic, Czech Republic
- Nosović, Novica, University of Sarajevo, Bosnia and Herzegovina
- Noyer, Arne, Ostfalia University of Applied Sciences, Germany
- Nutini, Francesco, Institute for Electromagnetic Sensing of the Environment, National Research Council, Milano, Italy
- Ó Cinnéide, Mel, National University of Ireland, Dublin, Ireland
- Ochodek, Mirosław, Poznan University of Technology, Poland
- Ogrodniczuk, Maciej, Institute of Computer Science, Polish Academy of Sciences, Poland
- Okarma, Krzysztof, West Pomeranian University of Technology in Szczecin, Poland
- Oppermann, Alexander, Physikalisch-Technische Bundesanstalt, Germany
- Ota, Daniel, Fraunhofer, Germany
- Ozkan, Necmettin, Kuveyt Turk Participation Bank, Turkey
- Ozkaya, Mert, Yeditepe University, Turkey
- Palau, Carlos, Universitat Politècnica de València, Spain
- Paliwoda-Pękosz, Grażyna, Cracow University of Economics, Poland
- Palma, Raul, Supercomputing and Networking Center, Poland
- Palmigiano, Alessandra, The Vrije Universiteit Amsterdam, the Netherlands
- Paluszyński, Wiesław, TIC sp. z o.o., Poland
- Pamin, Jerzy, Cracow University of Technology, Poland
- Pancerz, Krzysztof, Academy of Zamosc, Poland
- Panda, Subhrakanta, BITS-PILANI Hyderabad Campus, India
- Pandey, Dr. Rajiv, Amity University, India
- Pandey, Sushant Kumar, University of Gothenburg, Sweden
- Pankowska, Malgorzata, University of Economics in Katowice, Poland
- Papaspyrou, Nikolaos S., National Technical University of Athens, Greece
- Paszkiewicz, Andrzej, Politechnika Rzeszowska im. I. Łukasiewicza, Poland
- Pataricza, András, Budapest University of Technology and Economics, Hungary
- Pazienza, Andrea, Innovation Lab, Exprivia S.p.A., Italy
- Pekergin N., Nihal, Univ. Paris-Est-Creteil, France
- Peralta, Daniel, Ghent University, Belgium
- Perechuda, Kazimierz, Wroclaw University of Economics and Business, Poland
- Pereira Nunes, Bernardo, The Australian National University, Australia
- Pereira, Rui Humberto, ISCAP/IPP, Portugal
- Petcu, Dana, West University of Timisoara, Romania
- Peters, Georg, Hochschule München, Germany
- Petrik, Milan, Czech University of Life Sciences Prague, Czech Republic
- Petrovska, Biserka, Goce Delcev University, North Macedonia
- Pęzik, Piotr, University of Lodz, Poland
- Piasecki, Maciej, Wroclaw University of Science and Technology, Poland
- Piotr Kosiuczenko, Military University of Technology in Warsaw, Poland
- Pires, Ivan Miguel, Universidade da Beira Interior, Portugal
- Po, Laura, Universitá di Modena e Reggio Emilia, Italy
- Poniszewska-Maranda, Aneta, Lodz University of Technology, Poland
- Porta, Marco, University of Pavia, Italy
- Porubän, Jaroslav, Technical University of Košice, Slovakia
- Provotar, Oleksandr, Taras Shevchenko National University of Kyiv, Ukraine
- Przybyła-Kasperek, Małgorzata, Uniwersytet Śląski w Katowicach, Poland
- Przybylek, Michal, Warsaw University, Poland
- Ptaszynski, Michal, Kitami Institute of Technology, Japan
- Puime, Felix, Universidade de A Coruña, Spain
- Queirós, Ricardo, ESMAD-P.PORTO & CRACS – INESC TEC, Portugal

- Radlinski, Lukasz, West Pomeranian University of Technology, Poland
- Ramanna, Sheela, University of Winnipeg, Canada
- Rangel Henriques, Pedro, University of Minho, Portugal
- Rantanen, Petri, Tampere University of Technology, Finland
- Rauch, Jan, Prague University of Economics and Business, Czech Republic
- Rechavi, Amit, Ruppin Academic Center, Israel
- Reformat, Marek, University of Alberta, Canada
- Ristic, Sonja, University of Novi Sad, Serbia
- Rojek, Krzysztof, Czestochowa University of Technology, Poland
- Rollo, Federica, University of Modena and Reggio Emilia, Italy
- Roman, Adam, Jagiellonian University, Poland
- Rossi, Bruno, Masaryk University, Czech Republic
- Rossi, Bruno, Masaryk University, Czechia
- Roszczyk, Radosław, Warsaw University of Technology, Poland
- Rot, Artur, Wroclaw University of Economics, Poland
- Rozevskis, Uldis, University of Latvia, Latvia
- Rusho, Yonit, Shenkar College of Engineering and Design, Israel
- Rycerz, Katarzyna, AGH University of Science and Technology, Poland
- Sá, Juliana, Universidade da Beira Interior, Portugal Hospital Center of Cova da Beira, Portugal
- Saari, Mika, Tampere University of Technology, Finland
- Sachdeva, Shelly, NIT, DELHI, India
- Sachenko, Anatoly, Ternopil State Economic University, Ukraine
- Sadowska, Małgorzata, Politechnika Wrocławska, Poland
- Salem, Abdel-Badeeh, Ain Shams University, Egypt
- Salvetti, Ovidio, Institute of Information Science and Technologies, National Research Council, Pisa, Italy
- Samolej, Slawomir, Rzeszow University of Technology, Poland
- Samotyy, Volodymyr, Lviv State University of Life Safety, Ukraine
- Santiago, Joanna, ISEG - University of Lisbon, Portugal
- Santos, Pedro Miguel, Instituto Superior de Engenharia do Porto, ISEP, Portugal
- Saraiva, João, University of Minho, Portugal
- Sarwas, Grzegorz, Warsaw University, Poland
- Saurabh, Nishant, Utrecht University, the Netherlands
- Sawerwain, Marek, University of Zielona Góra, Poland
- Schaefer, Gerald, Loughborough University, Leicestershire, United Kingdom
- Schnepf, Timo, BIBB, Bonn, Germany
- Schön, Eva-Maria, University of Applied Sciences Emden/Leer, Germany
- Schreiner, Wolfgang, Johannes Kepler University Linz, Austria
- Schreiner, Wolfgang, Research Institute for Symbolic Computation (RISC), Austria
- Schreurs, Jeanne, Hasselt University, Belgium
- Scozzari, Andrea, Institute of Information Science and Technologies, National Research Council, Italy
- Segedinac, Milan, Faculty of Technical Scieneces, Serbia
- Sekerinski, Emil, McMaster University, Canada
- Sen, Jayanta, Government General Degree College, India
- Shen, Hong, The University of Adelaide, Australia
- Sidje, Roger B., University of Alabama, USA
- Sierra, Jose Luis, Universidad Complutense de Madrid, Spain
- Sifaleras, Angelo, University of Macedonia, School of Information Sciences, North Macedonia
- Sikorski, Marcin, Gdansk University of Technology, Poland
- Sikorski, Marcin, Gdansk University, Poland
- Silaghi, Gheorghe Cosmin, Babes-Bolyai University, Romania
- Silva, Lincoln, UERJ, Brazil
- Singer, Jeremy, University of Glasgow, Scotland
- Singh, Pradeep, KIET Group of Institutions, India
- Singh, Yashwant, Jaypee University of Information Technology Waknaghat, India
- Sinkala, Zipani Tom, Karlstad Univeristy, Sweden
- Skonieczny, Łukasz, Warsaw University of Technology, Poland
- Skórzewski, Paweł, Adam Mickiewicz University, Poland
- Skowron, Andrzej, Systems Research Institute, Polish Academy of Sciences and Cardinal Stefan Wyszynski University, Poland
- Skruch, Pawel, AGH University of Krakow, Poland
- Skubalska-Rafajłowicz, Ewa, Wrocław University of Science and Technology, Poland
- Slivnik, Bostjan, University of Ljubljana, Slovenia
- Smialek, Michal, Warsaw University of Technology, Poland
- Smywiński-Pohl, Aleksander, AGH University of Science and Technology in Krakow, Poland
- Soares, Michel, Federal University of Sergipe, Brazil
- Sobczak, Andrzej, SGH, Poland
- Sobińska, Małgorzata, Wroclaw University of Economics and Business, Poland
- Sojka, Michal, Czech Technical University in Prague, Czechia
- Solanki, Vijender Kumar, CMR Institute of Technology (Autonomous), India
- Soltysik-Piorunkiewicz, Anna, University of Economics in Katowice, Poland

- Sosnowski, Janusz, Institute of Computer Science, Poland
- Sosnowski, Zenon A., Bialystok University of Technology, Poland
- Sousa Pinto, Agostinho, Instituto Politécnico do Porto, Portugal
- Sozer, Hasan, Ozyegin University, Turkey
- Stanczyk, Urszula, Silesian University of Technology, Poland
- Stanislaw Jarzabek, Bialystok University of Technology, Poland
- Stankosky, Michael, The University of Scranton, United States
- Stark, Sandra, University Leipzig, Germany
- Stasiak, Andrzej, Wojskowa Akademia Techniczna, Poland
- Stavness, Ian, University of Saskatchewan, Canada
- Steinbrink, Nicholas, the Bertelsmann Stiftung, Germany
- Stencel, Krzysztof, University of Warsaw, Poland
- Štěpánek, Lubomír, Charles University; Prague University of Economics and Business, Czech Republic
- Stoean, Catalin, Romanian Institute of Science and Technology, Cluj-Napoca, and Universidad de Malaga, Romania
- Stoean, Catalin, University of Craiova, Romania
- Stoica, Cosmin, University of Craiova, Romania
- Stój, Jacek, Silesian University of Technology, Poland
- Stojanov, Riste, Faculty of Computer Science and Engineering, North Macedonia
- Subbotin, Sergey, National University "Zaporizhzhia Polytechnic", Ukraine
- Suraj, Zbigniew, Rzeszów University, Poland
- Swacha, Jakub, University of Szczecin, Poland
- Świechowski, Maciej, QED Software, Poland
- Sylwia Kopczyńska, Poznań University of Technology, Poland
- Symeonidis, Symeon, Democritus Univesity of Thrace, Greece
- Szafran, Bartlomiej, AGH University of Science and Technology, Poland
- Szczech, Izabela, Poznan University of Technology, Poland
- Szczerbicki, Edward, University of New Castle, Australia
- Szmit, Maciej, University of Lodz, Poland
- Szmuc, Tomasz, AGH University of Science and Technology, Poland
- Szpyrka, Marcin, AGH University of Science and Technology, Poland
- Szumski, Oskar, University of Warsaw Faculty of Management, Poland
- Szwoch, Mariusz, Gdansk University of Technology, Poland
- Szyjewski, Zdzislaw, Uniwersytet Szczeciński, Poland
- Taglino, Francesco, IASI-CNR, Italy
- Tanwar, Sudeep, Nirma University, India
- Terra, Marcus, University of Londrina, Brazil
- Timpel, Patrick, WIG2, Germany
- Tomasz, Andrysiak, University of Technology and Life Sciences (UTP), Poland
- Tomczyk, Łukasz, Jagiellonian University, Poland
- Toraldo, Gerardo, Università della Campania "l.-Vanvitelli", Italy
- Tormasi, Alex, Szechenyi Istvan University, Hungary
- Töreyin, Behçet Uğur, Technical University, Turkey
- Toscano, Piero, Institute of Bioeconomy, National Research Council, Italy
- Trendowicz, Adam, Fraunhofer, Germany
- Trocan, Maria, Institut Supérieur d'Électronique de Paris, France
- Trocan, Maria, Institut Supérieur d'Electronique de Paris, France
- Trybus, Bartosz, Rzeszów University of Technology, Poland
- Trybus, Leszek, Rzeszow University of Technology, Poland
- Tudoroiu, Nicolae, John Abbott College, Canada
- Tudruj, Marek, Institute of Computer Science, Polish Academy of Sciences, Poland
- Tyagi, Sudhanshu, Thapar Institute of Engineering & Technology, India
- Unland, Rainer, University of Duisburg-Essen, Germany
- Ustimenko, Vasyl, The University of Maria Curie Sklodowska in Lublin, Poland
- Van Landuyt, Dimitri, Katholieke Universiteit Leuven, Belgium
- Varanda Pereira, Maria João, Instituto Politécnico de Bragança, Portugal
- Vardanega, Tullio, University of Padua, Italy
- Vasiliev, Julian, University of Economics, Bulgaria
- Vazquez-Poletti, Jose Luis, Universidad Complutense de Madrid, Spain
- Vecchio, Massimo, Fondazione Bruno Kessler (FBK), Italy
- Vega Vega-Rodríguez, Miguel A., University of Extremadura, Spain
- Velev, Miroslav, Aries Design Automation, United States
- Verstraete, Jörg, Systems Research Institute, Polish Academy of Sciences, Poland
- Vescoukis, Vassilios, National Technical University of Athens, Greece
- Vitek, Jan, Northeastern University, USA
- Vogiatzis, Chrysafis, University of Illinois at Urbana-Champaign, United States
- Wahid, Khan Ferdous, Airbus Group, Germany
- Walkowiak-Gall, Anita, Politechnika Wroclawska, Poland
- Walkowska, Justyna, DeepL, Poland
- Waloszek, Wojciech, Gdańsk University of Technology, Poland

- Walter, Bartosz, PCSS & PPoz, Poland
- Wardziński, Andrzej, Gdańsk University of Technology, Poland
- Wasielewska, Katarzyna, Systems Research Institute, Polish Academy of Sciences, Poland
- Wątróbski, Jarosław, West Pomeranian University of Technology, Poland
- Weber, Richard, Universidad de Chile, Chile
- Węcel, Krzysztof, Poznań University of Economics and Business, Poland
- Weerasinghe, Shakthi, Deakin University, Australia
- Wegrzynowicz, Patrycja, NASK Research and Academic Computer Network, Poland
- Wei, Wei, Xi'an University of Technology, China
- Weil, Vera, University of Cologne, Germany
- Werewka, Jan, AGH University of Science and Technology, Poland
- Wielki, Janusz, Opole University of Technology, Poland
- Wierzchoń, Piotr, Adam Mickiewicz University, Poland
- Winnige, Stefan, BIBB, Bonn, Germany
- Wisniewski, Piotr, Nicolaus Copernicus University, Poland
- Wiszniewski, Bogdan, Gdansk University of Technology, Poland
- Wnuk, Krzysztof, BTH, Sweden
- Woliński, Marcin, Institute of Computer Science, Polish Academy of Sciences, Poland
- Wróblewska, Alina, Institute of Computer Science, Polish Academy of Sciences, Poland
- Wróblewska, Anna, Warsaw University of Technology, Poland
- Wrona, Konrad, NATO Communications and Information Agency, the Netherlands
- Wysocki, Marian, Rzeszow University of Technology, Poland
- Wyrzykowski, Krzysztof, NET PC, Poland
- Xenakis, Christos, University of Piraeus, Greece
- Xuetao, Jin, Communication University of China, China
- Yang, Yujiu, Tsinghua University, China
- Yao, Yiyu, University of Regina, Canada
- Zadrożny, Slawomir, Systems Research Institute, Poland
- Zaitsev, Dmitry
- Zajac, Mieczyslaw, Cracow University of Technology, Poland
- Zalewski, Andrzej, Warsaw University of Technology, Poland
- Zalewski, Janusz, Florida Gulf Coast University, USA
- Zawadzka, Teresa, Gdańsk University of Technology, Poland
- Zaytsev, Vadim, Universiteit Twente, the Netherlands
- Zborowski, Marek, University of Warsaw, Poland
- Zdravevski, Eftim, University "Ss.Cyril and Methodius", Macedonia
- Zhang, Hongyu, The University of Newcastle, United Kingdom
- Zhu, Yungang, Jilin University, China
- Zieliński, Zbigniew, Military University of Technology, Poland
- Zielosko, Beata, Uniwersytet Śląski w Katowicach, Poland
- Ziemba, Ewa, University of Economics in Katowice, Poland
- Ziemba, Paweł, University of Szczecin, Poland
- Zitouni, M. Sami, University of Dubai, United Arab Emirates

X. Acknowledgments

In conclusion, let us emphasize that delivery of FedCSIS 2023 required a dedicated effort of many people. We would like to express our warmest gratitude to all Topical Area Curators, to the members of the Senior Program Committee and to the members of the Program Committee, for their hard work in attracting and reviewing all submissions. We thank the authors of papers for their great contribution to the theory and practice of Computer Science and Intelligence Systems. We are grateful to Keynote, Invited, and Tutorial Speakers, for sharing their knowledge and wisdom with the participants. Last, but not least, we acknowledge one more time Jarosław Arabas, Sławomir Zadrożny, and Przemysław Biecek. We are very grateful for your efforts!

We hope that you all had an inspiring conference. We also hope to meet you again for the 19th Conference on Computer Science and Intelligence Systems (FedCSIS 2024) which will take place in Belgrade, Serbia on September 8-11, 2024. Finally, we hope that you will find the evolution of the FedCSIS Conference concept as something that properly addresses the current needs of research and applications. We want to continue looking at Computer Science from different angles but in the same time, acknowledging the topic Intelligence Systems as the central point of everything that we are considering.

Co-Chairs of the FedCSIS Conference Series:

Maria Ganzha, *Warsaw University of Technology, and Systems Research Institute Polish Academy of Sciences, Poland*
Leszek Maciaszek (Honorary Chair), *Macquarie University, Australia and Wrocław University of Economics, Poland*
Marcin Paprzycki, *Systems Research Institute Polish Academy of Sciences, and Warsaw University of Management, Poland*
Dominik Ślęzak, *University of Warsaw, Poland and QED Software, Poland and DeepSeas, USA*

Annals of Computer Science and Information Systems,
Volume 35

Proceedings of the 18th Conference on Computer Science and Intelligence Systems

September 17–20, 2023. Warsaw, Poland

TABLE OF CONTENTS

MAIN TRACK

MAIN TRACK INVITED CONTRIBUTIONS

MAIN TRACK REGULAR PAPERS

MAIN TRACK SHORT PAPERS

THEMATIC TRACKS

THEMATIC TRACKS REGULAR PAPERS

THEMATIC TRACKS SHORT PAPERS

Competitions

PolEval

Cybersecurity Threat Detection in the Behavior of IoT Devices

Center for Artificial Intelligence Challenge on Conversational AI Correctness

TEMPORAL IMAGE CAPTION RETRIEVAL COMPETITION

DOI: 10.15439/2023F6019
ISSN 2300-5963 ACSIS, Vol. 35

Measuring Trustworthiness in Neuro-Symbolic Integration

Andrea Agiollo, Andrea Omicini
0000-0003-0531-1978, 0000-0002-6655-3869
Alma Mater Studiorum – Università di Bologna, Italy
Email: {andrea.agiollo, andrea.omicini}@unibo.it

***Abstract*—*Neuro-symbolic* integration of symbolic and subsymbolic techniques represents a fast-growing AI trend aimed at mitigating the issues of neural networks in terms of decision processes, reasoning, and interpretability. Several state-of-the-art neuro-symbolic approaches aim at improving performance, most of them focusing on proving their effectiveness in terms of raw predictive performance and/or reasoning capabilities. Meanwhile, few efforts have been devoted to increasing model trustworthiness, interpretability, and efficiency—mostly due to the complexity of measuring effectively improvements in terms of trustworthiness and interpretability. This is why here we analyse and discuss the need for ad-hoc *trustworthiness metrics* for neuro-symbolic techniques. We focus on two popular paradigms mixing subsymbolic computation and symbolic knowledge, namely: *(i) symbolic knowledge extraction* (SKE), aimed at mapping subsymbolic models into human-interpretable knowledge bases; and *(ii) symbolic knowledge injection* (SKI), aimed at forcing subsymbolic models to adhere to a given symbolic knowledge. We first emphasise the need for assessing neuro-symbolic approaches from a trustworthiness perspective, highlighting the research challenges linked with this evaluation and the need for ad-hoc trust definitions. Then we summarise recent developments in SKE and SKI metrics focusing specifically on several trustworthiness pillars such as interpretability, efficiency, and robustness of neuro-symbolic methods. Finally, we highlight open research opportunities towards reliable and flexible trustworthiness metrics for neuro-symbolic integration.**

I. Introduction

A GROWING number of critical applications are being developed that rely on artificial intelligence (AI) solutions—mostly, on machine and deep learning (ML, DL), more specifically. In this realm, the most popular trend is by far the engineering of intelligent computational systems where hard-to-code tasks are automatically learned from data—promoting a data-driven problem-solving approach. Tasks that can be learned this way range from image [1], [2] to text processing [3], [4], stepping through graph learning [5], [6], [7] and time series forecasting [8], [9], among the many others. The popularity of (semi-)autonomous AI systems largely depends on their ability of outperforming humans in some specific tasks. Yet, AI agents – and especially ML agents – cannot be really trusted by humans, for the obscurity of their data processing and decision making pipeline, and for their limited interaction with human users as well. In the recent past, this lack of trustworthiness jumped to the news due to some AI systems' behaviour harming humans—such as chatbots suggesting deleterious practice[1] and facial-recognition technology recognising innocents as criminals.[2] Therefore, the need to assess the level of trustworthiness of AI system before its deployment it is nowadays apparent to all parties involved in the development of AI solutions. Targeting this need, the European Union (EU) has recently released the Ethics Guidelines for Trustworthy AI[3] as a part of its AI strategy.

While representing a fundamental stepping stone in the definition of AI trustworthiness, these ethics guidelines apparently focus on popular ML agents solutions in their definition process. Indeed, most trust requirements are clearly linked with the black-box nature of ML and DL solutions—such as the need for transparency, explanations, human interaction, and many others. However, AI is not just ML/DL, so AI systems are much more than ML/DL systems. Recent research efforts have focused on novel AI paradigms aiming at blending the subsymbolic perspective of ML and DL agents with symbolic AI solutions focusing on high-level symbolic (human-readable) representations of problems, logic, and search: this is where *neuro-symbolic integration systems* (NeSy) stand today. NeSy integrate neural (subsymbolic) and symbolic AI solutions aiming at suitably complementing their strengths and weaknesses, introducing reasoning and cognitive capabilities (the symbolic way) while preserving fast-learning capabilities (the subsymbolic way). The range of NeSy approaches is vastly distant from the AI systems accounted for in the definition of EU trustworthiness pillars, as they leverage symbolic (human-comprehensible) solutions which are in principle trustworthy by design. Therefore, NeSy introduces a further level of complexity in the definition of their trustworthiness value, given by the complex interaction between symbolic and subsymbolic elements. The result is the current lack of suitable definitions of the notion of trustworthiness in terms of NeSy systems.

This is why in this paper we deal with the definition of trustworthiness for NeSy systems, focusing specifically on two broad NeSy categories, namely:

[1] https://edition.cnn.com/2023/06/01/tech/eating-disorder-chatbot/

[2] https://www.nytimes.com/2020/12/29/technology/facial-recognition-misidentify-jail.html

[3] https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

- *symbolic knowledge injection* (SKI) models—that is, systems featuring symbolic knowledge that can be explicitly provided so that subsymbolic predictions are either computed as a function of it, or made consistent with it;
- *symbolic knowledge extraction (SKE)* models, represented by the set of approaches accepting subsymbolic predictors as input and producing symbolic knowledge as output; the aim for the SKE system is to extract symbolic knowledge reflecting the behaviour of the predictor with high fidelity.

The definition of requirements for trustworthy NeSy systems represents a fundamental step towards their safe adoption. However, requirements definition by itself can not be considered as an exhaustive measure to ensure and calibrate the trustworthiness of NeSy systems. Instead, it is of utmost significance to define NeSy *trustworthiness metrics* that allow to actually measure the level of a system trust, possibly enabling an in-depth analysis of the components raising trust concerns. Whereas a few trustworthiness metrics definition already exist, tackling specific components of NeSy models – such as accuracy, robustness and efficiency –, the vast majority of NeSy most relevant aspects are still unexplored.

This is why in this paper we:

- define how the AI trustworthiness requirements translate to the NeSy realm, analysing in detail each pillar of trust and its implication on NeSy models.
- analyse the available metrics for each of the novel NeSy trust requirements as well as the potential future directions to explore in the analysis of NeSy trust;
- suggest some novel metrics to measure specific NeSy elements, focussing on SKI and SKE as two well-defined broad categories of NeSy models.

This article is organised as follows. Section II presents the transition from trustworthy AI requirements to their corresponding NeSy trust pillars, analysing in depth how NeSy elements impact each requirement. Section III showcases the need for defining trust metrics – rather than requirements – and analyses the complexity of that definition, the reason behind it, and how we propose to tackle it. We then introduce the relevant concepts of SKI and SKE needed to design trust metrics in Section IV, and propose a detailed analysis of available and lacking metrics in Section V. Finally, we conclude and present the future directions in Section VI.

II. From Trustworthy AI to Trustworthy NeSy

As a fundamental step of its AI strategy, the European Union (EU) has defined seven key trustworthiness criteria to meet during the development, deployment, and use of AI systems, namely: *(i) human agency and oversight*, as the need for oversight mechanisms enabling the informed interaction between the AI agent(s) and the human(s) counterpart; *(ii) robustness and safety*, as the need for accuracy, reliability, resilience and security of AI agent(s); *(iii) privacy and data governance*, as the need for ensuring legitimised access to data, while taking into account data quality and integrity; *(iv) transparency*, as the need for providing human users with explanations of the AI agent(s)'s decision process; *(v) diversity, non-discrimination and fairness*, as the need for avoiding unfair bias while enable everyone's access to AI technology; *(vi) environmental and societal well-being*, as the need for sustainability of AI agent(s) and the transition to their environmentally friendly development; *(vii) accountability*, as the need for mechanisms that ensure responsibility and accountability for the behaviour and outcomes of AI systems. The above requirements define a broad umbrella of concepts and means to identify relevant components in the deployment of AI systems and ensure their trustworthiness. However, being designed to be general enough to be applicable to any – or at least as most as possible – AI systems, they are actually too general to be used to define actual metrics to effectively measure every sort of AI systems. Therefore, to make them actually working, a more detailed specification of trustworthiness requirements is needed: in particular, the general EU pillars should be *translated* into domain-specific pillars, promoting the definition of trustworthiness metrics for each specific AI domain. Such a translation should also account for the current bias of EU trustworthiness pillars towards subsymbolic AI systems—where, for instance, the black-box nature of all components is given as understood when dealing with issues such as transparency, explainability, human interaction, even though it mostly concerns subsymbolic components only.

Thus, in the remainder of this paper we define the pillars of trustworthiness for AI systems based on Neuro-Symbolic (NeSy) integration. We analyse the seven EU-defined trustworthiness criteria for AI, and translate each of them into its NeSy counterpart, leveraging on the aspects of NeSy that promote fairness and explainability by design. On the other hand, leveraging both symbolic and subsymbolic paradigms, NeSy systems may be affected by robustness, safety, and bias issues from both sides – i.e., symbolic and subsymbolic –, hindering their overall trustworthiness. Therefore, a fundamental issue in the NeSy context is to identify whether and to what extent the blending of symbolic and subsymbolic techniques can either help or hinder trustworthiness, in particular in the perspective of the definition of ad-hoc trustworthiness metrics.

a) Human agency and oversight: in its original formulation this requirement stresses the need for the introduction of oversight mechanisms enabling informed interaction between AI agents and humans counterparts. The underlying assumption here is that humans can not understand AI system at all – or, can understand interaction with AI systems in a very limited way – so AI systems can never be considered as trustworthy, as human agents are incapable to fix the AI system when issues arise. When taking into account NeSy mechanisms, the symbolic and subsymbolic fusion component clearly affects the interaction with with its human counterpart. Instead, the symbolic component could represent the enabling agent for meaningful interaction between human and the system, promoting human-in-the-loop, human-on-the-loop, and human-in-command approaches.

> NeSy version
>
> The need for assessing to what extent the symbolic and subsymbolic *interaction* of NeSy components helps *improving* informed human-AI interaction and human oversight.

b) Technical robustness and safety: in its original formulation this requirement stresses the need for accuracy, reliability, resilience, and security of AI agents. Indeed, an inaccurate or unstable AI agent can not be considered trustworthy, as its behaviour may fluctuate radically throughout its life cycle. Let us consider for instance adversarial examples [10], [11], where slight perturbations of the input fed to the AI system result in radically different outcomes: AI system of that sort are inherently unreliable—thus untrustworthy. Even though this has motivated some research efforts focused on the identification of robustness issues of ML/DL systems, very small light has been shed on the robustness and safety issues of NeSy systems. NeSy agents rely on both symbolic and subsymbolic components, the former being – with some exception – verifiable and stable by design while the latter lacks of stability, verifiability or strong mathematical modeling of their behaviour and properties. The interaction of such elements introduce non-trivial behaviour in NeSy systems, where the symbolic components can be used as a helping tool for stabilising subsymbolic elements or the subsymbolic tools can be used to produce imperfect – thus unreliable – symbolic knowledge. Therefore, we consider relevant studying to what extent the verifiability of symbolic components alters during the integration process, and how the (in)stability of the subsymbolic element is impacted by the symbolic knowledge.

> NeSy version
>
> The need for assessing the impact of both symbolic (verifiable) and subsymbolic (not verifiable) *interaction* on the *stability* of the NeSy system.

c) Privacy and data governance: in its original formulation this requirement stresses the need for legitimate access to data, while taking into account data quality and integrity. This requirement identifies the untrustworthy nature of systems optimised over unreliable data, and promotes the introduction of open data for testing AI systems and their behaviour. To this end, NeSy systems differ quite heavily from their pure subsymbolic AI counterparts, as they – in most cases – require the processing of symbolic knowledge and data at the same time. Therefore, it is relevant to notice that data quality issues extend to knowledge quality issues when considering NeSy systems—even though symbolic knowledge is typically managed explicitly by AI programmers and is often verifiable in automatic way.

> NeSy version
>
> The need for ensuring the *quality* of both *data* and *symbolic knowledge* of a NeSy system, along with its accessibility.

d) Transparency: in its original formulation this requirement stresses the need for producing explanations of the AI agents' decision processes, deeming as untrustworthy those AI systems for which it is complex or unfeasible to obtain an explanation of its decision process. The definition of an AI system transparency depends on the complexity of the process of obtaining explanations, and their understandability. Indeed, in most AI scenarios multiple explanations can be drawn to render transparent the system at hand, depending on the level of detail needed and the process used. While being conceptually similar, the transparency level of NeSy systems – with respect to their pure subsymbolic AI counterparts – may differ a lot in terms of extraction complexity and understandability. Indeed, most NeSy systems represent a more transparent solution by design, as they leverage symbolic components, inputs or outputs, which are – to some extent – intrinsically understandable by humans. Therefore, we consider relevant to assess if – and to what extent – the integration components of NeSy systems impacts the transparency of the obtained agent(s).

> NeSy version
>
> The need for assessing the *gain* in terms of *transparency* obtained by a NeSy system with respect to its pure subsymbolic components.

e) Diversity, non-discrimination, and fairness: in its original formulation this requirement stresses the need for avoiding unfair bias and enable everyone's access to the AI technology. Indeed, biased AI technologies must not be deployed as they have been proven to increase the chance of harmful events against human agents. Given the relevance of fairness, several efforts have been put in place to investigate the nature of AI mechanisms' bias. However, biases of pure subsymbolic models and their NeSy counterparts differ conceptually in terms of their root causes: bias can rise in NeSy models as the consequence of any unexpected behaviour of their subsymbolic components, or their interaction with their symbolic elements. Indeed, similarly to what done for NeSy robustness, here it is relevant to highlight that the bias and fairness of symbolic components represent a verifiable and provable variable, while its interaction with subsymbolic elements does not, as it is not possible to define a-priori how the subsymbolic interaction impact the overall system behaviour. Therefore, it is fundamental for NeSy systems to consider possible biases rooted in each step of the fusion between symbolic and subsymbolic components. This is also valid for possible bias benefits that can be obtained from the interaction between symbolic and subsymbolic components in NeSy, as the symbolic elements can be used to tune the subsymbolic components to avoid biases that may arise during their optimisation.

NeSy version

The need for *measuring* biased and discriminative behaviour of NeSy agents rooted in the *interaction* between their symbolic and subsymbolic components.

f) Environmental and societal well-being: in its original formulation this requirement stresses the need for sustainability of AI agents and the transition to their environmentally friendly development, deeming as untrustworthy those AI agents that do not benefit all human beings, including future generations. While measuring the impact of pure subsymbolic AI agents on the environment has been the focus of several works in the AI community, the in-depth analysis of how NeSy mechanism can help reducing the environmental impact of AI. The symbolic component of several NeSy mechanism can be leveraged as a helping tool for reducing the amount of resources required for the optimisation of its subsymbolic component. Moreover, it is also possible for some NeSy mechanism to leverage symbolic approaches to achieve comparable performance – w.r.t. pure subsymbolic AI agent – while requiring a smaller memory footprint—resulting in smaller latency and energy consumption. On the other hand, the complex interaction between symbolic and subsymbolic components may introduce an overhead in the NeSy system, causing the waste of resources and thus decreasing the efficiency of the agent. Therefore, it is necessary to define a novel resource efficiency requirement for NeSy agents.

NeSy version

The need for assessing the *gain* in terms of *sustainability* of NeSy systems with respect to their pure subsymbolic components.

g) Accountability: in its original formulation this requirement stresses the need for mechanisms that ensure responsibility and accountability for AI systems and their outcomes. At its core, accountability can be defined as an obligation to inform about, and justify the AI's conduct [12]. Therefore, the fundamental property for AI's accountability is represented by *answerability*, which is the property of an AI system to allow for *interrogation* concerning a decision process. Accountability is closely tight to transparency, as it requires for an AI system to produce justification – a.k.a. explanations – for its actions. Therefore, a similar analysis to the one done for transparency applies to this context, where we stress the relevance of analysing the accountability gains obtained through symbolic and subsymbolic integration in NeSy systems over ML/DL counterparts.

NeSy version

The need for assessing the *gain* in terms of *answerability* obtained by a NeSy system with respect to its pure subsymbolic components.

III. On the Relevance of Trustworthiness Metrics

The trustworthy requirements proposed for both general AI systems and NeSy agents represent a general umbrella of concepts that should be covered in the system at hand. Indeed, none of the requirements defined so far give a specific characterisation of a target level – e.g., target fairness – for that requirement to be considered satisfied. Such general characterisation of trustworthiness is mainly caused by two contributing factors, namely

- *High variability characterising AI systems.* AI agents optimised to solve different tasks are expected to differ largely in terms of inner working principles. Therefore, identifying a common trustworthiness definition with the due level of detail represents a complex task
- *Conceptual complexity of trustworthiness building blocks.* Trustworthiness is defined as a collection of diverse features of a systems to be achieved for it to be worthy of humans' trust. However, some – if not most – of the trustworthiness sub-components are not easy-to-grasp concepts in their definition. For example, taking into account bias, we immediately understand that bias must be one of the sub-components required to achieve trustworthiness. However, the definition of bias by itself represents a complex task that have bogged researchers with troublesome questions like what is bias?, when is a system biased?, what is the minimum amount of bias for a system to be considered as such?. Being complex in their definition, these building blocks are also complex to measure effectively, hindering the overall level of trust measurement.

The issues connected with the general characterisation of AI trustworthiness hinder the applicability of such trustworthiness requirements. Indeed, while representing a valid starting point for analysing AI trustworthiness, these requirements do not fully allow to comprehensively grasp the extent of a system's trustworthiness. To this end, the definition of trustworthiness metrics – rather than requirements or pillars – represents an open issue of the utmost importance. Trustworthiness metrics make it possible to evaluate the extent of a system trust, allowing for a more detailed classification of the AI components to be deployed and the ones to block. However, the definition of a single general, flexible, and ubiquitous trustworthiness metric is made almost impossible by the same issues that affect the generality of trustworthiness requirements. Therefore, we here consider to translate the trustworthiness requirements into a set of equivalent trustworthiness metrics, taking into account the *high variability characterising AI systems* and the *conceptual complexity of trustworthiness building blocks*.

We first consider the issue connected with the *high variability characterising AI systems*. To enable the definition of rigorous trustworthy metrics, we here propose to consider the transition from the general AI trustworthy requirements to the corresponding pillars for each AI branch. Section II presents a similar transition from trustworthy AI into trustworthy NeSy. A similar transition can be identified for each and every AI domain, obtaining domain-specific detailed trustworthiness requirements. This step enables a stricter definition of trustworthiness for each AI domain, making it possible to focus more specifically on the peculiar approaches, components, and aspects that characterise the domain under analysis.

To tackle the *conceptual complexity of trustworthiness building blocks*, we here propose to avoid focusing on the proposal of single, overly-complex trustworthy metrics with the aim of obtaining a general formulation applicable to any AI system. Rather, we suggest to tackle the measurement of systems' trustworthiness through the adoption of a broad set of highly-specialised metrics that analyse single components of the trustworthiness definition. In this context, we consider proposing a single metric or a set of metrics for each pillar/requirement of trustworthiness. The proposed metrics should focus on a specific issue or feature of the AI system at hand – such as its robustness to specific input perturbation, or the bias towards a specific group –, producing as output a single numeric value, describing its safety level—i.e., how much that issue is alarming for the system. Highly-specialised metrics can then be arbitrarily combined to obtain a dynamic trustworthiness score, depending on the trustworthiness components that are to be considered more relevant for the scenario under examination. This simplified process allows not just the easier definition of each set of trustworthiness metric – e.g., bias metrics, robustness metrics, etc. –, but also the evaluation of set based on a given relevance. Consider for example a scenario where the bias requirement should be considered as more relevant w.r.t. the human oversight requirement. Our approach allows a higher weight to be assigned to the bias metrics before its combination with the human oversight metrics to obtain the general trustworthy measurement. Therefore, we here propose to tackle the trustworthiness measurement issue by adopting a dynamic broad set of highly specific metrics that can be combined depending on the given measurement requirements.

IV. Background on SKI and SKE

In this section we provide an overview of the two NeSy mechanisms we focus on, namely Symbolic Knowledge Injection (Section IV-A) and Symbolic Knowledge Extraction (Section IV-B).

A. Symbolic Knowledge Injection (SKI)

Symbolic Knowledge Injection (SKI) defines the set of NeSy systems characterised by explicit procedures aiming at affecting how subsymbolic components draw their inferences for them to be made consistent with some given symbolic knowledge. In their definition, SKI mechanisms require having a subsymbolic predictor – a.k.a. model – and a given symbolic knowledge which always hold true for the considered context. The given symbolic knowledge should consist of logic formulæexpressed in any logic language of choice. In their scope, SKI mechanisms are designed to either *(i)* leverage the given input symbolic knowledge to enrich the training of the subsymbolic predictor; *(ii)* process the given symbolic knowledge via subsymbolic computations to achieve a novel, more meaningful symbolic knowledge; *(iii)* combine both of the previous processes. To achieve any of these scopes, SKI requires the given symbolic knowledge to be converted into a specific numeric form processable by the subsymbolic potion of the NeSy mechanism, to enable the injection process. More in detail, the converted symbolic knowledge is leveraged by the SKI approach to steer the learning process of the underlying subsymbolic model in any of the following way: *(i)* penalising the subsymbolic component during its training, whenever it violates the given symbolic knowledge, usually through defining a custom-made hybrid loss function; *(ii)* construct (a portion of) the subsymbolic component in such a way to make it reflect the given symbolic knowledge; *(iii)* convert the given symbolic knowledge into numeric-array form to be used as training data for the subsymbolic components of the NeSy system. In other words, SKI can be seen as the proces of optimising subsymbolic predictors in such a way that they are helped by the given symbolical knowledge.

B. Symbolic Knowledge Extraction (SKE)

Symbolic Knowledge Extraction (SKE) represents the set of NeSy approaches accepting subsymbolic predictors as input and producing symbolic knowledge as output. More in detail, SKE mechanisms aim at distilling the knowledge that a subsymbolic predictor has grasped from data into symbolic form, expressed by a set of logic formulæ. SKE enables the construction of a symbolic surrogate model that mimics the behaviour of a subsymbolic component. The obtained symbolic rules may then be exploited to either *(i)* understand and explain the behaviour of the original predictor; or *(ii)* replace subsymbolic components of the system while retaining its learning capabilities; To achieve symbolic knowledge construction, SKE can either *(i)* inspect (even partially) the parameters of the subsymbolic component – i.e., decompositional approaches –; or *(ii)* rely solely on the subsymbolic component's outputs—i.e., pedagogical approaches. Depending on the SKE approach the obtained symbolic knowledge can be under the form of lists of rules, decision trees or decision tables, each of them composed by any statement structure, such as propositional rules, fuzzy rules or any other kind of logic formulæ. In other words, SKE can be seen as the process of optimising symbolic AI components in such a way that their behaviour mimics given subsymbolic components.

V. NeSy Metrics for Trustworthiness

In this section we present the trustworthiness metrics (both available and missing ones) for NeSy systems, specifically focusing on SKI and SKE. We analyse each of the seven trustworthiness pillars/requirements separately to obtain a thorough representation of the state-of-the-art and future directions.

A. Human Oversight

NeSy version of human oversight requirement is defined as the need for assessing to what extent the symbolic and subsymbolic *interaction* of NeSy components helps *improving* informed human-AI interaction and human oversight.

1) Available Metrics: Most approaches to measure human oversight in AI scenarios focus on aspects of human-AI interaction, where explanation of behaviours represents the most important component of the interaction process. As a results, much attention has been paid to the measurement of how explanations could guide people to respond to and predict the AI system behaviour [13]. A large number of studies exist in this realm, which mainly leverage on users to subjectively rate system predictability, likability, etc.[14] While useful in order to define systems predictability, these studies lack the assessment of human influence and control on the AI system at hand. The reason for this is to be found mainly on the black-box and data-driven nature of subsymbolic models that these works take into account. Indeed, most – if not all – subsymbolic models allow for limited control by the human users, given mostly by the data gathering and selection process.

2) Missing Metrics: Unlike pure subsymbolic systems, NeSy models intrinsically enable higher level of human oversight via the integration of symbolic knowledge. However, the extent of such oversight capabilities should be studied in depth through the proposal of ad-hoc metrics that measure how much the behaviour of a NeSy system can be controlled by a human user. To this aim, in the SKI context, we consider proposing a novel metric assessing the impact of the injection process to the underlying model. The impact can be measured as the amount of injected knowledge that is effectively absorbed by the underlying model. The metric would assess the level of available human oversight in SKI systems, allowing for a precise definition of the extent of human control. Meanwhile, the SKE context emphasises the need for measuring the modifiability of the extracted symbolic knowledge from an initial subsymbolic predictor. Indeed, SKE approaches by themselves do not allow for an in-depth control of the model behaviour, but rather enable their inspection. In this context, a desirable solution is represented by refining the extracted knowledge and using it as input for a SKI system acting upon the same subsymbolic model. This process would enable a sort of debugging loop of NeSy systems leveraging both SKE and SKI, with an increased potential for human oversight. Here, we require the definition of an ad-hoc metric capable of assessing the portion of symbolic knowledge that can be extracted, refined and injected back in the system with it being correctly assimilated by the model.

B. Robustness

NeSy version of the robustness requirement is defined as the need for assessing the impact of symbolic (verifiable) and subsymbolic (not verifiable) *interaction* on the *stability* of the NeSy system.

1) Available Metrics: The state-of-the-art picture of NeSy robustness emphasises the lack of a common agreement on the definition of robustness itself, thus leading to diverging works focusing on opposite aspects of NeSy systems. Indeed, in this context, several works focus on highlighting the robustness of NeSy models in terms of their performance over complex or out-of-distribution inputs [15], [16], [17]. Although relevant for pointing out the potential of NeSy approaches, these works propose somehow misleading definitions of robustness, mostly focusing on NeSy flexibility rather than its stability. NeSy systems may perform well on complex and out-of-distribution samples, while suffering instability on small input perturbations—causing robustness collapse. Several other concepts have been taken into account when considering NeSy robustness such as prediction coherence and consistency [18], subsymbolic verification through neuro-symbolic integration [19], avoidance of reasoning shortcuts [20] and many more. However, the majority of these approaches not only assess an ad-hoc concept of robustness, but also focus on its qualitative evaluation thus failing to assess the quantitative aspect required to achieve robustness metrics.

While it is true that there exists some confusion concerning the definition of NeSy robustness, there are few relevant works aiming at defining precise robustness metrics. More in detail, Yang et al. [21] present a novel learning approach for neuro-symbolic programs, showing its robustness against input perturbations in terms of provably safe portion of the learned model. In this context, NeSy robustness against adversarial attacks represents a popular area of research with several works aiming at proving either qualitatively [22] or quantitatively [23] the safety of NeSy approaches. Most of these works define robustness in terms of accuracy degradation over varying input perturbation intensity, independently of the input perturbation type and magnitude.

2) Missing Metrics: As a result of the mixed focus given to NeSy aspects when tackling robustness, several aspect of NeSy robustness and stability have not been thoroughly analysed, yet. Indeed, there exists the need to study if – and to what extent – the stability and verifiability of symbolic AI components is preserved throughout the integration process in NeSy models. In this context, focusing on the SKI realm, we suggest that a measure of integration stability – as the portion of symbolic elements that are correctly integrated in the injected model – is needed here. Such a metric would basically represent the portion of symbolic control that a NeSy system can attain during its integration step. Secondly, also those scenarios where the symbolic elements of NeSy models suffer from some sort of imperfection have to be taken into account. Here, it is important to measure the stability of SKI models when the injected knowledge is altered as a result of some imperfect automation process. Finally, it is also relevant to measure the stability of NeSy systems over symbolic representation variability, to assess how different symbolic representations – e.g., logic formulæ, knowledge graphs, etc. – may impact the integration process. To this end, we propose to measure the performance of SKI integration when two syntactically different yet equivalent chunks of symbolic knowledge are exploited in the same integration process.

C. Data & Knowledge Quality

NeSy version of the data & knowledge quality requirement is defined as the need for ensuring the *quality* of both *data* and *symbolic knowledge* of a NeSy system, along with its accessibility.

1) Available Metrics: Given the impact of data quality on the optimisation process of ML and DL systems, several quality metrics are available, namely: *(i)* class overlap [24], *(ii)* boundary complexity [25], *(iii)* label noise [26], *(iv)* class imbalance [27], *(v)* missing value analysis [28], and many more. Although designed for subsymbolic AI models, these metrics translate to the data-driven component of NeSy systems without particular issues, especially in those systems that follow a neural to symbolic – neuro $\rightarrow$ symbolic [29] – pipeline such as SKE approaches. In this context, these metrics makes it possible to check the correctness of the information that the subsymbolic components of NeSy gather from the data.

2) Missing Metrics: Unlike pure subsymbolic approaches – which rely solely on data for optimisation –, NeSy models gather information from both a data-driven and a symbolic knowledge component. In this context, it is fundamental to assess the level of compatibility or overlap between the data and the symbolic knowledge to be combined. In most NeSy systems quite a strong overlap is required between data and symbolic knowledge in order to avoid optimisation drift issues, where the integrated knowledge contrasts concepts learnt from the data. Meanwhile, a perfect overlap would also not be ideal in NeSy systems, as the optimisation process would gather the same information from both data and symbolic knowledge. Therefore, we here stress the need for new metrics that could measure the conceptual and technical overlap between data and symbolic knowledge at hand. Another relevant aspect to measure in this context is represented by the quality of the symbolic component of the NeSy system. While symbolic AI approaches are verifiable and deemed trustworthy, several NeSy – especially SKI – approaches rely on the integration of knowledge bases given a-priori and defined by human experts. Although mostly reliable, knowledge bases may be either incomplete or imperfect due to the human-centred building process. Therefore, metrics are needed that would make it possible to score knowledge components exploited in NeSy systems.

D. Transparency

NeSy version of the transparency requirement is defined as the *transparency gain* obtained by a NeSy system with respect to its pure subsymbolic components.

1) Available Metrics: When focusing on transparency, most of the available metrics for AI and NeSy models focus on explanations quality evaluation. Generally speaking, explanations quality is characterised by several key attributes [30], namely: *(i)* understandability – i.e., explanation complexity –; *(ii)* completeness – i.e., explanation coverage –; *(iii)* sufficiency of detail – i.e., explanations depth –; *(iv)* usefulness – i.e., explanation applicability –; and *(v)* feeling of satisfaction—i.e., explanation interactivity. By focusing on some of the above attributes, several works propose explainability and transparency metrics for AI and NeSy. Authors in [31] introduce a set of metrics to evaluate interpretability methods through measurements of simplicity, broadness, and fidelity of explanations. Meanwhile, Holzinger et al. [32] introduce a system causability scale to measure explanations quality, based on the notion of causability [33] together with the notion of usability scale. Although designed for explanations in general, these metrics nicely fit in the SKE frame, where they can be used to assess the quality of the extraction mechanism, as done in [34], where authors focus on unambiguity, interpretability, and interactivity of explanations.

2) Missing Metrics: Available explainability metrics aim at measuring the quality of explanations in absolute terms—i.e., how good are my extracted explanations? Meanwhile, our definition of NeSy transparency requires to measure the *gain* in transparency obtained from symbolic and subsymbolic integration. Therefore, there is the need for novel metrics for NeSy systems comparing the quality of a system's explanations before and after symbolic and subsymbolic integration. Moreover, we here stress the unbalanced nature of explainability metrics, as most metrics focus solely on features of explanations that are automatically measurable – e.g., correctness, coverage, length, etc. –, whereas there are basically no metrics focusing on human oriented specifications. A relevant issue for future research in this are is the definition of metrics that account for the subjective human factor in explanations, assessing the level of explanations satisfaction and understandability via human-assisted experimentation. Finally, it should be noted that transparency should not just focus on measuring the quality of the explanations that can be obtained from a system, but should instead assess the complexity of the process for extracting those explanations, too. Indeed, explanations obtained from a DL model using SKE may be complete, understandable and useful, but require a high computational burden to be extracted, rendering the overall DL and SKE process less transparent.

E. Fairness

NeSy version of fairness requirement is defined as the need for *measuring* biased and discriminative behaviour of NeSy agents rooted in the *interaction* between their symbolic and subsymbolic components.

1) Available Metrics: Given the nuances characterising a context-dependent notion like fairness, developing quantitative formulations for fairness metrics is challenging [35]. In the general context of AI systems, fairness is generally regarded as *outcome fairness*, which is the definition of equality of the decision making process outcomes. Here, fairness can be categorised into individual vs. group notions of fairness, and observational vs. causal approaches to assess fairness [36]. Observational fairness approaches are characterised by a number of existing metrics, such as: *(i) independence metrics* – e.g., statistical parity, group fairness, demographic parity, etc. –; *(ii) separation metrics* – e.g., equal opportunity, equalised odds, predictive equality, etc. –; and *(iii) sufficiency metrics*—e.g., groups calibration, predictive parity, etc.

While representing a fundamental requirement, fairness in NeSy setups is yet to be explored in detail. Indeed, only a handful of works have investigated fairness in NeSy systems. Authors in [37] propose to leverage the combination of symbolic knowledge extraction from Logic Tensor Networks [38] and injection of fairness constraints via continual learning to enforce fairness. Gao et al. [39] inject a fairness-based component in the loss function of subsymbolic models during their optimisation process to achieve higher fairness. Beyond their obvious relevance, these work focus solely on possible fairness benefits obtained through NeSy, as they rely on the application of SKI and SKE to reduce bias issues, leveraging the general AI fairness metrics. Therefore, available NeSy-specific fairness metrics are still missing that would aim at measuring just the impact of symbolic and subsymbolic integration upon fairness. This deficit is probably due to two aspects: *(i)* most observational fairness metrics are considered to be applicable to NeSy systems without modification; and *(ii)* most research focuses on measuring the fairness and assess it, rather than aiming at identifying its root causes.

2) Missing Metrics: In its NeSy version, the fairness requirement highlights the need to assess the possible fairness issues or improvements that arise from the use of symbolic and subsymbolic integration. It is clear that this requirement is not satisfied by available fairness metrics. Indeed, although most observational fairness metrics apply to NeSy systems, they do not allow for identification of the root causes of bias. One approach to tackle this issue would be to measure NeSy fairness as a differential of observational fairness between a SKI/SKE model and its ML/DL counterpart. However, such an approach would be over-simplistic, as it would not allow the specific sub-components of the integration process or of the symbolic knowledge that impact fairness to be captured. One possible solution would be to measure the fairness of NeSy systems over a set of symbolic knowledge bases, each representing a specific set of fairness goal. This process would allow fairness goal to be decomposed into its components/elements, then measure how well a NeSy system can enforce each fairness element.

F. Resource Efficiency

NeSy version of the resource efficiency requirement is defined as the need for assessing the *gain* in terms of *sustainability* with respect to pure subsymbolic counterparts.

1) Available Metrics: When dealing with resource efficiency of AI systems in general, the detailed definition of the set of resources to take into account represents a fundamental aspect. Several elements of the system at hand can be identified as resources, ranging from the energy required by the system to be optimised to its scalability—e.g., overall complexity. In this context, Agiollo et al. [40], [41] propose a rigorous definition of resource efficiency improvements achievable by SKI systems spanning over four different resource components. More in detail, the authors focus on the definition of energy, latency, memory, and data efficiency of any SKI model, aiming at addressing its environmental impact – e.g., energy and data –, and its scalability—e.g., memory and latency. These metrics are defined as the relative difference in terms of resources – e.g., energy, etc. – required to optimise a SKI model to reach the same level of performance of a subsymbolic counterpart. To this end, the authors define each of the resource analysed, and provide for a tool to measure them in a SKI setup, showing how SKI can improve energy and data efficiency, while degrading the system latency. Latency increments are linked with the increased complexity of the system given by the interaction between symbolic and subsymbolic components, which is however beneficial in terms of number of data required for optimising the model. Indeed, several other works show the data efficiency of NeSy models – such as [42], [43], [44] – even though they lack a proper definition for efficiency.

2) Missing Metrics: As data efficiency represents one of the declared advantages of NeSy systems, most of the literature focuses specifically on this aspect, leaving some space for investigation about other relevant aspects of resource efficiency. More in detail, detailed analysis of the environmental impact of AI and NeSy models development in terms of their carbon footprint are still mostly missings. Studying the energy consumption of the development of a single NeSy model is not enough, as the computation infrastructure used throughout this development – such as clusters and cloud infrastructures – strongly impact its environmental footprint. Moreover, whereas few metrics exist that assess the efficiency of NeSy under the SKI perspective, there are basically no metrics for resource efficiency in the SKE area. In this context, it would be desirable to have metrics similar to the ones obtained for SKI comparing the resource usage of the original subsymbolic model and its symbolic emulation. Depending on the SKE approach at hand, it is possible to consider extracting a small symbolic AI models mimicking the behaviour big DL frameworks. The small symbolic model obtained may help hugely reducing the amount of resources – especially energy, latency, and memory – required to deploy the AI system. Therefore, we here suggest as a future direction to investigate whether – and to what extent – SKE can produce small and fast counterparts of DL models. Here, the resource efficiency metric could be simply designed as the relative difference between the amount of resources required to run the original DL model and its symbolic emulation.

G. Accountability

NeSy version of the accountability requirement is defined as the need for assessing the *gain* in terms of *answerability* obtained by a NeSy system with respect to its pure subsymbolic components.

1) Available Metrics: As it is represented by the answerability of an AI system, accountability is closely tight to transparency. Indeed, accountability requires the underlying system to be explainable, and the explanations to be correct, reliable, and comprehensible. Correctness and reliability of explanations depend on the precision of the AI system and its explanation construction counterpart. Therefore, most efforts in this field focus on the explainability of the AI/NeSy system at hand. As a result, the set of available AI and NeSy metrics for accountability is basically represented by the same set of metrics presented in Section V-D.

2) Missing Metrics: While being tightly linked with explainability, accountability also requires the extracted explanations to be correct and reliable. As correctness and reliability mostly depend on the precision of the AI/NeSy system, we here propose to define novel accountability metrics by opportunistically mixing transparency metrics (Section V-D) and robustness metrics (Section V-B). Therefore, accountability metrics should be defined as the result of explainability metrics applied over a set of input perturbations, measuring the rate of change of the obtained explanations.

VI. Conclusions

Trustworthiness of AI systems represents a fundamental requirement for their ubiquitous deployment. The notion of Trustworthy AI as defined by the EU is mostly a general one, yet implicitly accounting for issues coming from popular ML and DL techniques—so it fits well subsymbolic AI systems. A set of novel NeSy systems calls for a more specific definition of trustworthiness, as they rely on the integration of subsymbolic and symbolic AI where the symbolic components may affect – either positively or negatively – the trust level of the system. Accordingly, in this paper we analyse how the AI trustworthiness requirements defined by the EU translate to the NeSy realm, focusing on the relevant elements of the NeSy integration process impacting trust. First we analyse in detail each pillar of trust and its implication on NeSy models, then we focus on the available metrics for measuring such requirements. The state-of-the-art analysis highlights a lack of available metrics for most trustworthiness aspects when specifically considering NeSy systems. Therefore, we suggest potential future directions to explore in the analysis of NeSy trust along with with related metrics definitions. We believe that the rigorous definition of novel trust metrics tailored to NeSy systems is going to represent an essential step towards measurably reliable and trustworthy AI systems based on neuro-symbolic integration.

Acknowledgment

This work was partially supported by PNRR – M4C2 – Investimento 1.3, Partenariato Esteso PE00000013 – “FAIR—Future Artificial Intelligence Research” – Spoke 8 “Pervasive AI”, funded by the European Commission under the NextGenerationEU programme, and by the CHIST-ERA IV project “Expectation” – CHIST-ERA-19-XAI-005 –, co-funded by EU and the Italian MUR (Ministry for University and Research).

References

[1] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8627998

[2] A. Agiollo, G. Ciatto, and A. Omicini, “*Shallow2Deep*: Restraining neural networks opacity through neural architecture search,” in *Explainable and Transparent AI and Multi-Agent Systems*, ser. Lecture Notes in Computer Science, D. Calvaresi, A. Najjar, M. Winikoff, and K. Främling, Eds. Cham: Springer, 2021, vol. 12688, pp. 63–82. [Online]. Available: http://link.springer.com/10.1007/978-3-030-82017-6_5

[3] D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9075398

[4] A. Agiollo, L. C. Siebert, P. K. Murukannaiah, and A. Omicini, “The quarrel of local post-hoc explainers for moral values classification in natural language processing,” in *Explainable and Transparent AI and Multi-Agent Systems*, ser. Lecture Notes in Computer Science, D. Calvaresi, A. Najjar, A. Omicini, R. Aydoğan, R. Carli, G. Ciatto, Y. Mualla, and K. Främling, Eds. Springer, 2023, vol. 14127, ch. 6. [Online]. Available: http://link.springer.com/10.1007/978-3-031-40878-6_6

[5] Z. Zhang, P. Cui, and W. Zhu, “Deep learning on graphs: A survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 249–270, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9039675

[6] A. Agiollo and A. Omicini, “GNN2GNN: Graph neural networks to generate neural networks,” in *Uncertainty in Artificial Intelligence*, ser. Proceedings of Machine Learning Research, J. Cussens and K. Zhang, Eds., vol. 180. ML Research Press, Aug. 2022. ISSN 2640-3498 pp. 32–42, proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands. [Online]. Available: https://proceedings.mlr.press/v180/agiollo22a.html

[7] A. Agiollo, E. Bardhi, M. Conti, R. Lazzeretti, E. Losiouk, and A. Omicini, “GNN4IFA: Interest flooding attack detection with graph neural networks,” in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, IEEE Computer Society. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2023. ISBN 978-1-6654-6512-0 pp. 615–630. [Online]. Available: https://www.computer.org/csdl/proceedings-article/eurosp/2023/651200a615

[8] K. Benidis, S. S. Rangapuram, V. Flunkert, Y. Wang, D. C. Maddix, A. C. Türkmen, J. Gasthaus, M. Bohlke-Schneider, D. Salinas, L. Stella, F. Aubet, L. Callot, and T. Januschowski, “Deep learning for time series forecasting: Tutorial and literature survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 121:1–121:36, 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3533382

[9] A. Agiollo, M. Conti, P. Kaliyar, T. Lin, and L. Pajola, “DETONAR: Detection of routing attacks in RPL-based IoT,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1178 – 1190, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9415869

[10] J. Zhang and C. Li, “Adversarial examples: Opportunities and challenges,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2578–2593, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8842604

[11] A. C. Serban, E. Poll, and J. Visser, “Adversarial examples on object recognition: A comprehensive survey,” *ACM Computing Surveys*, vol. 53, no. 3, pp. 66:1–66:38, 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3398394

[12] C. Novelli, M. Taddeo, and L. Floridi, “Accountability in artificial intelligence: what it is and how it works,” *AI & SOCIETY*, pp. 1–12, 2023. [Online]. Available: https://link.springer.com/10.1007/s00146-023-01635-y

[13] M. M. A. de Graaf and B. F. Malle, “How people explain action (and autonomous intelligent systems should too),” in *2017 AAAI Fall Symposia, Arlington, Virginia, USA, November 9-11, 2017*. AAAI Press, 2017, pp. 19–26. [Online]. Available: https://aaai.org/ocs/index.php/FSS/FSS17/paper/view/16009

[14] C. Huang and B. Mutlu, “Robot behavior toolkit: generating effective social behaviors for robots,” in *International Conference on Human-Robot Interaction, HRI'12, Boston, MA, USA - March 05 - 08, 2012*, H. A. Yanco, A. Steinfeld, V. Evers, and O. C. Jenkins, Eds. ACM, 2012, pp. 25–32. [Online]. Available: https://dl.acm.org/doi/10.1145/2157689.2157694

[15] Z. Li, X. Wang, E. Stengel-Eskin, A. Kortylewski, W. Ma, B. V. Durme, and A. L. Yuille, “Super-CLEVR: A virtual benchmark to diagnose domain robustness in visual reasoning,” *CoRR*, vol. abs/2212.00259, 2022. [Online]. Available: https://arxiv.org/abs/2212.00259

[16] C. W. Wu, A. C. Wu, and J. Strom, “DeepTune: Robust global optimization of electronic circuit design via neuro-symbolic optimization,” in *IEEE International Symposium on Circuits and Systems, ISCAS 2021, Daegu, South Korea, May 22-28, 2021*. IEEE, 2021, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/9401488

[17] A. Liu, H. Xu, G. Van den Broeck, and Y. Liang, "Out-of-distribution generalization by neural-symbolic joint training," in *AAAI Conference on Artificial Intelligence*, vol. 37, no. 10, 2023, pp. 12 252–12 259. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/26444

[18] M. I. Nye, M. H. Tessler, J. B. Tenenbaum, and B. M. Lake, "Improving coherence and consistency in neural sequence models with dual-system, neuro-symbolic reasoning," in *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021, pp. 25 192–25 204. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/d3e2e8f631bd9336ed25b8162aef8782-Abstract.html

[19] X. Xie, K. Kersting, and D. Neider, "Neuro-symbolic verification of deep neural networks," in *Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022*, L. De Raedt, Ed. ijcai.org, 2022, pp. 3622–3628. [Online]. Available: https://www.ijcai.org/proceedings/2022/503

[20] E. Marconato, G. Bontempo, E. Ficarra, S. Calderara, A. Passerini, and S. Teso, "Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal," *CoRR*, vol. abs/2302.01242, 2023. [Online]. Available: https://arxiv.org/abs/2302.01242

[21] C. Yang and S. Chaudhuri, "Safe neurosymbolic learning with differentiable symbolic execution," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. [Online]. Available: https://openreview.net/forum?id=NYBmJN4MyZ

[22] M. R. Vilamala, T. Xing, H. Taylor, L. Garcia, M. Srivastava, L. M. Kaplan, A. D. Preece, A. Kimmig, and F. Cerutti, "DeepProbCEP: A neuro-symbolic approach for complex event processing in adversarial settings," *Expert Systems with Applications*, vol. 215, pp. 119 376:1–26, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422023946

[23] G. Ibarra-Vázquez, G. Olague, M. Chan-Ley, C. Puente, and C. Soubervielle-Montalvo, "Brain programming is immune to adversarial attacks: Towards accurate and robust image classification using symbolic learning," *Swarm and Evolutionary Computation*, vol. 71, p. 101059, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650222000311

[24] M. Denil and T. P. Trappenberg, "Overlap versus imbalance," in *Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, A. Farzindar and V. Keselj, Eds., vol. 6085. Springer, 2010, pp. 220–231. [Online]. Available: https://link.springer.com/10.1007/978-3-642-13059-5_22

[25] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. de Souto, and T. K. Ho, "How complex is your classification problem?: A survey on measuring classification complexity," *ACM Computing Surveys*, vol. 52, no. 5, pp. 107:1–107:34, 2019. [Online]. Available: https://dl.acm.org/doi/10.1145/3347711

[26] C. G. Northcutt, L. Jiang, and I. L. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021. [Online]. Available: https://jair.org/index.php/jair/article/view/12125

[27] Y. Lu, Y. Cheung, and Y. Y. Tang, "Bayes imbalance impact index: A measure of class imbalanced data set for classification problem," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3525–3539, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8890005

[28] D. C. Corrales, J. C. Corrales, and A. Ledezma, "How to address the data quality issues in regression models: A guided process for data cleaning," *Symmetry*, vol. 10, no. 4, p. 99, 2018. [Online]. Available: https://www.mdpi.com/2073-8994/10/4/99

[29] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, "Neuro-symbolic artificial intelligence," *AI Communications*, vol. 34, no. 3, pp. 197–209, 2021. [Online]. Available: https://content.iospress.com/articles/ai-communications/aic210084

[30] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable AI: challenges and prospects," *CoRR*, vol. abs/1812.04608, 2018. [Online]. Available: http://arxiv.org/abs/1812.04608

[31] A. Nguyen and M. R. Martínez, "On quantitative aspects of model interpretability," *CoRR*, vol. abs/2007.07584, 2020. [Online]. Available: https://arxiv.org/abs/2007.07584

[32] A. Holzinger, A. M. Carrington, and H. Müller, "Measuring the quality of explanations: The system causability scale (SCS)," *KI - Künstliche Intelligenz*, vol. 34, no. 2, pp. 193–198, 2020. [Online]. Available: https://link.springer.com/10.1007/s13218-020-00636-z

[33] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, "Causability and explainability of artificial intelligence in medicine," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 4, pp. e1312:1–13, 2019. [Online]. Available: https://wires.onlinelibrary.wiley.com/doi/full/10.1002/widm.1312

[34] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Interpretable & explorable approximations of black box models," *CoRR*, vol. abs/1707.01154, 2017. [Online]. Available: http://arxiv.org/abs/1707.01154

[35] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Matroids, matchings, and fairness," in *22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 2019, pp. 2212–2220. [Online]. Available: http://proceedings.mlr.press/v89/chierichetti19a.html

[36] R. Calegari, G. G. Castañé, M. Milano, and B. O'Sullivan, "Assessing and enforcing fairness in the AI lifecycle," in *32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)*. Macau, China: IJCAI, August 19–25 2023.

[37] B. Wagner and A. d'Avila Garcez, "Neural-symbolic integration for fairness in AI," in *AAAI-MAKE 2021 – Combining Machine Learning and Knowledge Engineering*, ser. CEUR Workshop Proceedings, A. Martin, K. Hinkelmann, H. Fill, A. Gerber, D. Lenat, R. Stolle, and F. van Harmelen, Eds., vol. 2846. CEUR-WS.org, 2021. [Online]. Available: https://ceur-ws.org/Vol-2846/paper5.pdf

[38] S. Badreddine, A. S. d'Avila Garcez, L. Serafini, and M. Spranger, "Logic tensor networks," *Artificial Intelligence*, vol. 303, pp. 103 649:1–39, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221002009

[39] X. Gao, J. Zhai, S. Ma, C. Shen, Y. Chen, and Q. Wang, "FairNeuron: improving deep neural network fairness with adversary games on selective neurons," in *44th International Conference on Software Engineering, ICSE 2022*. ACM, 2022, pp. 921–933. [Online]. Available: https://dl.acm.org/doi/10.1145/3510003.3510087

[40] A. Agiollo, A. Rafanelli, and A. Omicini, "Towards quality-of-service metrics for symbolic knowledge injection," in *WOA 2022 – 23rd Workshop "From Objects to Agents"*, ser. CEUR Workshop Proceedings, A. Ferrando and V. Mascardi, Eds., vol. 3261. Sun SITE Central Europe, RWTH Aachen University, 2022. ISSN 1613-0073 pp. 30–47. [Online]. Available: http://ceur-ws.org/Vol-3261/paper3.pdf

[41] A. Agiollo, A. Rafanelli, M. Magnini, G. Ciatto, and A. Omicini, "Symbolic knowledge injection meets intelligent agents: QoS metrics and experiments," *Autonomous Agents and Multi-Agent Systems*, vol. 37, no. 2, pp. 27:1–27:30, Jun. 2023. [Online]. Available: https://link.springer.com/10.1007/s10458-023-09609-6

[42] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=rJgMlhRctm

[43] Q. Zhang, L. Wang, S. Yu, S. Wang, Y. Wang, J. Jiang, and E. Lim, "NOAHQA: Numerical reasoning with interpretable graph question answering dataset," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. ACL, 2021, pp. 4147–4161. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.350/

[44] B. Škrlj, M. Martinc, N. Lavrač, and S. Pollak, "autoBOT: evolving neuro-symbolic representations for explainable low resource text classification," *Machine Learning*, vol. 110, no. 5, pp. 989–1028, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10994-021-05968-x

DOI: 10.15439/2023F3385
ISSN 2300-5963 ACSIS, Vol. 35

Deciphering Clinical Narratives – Augmented Intelligence for Decision Making in Healthcare Sector

Lipika Dey
TCS Research
India
lipika.dey@tcs.com

Sudeshna Jana
TCS Research
India
sudeshna.jana@tcs.com

Tirthankar Dasgupta
TCS Research
India
dasgupta.tirthankar@tcs.com

Tanay Gupta
TCS Research
India
gupta.tanay@tcs.com

***Abstract*—Clinical notes that describe details about diseases, symptoms, treatments, and observed reactions of patients to them, are valuable resources to generate insights about the effectiveness of treatments. Their role in designing better clinical decision making systems is being increasingly acknowledged. However, the availability of clinical notes is still an issue due to privacy violation concerns. Hence most of the work done are on small datasets and neither the power of machine learning is fully utilized, nor is it possible to validate the models properly. With the availability of the Medical Information Mart for Intensive Care (MIMIC-III v1.4) dataset for researchers though, the problem has been somewhat eased. In this paper we have presented an overview of our earlier work on designing deep neural models for prediction of outcomes and hospital stay for patients using MIMIC data. We have also presented new work on patient stratification and explanation generation for patient cohorts. This is early work targeted towards studying trajectories for treatment for different cohorts of patients, which can ultimately lead to discovery of low-risk models for individual patients to ensure better outcomes.**

***Index Terms*—Clinical Notes, BioNER, Clustering, Anomaly Detection, Autoencoder, Shapley Value**

I. Introduction – Clinical Narratives and Their Uses

COMMUNICATION within healthcare systems is dominated by textual narratives. This includes a diverse array of documents generated by different sources for various purposes. These texts can be broadly classified as follows:

(a). Clinical notes, especially nursing notes attached to Electronic Health Records (EHR) of patients admitted to the hospital for treatment, contain valuable information about the patients, symptoms, diagnoses, treatments, chronic and past ailments, drug prescriptions, and adverse drug effects, if any, for the patient. Collections of such texts can be effectively mined to gather insights to improve healthcare for other patients [1]. If not explicitly, these texts also provide insights about a physician's possible reasons for following a path of treatment. Nursing notes are time-stamped and therefore can provide a view into the trajectories of recovery. Clinical notes are also highly sensitive in nature since they contain personal identification details. Hence, though large volumes of clinical data are generated across the world, restricted access to the data due to security and privacy concerns is a major bottleneck for researchers. The publicly available Medical Information Mart for Intensive Care (MIMIC) database [2] has eased out this problem to some extent. This data has been anonymized and specially made available for research purposes only. This database contains daily records of over $40,000$ patients with details about their illness, symptoms, medical history, results of diagnostic tests, treatments, nursing observations, discharge summaries along with some patient demographic details like age, gender etc. along with the number of days spent in ICU and other wards for each admission.

(b). Pathological and radiology reports - These are semi-structured narratives resulting from various targeted examinations like findings from radiology images, blood tests, etc. Researchers in the area of computer vision have been long engaged in developing predictive systems from the images for automated detection of diseased cells, tissues, or organs. Recent developments in the area of multi-modal analytics have spurred interest in using textual descriptions along with images for better predictive models.

(c). Bio-medical literature or technical publications that report scientific advances in the area of life sciences and healthcare, include documents like journal articles, case studies, systematic reviews, and clinical guidelines published by regulatory bodies. These documents are important sources of information for those who work in the areas of drug discovery, designing new treatment protocols, etc. Text mining of bio-medical scientific literature is an old established area. The aim is to come up with systems that can help in the easy assimilation of knowledge from this vast incremental repository using sophisticated information extraction and reasoning methodologies. A comprehensive review of text mining techniques for extraction information from bio-medical texts is presented in [3].

(d). Social Media texts - Patient-generated text like tweets or blog posts play a critical role in gathering insights about individual and collective experiences about a drug, treatment, clinical trials, or care facility. Social media text analysis has played a critical role in detecting and assimilating adverse drug effects, especially in obtaining new knowledge about preconditions or co-occurring conditions that cause adverse effects of a drug. With the increasing popularity of patient support groups like PatientsLikeMe, social media content analysis is gaining new heights. There are dedicated groups for different diseases offering hitherto undiscovered insights about rare diseases to the entire community [4].

The repository of clinical texts is huge, analysis of which can yield deep insights for clinical decision-making, drug discovery, and healthcare management systems. While text mining from reports, literature, and social media has been actively pursued for some time now, the mining of clinical notes from EHRs is a fairly new area, primarily due to the non-availability of such texts earlier. Even to this day, the volume of clinical records of patients available for study purposes is fairly low. This is a severe impediment to the development of machine learning systems, as these are known to require large volumes of data. Nevertheless, with increasing focus on personalized and optimized healthcare management, the analysis of clinical texts is gaining importance. Analytical applications of clinical documents can be broadly classified as -

- Descriptive analytics: which is targeted at knowledge discovery from the scientific literature. This is a fairly mature area of research actively pursued by researchers working in the areas of natural language processing and text mining. They often aim to discover information about new entities and relations reported in the literature. Dalianis provides a detailed review of clinical text mining and its applications along with the challenges of analyzing such data in the book [1]. This book also presents a comprehensive study of clinical text mining in non-English languages.
- Diagnostic analytics: this area digs deeper into clinical texts to unearth causal explanations about events.
- Predictive analytics: this area seeks to employ predictive models to predict possibilities of repeat occurrence of known events. One of the major consumers of predictive models designed using past patient data are hospital management authorities. Hospital admission notes have turned out to be very useful for the purpose, as these can be used to predict ICU length of stays [5], hospital readmissions, procedure requirements, etc. ICU stay prediction is an important problem for hospitals, since ICU facilities are expensive to set up, and their optimal use and availability are imperative to ensure better outcomes through proper resource planning [6]–[8]. Obtaining advance information about the possible length of ICU stay or duration of hospitalization are also useful for patients and their families, as it helps in better expectation management and planning from their side also. Predicting individual patient outcomes is gaining importance as clinical decision making is increasingly focusing on individualized care. Given the wide variability among individuals however, a step forward in this direction is to move towards understanding patient cohorts - that is group of patients who are more similar to each other, than to rest of the patients suffering from the same disease. This is known as patient stratification. Patient stratification is gaining interest from medical as well as machine learning researchers, as it holds the promise to deliver better outcomes to the exceptions. These are early days in this area, and most studies have been conducted on very focused dataset.
- Prescriptive analytics: this is an extension of predictive analytics, where the intent is to prescribe the best possible actions from among a set of possible actions, to achieve a desired outcome under a given state. Prescriptive analytics can be viewed as a natural follow-up of patient stratification. It ideally requires simulation of future possibilities to arrive at the best possible decision for an individual, by studying the current state, possible interventions and their effects using a simulation and then arriving at a feasible conclusion.

In this paper, we shall be primarily presenting our work done in the areas of predictive analytics and patient stratification using clinical text like hospital admission notes, with an end goal of providing decision intelligence for better care management and increased visibility into patient cohorts respectively. The rest of the paper is organized as follows. Section II provides an overview of the MIMIC dataset. Section III presents related work done in the area of predictive analytics, along with our earlier work in the area. It presents an overview of different types of predictions done with clinical data including text by different groups, and also an overview of our earlier work done for predicting ICU length of stay and procedure requirements for patients based on the first day's admission notes. Section III-A provides a comparative study of the performance of all the methods. This is followed by patient stratification work. Section IV presents an overview of related work in the area of patient stratification by other researchers. From section V onwards we present our work done in the area of patient stratification using clinical texts, which to the best of our knowledge has not been attempted before. We have presented deep learning based methods to generate explainable clusters from clinical notes of patients admitted with a specific disease. The explanations generated for the clusters provide more insights into the co-morbidities of each cohort present within the group. The cohort-treatment associations are also obtained. We present results of experiments done with pneumonia patients of the MIMIC dataset in section VII. Detailed insights into the cohorts obtained are presented in the form of cluster-wise patient statistics in terms of age and hospital stay, along with cluster-specific association of symptoms, treatments, and final recovery information obtained from the discharge summaries. We believe the insights can pave the way for analyzing the effectiveness of the treatment trajectories and thereby customizing them in future, to improve treatment effectiveness and reduce mortality, if possible. We conclude with a lot of future possibilities in section VIII.

II. The Dataset

As our primary data source, we have used the MIMIC-III v1.4 database [2], which contains the details of over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center (BIDMC) between 2001 and 2012. This database has pre-existing Institutional Review Board (IRB) approval, and researchers can access the data after successfully completing the training course "Data or Specimens Only Research" provided by the Collaborative Institutional Training Initiative (CITI).

The MIMIC-III database contains details of 46,520 distinct patients with 58,976 hospital admissions. This database includes both structured and unstructured clinical events documented for patients during hospital admissions. Importantly, the database adheres to stringent anonymization protocols, meticulously safeguarding patient privacy. Moreover, to ensure heightened privacy protection, precise dates and times of events have been intentionally obscured. Instructions for accessing this dataset are available in the website https://mimic.mit.edu/docs/gettingstarted/.

III. Building Predictive Models for Patient Care with Clinical Notes

In this section, we present an overview of work done in the area of predictive analytics with electronic health records.

In [9], a neural network based model is used to predict the length of remaining hospital stay for a patient at the time of exit from the ICU unit. In this study, authors used several medical attributes like patients' demography, CPT events, services, procedures, diagnosis, etc. of 31,018 patients from the MIMIC database. In another study [13], Harutyunyan et al. proposed a channel-wise LSTM model using multitask training for predicting mortality along with a forecast of the remaining time to be spent in ICU made at each hour of stay. Predictions were generated from 17 clinical variables like Capillary refill rate, blood pressure, fraction inspired oxygen, Glasgow coma scale, glucose level, heart rate, etc. of patients from the mimic database. Afterward, in 2020 [12], a deep learning architecture based on the combination of temporal convolution and pointwise convolution was proposed to predict the length of ICU stay. This work used the eICU critical care dataset [18], which contained records of 118,534 unique patients, and predictions were based on structured features like patients' gender, age, hour of admission, height, weight, ethnicity, Unit Stay, Physician Speciality, etc. In [10], a study was presented on the prediction of length of stay in ICU and mortality, using several machine learning algorithms on a set of patients from the MIMIC database based on their vital signs like heart rate, blood pressure, temperature, respiratory rate, and patient's demography like age, gender, height, weight, etc. In 2021 [11], Su et al. developed several machine learning models for predicting mortality, severity, and length of stay for a set of 2224 Sepsis patients who were admitted to the ICU of Peking Union Medical College Hospital. In their predictive models, authors used patients' clinical parameters such as age, $P(v\text{-}a)CO_2$ $/C(a\text{-}v)O_2$, SO_2, oxygenation index, white blood cell count, oxygen concentration, temperature, etc. from the first 6h in the ICU.

It is worth mentioning here that none of the above-mentioned works used textual data for prediction. Most of them have used only structured clinical parameters for predicting various clinical events. The richness of clinical notes has not been fully exploited for prediction. In particular, nursing notes play a crucial role in capturing essential patient information that extends beyond the physiological metrics recorded by laboratory tests or radiology reports. These notes encompass a wide range of details, including symptoms, overall health condition, administered medications, performed procedures, and devised treatment strategies. Moreover, they occasionally encompass insights into a patient's response to care and treatment, often described through behavioral observations meticulously documented by the caregiving professional.

Figure 1 shows a sample nursing note with different portions of text color-coded, to highlight the different categories of information that a note may contain. Use of linguistic expressions like *"severe multilobar pnx"*, *"worsening multifocal pnx"*, *"No abdominal pain, no further bleeding"* provide an added dimension of human assessment, that cannot be captured through numbers only, but can be important while distinguishing between two similar patients who are possibly responding differently to the treatment. The notes are very comprehensive in nature. With the database containing almost as many notes for each patient as the number of days of admission, this offers quite a rich collection to work with. Additionally, within the EHR system, a discharge summary is also a crucial component of the patient's medical records that provides a concise overview of a patient's hospital stay, their medical condition, treatments received, and instructions for follow-up care upon their discharge from the hospital. Often the information is repeated across these sources. The redundancy helps in data verification, especially since the data can be quite noisy. Other fields which are more structured in nature like age, gender, admission diagnosis, medications, mortality, etc. are also used for predictive modeling.

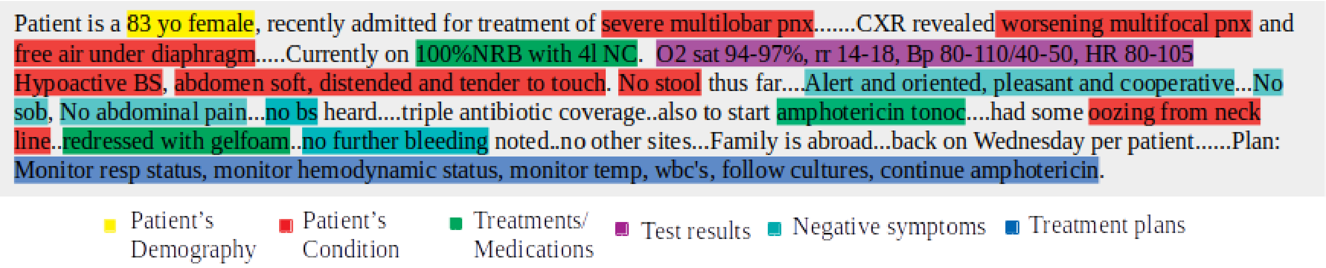
Recently, in 2021 Aken et al. [15], have introduced a model, called CORe, on top of BioBERT for predicting multiple clinical outcomes along with the duration of ICU stay. The authors devised a distinct note extracted from the discharge summary, leveraging it within their predictive framework.

In 2022 [16], we proposed a model which utilized nursing notes of the first day of ICU along with clinical parameters from laboratory tests, to predict whether a patient would need an ICU stay of short or long duration, where the partition of short or long stays was decided by median length of stay recorded in data. Further in [17], this work was extended to additionally predict the need for critical procedures such as bypass surgery, stenting, tracheotomy, and cholecystectomy - which were the most commonly occurring ones in the dataset, along with an ICU stay. The objective was to predict these procedures based on the first day's nursing notes in which these were not explicitly mentioned. We also proposed using a framework called "Local Interpretable Model-agnostic

TABLE I
PERFORMANCE ANALYSIS OF DIFFERENT MODELS FOR ICU LENGTH OF STAY PREDICTION.

Earlier Works	Dataset	Features used	ICU stay classes	Methods	Best Result
[9]	31,018 patients from MIMIC database	patients' demography, CPT events, services, procedures, diagnosis, etc.	ICU stay$\leq$5 days, class 0; ICU stay$>$5 days, class 1	neural network based model	80% accuracy
[10]	44,000 ICU stays from MIMIC database	patient's vital signs - heart rate, BP, temp, resp rate, age, gender, height, weight, etc.	ICU stay$\leq$2.64 days, class 0; ICU stay$>$2.64 days, class 1	Machine learning algorithms	65% accuracy using random forest algorithm
[11]	2224 Sepsis patients from Peking Union Medical College Hospital Intensive Care Medical Information System and Database (PICMISD)	patient's age, P(v-a)CO$_2$ /C(a-v)O$_2$, SO$_2$, oxygenation index, WBC count, oxygen concentration, bpm, temp, etc.	ICU stay (LOS) ($>$6 days, $\leq$ 6 days)	logistic regression, random forest, and XGBoost model	sensitivity = 0.79, specificity = 0.66, F1 score = 0.69, AUC = 0.76 using Random forest
[12]	eICU critical care dataset	patient's gender, age, hour of admission, height, weight, ethnicity, unit stay, Physician Speciality, etc.	classifying in 10 classes - one for ICU stays shorter than a day, seven day-long buckets for each day of the first week, one for stays over one week but less than two, and one for stays over two weeks	combination of temporal convolution and pointwise convolution	Kappa score = 0.58
[13]	MIMIC database 42276 ICU stays	17 structured clinical variables - capillary refill rate, blood pressure, fraction inspired oxygen, Glasgow coma scale, glucose, heart rate, etc from first 24 hours of admission	classifying in 10 classes - one for ICU stays shorter than a day, seven day-long buckets for each day of the first week, one for stays over one week but less than two, and one for stays over two weeks	LSTM-based neural network models	AUC-ROC = 0.84
[14]	22,353 patients from MIMIC database	Clinical Notes	Remaining ICU stay time is discretized into 10 classes {0-1, 1-2, 2-3, 3-4, 4-5, 5-6, 6-7, 7-8, 8-14, 14+}	multi-model neural network	Kappa score = 0.453
[15]	38,013 admissions from MIMIC database	created notes from discharge summaries	four categories - Under 3 days, 3 to 7 days, 1 week to 2 weeks and more than 2 weeks	pre-trained CORe model on top of BioBERT	AUC-ROC = 72.53%
[16]	**22,789 admissions from MIMIC database**	**nursing notes from first 24h along with 20 vital signs and lab measurements available in first 24h of ICU stay**	**"Short" if ICU LOS$<$4days and "Long" if ICU LOS$\geq$4days**	**trans-former based deep neural network model**	**Accuracy = 79.20%, AUC-ROC = 87.33% Kappa score = 0.594**
[17]	**28,659 admissions from MIMIC database**	**nursing notes from first 24h along with 20 vital signs and lab measurements available in first 24h of ICU stay**	**"Short" if ICU LOS$<$3days and "Long" if ICU LOS$\geq$3days**	**multimodal multiobjective transformer network**	**Accuracy = 84%**

Explanations" (LIME) to obtain explanations about the prediction outcome. LIME creates several subsets from the original data containing only a part of the original attributes. It then computes the influence of the attributes on the classification, based on the presence or absence of certain features in the selected text. To capture the details and the various nuances of a nursing note, we proposed using transformer-based models for representing them. Embeddings for the nursing notes were generated using BlueBERT [19] and Clinical BioBERT [20]. Along with this, for each admission, we computed four types of Severity of Illness (SOI) scores: APACHE II, SAPS II, SOFA, and OASIS based on data collected within 24 hours of ICU admission. This way, the proposed model made use of both structured as well as unstructured data. The novelty of the



Patient is a 83 yo female, recently admitted for treatment of severe multilobar pnx.......CXR revealed worsening multifocal pnx and free air under diaphragm.....Currently on 100%NRB with 4l NC. O2 sat 94-97%, rr 14-18, Bp 80-110/40-50, HR 80-105 Hypoactive BS, abdomen soft, distended and tender to touch. No stool thus far....Alert and oriented, pleasant and cooperative...No sob, No abdominal pain...no bs heard....triple antibiotic coverage..also to start amphotericin tonoc....had some oozing from neck line..redressed with gelfoam..no further bleeding noted..no other sites...Family is abroad...back on Wednesday per patient......Plan: Monitor resp status, monitor hemodynamic status, monitor temp, wbc's, follow cultures, continue amphotericin.

Fig. 1. A Sample nursing note collected from MIMIC-III. Colored components are the extracted events and entities.

proposed model was its design as a multi-objective prediction model, where critical procedures also were predicted along with the duration of ICU stay. Different kinds of networks were experimented with, namely the Convolutional Neural Network (CNN) [21] and Long Short Term Memory (LSTM) [22] for prediction. Additionally, it was observed that the use of a Term Frequency - Inverted Document Frequency (TF-IDF) vector, impacts the performance of the model significantly. It might be due to the fact that the TF-IDF vector can capture the common and distinct features across the classes very effectively. Figure 2 presents the prediction architecture that performed the best along with the proposed representation scheme.

The input representation for a patient is the vector generated by concatenating the outputs of the BiLSTM layer as a result of embedding the corresponding first admission note, the TF-IDF feature vector for clinical entities extracted from the note, and the SOI scores of the patient. The concatenated vector embedding is simultaneously fed into two task-specific fully connected layers, one for predicting ICU length-of-stay and another for predicting the possibilities of each of the four surgical interventions – bypass surgery, stenting, tracheotomy, and cholecystectomy. For the length-of-stay classification, we have used the softmax activation function with binary cross-entropy loss L_{LOS}. For the intervention prediction tasks, since these are not mutually exclusive outcomes, we have trained the prediction layer using the sigmoid activation function with binary cross-entropy loss functions $L_{intervention}$. Finally, we have defined a joint loss function using a linear combination of the loss functions for the two tasks as:

$$L_{Joint} = \lambda * L_{loss} + (1 - \lambda) * L_{intervension} \qquad (1)$$

, where λ controls the contribution of losses of the individual tasks in the overall joint loss.

A. Evaluation of Predictive Models for ICU LOS prediction

As discussed earlier, one major aspect that distinguishes the models from each other is the set of predictor variables used. The choice of prediction method is often guided by the choice of the predictor features. The evaluation metrics used by the studies are also not always the same. This makes comparison of methods a little tricky. However, we have presented a compilation of the models, the metrics, and the reported performances of these models in Table I. The last two rows of the table I present the performance of our proposed models over a set of ICU patients from the MIMIC database. The prediction accuracies for bypass surgery, stenting, tracheotomy, and cholecystectomy were found to be $89\%, 83\%, 55\% and 54\%$ respectively. The performance of the last two categories were not so good due to lack of enough data in the set. Interestingly however, while only 3% of the total tracheotomy procedures done later were mentioned in first day's notes, our model could predict 70% of them in the first day. However, the false positive rates were high for the proposed model. This can be reduced with more data. Overall, the significant gains for acquiring valuable insights into procedure requirements on the first day itself are quite significant. Consequently, such an approach offers prospects for enhanced planning and decision-making. It was observed that a total of 15691 unique diseases are recorded in the MIMIC database as key reasons for which patients were admitted to ICU. It was found that in the dataset we used, the topmost category, which constituted about 4% of the entire set were patients of Pneumonia, followed by around 2% each of Sepsis, Coronary Artery Disease, Congestive Heart Failure, and Gastrointestinal bleeding. This somewhat explains the observation of the cardiac procedures as most frequent followed by tracheotomy and cholecystectomy.

As seen in Table I, the performances of predictive systems are improving over time. While these systems are good for hospital management, particularly in efficiently managing resources for heterogeneous sets of patients, when it comes to providing better patient experience, the trajectory of clinical decision-making is moving towards providing more personalized care management. While these are early days of designing personalized care management systems, existing literature strongly suggests that rather than working with very large groups, the stratification of patients into homogeneous subgroups or clusters is likely to play a major role in enabling personalized treatments. In the next section, after describing novel techniques for patient clustering, we will demonstrate the effectiveness of the methods utilizing patient data afflicted with Pneumonia, which was the predominant disease present in the dataset.

IV. Patient Stratification - a review of related works

Most of the predictive models that have been proposed earlier in literature, have worked on large heterogeneous patient

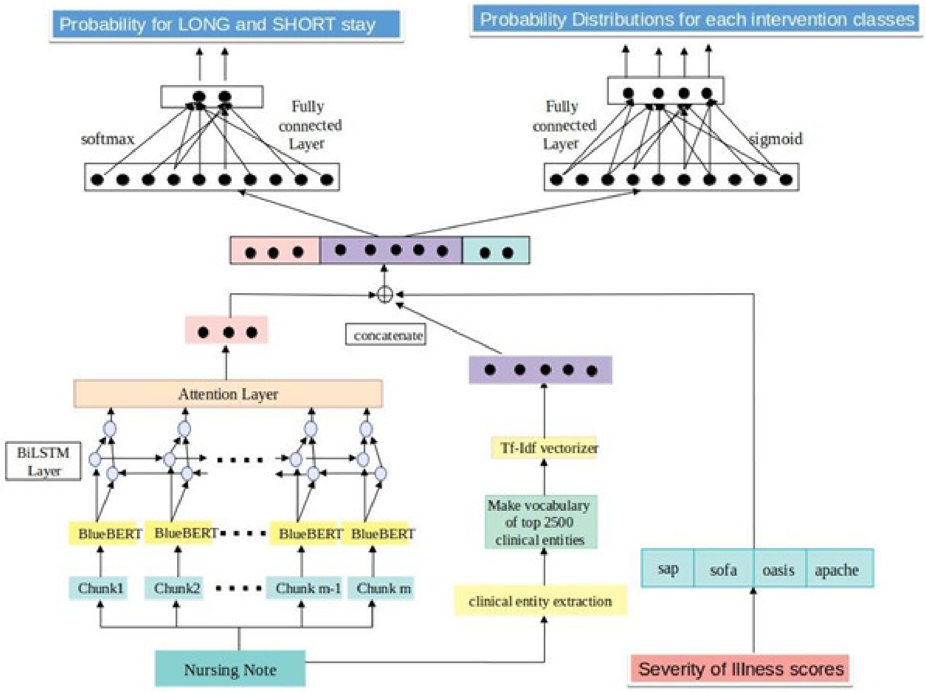


Fig. 2. Overview of proposed multimodal multitask framework for predicting the ICU length of stay and necessity of the interventions. Process the nursing notes in chunks by the BlueBERT model and add a BiLSTM-attention layer on the top. We also extract 2500 medical entities from these notes and make a TF-IDF representation. Then the note representations from the BlueBERT-BiLSTM -Attention network, TF-IDF representation, and four severity of illness scores are concatenated and two task-specific fully connected layers are applied to obtain the final predictions.

populations, which were not able to discover obtain much insights into risk factors for individual patients and hence not very suitable for personalized decision making. Patient stratification techniques that can identify homogeneous groups and thereafter group-specific risk factors, is therefore proposed as a better way to gain insights about outcome differences.

In recent years, several data clustering approaches have been proposed for patient stratification and subsequent analysis of the clusters using different cluster validity indices [23], [24]. In 2021, Alexander et al. [25] investigated the clinical heterogeneity of Alzheimer's disease patients using electronic health records (EHR). In 2022, Angelini et al. [26] proposed an explainable clustering method to identify dominant osteoarthritis endotypes using different biochemical markers, to design tailored treatments and drive drug development. In a recent paper [27], Bhavani et al. have applied hierarchical clustering (DTW-HC) and partitioning around medoids (DTW-PAM) from the first 8 hours of hospitalization records, to identify sub-phenotypes of infected patients. Chen et al. In [28] have presented a model for prediction and risk stratification of kidney outcomes in IgA Nephropath, which is a common disease worldwide. The intent is to predict long-term outcomes and stratifying risk for clinical decision-making. They have also stated that these kinds of work can be useful for designing future clinical trials. The model used was gradient tree boosting implemented in the eXtreme Gradient Boosting (XGBoost) system. The dataset itself was quite small. Kanwal et al. in [29] present stratification of patients suffering from Non-Fatty Liver Disease (NAFLD), which is largely asymptomatic, and success of treatment depends on optimal timing and accurate assessment of fibrosis risk. The work describes the NAFLD Clinical Care Pathway that was developed to assist clinicians in diagnosing and managing NAFLD with clinically significant fibrosis (stage F2–F4) based on the best available evidence using a statistical approach. In [30] Seinen et al. have presented a detailed review of prognostic prediction models that use unstructured clinical text.

V. Patient Stratification Using Nursing Notes - obtaining insights about patient cohorts and exceptions

We now present methods for generating explainable patient cohorts based on their condition at admission, by clustering first day's nursing notes of patients. Rather than using the entire dataset of all patients, our focus will now be on individual diseases. We propose the use of auto-encoders to represent patient health conditions, which is a different approach from using the transformer-based representations presented earlier. Further, we propose the use of SHAP values for explaining the clusters. We also show that the use of autoencoders within a disease category improves the accuracy of prediction of the duration of hospitalization.

While analyzing the prediction results using the LIME framework described in section III, we found that clinical notes, one factor that could affect the performance of prediction could be highly variable in style and content. While some caregivers record only the symptoms that are present on a given day, some others meticulously note down the absence of common symptoms, adverse reactions, psychological state of patients, appetite, etc. The use of non-standard terminology and abbreviations are known to be quite common. Using widely variable terms to describe the same condition is very common in clinical texts. For example - *icterus* and *jaundice* refer to the same disease. Similarly, the terms *brain abscess, intracerebral abscess, cerebral abscess* all refer to the same state. Over the years, bio-medical dictionaries like UMLS [31] have been prepared to document these. Though the BlueBERT embeddings that we used earlier, could capture linguistic nuances like difference between *severe pain* and *mild pain*, these could not always capture the similarities or differences between two notes based on the medical terms used. Therefore, before getting into the stratification work, where the similarity would play a significant role, we implemented an additional processing layer, wherein every clinical note underwent initial processing through the Biomedical dictionaries for standardization of terms. Using this step it was now possible to also distinguish different types of entities like symptoms, diseases, drugs, etc. which could help in grouping patients better. For example, people suffering from the same disease and undergoing the same treatment, but who showed different reactions to it, could now be put into different groups. This led to the idea of using a different embedding altogether which we shall now explain.

The details of the processing pipeline using the biomedical dictionaries are presented below.

Entity extraction: We have used Scispacy [32] and MetaMap [33] for extracting health conditions from patients' clinical notes.

Scispacy: ScispaCy is one of the most robust model for processing biomedical, scientific, and clinical texts on several NLP tasks such as part-of-speech tagging, dependency parsing, named entity recognition, etc. In our work, we have used the pre-trained scispaCy model *en_ner_bc5cdr_md*, which was trained on the BC5CDR corpus for recognizing disease names mentioned in a clinical note.

MetaMap: Nowadays, another entity extraction tool MetaMap is widely used for identifying medical entities. This was developed by the National Library of Medicine (NLM) to map biomedical text to concepts in the Unified Medical Language System (UMLS) [31]. We have processed clinical notes using MetaMap and extracted eight medical entities such as "Sign or Symptom", "Disease or Syndrome", "Acquired Abnormality", "Anatomical Abnormality", "Congenital Abnormality", "Injury or Poisoning", "Mental Process", and "Mental or Behavioral Dysfunction".

Detecting Negations: The Negex algorithm [34] which detects the presence of negative modifiers like "no", "not", etc. is then applied to detect negative mentions of entities in a text. The original list was enhanced to accommodate commonly appearing negation concepts such as "deny", "refuse", "absent", "decline" etc. that occur frequently in clinical notes. For example, given a sentence "*The patient has shortness of breath but denies any chest pain*", two symptoms identified should be *shortness of breath, neg chest pain*. These negative symptoms have a significant contribution to describing individual patients.

Entity Standardization: All the extracted entities are then standardized using the UMLS Metathesaurus [31], [35]. This is important since a single medical condition like "Hypertension" may be referred to as "High blood pressure", "Arterial hypertension" or "Hypertensive disorder" by different professionals. UMLS contains an exhaustive list of such situations and assigns a "Concept Unique Identifier (CUI)" to each. However, we have observed that several entities still did not have an exact match with any UMLS concept. These entities were mapped using an approximate string-matching algorithm [36], that found the closest UMLS concept based on Levenshtein distance measure [37]. For entities that could not be mapped to any UMLS concept, unique identifiers were created to ensure that no health condition was ignored. Examples of such entities from the MIMIC dataset include terms like "airway swelling", "overdistention of lung" etc. To avoid confusion, we refer to these also as CUIs.

Each clinical note can now be represented in terms of the CUIs present in it, either to indicate the presence or absence of a symptom. Consequently, a patient's status at a particular point in time can also be expressed in terms of these CUIs.

Let the collective list of CUIs describing the diseases and symptoms for a particular study be denoted by *health status (H)*. Given a patient p, the health condition at time t is denoted by a vector of $h_i \in H$, where the value of h_i is set to 1 if h_i mentioned in the corresponding clinical notes, -1 if it is mentioned negatively, and 0 if h_i is not mentioned. It may be noted that the CUIs associated with a patient are expected to change over time as treatment progresses. Consequently, the patient may be represented by different vectors over the same space as t changes.

Patient Medication Information: Besides health conditions, each patient also has medications that are prescribed based on these conditions. Considering a unified set of medicines M, at a given time t, each patient p is also associated with a binary vector $< m_i >, i = 1..|M|$, where $m_i = 1$, if medicine m_i is ongoing for p at time t, otherwise 0. This binary vector also changes over time.

A. Creating Dense Representation of Patients in Terms of Health Conditions using Autoencoders

While the number of unique diseases and symptoms obtained from any patient database is very high since all people do not exhibit all symptoms or diseases, the above vectors are high-dimensional and sparse. An autoencoder-based transformation is applied to obtain a dense representation in a lower-dimensional space [38], [39]. In an autoencoder (AE) model, the "encoder" network creates a compressed representation of the input data by capturing the essential characteristics and underlying patterns, while the "decoder" network learns to reconstruct the original input data from the compressed representation while minimizing the loss of information [40]. We next show how dense representations are used for prediction as well as clustering purposes.

B. Autoencoder-based Prediction of Duration of Hospitalization

Before getting into the details of clustering, we validated the representation using it in a predictive framework similar to the ones described earlier first. Figure 3 presents a CNN-based architecture that is used for the prediction of hospital / ICU stay duration using the autoencoded representation of the first day's clinical notes. The network performance was tested with a dataset of 2106 "Pneumonia" patients who had undergone one admission for the disease. The set comprises individuals both with and without ICU admission. Since the number of total patients in this category is still quite small for a deep neural architecture, we have used all the data for predicting long or short hospital stay, rather than ICU stay. The median stay for this set of patients was 9 days. We conducted two experiments to check the performance. For the first experiment, the cut-off between short and long stays was assumed to be 7 days, while for the second experiment, it was assumed to be 9 days. While the accuracy is found to be 83% for 7 days, it is 86% for 9 days, which is the median. Thus median appears to be a good estimator for deciding long or short stays. It was found that the system performed more errors for patients with short stays, which were erroneously classified as long. Some of these patients, had deceased after a short stay. Though not exactly comparable, the prediction performance is found to be much higher than all reported works presented earlier, which may be due to the focused dataset or the representation, or both. Whether this kind of performance can be observed for all other diseases also, needs to be further explored, but this would also need more data for each of the corresponding categories.

VI. Patient Stratification using autoencoders - Generating explainable patient clusters

The autoencoded vector representations are clustered using the k-means clustering algorithm [41], with Euclidean distance as the distance metric to identify similarity among patients. For a given value of k, a set of k cluster centers are chosen randomly, and then each data point is assigned to the cluster that is found by iteratively minimizing the within-cluster distance among the points. In order to determine the right value of k, we have made use of *silhoutte coefficient* [42]. The silhouette coefficient of each point measures how similar it is to other points within the same cluster in comparison to points in other clusters. The average silhouette coefficient computed from all the points provides a measure of the cohesiveness of each cluster along with their separation or distinctiveness from each other. Starting with 2, the value of k is iteratively increased as long as the silhouette score also increases with it. The ideal value of k is the one that yields the highest average silhouette score, beyond which the score starts to decrease steadily.

In order to generate human-interpretable explanations for the clusters, we used Shapley values [43], which can measure the contribution of each feature of each individual towards the final outcome, while preserving the sum of contributions of all. We wanted the explanations to be in terms of diseases and symptoms, including the dominant symptoms in a cluster, as well as the distinguishing aspects between the clusters. The CUI-based representation was used for the purpose. Treating the cluster labels as the target outcomes, a Random Forest classifier [44] was trained to predict the target labels, using the CUI vector-based representation of the patients. The trained model was then analyzed using SHAP TreeExplainer [43], [45], [46], to gain insights into the decision-making process. This method provides not only the contribution of each symptom to a particular label but also the SHAP values for each patient, thereby helping with the interpretation of why a patient has been assigned to a particular cluster. They also help in the interpretation of misclassifications by the model, if any.

A similar approach was followed to derive Shapley values for patients and clusters in terms of medicines. In this case, the Random Forest classifier was trained to predict the target cluster labels using the medicine vectors and then analyzed with the SHAP TreeExplainer. It helps in identifying the most significant medicines for each cluster, thereby providing insights about the common and distinct medicines administered to patients belonging to different clusters.

A. Identifying Anomalies within Clusters

The obtained SHAP values for patients' health conditions are now used to compute the anomaly score for each individual. Anomalous individuals - within a cluster who may demonstrate distinctive characteristics in terms of certain symptoms or response patterns, different from other members, should have higher scores. For a cluster C, the health condition anomaly score of a patient p is denoted by $\alpha_C(p)$ and computed as follows:

$$\alpha_C(p) = \sum_{h \in H} |\omega_h(p, C) * (m_h(C) - v_h(p))|,$$

where H denotes the set of all CUIs, $v_h(p)$ is either 1, 0, or -1 based on whether the symptom h is present for p or not, as described earlier, $m_h(C)$ is the median value of the symptom h in C and $\omega_h(p, C)$ denotes the SHAP value for symptom h for patient p with respect to C. $\alpha_C(p)$ is normalized for each cluster to keep the scores within 0 to 1. The higher anomaly score indicates that the patient is more distinctive from his/her neighbors.

Since the duration of sickness also varies largely among patients, a normalized anomaly score is also computed in terms of duration. For a patient p with duration of sickness d, the

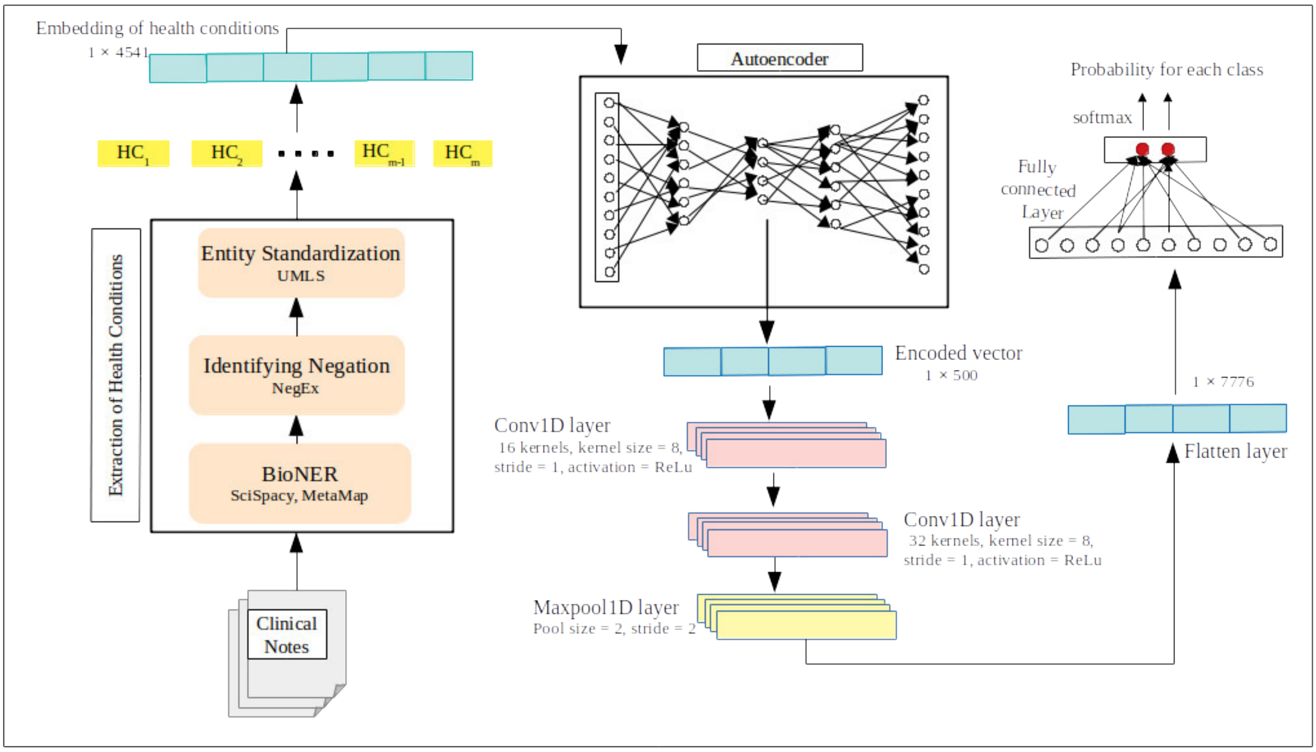


Fig. 3. Overview of the proposed framework for processing clinical notes, generating representations with an autoencoder, and subsequently applying CNN to predict hospitalization duration.

duration anomaly score denoted by $\gamma_C(p)$ with respect to other members of C is computed as follows :

$$\gamma_C(p) = |d - m_C| / \max_{i \in C} \gamma_C(i),$$

where m_C is the mode value of duration of disease for cluster C.

Absolute anomaly score $\chi(p)$ is computed using the following formula:

$$\chi_C(p) = 0.5 * \sqrt{\alpha_C(p)^2 + \gamma_C(p)^2},$$

This distributes the scores within the first quadrant of a unit circle, centered at the origin. The anomaly score is high for the set of points that are far away from the origin, the higher with the x and the y axes values providing insights about the symptom anomalies and duration anomaly respectively.

B. Assessing Causal effects of Significant Medications within Clusters

We further propose the use of causal analysis of clusters to analyze the effect of medications on patients. Causal analysis attempts to identify the effect of different treatments on groups of patients, based on observed outcomes. For patient stratification, the duration of sickness may be considered as an observed outcome, while medications or procedures are the treatment variables.

Causal analysis was done using the DoWhy package [47]. Given the diseases and symptoms as common cause variables, the duration of sickness as the outcome, and the medications as treatments administered, DoWhy generates an initial causal graph then estimands are identified using the graph. The final causal estimates are obtained using Propensity Score Weighting [48] as the estimator, and refutation as the validation technique. The causal estimates are validated using two different methods namely "adding random common cause" and "data subset validation" [49]. While the first method estimates the effect of a treatment by adding random independent variables, the second one does the same taking subsets of data. Either way, causal estimates are assumed to be good if the results don't show high perturbations, indicated by the p *value*.

VII. Results of Patient Stratification for Pneumonia Patients in MIMIC dataset

In this section, we present results for analyzing data of 2106 patients from the MIMIC-III v1.4 database [2], who were all admitted to the hospital and diagnosed as suffering from "Pneumonia", using the above methods. The intent was to:

(a) Obtain explainable clusters of patients based on their health conditions recorded in the first set of clinical notes on admission.

(b) Identify anomalies within each cluster and also generate explanations for the anomalous behaviors.

(c) Identify the significant medicines for each cluster using the SHAP values.

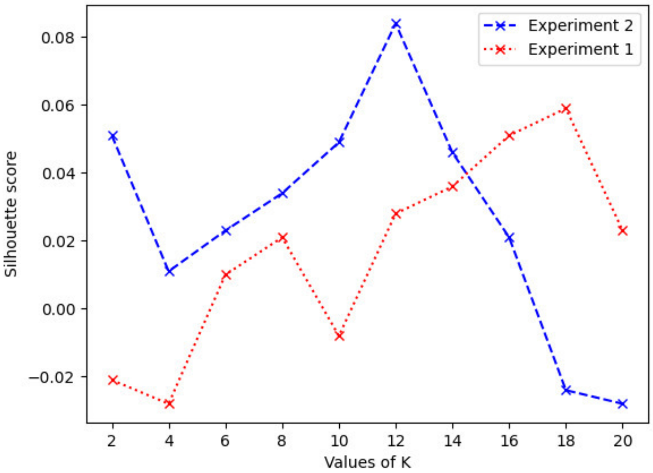


Fig. 4. Average silhouette scores for k = 2 to 20 for Experiment 1 (red) and Experiment 2 (blue).

(d) Study the causal effect of significant medicines on the duration of disease for the clusters.

As mentioned earlier, each of these patients had multiple clinical notes attached to them, approximately one per day of admission. The notes varied quite a bit in content. While some contained only incremental information, some were detailed and overlapped with earlier ones. A total of $23,737$ notes were obtained. After pre-processing all the notes as described in section V, a comprehensive list of 4800 unique health conditions was identified. Initially, the entire set of $23,737$ records was used to generate 500-dimensional autoencoded representations for the patients. However, the clustering results were not satisfactory. Hence, after some experimentation, this was changed to use only the first day's notes. The number of unique health status reduced to 2803. The 500 dimensional auto-encoders were then generated using the vectors for the first-day notes only. Figure 4 presents the average silhouette scores for k values ranging from 2 to 20 for both experiments. The best score was achieved for k equal to 12, using the second method. Figure 5 presents the distribution of these 12 clusters plotted using tSNE [50]. It shows that the clusters are fairly distinct and well-separated. Analysis of the SHAP summaries also revealed that the clusters were distinct and well-segregated from each other.

Since the trade names of drugs varied a lot, though their compositions were same, therefore the drugs corresponding to the notes were obtained from the database and mapped to their drug classes using the pre-trained *gpt-3.5-turbo* model [51]. However, the drug summaries generated by SHAP were not found to be very distinct from each other. This can be because the same drug might be prescribed for two different health conditions, or two different drugs might have been administered for the same health condition by different physicians. This needs to be analyzed in depth further, which remains a future task.

Figure 6 presents the SHAP summaries in terms of major health conditions present and absent, and the drugs identified for a few clusters. It shows that cluster 0 predominantly consists of patients suffering from *Endometriosis* and *Diabetes* along with pneumonia, and do not exhibit *Paroxysmal familial ventricular fibrillation*. Similarly, patients belonging to cluster 4 are found to suffer from *Lung consolidation* and do not have *Paroxysmal familial ventricular fibrillation*. Cluster 6 predominantly comprises patients with *Atrial Premature Complexes* and does not exhibit *Lung Consolidation* or *Paroxysmal familial ventricular fibrillation*. It may be observed that the most common co-morbidity was some or the other form of cardiac disease. The SHAP summary for drugs for clusters 0, 4, and 6, show that the most significant medicine administered to cluster 0 patients is *antidiabetic hormones*, which is obviously to handle diabetes, for patients of cluster is *proton pump inhibitor* and *hypoglycemic agent*, and the most prevalent medication for patients of cluster 6 is found to be *Vasopressor* which is for regulating blood pressure.

Anomaly scores revealed that the most anomalous patient in cluster 0 is a 67-year-old patient with a hospital stay of 62 days, deviating significantly from the cluster's mode value of 7 days. This anomaly can be attributed to the coexistence of Endometriosis, Paroxysmal familial ventricular fibrillation, and Acquired abnormality of atrium, a combination not observed in other members of this group. For cluster 4, while most patients were aged between 60 to 80, the most anomalous patient was a 33-year-old person who spent 81 days in the hospital, against the cluster mode value of 6 days. This person had all significant common symptoms of the cluster along with *Paroxysmal familial ventricular fibrillation*. The top two most anomalous persons of cluster 6 are aged 72 and 52 years, with 58 and 46 days of stays against cluster mode of 8 days. The anomalous symptom for the first person is *renal cyst*, while for the second patient, they are *lung consolidation* and *liver failure*. While most patients in this cluster are aged between 60 to 80, a third anomalous patient is aged 33 and suffered from multiple comorbidities but was admitted for only 3 days. Thus, it can be seen that the anomaly score is able to identify patients who are demographically outliers, even though age was not taken into account for computing the score.

Consequently, causal analysis was done for the drugs that were found to be significant for these clusters. For cluster 0, the effect of *antidiabetic hormone* is -0.459. The negative values indicate that these medicines contributed towards reduced duration of stay. Also, we have observed that this medicine is given 83% of short-stayed patients. On the other hand, the effect of *proton pump inhibitor* is 2.746 and this medicine is given to 89% of long-stayed patients. For cluster 4, the causal effect values of medicine *beta-blocker* was -2.7, for *Analgesic anti-platelet* it was -2.02, and for *Bronchodilator* it was -1.51, contributed towards in decreasing in the duration of hospital stay. *Opioid analgesic* had the highest positive value, indicating that this did not play a role in reducing the duration. For cluster 6, in which 84% of patients suffer from *arterial premature complexes*, medicines *Corticosteroid (-3.17), Anticonvulsant/ neuropathic pain agent (-3.02), Opioid*

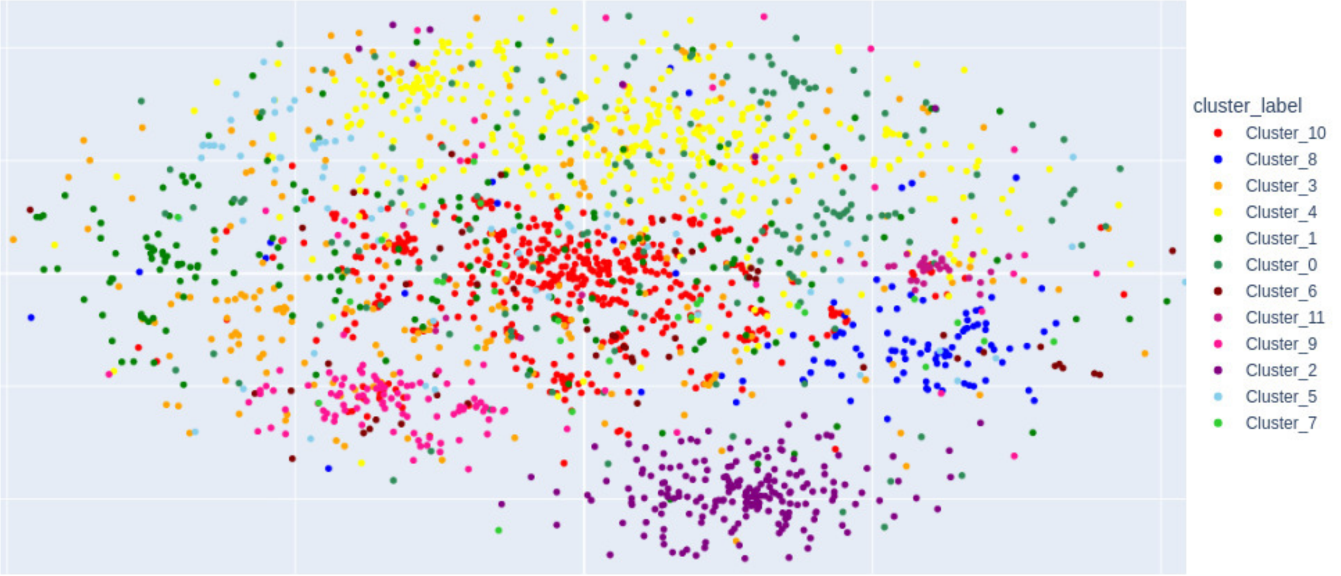


Fig. 5. t-SNE visualization of 12 clusters.

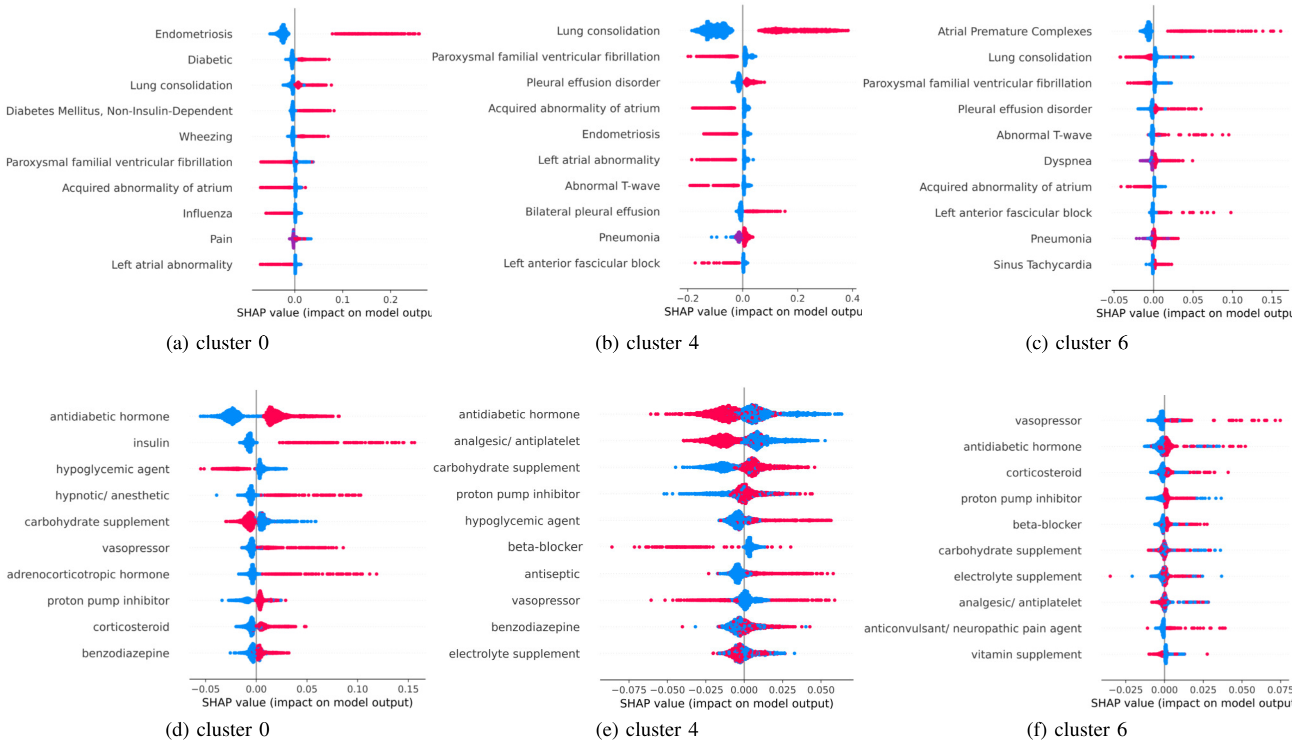
analgesic (-2.4), Vasopressor (-1.02) are found to causally reduce the duration of stay. Further analysis for cluster 6 reveals that 58% of patients with less than 6 days of hospital stay, 52% of those with 6 - 10 days, and 50% of those with higher than 10 days, were administered with *Corticosteroid* on the first day. The manual inspection also reveals that *Corticsteroid* was not administered to the top three anomalous patients, whose details were presented earlier, on the first day. Likewise, we conducted similar analyses for the remaining clusters. Though the exact implications of the results are best analyzed by healthcare experts, our analysis reveals that the results obtained from anomaly detection, their explanations, and causal analysis are all consistent with each other.

We also extracted the final recovery status of the patients from the discharge summaries. Based on the descriptions, we identified three major states at the time of discharge - (a). deceased patients, (b). patients whose vital signs were stable, could ambulate independently, and were coherent, (c). patients who were lethargic but mentally alert and ambulated with assistance. Figure 7 shows the distribution of the different clusters of patients identified earlier (based on their initial states) across these three categories. The top three categories of deceased patients are from clusters 7, 0 and 11, who suffered from comorbidities like ventricular hypertrophy, endometriosis or showed severe signs of edema. The results indicate that there is a need to look deeper into these cohorts, especially into the symptoms presented by the deceased patients. Comparative analysis of lengths of hospital admission for all patient clusters and deceased patients of each cluster are shown in Figure 8. It can be seen that the medians of the deceased patients for each cluster are almost same as those of other patients, which is around 10. While majority of the patients survived, a small percentage could not, and future work would be to delve deeper into the reasons for these differing outcomes. Superficial analysis at this point reveals that these patients had higher number of co-morbidities at admission time, some uncommon ones like *Septic embolus, Kidney Failure, AIDS-Associated Nephropathy, sepsis* etc. Data and the trajectory of treatment of these patients can be analyzed further to see what could have been done to ensure a positive outcome.

Figure 9 presents cluster-wise classification accuracy of predicting LOS for test patients, obtained by the classifier mentioned in section V-B. It is observed that there were no classification error for patients of cluster 6, which also incidentally had the least variation in terms of length of stay for patients as shown in 8. In future we would like to work on prediction of outcomes for patients within individual clusters. We are also exploring the role of SHAP values in explaining the prediction outcomes since though the LIME framework used earlier provides measure of associations between words and classes, it does not provide any explanation about why a particular class was assigned to an individual based on a collection of features.

VIII. Conclusion

Clinical notes are the backbone of healthcare systems. Well-written documents can provide valuable insights about diseases, patients and treatment effectiveness. They can provide a wealth of information about the similarities and diversity of situations across different situations. Deciphering the notes themselves however is a difficult problem due to the inherent variations in terms of style and content, which result from individual and organizational preferences. Obtaining these notes for analytical tasks is also a difficult problem. Though their utility is well-acknowledged, still these are not available in volumes that can help in designing machine-learning models for analyzing them. The key concerns are those of security



(a) cluster 0 (b) cluster 4 (c) cluster 6

(d) cluster 0 (e) cluster 4 (f) cluster 6

Fig. 6. SHAP values for Cluster 0, 4, and 6 of health conditions (top) and drugs (bottom). Red indicates high significance and blue low. Right side indicates presence and left side indicates absence of a feature.

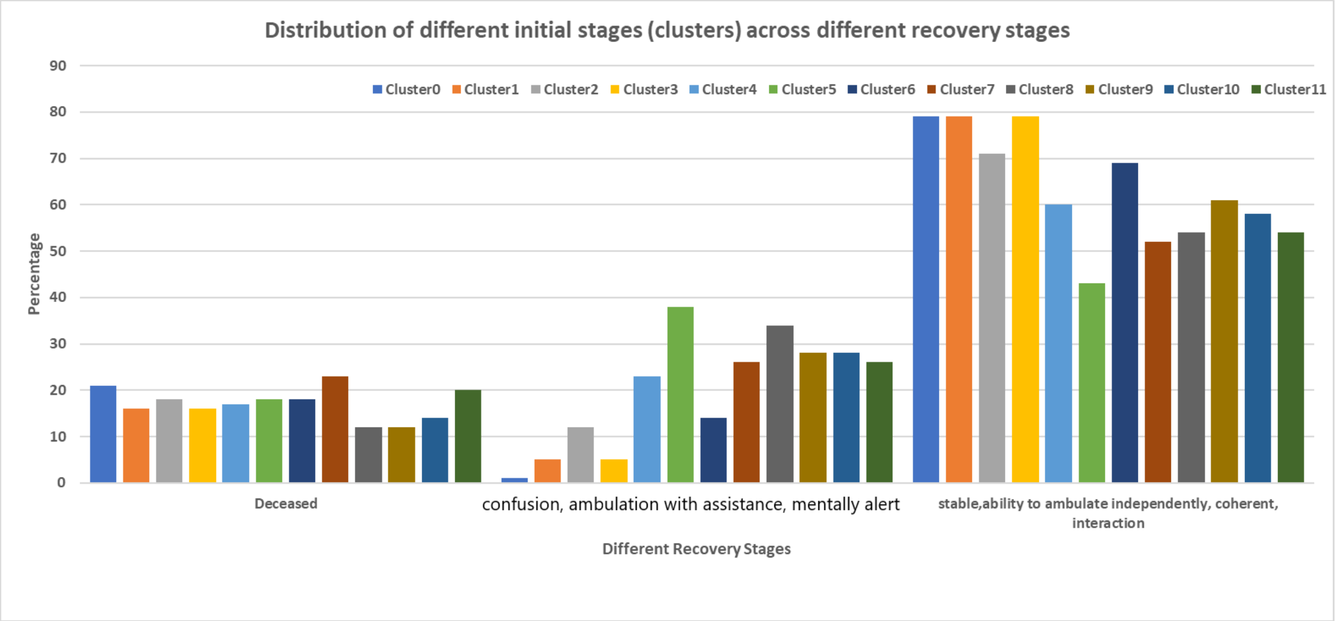


Fig. 7. Distribution of patients of twelve clusters derived from initial stages across recovery classes.

and privacy violations. Various groups however have started reporting their work on proprietary data. While this does establish the legitimacy of the problem, it is often not easy to reproduce the results in another setting or conduct a comparative analysis of the results obtained. The MIMIC dataset available for researchers alleviates some of the problems. It is a fairly large dataset with substantial number of clinical notes associated with details about diseases, treatment and outcomes.

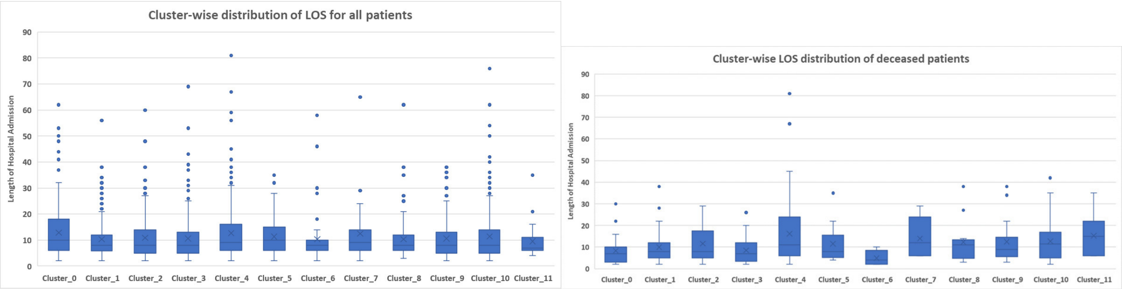


Fig. 8. (a) Clusterwise distribution of length of stay for all patients (b) Clusterwise distribution of length of stay for deceased patients.

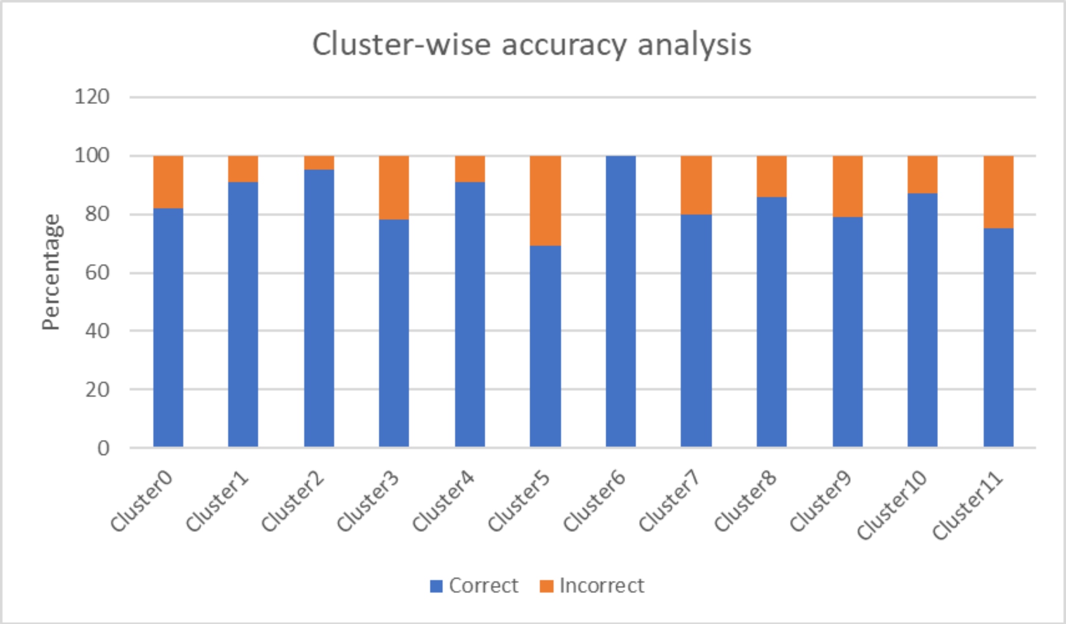


Fig. 9. Clusterwise accuracy of length of stay classification.

The use of clinical notes in predicting length of hospital admission, readmission possibilities, treatments as well as expected clinical outcomes has been prevalent for quite some time. Presently, the use of unstructured clinical notes is on the rise for development of prognostic prediction models. The focus is on developing explainable models. It is expected that robust and trustworthy prediction models will change the course of clinical practice as treatment procedures will move from majority focused designs to more customized designs.

In this paper, we have presented some initial work that we have started for explainable patient stratification. We have shown that deep learning based representations can effectively capture the richness of clinical notes and thereby be used to provide valuable insights about patient cohorts as well as exceptions within them. In future, we intend to extend our work in generating complete trajectories using the proposed representations. These trajectories in conjunction with the outcomes can help in risk assessment for patients, and thereby help in steering towards low-risk trajectories, especially for patients who are outliers.

References

[1] H. Dalianis, *Clinical text mining: Secondary use of electronic patient records*. Springer Nature, 2018.

[2] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[3] B. Percha, "Modern clinical text mining: a guide and review," *Annual review of biomedical data science*, vol. 4, pp. 165–187, 2021.

[4] T. L. Rodziewicz and J. E. Hipskind, "Medical error prevention," *StatPearls. Treasure Island (FL): StatPearls Publishing*, 2020.

[5] K. Stone, R. Zwiggelaar, P. Jones, and N. Mac Parthaláin, "A systematic review of the prediction of hospital length of stay: Towards a unified framework," *PLOS Digital Health*, vol. 1, no. 4, p. e0000017, 2022.

[6] K.-C. Chang, M.-C. Tseng, H.-H. Weng, Y.-H. Lin, C.-W. Liou, and T.-Y. Tan, "Prediction of length of stay of first-ever ischemic stroke," *Stroke*, vol. 33, no. 11, pp. 2670–2674, 2002.

[7] A. Lim, "Statistic methods for modeling incidence of infectious diseases mortality and length of stay in hospital fot patients dying in southern thailand," Ph.D. dissertation, Prince of Songkla University, Pattani Campus, 2009.

[8] D. A. Huntley, D. W. Cho, J. Christman, and J. G. Csernansky, "Predicting length of stay in an acute psychiatric hospital," *Psychiatric services*, vol. 49, no. 8, pp. 1049–1053, 1998.

[9] T. Gentimis, A. Ala'J, A. Durante, K. Cook, and R. Steele, "Predicting hospital length of stay using neural networks on mimic iii data," in *2017*

IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE, 2017, pp. 1194–1201.

[10] K. Alghatani, N. Ammar, A. Rezgui, A. Shaban-Nejad *et al.*, "Predicting intensive care unit length of stay and mortality using patient vital signs: Machine learning model development and validation," *JMIR Medical Informatics*, vol. 9, no. 5, p. e21347, 2021.

[11] L. Su, Z. Xu, F. Chang, Y. Ma, S. Liu, H. Jiang, H. Wang, D. Li, H. Chen, X. Zhou *et al.*, "Early prediction of mortality, severity, and length of stay in the intensive care unit of sepsis patients based on sepsis 3.0 by machine learning models," *Frontiers in Medicine*, vol. 8, p. 883, 2021.

[12] E. Rocheteau, P. Liò, and S. Hyland, "Predicting length of stay in the intensive care unit with temporal pointwise convolutional networks," *arXiv preprint arXiv:2006.16109*, 2020.

[13] H. Harutyunyan, H. Khachatrian, D. C. Kale, G. Ver Steeg, and A. Galstyan, "Multitask learning and benchmarking with clinical time series data," *Scientific data*, vol. 6, no. 1, pp. 1–18, 2019.

[14] S. Khadanga, K. Aggarwal, S. Joty, and J. Srivastava, "Using clinical notes with time series data for icu management," *arXiv preprint arXiv:1909.09702*, 2019.

[15] B. van Aken, J.-M. Papaioannou, M. Mayrdorfer, K. Budde, F. A. Gers, and A. Löser, "Clinical outcome prediction from admission notes using self-supervised knowledge integration," *arXiv preprint arXiv:2102.04110*, 2021.

[16] S. Jana, T. Dasgupta, and L. Dey, "Using nursing notes to predict length of stay in icu for critically ill patients," in *Multimodal AI in healthcare: A paradigm shift in health intelligence*. Springer, 2022, pp. 387–398.

[17] ——, "Predicting medical events and icu requirements using a multimodal multiobjective transformer network," *Experimental Biology and Medicine*, vol. 247, no. 22, pp. 1988–2002, 2022.

[18] T. J. Pollard, A. E. Johnson, J. D. Raffa, L. A. Celi, R. G. Mark, and O. Badawi, "The eicu collaborative research database, a freely available multi-center database for critical care research," *Scientific data*, vol. 5, no. 1, pp. 1–13, 2018.

[19] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[20] E. Alsentzer, J. R. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," *arXiv preprint arXiv:1904.03323*, 2019.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] X. Li and K.-C. Wong, "Evolutionary multiobjective clustering and its applications to patient stratification," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1680–1693, 2018.

[24] R. W. Grant, J. McCloskey, M. Hatfield, C. Uratsu, J. D. Ralston, E. Bayliss, and C. J. Kennedy, "Use of latent class analysis and k-means clustering to identify complex patient profiles," *JAMA network open*, vol. 3, no. 12, pp. e2 029 068–e2 029 068, 2020.

[25] N. Alexander, D. C. Alexander, F. Barkhof, and S. Denaxas, "Identifying and evaluating clinical subtypes of alzheimer's disease in care electronic health records using unsupervised machine learning," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 1–13, 2021.

[26] F. Angelini, P. Widera, A. Mobasheri, J. Blair, A. Struglics, M. Uebelhoer, Y. Henrotin, A. C. Marijnissen, M. Kloppenburg, F. J. Blanco *et al.*, "Osteoarthritis endotype discovery via clustering of biochemical marker data," *Annals of the Rheumatic Diseases*, vol. 81, no. 5, pp. 666–675, 2022.

[27] S. V. Bhavani, L. Xiong, A. Pius, M. Semler, E. T. Qian, P. A. Verhoef, C. Robichaux, C. M. Coopersmith, and M. M. Churpek, "Comparison of time series clustering methods for identifying novel subphenotypes of patients with infection," *Journal of the American Medical Informatics Association*, vol. 30, no. 6, pp. 1158–1166, 2023.

[28] T. Chen, X. Li, Y. Li, E. Xia, Y. Qin, S. Liang, F. Xu, D. Liang, C. Zeng, and Z. Liu, "Prediction and risk stratification of kidney outcomes in iga nephropathy," *American journal of kidney diseases*, vol. 74, no. 3, pp. 300–309, 2019.

[29] F. Kanwal, J. H. Shubrook, L. A. Adams, K. Pfotenhauer, V. W.-S. Wong, E. Wright, M. F. Abdelmalek, S. A. Harrison, R. Loomba, C. S. Mantzoros *et al.*, "Clinical care pathway for the risk stratification and management of patients with nonalcoholic fatty liver disease," *Gastroenterology*, vol. 161, no. 5, pp. 1657–1669, 2021.

[30] T. M. Seinen, E. A. Fridgeirsson, S. Ioannou, D. Jeannetot, L. H. John, J. A. Kors, A. F. Markus, V. Pera, A. Rekkas, R. D. Williams *et al.*, "Use of unstructured text in prognostic clinical prediction models: a systematic review," *Journal of the American Medical Informatics Association*, vol. 29, no. 7, pp. 1292–1302, 2022.

[31] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[32] M. Neumann, D. King, I. Beltagy, and W. Ammar, "Scispacy: Fast and robust models for biomedical natural language processing. arxiv 2019," *arXiv preprint arXiv:1902.07669*.

[33] A. R. Aronson, "Metamap: Mapping text to the umls metathesaurus," *Bethesda, MD: NLM, NIH, DHHS*, vol. 1, p. 26, 2006.

[34] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries," *Journal of biomedical informatics*, vol. 34, no. 5, pp. 301–310, 2001.

[35] B. McInnes, Y. Liu, T. Pedersen, G. Melton, and S. Pakhomov, "Umls:: similarity: Measuring the relatedness and similarity of biomedical concepts." Association for Computational Linguistics, 2013.

[36] P. A. Hall and G. R. Dowling, "Approximate string matching," *ACM computing surveys (CSUR)*, vol. 12, no. 4, pp. 381–402, 1980.

[37] R. Haldar and D. Mukhopadhyay, "Levenshtein distance technique in dictionary lookup methods: An improved approach," *arXiv preprint arXiv:1101.1232*, 2011.

[38] Y. Wang, H. Yao, and S. Zhao, "Auto-encoder based dimensionality reduction," *Neurocomputing*, vol. 184, pp. 232–242, 2016.

[39] W. Wang, Y. Huang, Y. Wang, and L. Wang, "Generalized autoencoder: A neural network framework for dimensionality reduction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 490–497.

[40] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv preprint arXiv:2003.05991*, 2020.

[41] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[42] T. M. Kodinariya, P. R. Makwana *et al.*, "Review on determining number of cluster in k-means clustering," *International Journal*, vol. 1, no. 6, pp. 90–95, 2013.

[43] L. Merrick and A. Taly, "The explanation game: Explaining machine learning models using shapley values," in *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*. Springer, 2020, pp. 17–38.

[44] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," *Ensemble machine learning: Methods and applications*, pp. 157–175, 2012.

[45] I. Ekanayake, D. Meddage, and U. Rathnayake, "A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using shapley additive explanations (shap)," *Case Studies in Construction Materials*, vol. 16, p. e01059, 2022.

[46] S. Cohen, E. Ruppin, and G. Dror, "Feature selection based on the shapley value," *other words*, vol. 1, p. 98Eqr, 2005.

[47] A. Sharma and E. Kiciman, "Dowhy: An end-to-end library for causal inference," *arXiv preprint arXiv:2011.04216*, 2020.

[48] P. C. Austin, "An introduction to propensity score methods for reducing the effects of confounding in observational studies," *Multivariate behavioral research*, vol. 46, no. 3, pp. 399–424, 2011.

[49] A. Sharma, V. Syrgkanis, C. Zhang, and E. Kıcıman, "Dowhy: Addressing challenges in expressing and validating causal assumptions," *arXiv preprint arXiv:2108.13518*, 2021.

[50] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[51] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen *et al.*, "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," *arXiv preprint arXiv:2303.10420*, 2023.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 25–37

DOI: 10.15439/2023F2324
ISSN 2300-5963 ACSIS, Vol. 35

When to Trust AI: Advances and Challenges for Certification of Neural Networks

Marta Kwiatkowska, Xiyue Zhang
0000-0001-9022-7599
0000-0003-1649-7165
University of Oxford, UK
Email: {marta.kwiatkowska, xiyue.zhang}@cs.ox.ac.uk

***Abstract*—Artificial intelligence (AI) has been advancing at a fast pace and it is now poised for deployment in a wide range of applications, such as autonomous systems, medical diagnosis and natural language processing. Early adoption of AI technology for real-world applications has not been without problems, particularly for neural networks, which may be unstable and susceptible to adversarial examples. In the longer term, appropriate safety assurance techniques need to be developed to reduce potential harm due to avoidable system failures and ensure trustworthiness. Focusing on certification and explainability, this paper provides an overview of techniques that have been developed to ensure safety of AI decisions and discusses future challenges.**

I. Introduction

ARTIFICIAL intelligence (AI) has advanced significantly in recent years, largely due to the step improvement enabled by deep learning in data-rich tasks such as computer vision or natural language processing. AI technologies are being widely deployed and enthusiastically embraced by the public, as is evident from the take up of ChatGPT and Tesla. However, deep learning lacks robustness, and neural networks (NNs), in particular, are unstable with respect to so called *adversarial perturbations*, often imperceptible modifications to inputs that can drastically change the network's decision. Many such examples have been reported in the literature and the media. Figure 1 (left) shows a dashboard camera image from [1], for which a change of a single pixel to green changes the classification of the image from red traffic light to green, which is potentially unsafe if there is no fallback safety measure; while this is arguably an artificial example, some modern cars have been observed to mis-read traffic signs, including the physical attack in Figure 1 (middle), where the digit 3 has been modified. Traffic sign recognition is a complex problem to specify and solve, see Figure 1 (right), which shows a real traffic sign in Alaska. As with any maturing technology, it is natural to ask if AI is ready for wide deployment, and what steps – scientific, methodological, regulatory, or societal – can be taken to achieve its trustworthiness and reduce potential for harm through rushed roll-out. This is particularly important given the fast-paced development of AI technologies and the natural propensity of humans to overtrust automation.

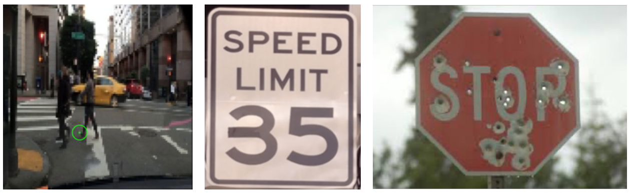


Fig. 1: Challenges of safe traffic sign recognition. Single-pixel adversarial attack from [1] (left), physical attack (middle) and a real traffic sign (right).

For AI to be trusted, particularly in high-stakes situations, where avoidable failure or wrong decision can lead to harm or high cost being incurred, it is essential to provide *provable guarantees* on the critical decisions taken autonomously by the system. Traditionally, for software systems this has been achieved with *formal verification* techniques, which aim to formally prove whether the system satisfies a given specification, and if not provide a diagnostic counter-example. Founded on logic, automated verification, also known as model checking, achieves this goal by means of executing a verification algorithm on a suitably encoded model of the system. Software verification has become an established methodology and a variety of tools of industrial relevance are employed in application domains such as distributed computation, security protocols or hardware. Beginning with [2], [3], over the past few years a number of formal verification techniques have been adapted to neural networks, which are fully data-driven and significantly differ from the state-based transition system models of conventional software, and have given rise to practical, algorithmic techniques that provide provable guarantees on neural network decisions [4].

This paper aims to provide an overview of existing techniques that can be used to increase trust in AI systems and outline future scientific challenges, while at the same time raising awareness of potential risks with early adoption. It is taken as granted that safety assurance of AI systems is complex and needs to involve appropriately regulated processes and assignment of accountability. The topics discussed in this paper are by no means exhaustive, but offer a representative selection of techniques and tools that can be used within such safety assurances processes, and can be adapted, extended or built upon to increase robustness and trustworthiness of AI systems. The paper will focus on highlighting the following two aspects:

- **Certification**: focusing on individual decisions (possibly critical to the integrity of the system) that are made by neural networks, we provide an overview of the main methodological approaches and techniques that have been developed to obtain provable guarantees on the correctness of the decision, which can thus be used for certification. The sources of computational complexity of neural network verification will be discussed, as well as limitations of existing methods and ways to address them.
- **Explainability**: neural networks are 'black boxes' that are trained from data using obscure optimization processes and objectives, and it is argued that users of AI systems will benefit from the ability to obtain explanations for the decisions. We summarise the main approaches to producing explanations and discuss that they may lack robustness and how this issue can be addressed.

The overview includes high-level description of main algorithms, which are illustrated by worked examples to explain their behaviour to the interested reader. This is followed by a selection of case studies of robustness analysis and/or certification drawn from a variety of application domains, with the aim to highlight the strengths and weaknesses of the approaches. Finally, future challenges and suggestions for fruitful directions to guide the developments in this actively studied and important area will be outlined.

The paper is organised as follows. Section II introduces the main concepts, focusing on neural networks in the supervised learning setting. Section III provides an overview of the main (forward and backward) analysis approaches, with a description of the working for a selection of algorithms illustrated by worked examples. Section IV includes a few excerpts from a selection of verification and certification experiments, aimed at highlighting the uses of the main methods, and Section V outlines future challenges. Finally, Section VI concludes the paper.

II. Safety, Robustness and Explainability

In the context of safety-critical systems, safety assurance techniques aim to prevent, or minimise the probability of, a hazard occurring, and appropriate safety measures are invoked in case of failures. In this paper, we focus on critical decisions made by neural networks, which we informally refer to as safe if they satisfy a given property, which can be shown or disproved by formal verification. Before discussing formal verification techniques, we begin with background introduction to the main concepts of deterministic neural networks, their (local) robustness and explanations.

A. Neural Networks

We consider neural networks in the supervised learning setting. A neural network is a function $f : \mathbb{R}^n \to \mathbb{R}^m$ mapping from the input space to the output space, which is typically trained based on a dataset $\mathcal{D}$ of pairs (x, y) of input x and ground truth label y. A neural network consisting of $L+1$ layers (including the input layer) can be characterized by a set of matrices $\{W^{(i)}\}_{i=1}^L$ and bias vectors $\{b^{(i)}\}_{i=1}^L$ for linear (affine) transformations, followed by pointwise activation functions, such as $ReLU$, $Sigmoid$, and $Tanh$, for nonlinear transformations. We use $\hat{z}^{(i)}$ and $z^{(i)}$ to denote the pre-activated and activated vectors of the i-th layer, respectively. The layer-by-layer forward computation of neural networks can be described as follows:

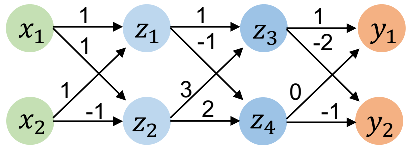


Fig. 2: A feed-forward neural network.

- *Linear transformation.* The linear transformation generates a pre-activated vector $\hat{z}^{(i)} = W^{(i)} \cdot z^{(i-1)} + b^{(i)}$ $(i \in [1, L])$ from the output of the previous layer, and $z^{(0)} = x$ denotes the input vector.
- *Pointwise nonlinear transformation.* The pointwise nonlinear transformation generates the activation vector $z^{(i)} = \sigma(\hat{z}^{(i)})$ $(i \in [1, L])$. In practice, $softmax$ is usually employed as the activation function for the output layer in classification tasks, which provides the normalised relative probabilities of classifying the input into each label.

Given an input $x \in \mathbb{R}^n$, the output of f on x is defined by $f(x) = f^{(L)} \circ \cdots \circ f^{(1)}(x)$, where $f^{(i)}$ denotes the mapping function of the i-th layer, which is the composition of linear and pointwise nonlinear transformations.

Example 1. *Figure 2 shows a simple feed-forward (and fully connected (FC)) neural network with four layers and ReLU as the activation function. x_1, x_2 represent two input neurons. z_1, z_2 and z_3, z_4 represent the activated neurons of the two hidden layers. y_1, y_2 are two output neurons. The forward computation from the input layer to the output layer is as follows (ReLU is denoted as σ).*

$$z_1 = \sigma(x_1 + x_2), \quad z_2 = \sigma(x_1 - x_2) \tag{1}$$

$$z_3 = \sigma(z_1 + 3z_2), \quad z_4 = \sigma(-z_1 + 2z_2) \tag{2}$$

$$y_1 = z_3, \quad y_2 = -2z_3 - z_4 \tag{3}$$

We will use this neural network as a running example to illustrate different problem formulations and methods to address them.

B. Robustness

Robustness focuses on neural networks' resilience to adversarial attacks, noisy input data, etc., at test time, known as evasion attacks [6], [7], [8]. Attacks at training time are known as poisoning attacks [9], which have been omitted from this overview.

Adversarial robustness [7] of neural networks formalizes the desirable property that a well-trained model makes consistent predictions when its input data point is subjected to

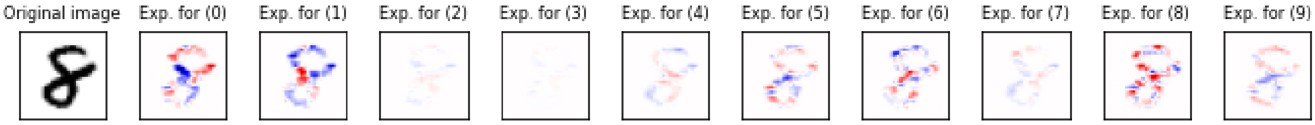


Fig. 3: The IG explanation for each of the classes of the MNIST dataset, where red indicates a positive contribution and blue a negative. Figure taken from [5].

small adversarial perturbations. *Local (adversarial) robustness* pertains to a given input point x with ground truth label y, and is usually defined in terms of invariance of the network's decision within a small neighbourhood $\mathbb{B}_p(x,\epsilon)$ of x, for a class of perturbations bounded by ϵ with respect to the ℓ_p norm.

Definition 1. *Given a (deterministic) neural network f, a labelled input data point (x, y), and a perturbation bound ϵ, the local robustness property of f on x is defined as*

$$\forall x' \in \mathbb{B}_p(x,\epsilon).\ \arg\max_{i=1,\cdots,m} f_i(x') = y,$$

where $\mathbb{B}_p(x,\epsilon)$ denotes the adversarial ℓ_p-ball of radius ϵ around input x.

Should there exist a point x' in the neighbourhood whose class is different than y, it is referred to as an *adversarial example*.

A related concept is that of a *maximal safe radius (MSR)* [10], denoted $MSR(x)$, which is the minimum distance from $x \in \mathbb{R}^n$ to the decision boundary, and is defined as the largest $\epsilon > 0$ such that $\forall x' \in \mathbb{B}_p(x,\epsilon).\ \arg\max_{i=1,\cdots,m} f_i(x') = \arg\max_{i=1,\cdots,m} f_i(x)$. Computing the value of MSR, say γ, provides a *guarantee* that the decision is robust (safe) for perturbations up to γ. On the other hand, finding an adversarial example at distance γ' is witness to the failure of robustness.

Global robustness [11] concerns the stability of predictions over the whole input space and is omitted.

C. Explainability

Explainability [12], [13] aims to understand and interpret why a given neural network makes certain predictions. The term explainability is often used interchangeably with interpretability in the literature, though interpretability usually refers to explaining how the model works. In this overview, we focus on local (pointwise) explainability for an individual *model decision*, which is categorised into *feature attribution* methods, which heuristically estimate feature attribution scores for model predictions and include *gradient-based* [14], [15] and *perturbation-based* techniques [16], [17], and *abduction-based* methods [18], [19], which identify the features that imply the decision and can thus provide (safety) guarantees. Attribution scores can also be used for feature importance ranking [20] to provide an overall understanding of the importance of different input attributes on the model decisions.

1) Gradient-based methods: Gradient-based methods aim to estimate feature attribution scores for model predictions. Among these, a prominent method is the integrated gradients (IG) [15], which measures the attribution score of each input feature to the model's prediction by integrating the gradients of the model's output with respect to the input features along the path from a baseline input to the actual input.

Definition 2. *Given a neural network f, an input x and a baseline input x', the integrated gradients for each input feature $i \in [1,\cdots,n]$ are defined as the weighted (by input feature difference) integral of the gradients over the straight line path between x and x':*

$$IG_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \quad (4)$$

Figure 3 from [5] presents an illustrative example of IG explanations, showing the explanation for each class of a correctly classified handwritten-digit “8” from the MNIST dataset. In this example, positive contributions are highlighted in red, while negative contributions are indicated by blue.

2) Perturbation-based methods: LIME [16] and its successor Anchors [17] are representatives of explainability methods that deploy a perturbation-based strategy to generate local explanations for model predictions. LIME assumes local linearity in a small area around an input instance and generates a set of synthetic data by perturbing the original input. Anchors [17] explains the model predictions by identifying a set of decision rules that “anchors” the prediction. Compared with LIME, Anchors generates more explicit decision rules and derives local explanations by consulting x's perturbation neighbourhood in different ways. In particular, Anchors evaluates the coverage fraction of the perturbed data samples sharing the same class as x, matching the decision rules.

3) Robust explanations: The explanation techniques mentioned above use different heuristics to derive local explanations, demonstrating effective generality beyond the given input but lacking robustness to adversarial perturbations. The robustness notion for explanation is important to ensure the stability of the explanation in the sense that the explanation is logically sufficient to imply the prediction. Intuitively, the computed explanation for a perturbed input should remain the same as the original input.

To this end, [18], [19] introduce a principled approach to derive explanations with formal guarantees by exploiting abduction reasoning. This ensures the robustness of the explanation by requiring its invariance w.r.t. any perturbation of the remaining features that are left out. The explanation method of [19] focuses on *optimal robust explanations (OREs)*, to provide both robustness guarantees and optimality w.r.t. a cost function. Optimality provides the flexibility to control the desired properties of an explanation. For instance, the cost function could be defined as the length of the explanation to derive minimal but sufficient explanations.

III. Certification for Neural Networks

In this section, we present an overview of recent advances for certification of neural networks, with a focus on formal verification. Given a neural network $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, we consider the formal verification problem [4], defined for a property specified as a pair $(\phi_{\mathsf{pre}}, \phi_{\mathsf{post}})$ of precondition and postcondition, by requiring that $\forall x \in \mathbb{R}^n. x \models \phi_{\mathsf{pre}} \implies f(x) \models \phi_{\mathsf{post}}$, that is, for all inputs satisfying the precondition the corresponding (optimal softmax) decision must satisfy the postcondition. Typically, $\phi_{\mathsf{pre}} \subseteq \mathbb{R}^n$ and $\phi_{\mathsf{post}} \subseteq \mathbb{R}^m$, but can be respectively induced from subsets of input features or sets of labels. Formal verification then aims to establish algorithmically whether this property holds, thus resulting in a *provable guarantee*. Otherwise, the property may be falsified, in which case a witness is provided, or inconclusive. Sometimes, we may wish to compute the proportion of inputs that satisfy the postcondition, known as *quantitative verification* [21].

Various formal verification methods have been proposed to provide provable guarantees for neural networks. We classify existing verification methods into forward and backward analysis, depending on whether they start from the input or output space.

- *Forward analysis:* Forward analysis methods start from the precondition $X = \{x \in \mathbb{R}^n \mid x \models \phi_{\mathsf{pre}}\}$ defined on the input space, and check whether the outputs (corresponding to the input region) satisfy the postconditions ϕ_{post}. For example, robustness verification approaches [22], [23], [24], [25] start from the perturbation neighbourhood of a given input, e.g., an l_∞ ball around an input point x, and compute bounds on the outputs to check whether the predicted labels over the adversarial region are preserved.
- *Backward analysis:* Backward analysis methods start from the postcondition $Y = \{y \in \mathbb{R}^m \mid y \models \phi_{\mathsf{post}}\}$ and aim to find the set of inputs that lead to such outputs. For example, preimage generation (inverse abstraction) approaches [26], [27], [28], [29] start from the output constraints, e.g., a polytope constraining the probability of the target label is greater than the other labels, and derive the input set that provably leads to this particular decision.

We remark that, similarly to formal verification for conventional software, certification for machine learning models is computationally expensive, and it is therefore recommended for use in safety- or security-critical settings. In less critical situations, diagnostic methods [30], which approximate model decisions to analyse their predictions, can be employed to investigate both model- and data-related issues.

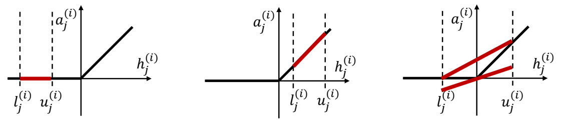


Fig. 4: Illustration of the convex relaxation for inactive (left), active (middle) and unstable (right) ReLU neurons.

A. Forward Analysis Methods

We categorize the forward analysis methods into two groups: *sound but incomplete* and *complete* methods. Soundness and completeness are essential properties of verification algorithms, which are defined as follows.

- *Soundness:* A verification algorithm is sound if the algorithm returns True and the verified property holds.
- *Completeness:* A verification algorithm is complete if (i) the algorithm never returns unknown; and (ii) if the algorithm returns False, the property is violated.

1) Incomplete methods: Incomplete verification methods leverage approximation techniques, such as search [1], [10], convex relaxation [31] and abstract interpretation [32], respectively to compute lower/upper bounds on MSR or the non-convex optimization problem. A safety property is verified when the reachable outputs satisfy the postcondition; otherwise, no conclusion can be drawn. At the same time, due to the relaxation introduced by the approximation techniques, incomplete methods have better scalability than complete ones.

a) Game-based search: Knowledge of the maximum safe radius (MSR) can serve as a guarantee on the maximum magnitude of the allowed adversarial perturbations. Unfortunately, MSR computation is intractable, and instead approximate algorithms have been developed for images in [10], and extended to videos in [33], that compute lower and upper bounds on MSR with provable guarantees, i.e., bounded error. The method relies on the network satisfying the Lipschitz condition and can be configured with a variety of feature extraction methods, for example SIFT. Given an over-approximation of the Lipschitz constant, the computation is reduced to a finite optimization over a discretisation of the input region X corresponding to the precondition ϕ_{pre}. The resulting finite optimization is solved in anytime fashion through a two-player game, where player 1 selects features and player 2 perturbs the image representation of the feature, and the objective is set to minimise the distance to an adversarial example. Under the assumptions, the game can be unfolded into a finite tree and Monte Carlo Tree Search (MCTS) used to approximate MSR upper bound, and Admissible A* MSR lower bound, respectively.

b) Bound propagation: A common technique for incomplete verification is applying convex relaxation to bound nonlinear constraints in neural networks. This way, the original non-convex optimization problem is transformed into a linear programming problem. With the relaxed linear constraints, the global lower and upper bounds can be computed more

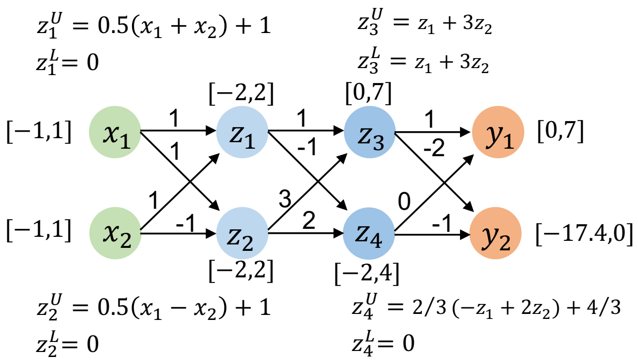


Fig. 5: Verification via bound propagation.

efficiently for the associated (relaxed) linear program. Representative methods that adopt efficient bound propagation include convex outer adversarial polytope [31], CROWN [34] and its generalization [35], [25]. Figure 4 illustrates convex relaxation using linear bounding functions to bind ReLU neurons. Note that relaxation is only introduced for unstable neurons, while the ReLU constraints for inactive and active ones are exact. For unstable neurons, the lower and upper bounding function for the j-th neuron of the i-th layer $a_j^{(i)}(x)$ (activated value) with regard to $h_j^{(i)}(x)$ (before activation) are:

$$\alpha_j^{(i)} h_j^{(i)}(x) \leq a_j^{(i)}(x) \leq -\frac{u_j^{(i)} l_j^{(i)}}{u_j^{(i)} - l_j^{(i)}} + \frac{u_j^{(i)}}{u_j^{(i)} - l_j^{(i)}} h_j^{(i)}(x) \quad (5)$$

where a flexible lower bound function with parameter $\alpha_j^{(i)}$ as in [35] is used, which leads to a valid lower bound for any parameter value within $[0, 1]$.

By propagating the linear (symbolic) upper and lower bounds layer by layer, we can obtain the linear bounding functions f^L, f^U for the entire neural network f, and it holds that $\forall x \in X.\, f^L(x) \leq f(x) \leq f^U(x)$. The non-convex verification problem is thus transformed into a linear program with the objective linear in the decision variables. The certified upper and lower bounds can be computed by taking the maximum, $\max_{x \in \mathbb{B}_p(x,\epsilon)} f^U(x)$, and the minimum, $\min_{x \in \mathbb{B}_p(x,\epsilon)} f^L(x)$, which have *closed-form* solutions for linear objectives (f^U, f^L) and convex norm constraints $\mathbb{B}_p(x, \epsilon)$.

Example 2. *Consider the neural network illustrated in Example 1. The verification problem we consider is given by the pre-condition* $\phi_{\text{pre}} = \{x \in \mathbb{R}^2 | x \in [-1, 1] \times [-1, 1]\}$ *and the post-condition* $\phi_{\text{post}} = \{y = f(x) \in \mathbb{R}^2 \mid y_1 \geq y_2\}$, *and we want to prove that* $\forall x.\, x \models \phi_{\text{pre}} \implies f(x) \models \phi_{\text{post}}$.

Figure 5 shows the overall bound propagation procedure for this verification problem, where the interval $[\cdot, \cdot]$ *represents the concrete value range computed for each neuron.* z_i^U, z_i^L *represent the linear upper and lower bounding functions for nonlinear neurons, which are computed according to Equation 5 based on the concrete value intervals. Starting from the input layer, we can first compute the concrete bounds (*$[-2, 2]$*) for* z_1 *and* z_2 *(before activation). The bounding functions* (z_1^L, z_1^U), (z_2^L, z_2^U) *are then computed according to Equation 5, where* $\alpha = 0$ *is taken as the lower bounding function coefficient. The linear bounding functions can directly propagate to the next layer via the linear matrix transformation. Then, by taking the minimum value of the lower bounding function and the maximum of the upper one, concrete value ranges (*$[0, 7]$ *and* $[-2, 4]$*) are computed for* z_3 *and* z_4*, based on which symbolic functions* (z_3^L, z_3^U), (z_4^L, z_4^U) *can be derived and further propagated to the output layer. In the end, we compute the global lower and upper bounds for* y_1 *and* y_2*, which are* $[0, 7]$ *and* $[-17.4, 0]$*, respectively. From the certified bounds on the output layer, it holds that* $\min(y_1) \geq \max(y_2)$ *for any input* $(x_1, x_2) \in [-1, 1] \times [-1, 1]$*. Therefore, the bound propagation method certifies that the neural network is robust in the input domain with respect to the ground-truth label* y_1.

c) Abstract interpretation: Abstract interpretation [32], [36] is a classic framework that can provide sound and computable finite approximations for infinite sets of behaviours. To provide sound analysis of neural networks, several works [22], [37], [38], [24] have exploited this technique to reason about safety properties. These methods leverage numerical abstract domains to overapproximate the inputs and compute an over-approximation of the outputs layer by layer. To this end, an abstract domain is selected to characterize the reachable output set for each layer as an abstract element. The choice of abstract domain is essential to balance the analysis precision and scalability. Commonly used abstract domains for neural network verification [39] include *Interval*, *Zonotope*, and *Polytope*, of which the general formulations are summarized in the following (increasing in precision):

$$\text{Interval:} \quad \{x \in \mathbb{R}^n | l_i \leq x_i \leq u_i\}$$

$$\text{Zonotope:} \{x \in \mathbb{R}^n | x_i = c_{i0} + \sum_{j=1}^{m} c_{ij} \cdot \epsilon_j, \epsilon_j \in [-1, 1]\}$$

$$\text{Polytope:} \{x \in \mathbb{R}^n | x_i = c_{i0} + \sum_{j=1}^{m} c_{ij} \cdot \epsilon_j, F(\epsilon_1, \cdots, \epsilon_m)\}$$

where ϵ_j ($j = 1, \cdots, m$) denote m generator variables. The generator variables are bounded within the interval $[-1, 1]$ for zonotopes and constrained by F for polytopes, where F takes in the form of a convex polytope $\mathbf{c}\mathbf{x} \leq \mathbf{d}$.

With the abstract domain capturing the reachable outputs of each layer, abstract transformers are defined to compute the effect of different layers on propagating the abstract element. Affine transformers are usually supported by the underlying abstract domain, such as Zonotope and Polytope, to abstract the linear functions. For nonlinear functions, case splitting and unifying is proposed in [22] by defining the *meet* and *join* operators to propagate zonotope abstraction through piecewise-linear layers. Convex approximations are adopted in [24] for abstract transformers of nonlinear functions where the approximation can be captured with the proposed polyhedra abstraction. At the end of the analysis, the abstract element of the output layer is an over-approximation of all possible concrete outputs corresponding to the input set. Then we can directly verify the over-approximation of the outputs against the postcondition ϕ_{post}, i.e., check whether the over-approximation is fully contained within ϕ_{post}. One drawback of this method is that the over-approximation may be quite loose.

2) Complete methods: Early complete verification approaches for neural networks [40], [3] encode the neural network into a set of constraints exactly and then check the satisfaction of the property with constraint solvers, e.g, SMT (Satisfiability Modulo Theory) or MILP (Mixed Integer Linear Programming) solvers. Since such constraint-solving methods encode the neural network in an exact way, they are able to ensure both soundness and completeness in providing certification guarantees. One limitation is that these methods suffer from exponential complexity in the worst case. To address the computational intractability, Branch and Bound techniques are adopted and customized for neural network verification, where efficient incomplete methods can be exploited to speed up the bound computation.

a) SMT solver: Reluplex [3] is proposed as a customized SMT solver for neural network verification. The core idea is to extend the simplex algorithm, a standard algorithm to solve linear programming problems, with additional predicates to encode (piecewise linear) ReLU functions and transition rules (*Pivot* and *Update*) to handle ReLU violations. The extended Reluplex algorithm allows variables that encode ReLU nodes to temporarily violate the ReLU constraints. Then, as the iteration proceeds, the solver picks variables that violate a ReLU constraint and modifies the assignment to fix the violation using *Pivot* and *Update* rules. When the attempts to fix a ReLU constraint using *Update* rules exceed a threshold, a ReLU splitting mechanism is applied to derive two sub-problems. Reluplex is then invoked recursively on these two sub-problems. Compared with the eager splitting on all ReLU neurons, Reluplex proposes a splitting-on-demand strategy to reduce unnecessary splitting and limit splits to ReLU constraints that are more likely to cause violation problems. Due to the exact encoding nature, Reluplex suffers from exponential complexity in the worst case and thus cannot scale to large neural networks.

b) MILP: MILP-based verification methods [41], [42], [43] encode a neural network with piecewise-linear functions as a set of mixed integer linear constraints. To encode the nonlinearities, they introduce an indicator decision variable δ to characterize the two statuses of unstable ReLU neurons. An unstable ReLU neuron $z = \max(\hat{z}, 0)$ with concrete bounds (l, u) can be encoded exactly using the following constraints:

$$\begin{aligned} & z \geq 0, \quad z \leq u \cdot \delta, \\ & z \geq \hat{z}, \quad z \leq \hat{z} - l \cdot (1 - \delta), \\ & \delta \in \{0, 1\} \end{aligned}$$

Note that the MILP constraints require the pre-computation of finite bounds for the nonlinear neurons, i.e., (l, u). It is known that the tightness of lower and upper bounds in the indicator constraints is crucial to the resolution of the MILP problem [44], [42], and consequently, the verification efficiency. MIPVerify [43] thus proposes a progressive bound tightening approach to improve upon existing MILP-based verifiers. The algorithm starts with coarse bounds computed using efficient bound computation procedures such as *Interval Arithmetic*. Bound refinement is performed only when the MILP problem can be further tightened. In such a case, more precise but less efficient bound computation procedures, e.g., *Linear Programming* (LP), are adopted to derive tighter bounds. This progressive bounding procedure can also be extended to other bound computation methods, such as dual optimization, to achieve a trade-off between tightness and computational complexity.

Example 3. *In this example, we encode the neural network, shown in Example 1, into the exact MILP formulation. The verification problem is the same as shown in Example 2, i.e., to determine whether $\phi_{\text{post}} = \{y \in \mathbb{R}^2 \mid y_1 \geq y_2\}$ holds for all inputs in the input domain $[-1, 1] \times [-1, 1]$. We encode the output property by specifying its negation, i.e., $\phi'_{\text{post}} = \neg\phi_{\text{post}}$. If there exists an instance where ϕ'_{post} does hold, then a witness to ϕ'_{post} is the counter-example for ϕ_{post}. If ϕ'_{post} is unsatisfiable, then the property ϕ_{post} is proved.*

Assume we have computed the concrete value of lower and upper bounds of z_i employing efficient bound propagation techniques. Then the neural network and the verification problem can be formulated as follows:

$$x_1 \geq -1,\, x_1 \leq 1,\, x_2 \geq -1, x_2 \leq 1\, (\phi_{\text{pre}}) \tag{6}$$

$$\hat{z}_1 = x_1 + x_2, \quad \hat{z}_2 = x_1 - x_2, \quad \delta_1, \delta_2 \in \{0, 1\} \tag{7}$$

$$z_1 \geq 0,\, z_1 \leq 2\delta_1,\, z_1 \geq \hat{z}_1,\, z_1 \leq \hat{z}_1 + 2(1 - \delta_1), \tag{8}$$

$$z_2 \geq 0,\, z_2 \leq 2\delta_2,\, z_2 \geq \hat{z}_2,\, z_2 \leq \hat{z}_2 + 2(1 - \delta_2), \tag{9}$$

$$\hat{z}_3 = z_1 + 3z_2, \quad \hat{z}_4 = -z_1 + 2z_2, \quad \delta_4 \in \{0, 1\} \tag{10}$$

$$z_3 = \hat{z}_3, \textit{(stable neuron)} \tag{11}$$

$$z_4 \geq 0,\, z_4 \leq 4\delta_4,\, z_4 \geq \hat{z}_4,\, z_4 \leq \hat{z}_4 + 2(1 - \delta_4), \tag{12}$$

$$y_1 = z_3, \qquad y_2 = -2z_3 - z_4 \tag{13}$$

$$y_1 < y_2\, (\neg\phi_{\text{post}}) \tag{14}$$

The lower and upper bounds l_i and u_i ($i \in \{1, 2, 3, 4\}$) are derived as shown in Example 2. The binary variables δ_i ($i \in \{1, 2, 4\}$) are introduced to indicate the status of unstable ReLUs and it holds that $\delta_i = 0 \Leftrightarrow z_i = 0$ and $\delta_i = 1 \Leftrightarrow z_i = \hat{z}_i$. Checking the feasibility of the above model using MILP solvers (e.g., Gurobi) will return infeasible, thus proving the original property.

c) Branch and Bound: To improve the scalability of verification algorithms to larger neural networks, a branch and bound framework (BaB) [45] has been proposed. The BaB framework mainly consists of two components: a branching method that splits the original verification problem into multiple subproblems and a bounding method to compute the upper and lower bounds of the subproblems. This modularized design provides a unifying formulation paradigm for different verifiers, with the main difference lying in the splitting function and the bounding method. For example, the verifier

ReluVal [46] performs splitting on the input domain according to sensitivity analysis, e.g., input-output gradient information, and computes bounds using symbolic interval propagation. The aforementioned SMT-based verifier *Reluplex* [3] performs splitting on ReLU neurons guided by the violation frequency of the ReLU constraints and computes the bounds on the relaxed problems by dropping some constraints on the nonlinearities (which yields an over-approximation of the constraint optimization problems).

To further improve neural network verification, the BaB method introduces two new branching strategies: BaBSB for branching on input domains and BaBSR for branching on ReLU neurons. Both branching methods adopt a similar heuristic to decide which dimension or ReLU neuron to split on. BaBSB computes a rough estimate of the improvement on the bounds obtained with regard to every input dimension, where the estimation makes the split decision be set more efficiently. On the other hand, BaBSR estimates the bound improvement with regard to each unfixed ReLU neuron by computing the ReLU scores. The bounding methods resort to LP solvers to tighten the intermediate bounds on the subdomains or use more computationally efficient methods such as Interval Arithmetic.

B. Backward Analysis Methods

Backward analysis methods for neural networks, also known as preimage generation or inverse abstraction, aim at computing the input set that will lead the neural network to a target set, e.g., a safe or unsafe region. They complement the forward analysis methods, which may result in over-approximated bounds worsening as the computation progresses through the layers of the network. In the following, we categorize the representative approaches broadly into two groups: *exact* and *approximate* methods.

1) Exact methods: Exact backward analysis methods reason about the preimage of a target output set by encoding the neural network behaviours in an exact manner. These methods are able to compute the exact symbolic representation of the preimage for different output properties. One limitation suffered by these methods is that they can only process neural networks with piecewise-linear activation functions (e.g., $ReLU$), as they aim at an exhaustive decomposition of the non-convex function (the neural network) into a set of linear functions. The preimage (input set) for a target output set with regard to a neural network f is characterized as a union of polytopes, where the mapping functions are completely linear on each subregion.

a) Exact preimage: The exact preimage generation method [26] complements the forward analysis methods to reason about the inputs that lead to target outputs. The algorithm computes the exact preimage by relying on two elementary properties: (1) preimage of the composite functions is the reversed composition of preimages for each layer, i.e., $(f^{(L)} \circ \cdots \circ f^{(1)})^{-1} = (f^{(1)})^{-1} \circ \cdots \circ (f^{(L)})^{-1}$, and (2) preimage of a union set can be built up from the preimages of each subset in the union, i.e., $f^{-1}(\cup_j S_j) = \cup_j f^{-1}(s_j)$. This method assumes that the output set, e.g., a safe region, can be formulated as a polytope (intersection of half-planes) $\{y \in \mathbb{R}^m | Ay - b \leq 0\}$. It then propagates the polytope backwards through the layers.

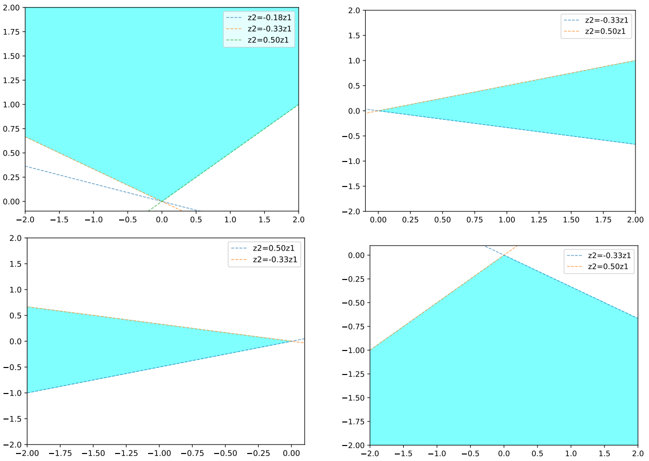


Fig. 6: Preimage polytopes by exact method.

For linear layers, the preimage is computed by applying the linear operations corresponding to the layer. Suppose we have a linear mapping in the form of $y = Wz + a$, then the preimage of the output polytope under this linear operation can be formulated as $\{z \in \mathbb{R}^{n_{L-1}} | AWz + (Aa - b) \leq 0\}$. For nonlinear layers, the algorithm restricts the backward propagation to a subset where the activation pattern of the ReLU neurons is fixed. Let $s(z)$ denote the activation status vector of the nonlinear neurons where $s(z)_j = 1$ if $z_j \geq 0$ and $s(z)_j = 0$ otherwise. A diagonal matrix $diag(s(z))$ is introduced to restrict to a fixed activation pattern, on which only linear computation is required to compute the preimage subset. The exact preimage can then be computed by taking the union of each partition (preimage property (2)).

$$\begin{aligned}
&ReLU^{-1}(\{y \in \mathbb{R}^m | Ay - b \leq 0\}) \\
&= \bigcup_{s \in \{0,1\}^{n_i}} \{z \in \mathbb{R}^{n_i} | Adiag(s)z - b \leq 0, -diag(s)z \leq 0, \\
&\qquad diag(1-s)z \leq 0\}
\end{aligned}$$

Example 4. *In this example, we consider the same verification problem as in Example 2 and 3, but from the backward perspective. Preimage analysis aims to investigate whether the input region* $[-1, 1] \times [-1, 1]$*, which is expected to result in decision* y_1*, fails the safety check. We first formulate the target output region as a polytope. Since we only have two labels, the output constraint is, therefore, a single half-plane encoded as* $\{y \in \mathbb{R}^2 | y_1 - y_2 \geq 0\}$*. We then proceed to compute the preimage of the target polytope under the linear mapping (from the* 2^{nd} *hidden layer to the output layer), of which the result is* $\{(z_3, z_4) \in \mathbb{R}^2 \mid 3z_3 + z_4 \geq 0\}$*. Next,* *preimage computation for ReLU starts with partitioning the neuron vector space $\mathbb{R}^2$ into 2^2 sets where, for each subset, the status of nonlinear neurons is fixed and preimage computation proceeds similarly to the linear mapping. The partition leads to four result polytopes.*

Figure 6 shows the result of four preimage polytopes derived in the two-dimensional space (z_1, z_2). As an example, the preimage polytope derived corresponding to the partition where both neurons are active is (upper left of Figure 6):

$$\{z^{(1)} \in \mathbb{R}^2 \ : A^{(1)} z^{(1)} \geq 0\} \quad \textit{where}$$
$$A^{(1)} = \begin{bmatrix} 2 & 11 \\ 1 & 3 \\ -1 & 2 \end{bmatrix}, \quad z^{(1)} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

The other three polytopes are derived in the same way. The four preimage polytopes are then partitioned further into 16 polytopes to characterize the exact preimage of the input layer. Note that the combination of the four polytopes actually covers the hidden vector space $[-2, 2] \times [-2, 2]$, and the resulting preimage polytopes on the input layer cover the region $[-1, 1] \times [-1, 1]$, which certifies that the correct decision is taken for the entire region under investigation.

b) SyReNN: SyReNN is proposed in [47] to compute the symbolic representation of a neural network so as to understand and analyze its behaviours. It targets *low-dimensional* input subspaces and computes their exact symbolic partitioning, on which the mapping function is completely linear. This methodological design is also referred to as neural network decomposition. We classify it as a backward analysis method, as this method provides a symbolic representation in the input space. SyReNN focuses on neural networks with piecewise-linear activation functions. This restriction enables a precise characterisation of the input space X as a finite set of polytopes $\{X_1, \cdots, X_n\}$. Within each input polytope X_i, the neural network is equivalent to a linear function. By means of such a symbolic representation, safety verification is reduced to checking whether the vertices of every bounded convex polytope X_i satisfy the output property.

To compute the symbolic decomposition on the input domain, this algorithm starts with the trivial partition X and derives the linear partitions layer by layer. Given the partition hyperplanes of the nonlinear layer i, e.g., $z_1 = 0, z_2 = 0, \cdots, z_{n_i} = 0$ with n_i ReLUs, and the symbolic representation $\hat{f}_{i-1}$ (a set of polytopes) computed until layer $i-1$, $\hat{f}_i$ is computed by recursively partitioning the current polytopes based on the newly-added hyperplanes. For example, given a polytope Z_{i-1}, if an orthant boundary (e.g., hyperplane $z_i = 0$) is hit when traversing the boundary of Z_{i-1}, then Z_{i-1} is further partition into $Z_{i-1,1}$ and $Z_{i-1,2}$ which lie on the opposite sides of the hyperplane. This procedure terminates until all resulting polytopes lie within a completely linear region of the neural network f.

2) Approximate methods: Exact methods for preimage analysis suffer from exponential complexity in the worst case. Similarly to the development of incomplete verifiers, preimage approximation techniques begin to emerge by leveraging different approximation (relaxation) techniques. They compute a symbolic approximation of the preimages to bypass the intractability of computing exact preimage representations. Computational efficiency and scalability can be greatly improved with the sacrifice of precision.

a) Symbolic interpolation: Symbolic interpolation [48] has been used for program verification and SMT solving. To compute provable preimage approximations, [27] leverages interpolants, especially those with simple structures, and computes preimages from the output space through hidden layers to the input space. The generated approximations can then be applied to reason about the properties of the neural network itself. For example, in the case that a desired property (a target output set) Y should be satisfied when starting from a certain input set X, an under-approximation $\underline{X}$ of the preimage for Y can be computed. Then the property can be verified by checking whether $\underline{X} \rightarrow X$ holds.

[48] proposes an algorithm to compute the preimage approximation by iterating backwards through the layers. It encodes the neural network as constraints in the theory of *quantifier-free linear rational arithmetic* (QFLRA) and requires the output set to be encoded as a Boolean combination of atoms in the form of half-spaces. Suppose we now focus on deriving preimage over-approximations of the target output set Y. The algorithm starts by computing the (overapproximated) set of inputs to the last layer, denoted as $p_L^{f,Y}$, which leads to the output set Y, i.e., $f^{(L)}(p_L^{f,Y}) \models Y$. The algorithm then iteratively computes preimages of the other layers that satisfy $p_i^{f,Y} = \{z \mid f^{(i)}(z) \models p_{i+1}^{f,Y}\}$. This procedure leverages sampling techniques to construct a set of points mapped to the complement of Y, which are used to tighten the over-approximations. The algorithm relies on Craig's Interpolation theorem to guarantee the existence of an (over- and under-) approximation. It also leverages the bound propagation framework to compute a bounded domain on each layer, which speeds up the interpolation condition checking.

b) Inverse bounding: [28] points out two important applications based on preimage analysis: safety verification for dynamical systems and out-of-distribution input detection. Motivated by these use cases, an inverse bound propagation method is proposed to compute the over-approximation of the preimage. Bound propagation has been widely employed to build efficient verifiers in the forward direction (certified output bound computation). Compared with forward analysis, it is challenging to adopt bound propagation methods directly to compute tight intermediate bounds, and thus difficult to compute tight over-approximations. This is because, for the inverse problem, the constraints on the input are quite loose and even unbounded in some control applications. Simply applying the bound propagation procedure will not lead to useful intermediate bounds, which further impacts the tightness of the symbolic relaxation on nonlinear neurons.

Given this, an inverse propagation algorithm is proposed in [28] to compute a convex over-approximation of the preimage represented by a set of cutting planes. It first transforms

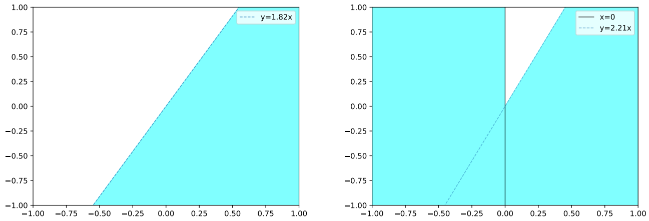


Fig. 7: Preimage approximation.

the preimage over-approximation problem to a constrained optimization problem over the preimage and further relaxes it to Lagrangian dual optimization. To tighten the preimage and intermediate bounds, they introduce a dual variable with respect to the output constraints and tighten these bounds iteratively, leveraging standard gradient ascent algorithm.

c) Preimage approximation: Motivated by the practical needs of global robustness analysis [49], [11], [21] and quantitative verification [50], [51], an anytime algorithm is proposed in [29] to compute provable preimage approximation. The generated preimage is further applied to verify quantitative properties of neural networks, which is defined by the relative proportion of the approximated preimage volume against the input domain under analysis, formally defined as follows.

Definition 3. *Given a neural network $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, a measurable input set with non-zero measure (volume) $X \subseteq \mathbb{R}^n$, a measurable output set $Y \subseteq \mathbb{R}^m$, and a rational proportion $p \in [0, 1]$, the neural network satisfies the quantitative property (X, Y, p) if $\frac{\mathrm{vol}(f_X^{-1}(Y))}{\mathrm{vol}(X)} \geq p$.*

This approach targets safety properties that can be represented as polytopes and characterizes preimage under-approximation using a disjoint union of polytopes. To avoid the intractability of the exact preimage generation method, convex relaxation is used to derive sound under-approximations. However, one challenge is that the generated preimage under-approximation can be quite conservative when reasoning about properties in large input spaces with relaxation errors accumulated through each layer. To refine the preimage abstraction, a global branching method is introduced to derive tighter approximations on the input subregions. This procedure proposes a (sub-)domain search strategy prioritizing partitioning on most uncovered subregions and a greedy splitting rule leveraging GPU parallelization to achieve better per-iteration improvement. To further reduce the relaxation errors, this method formulates the approximation problem as an optimization problem on the preimage polytope volume. Then it proposes a differentiable relaxation to optimize bounding parameters using projected gradient descent.

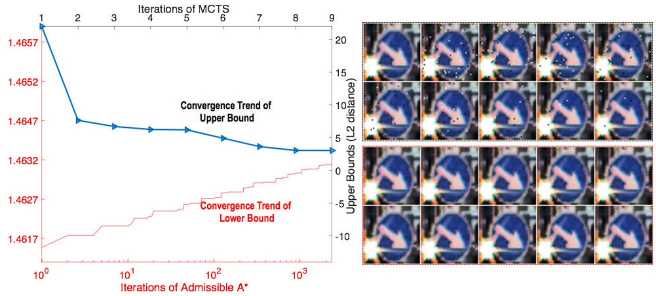


Fig. 8: Convergence of maximum safe radius computed using the game-based method for a traffic sign image from the GTSRB dataset originally classified as "keep right". Left: The convergence trends of the upper bound obtained with Monte Carlo Tree Search and the lower bound with Admissible A*. Right: unsafe images (top two rows) and certified safe images (bottom two rows). Figure taken from [10].

Example 5. *In this example, we demonstrate how to construct a provable preimage (under-)approximation for the target output region, and apply it to quantitative analysis of the verification problem shown in previous examples. Consider the quantitative property with input set $\phi_{\mathsf{pre}} = \{x \in \mathbb{R}^2 \mid x \in [-1, 1]^2\}$, output set $\phi_{\mathsf{post}} = \{y \in \mathbb{R}^2 \mid y_1 - y_2 \geq 0\}$, and quantitative proportion $p = 0.9$. We apply the preimage approximation algorithm to verify this property. Figure 7 presents the computed preimage before (left) and after one-iteration refinement (right). Note that the partition is performed w.r.t. input x_1, which results in two polytopes for the subregions. We compute the exact volume ratio of the refined under-approximation against the input set. The quantitative proportion reached with the refinement is 94.3%, which verifies the quantitative property.*

IV. Application Examples

In this section we provide a selection of experimental results and lessons learnt from applying formal verification and certification approaches described in the previous section to neural network models drawn from a range of classification problems. These include image and video recognition, automated decisions in finance and text classification. In addition to adversarial robustness of the models, we demonstrate certification of individual fairness of automated decisions and discuss robust explanations.

A. MSR-based Certification for Images and Videos

The game-based method [10] has been applied to analyse and certify the robustness of image classification models to adversarial perturbations with respect to the maximal safe radius, working with a range of feature extraction methods and distance metrics. Figure 8 shows a typical outcome of such analysis, with converging lower and upper MSR bounds for an image of a traffic sign for l_2 distance and features extracted from the latent representation computed by a convolutional neural network (CNN) model. It can be seen that the image is certified safe for adversarial perturbations of up to 1.463 in l_2 distance, which is some distance away from the best upper bound at approx. 3, but can be improved with more iterations since the method is anytime.

An extension of the game-based method was developed in [33] to provide MSR-based certification for videos, and

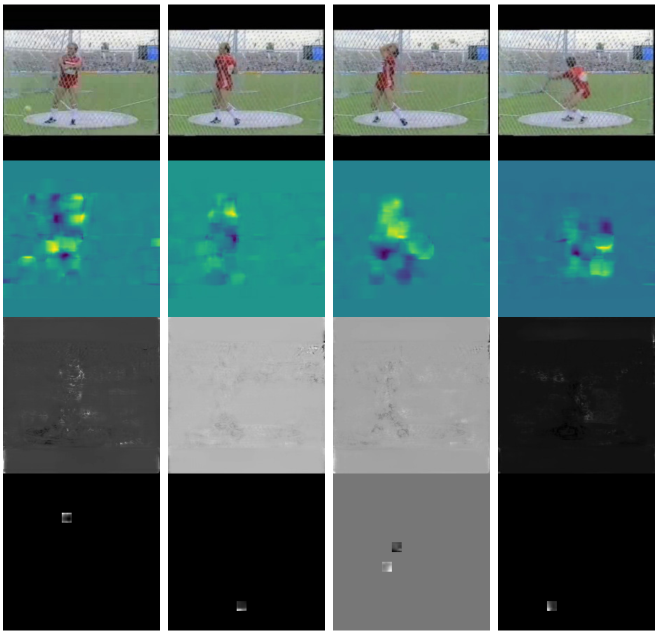


Fig. 9: Shown in top row are sampled frames of a HammerThrow video and the corresponding optical flows are in the 2nd row. Unsafe perturbations of flows are in 3rd row and safe in 4th. Figure taken from [33].

specifically for neural network models consisting of a CNN to perform feature extraction and a recurrent neural network (RNN) to process video frames. Adversarial perturbations were defined with respect to optical flow, and the algorithmic techniques involve tensor-based computation. Examples of safe and unsafe perturbations are shown in Figure 9, and convergence trends for lower and upper bounds similar to those in Figure 8 can be observed.

B. *Robustness of Language Models*

As an example of application of convex relaxation tools (variants of CROWN [34]), we mention the study of [52], which aims to assess the robustness of Natural Language Processing tasks (sentiment analysis and text classification) to word substitution. It was reported that standard fully connected (FC) and CNN models are very brittle to such perturbations, which may make their certification unworkable. [53] critiqued the appropriateness of the classical concept of adversarial robustness defined in terms of word substitution in the context of NLP models. It was observed in an empirical study that models trained to be robust in the classical sense, for example, trained using interval bound propagation (IBP), lack robustness to syntax/semantic manipulations. It was then argued in [53] that a *semantic* notion of robustness that better captures linguistic phenomena such as shallow negation and sarcasm is needed for language models, where a framework based on templates was developed for evaluation of semantic robustness.

C. *Robust Explanations for Language Models*

Explainability of language models was studied in [19], with a focus on robust optimal explanations that imply the model prediction. Figure 10 shows examples of high-quality robust optimal explanations (using the minimum length of explanation as the cost function). In contrast, heuristic explanations such as integrated gradients or Anchors my lack of robustness, but it is possible to repair non-robust Anchors explanations by minimally extending them, see Figure 11.

D. *Fairness Certification Using MILP*

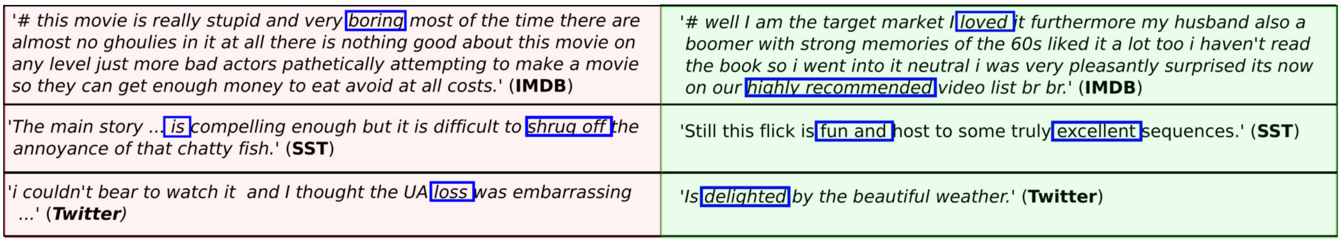
[54] developed methods for certification of individual fairness of automated decisions, defined, given a neural network and a similarity metric learnt from data, as requiring that the output difference between any pair of ϵ-similar individuals is bounded by a maximum decision tolerance $\delta \geq 0$. Working with a range of similarity metrics, including Mahalanobis distance, a MILP-based method was developed not only to compute certified bounds on individual fairness, but also to train certifiably fair models. The computed certified bounds δ_* are plotted in Figure 12 for the Adult and the Crime benchmarks. Each heat map depicts the variation of δ_* as a function of ϵ and the NN architecture. It can be observed that increasing ϵ correlates with an increase in the values for δ_*, as higher values of ϵ allow for greater feature changes.

V. Future Challenges

Formal verification and certification of neural network models has made steady progress in recent years, with several tools released to the community and an established tool competition [4]. Nevertheless, considerable scientific and methodological progress is needed before these tools are adopted by developers. Below we outline a number of research challenges.

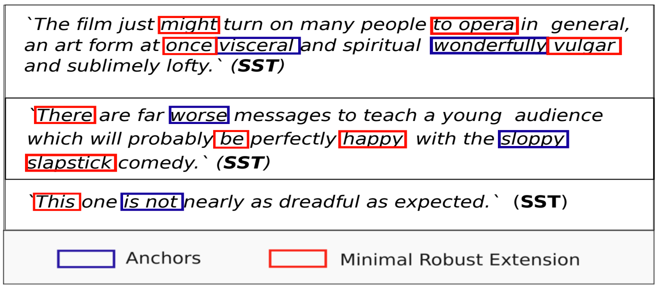
a) Beyond ℓ_p-norm robustness: The vast majority of robustness evaluation frameworks consider bounded ℓ_p-norm perturbations. While these suffice as proxies for minor visual image perturbations, real-world tasks rely on similarity measures, for example cosine similarity for word embeddings or Mahalanobis distance for images. It is desirable to define measures and certification algorithms for semantic robustness, which considers such similarity measures as first class citizens, and works with perturbations that reflect visual or geometric aspects characteristic of the application, such as object movement or lighting conditions. More generally, robustness evaluation frameworks for more complex properties induced by the use cases will be needed.

b) Beyond supervised robustness: Existing robustness formulations focus on the supervised learning setting. However, collecting and labelling large datasets that are necessary to ensure the high robustness performance needed in safety-critical applications is costly and may not be feasible for use cases such as autonomous driving. Instead, it is desirable to formulate robustness measures and evaluation frameworks directly in some appropriate semi-supervised, or even unsupervised, setting, where the definition of robustness needs to focus on the quality of the learned representations rather than



'# this movie is really stupid and very boring most of the time there are almost no ghoulies in it at all there is nothing good about this movie on any level just more bad actors pathetically attempting to make a movie so they can get enough money to eat avoid at all costs.' (**IMDB**)	'# well I am the target market I loved it furthermore my husband also a boomer with strong memories of the 60s liked it a lot too i haven't read the book so i went into it neutral i was very pleasantly surprised its now on our highly recommended video list br br.' (**IMDB**)
'The main story ... is compelling enough but it is difficult to shrug off the annoyance of that chatty fish.' (**SST**)	'Still this flick is fun and host to some truly excellent sequences.' (**SST**)
'i couldn't bear to watch it and I thought the UA loss was embarrassing ...' (***Twitter***)	'Is delighted by the beautiful weather.' (**Twitter**)

Fig. 10: Optimal robust explanations (highlighted in blue) for IMDB, SST and Twitter datasets (all the texts are correctly classified). Figure taken from [19].



`The film just might turn on many people to opera in general, an art form at once visceral and spiritual wonderfully vulgar and sublimely lofty.` (***SST***)

`There are far worse messages to teach a young audience which will probably be perfectly happy with the sloppy slapstick comedy.` (***SST***)

`This one is not nearly as dreadful as expected.` (**SST**)

□ Anchors □ Minimal Robust Extension

Fig. 11: Examples of Anchors explanations (in blue) along with the minimal extension required to make them robust (in red). Figure taken from [19].

classification (prediction) because of the lack of labels. This may involve working with similarity measures such as Mahalanobis distance and will be challenging both theoretically and computationally to achieve provable robustness guarantees.

c) Scalability in network width and depth: Despite much progress, the scalability of robustness certification and evaluation frameworks remains limited to low-dimensional models. In order to apply certification to realistic use cases (such as object detection) will necessitate significant improvements with respect to input dimensionality and network depth, as well as the types of activation functions that can be handled.

d) Efficiency and precision trade-off: Robustness certifications and evaluation involves a variety of methods, including exact, approximate and statistical. While exact methods offer completeness, trading off exact precision for approximate bounding results in more efficient any time methods, and completeness can be recovered by combining fast approximate methods such as convex relaxation with branch-and-bound computation. Statistical methods provide estimates of robustness that may be unsound but fast and, in many cases, sufficient for the application being considered.

e) Compositionality and modularity of AI systems: Certification tools that have been developed to date are monolithic, which matches the monolithic structure of the vast majority of neural network models. Yet, similarly to safety-critical systems, it is anticipated that better structuring of models and tools is likely to improve their reliability and maintainability. Therefore, modularity, compositionality and, in particular, assume-guarantee compositional frameworks, are desirable future directions.

f) Calibrating uncertainty: It is recognised that deterministic neural networks can be overconfident in their decisions, and instead a variant known as Bayesian neural networks (BNNs), which admits a distribution over the weights and provides outputs in the form of the posterior distribution, is preferred, as it allows for a principled means to return an uncertainty measure alongside the network output. BNN certification methodologies are much more complex than for deterministic NNs, and still in early stages of development, including uncertainty quantification [55], computing lower bounds on safety probability [56] and certifiable adversarial robustness [57]. Unfortunately, the methods do not scale beyond small networks and standard Bayesian inference tends to underestimate uncertainty.

g) Robust learning: A drawback of certification as presented in this paper is that it pertains to trained models, and if the model fails certification, it is not clear how it can be repaired, and expensive retraining may be needed. A natural question then arises as to whether one can learn a model that is guaranteed to be robust. Building on the positive and negative theoretical results in the case of robust learning against evasion attacks [58], [59], [60], [61], it would be interesting to generalise these results to neural network models and development of implementable frameworks that can provide provable guarantees on robustness.

VI. Conclusion

We have provided a brief overview of formal verification approaches that can be employed to certify neural network models at test time, to train certifiably robust or fair models, and to provide meaningful explanations for network predictions. The methods can be categorised into forward and backward analysis, and involve techniques such as search, bound propagation, constraint solving and abstract interpretation. Both forward and backward analysis have the potential to support more complex verification properties, which have been little explored to date. Empirical results obtained on a range of standard benchmarks show that neural network models are often brittle to adversarial perturbations, but verification approaches can be used to strengthen their robustness and compute certification guarantees, thus improving trustworthiness of AI decisions.

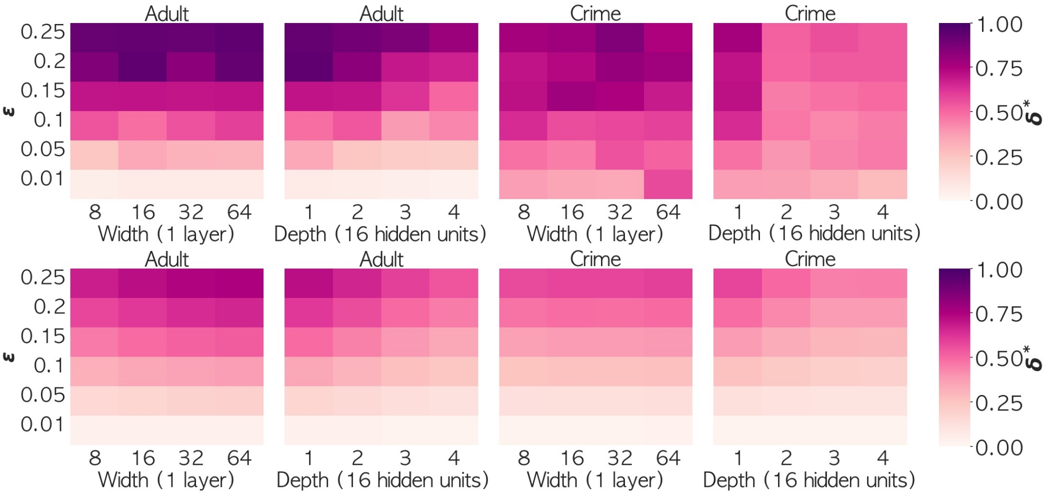


Fig. 12: Certified bounds on individual fairness (δ_*) for different architecture parameters (widths and depths) and maximum similarity (ϵ) for the Adult and the Crime datasets. Similarity metrics used are Mahalanobis (top row) and weighted ℓ_∞ metric (bottom row). Figure taken from [54].

Acknowledgments

This project received funding from the ERC under the European Union's Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No. 834115) and ELSA: European Lighthouse on Secure and Safe AI project (grant agreement No. 101070617 under UK guarantee).

References

[1] M. Wicker, X. Huang, and M. Kwiatkowska, "Feature-guided black-box safety testing of deep neural networks," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2018, pp. 408–426.

[2] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *29th International Conference on Computer Aided Verification*, ser. Lecture Notes in Computer Science, vol. 10426. Springer, 2017. doi: 10.1007/978-3-319-63387-9_1 pp. 3–29.

[3] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *29th International Conference on Computer Aided Verification*, ser. Lecture Notes in Computer Science, vol. 10426. Springer, 2017. doi: 10.1007/978-3-319-63387-9_5 pp. 97–117.

[4] C. Brix, M. N. Müller, S. Bak, T. T. Johnson, and C. Liu, "First three years of the international verification of neural networks competition (VNN-COMP)," *CoRR*, vol. abs/2301.05815, 2023. doi: 10.48550/arXiv.2301.05815

[5] R. Falconmore, "On the role of explainability and uncertainty in ensuring safety of AI applications," Ph.D. dissertation, University of Oxford, UK, 2022.

[6] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Srndic, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *European Conference on Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science, vol. 8190. Springer, 2013. doi: 10.1007/978-3-642-40994-3_25 pp. 387–402.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[8] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018. doi: 10.1145/3243734.3264418 pp. 2154–2156.

[9] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *29th International Conference on Machine Learning*. icml.cc / Omnipress, 2012.

[10] M. Wu, M. Wicker, W. Ruan, X. Huang, and M. Kwiatkowska, "A game-based approximate verification of deep neural networks with provable guarantees," *Theoretical Computer Science*, vol. 807, pp. 298–329, 2020. doi: 10.1016/j.tcs.2019.05.046

[11] K. Leino, Z. Wang, and M. Fredrikson, "Globally-robust neural networks," in *38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 6212–6222.

[12] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2020. doi: 10.1145/3359786

[13] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[14] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should I trust you?": Explaining the predictions of any classifier," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016. doi: 10.1145/2939672.2939778 pp. 1135–1144.

[17] ——, "Anchors: High-precision model-agnostic explanations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[18] A. Ignatiev, N. Narodytska, and J. Marques-Silva, "Abduction-based explanations for machine learning models," in *Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33011511 pp. 1511–1519.

[19] E. L. Malfa, R. Michelmore, A. M. Zbrzezny, N. Paoletti, and M. Kwiatkowska, "On guaranteed optimal robust explanations for NLP models," in *Thirtieth International Joint Conference on Artificial Intelligence*. ijcai.org, 2021. doi: 10.24963/ijcai.2021/366 pp. 2658–2665.

[20] A. Janusz, D. Slezak, S. Stawicki, and K. Stencel, "A practical study of methods for deriving insightful attribute importance rankings using decision bireducts," *Inf. Sci.*, vol. 645, p. 119354, 2023. doi: 10.1016/j.ins.2023.119354

[21] B. Wang, S. Webb, and T. Rainforth, "Statistically robust neural network classification," in *Uncertainty in Artificial Intelligence*. PMLR, 2021, pp. 1735–1745.

[22] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev, "AI2: safety and robustness certification of neural networks with abstract interpretation," in *IEEE Symposium on Security and*

Privacy. IEEE Computer Society, 2018. doi: 10.1109/SP.2018.00058 pp. 3–18.

[23] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Efficient formal safety analysis of neural networks," in *Annual Conference on Neural Information Processing Systems*, 2018, pp. 6369–6379.

[24] G. Singh, T. Gehr, M. Püschel, and M. T. Vechev, "An abstract domain for certifying neural networks," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 41:1–41:30, 2019. doi: 10.1145/3290354

[25] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, and J. Z. Kolter, "Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Annual Conference on Neural Information Processing Systems*, 2021, pp. 29 909–29 921.

[26] K. Matoba and F. Fleuret, "Exact preimages of neural network aircraft collision avoidance systems," in *Proceedings of the Machine Learning for Engineering Modeling, Simulation, and Design Workshop at Neural Information Processing Systems*, 2020, pp. 1–9.

[27] S. Dathathri, S. Gao, and R. M. Murray, "Inverse abstraction of neural networks using symbolic interpolation," in *Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33013437 pp. 3437–3444.

[28] S. Kotha, C. Brix, Z. Kolter, K. Dvijotham, and H. Zhang, "Provably bounding neural network preimages," *CoRR*, vol. abs/2302.01404, 2023. doi: 10.48550/arXiv.2302.01404

[29] X. Zhang, B. Wang, and M. Kwiatkowska, "On preimage approximation for neural networks," *CoRR*, vol. abs/2305.03686, 2023. doi: 10.48550/arXiv.2305.03686

[30] A. Janusz, A. Zalewska, L. Wawrowski, P. Biczyk, J. Ludziejewski, M. Sikora, and D. Slezak, "Brightbox - A rough set based technology for diagnosing mistakes of machine learning models," *Appl. Soft Comput.*, vol. 141, p. 110285, 2023. doi: 10.1016/j.asoc.2023.110285

[31] E. Wong and J. Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 5283–5292.

[32] P. Cousot and R. Cousot, "Abstract interpretation frameworks," *J. Log. Comput.*, vol. 2, no. 4, pp. 511–547, 1992. doi: 10.1093/logcom/2.4.511

[33] M. Wu and M. Kwiatkowska, "Robustness guarantees for deep neural networks on videos," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00039 pp. 308–317.

[34] H. Zhang, T. Weng, P. Chen, C. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Annual Conference on Neural Information Processing Systems*, 2018, pp. 4944–4953.

[35] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C. Hsieh, "Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," in *9th International Conference on Learning Representations*. OpenReview.net, 2021.

[36] P. Cousot and R. Cousot, "Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Fourth ACM Symposium on Principles of Programming Languages*. ACM, 1977. doi: 10.1145/512950.512973 pp. 238–252.

[37] M. Mirman, T. Gehr, and M. T. Vechev, "Differentiable abstract interpretation for provably robust neural networks," in *35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 3575–3583.

[38] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. T. Vechev, "Fast and effective robustness certification," in *Annual Conference on Neural Information Processing Systems*, 2018, pp. 10 825–10 836.

[39] A. Albarghouthi, "Introduction to neural network verification," *Found. Trends Program. Lang.*, vol. 7, no. 1-2, pp. 1–157, 2021. doi: 10.1561/2500000051

[40] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *Automated Technology for Verification and Analysis*. Springer International Publishing, 2017, pp. 269–286.

[41] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *10th International Symposium on NASA Formal Methods*, ser. Lecture Notes in Computer Science, vol. 10811. Springer, 2018. doi: 10.1007/978-3-319-77935-5_9 pp. 121–138.

[42] M. Fischetti and J. Jo, "Deep neural networks and mixed integer linear optimization," *Constraints An Int. J.*, vol. 23, no. 3, pp. 296–309, 2018. doi: 10.1007/s10601-018-9285-6

[43] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *7th International Conference on Learning Representations*. OpenReview.net, 2019.

[44] J. P. Vielma, "Mixed integer linear programming formulation techniques," *SIAM Rev.*, vol. 57, no. 1, pp. 3–57, 2015. doi: 10.1137/130915303

[45] R. Bunel, J. Lu, I. Turkaslan, P. H. S. Torr, P. Kohli, and M. P. Kumar, "Branch and bound for piecewise linear neural network verification," *J. Mach. Learn. Res.*, vol. 21, pp. 42:1–42:39, 2020.

[46] S. Wang, K. Pei, J. Whitehouse, J. Yang, and S. Jana, "Formal security analysis of neural networks using symbolic intervals," in *27th USENIX Security Symposium*. USENIX Association, 2018, pp. 1599–1614.

[47] M. Sotoudeh, Z. Tao, and A. V. Thakur, "Syrenn: A tool for analyzing deep neural networks," *Int. J. Softw. Tools Technol. Transf.*, vol. 25, no. 2, pp. 145–165, 2023. doi: 10.1007/s10009-023-00695-1

[48] A. Albarghouthi and K. L. McMillan, "Beautiful interpolants," in *25th International Conference on Computer Aided Verification*, ser. Lecture Notes in Computer Science, vol. 8044. Springer, 2013. doi: 10.1007/978-3-642-39799-8_22 pp. 313–329.

[49] W. Ruan, M. Wu, Y. Sun, X. Huang, D. Kroening, and M. Kwiatkowska, "Global robustness evaluation of deep neural networks with provable guarantees for the hamming distance," in *Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 5944–5952.

[50] T. Baluta, Z. L. Chua, K. S. Meel, and P. Saxena, "Scalable quantitative verification for deep neural networks," in *43rd IEEE/ACM International Conference on Software Engineering*. IEEE, 2021. doi: 10.1109/ICSE43902.2021.00039 pp. 312–323.

[51] P. Yang, R. Li, J. Li, C. Huang, J. Wang, J. Sun, B. Xue, and L. Zhang, "Improving neural network verification through spurious region guided refinement," in *27th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, vol. 12651. Springer, 2021, pp. 389–408.

[52] E. L. Malfa, M. Wu, L. Laurenti, B. Wang, A. Hartshorn, and M. Kwiatkowska, "Assessing robustness of text classification through maximal safe radius computation," in *Findings of the Association for Computational Linguistics*, ser. Findings of ACL, vol. EMNLP 2020. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.266 pp. 2949–2968.

[53] E. L. Malfa and M. Kwiatkowska, "The king is naked: On the notion of robustness for natural language processing," in *Thirty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2022, pp. 11 047–11 057.

[54] E. Benussi, A. Patanè, M. Wicker, L. Laurenti, and M. Kwiatkowska, "Individual fairness guarantees for neural networks," in *Thirty-First International Joint Conference on Artificial Intelligence*. ijcai.org, 2022. doi: 10.24963/ijcai.2022/92 pp. 651–658.

[55] R. Michelmore, M. Wicker, L. Laurenti, L. Cardelli, Y. Gal, and M. Kwiatkowska, "Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control," in *IEEE International Conference on Robotics and Automation*. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9196844 pp. 7344–7350.

[56] M. Wicker, L. Laurenti, A. Patane, and M. Kwiatkowska, "Probabilistic safety for bayesian neural networks," in *Thirty-Sixth Conference on Uncertainty in Artificial Intelligence*, ser. Proceedings of Machine Learning Research, vol. 124. AUAI Press, 2020, pp. 1198–1207.

[57] M. Wicker, L. Laurenti, A. Patane, N. Paoletti, A. Abate, and M. Kwiatkowska, "Certification of iterative predictions in bayesian neural networks," in *Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, ser. Proceedings of Machine Learning Research, vol. 161. AUAI Press, 2021, pp. 1713–1723.

[58] P. Gourdeau, V. Kanade, M. Kwiatkowska, and J. Worrell, "On the hardness of robust classification," in *Advances in Neural Information Processing Systems*, 2019, pp. 7444–7453.

[59] ——, "On the hardness of robust classification," *Journal of Machine Learning Research*, vol. 22, 2021.

[60] ——, "Sample complexity bounds for robustly learning decision lists against evasion attacks," in *Thirty-First International Joint Conference on Artificial Intelligence*. ijcai.org, 2022. doi: 10.24963/ijcai.2022/419 pp. 3022–3028.

[61] ——, "When are local queries useful?" in *Advances in Neural Information Processing Systems*, 2022.

DOI: 10.15439/2023F6419
ISSN 2300-5963 ACSIS, Vol. 35

Multiple Criteria Decision Aiding by Constructive Preference Learning (Keynote Lecture – Extended Abstract)

Roman Słowiński*†
*Poznań University of Technology, Poland
†Systems Research Institute, Polish Academy of Sciences, Poland

The notion of preference is relevant across a variety of scientific disciplines, including economics and social sciences, operational research and decision sciences, artificial intelligence, psychology, and philosophy. Preferences provide a means for specifying desires in a declarative and intelligible way, a key element for the effective representation of knowledge and reasoning respecting the value systems of Decision Makers (DMs) [1].

Recognizing the preferences of DMs is also crucial for multi-criteria decision aiding. We present a constructive preference learning methodology, called robust ordinal regression (ROR) [2]. This methodology links operational research (OR) with artificial intelligence (AI), and as such, it confirms the current trend in mutual relations between OR and AI [3].

The lecture starts from an observation that the dominance relation established in the set of alternatives evaluated on multiple attributes (criteria, or voters, or states of the nature) is the only objective information that stems from the formulation of a multiple attribute decision problem (ordinal classification, or ranking, or choice – with multiobjective optimization being a particular case). While it permits to eliminate many irrelevant (i.e., dominated) alternatives, it leaves many alternatives incomparable. This situation may be addressed by taking into account preferences of the DM. Therefore, decision aiding methods require some preference information exhibiting a value system of a single or multiple DMs.

In ROR, the preference information has the form of decision examples. They may either be provided by the DM on a set of real or hypothetical alternatives, or may come from observation of DM's past decisions. This information is used to build a preference model, which is then applied on a non-dominated set of alternatives to arrive at a recommendation presented to the DM(s).

In practical decision aiding, the process composed of preference elicitation, preference modeling, and DM's analysis of a recommendation, loops until the DM (or a group of DMs) accepts the recommendation or decides to change the problem setting. Such an interactive process is called constructive preference learning. We describe this process for three types of preference models:

1) utility functions,
2) outranking relations, and
3) sets of monotonic decision rules.

The case of a hierarchical structure of the set of criteria will be discussed [4], and the transparency and explainability features required from preference learning will be discussed on the example of interactive multiobjective optimization [5].

References

[1] E. Hüllermeier and R. Słowiński, "Preference learning and multiple criteria decision aiding: differences, commonalities, and synergies," *Submitted*, 2023.

[2] S. Greco, V. Mousseau, and R. Słowiński, "Ordinal regression revisited: multiple criteria ranking using a set of additive value functions," *European Journal of Operational Research*, vol. 191, no. 2, pp. 415–435, 2008.

[3] S. Corrente, S. Greco, M. Kadziński, and R. Słowiński, "Robust ordinal regression in preference learning and ranking," *Machine Learning*, vol. 93, pp. 381–422, 2013.

[4] S. Corrente, S. Greco, and R. Słowiński, "Multiple criteria hierarchy process in robust ordinal regression," *Decision Support Systems*, vol. 53, no. 3, pp. 660–674, 2012.

[5] S. Corrente, S. Greco, B. Matarazzo, and R. Słowiński, "Explainable interactive evolutionary multiobjective optimization," *Omega*, p. 102925, 2023.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 41–48
DOI: 10.15439/2023F8996
ISSN 2300-5963 ACSIS, Vol. 35

Online Learning Framework for Radio Link Failure Prediction in FANETs

Kiril Danilchenko
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada
kdanilch@uwaterloo.ca

Nir Lazmi
School of Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
nirlazm@post.bgu.ac.il

Michael Segal, *Senior Member*, IEEE
School of Electrical and Computer Engineering
Ben-Gurion University of the Negev
Beer-Sheva, Israel
segal@bgu.ac.il

***Abstract*—In this paper, we consider the problem of prediction of Radio Link Failures (RLF) in flying ad hoc networks (FANETs). Many environmental factors that influence the quality of radio wave propagation are dynamic, and thus, drones must continually learn and update their radio link quality prediction model while they operate online.**

Online machine learning algorithms can be used to build adaptive RLF predictors without requiring a pre-deployment effort. To predict the RLF, we use an online machine learning algorithm and information gathering by message-passing from the neighbors. We propose an algorithm called *ML-Net* (Machine Learning and Network algorithm) to predict RLF. To the best of our knowledge, the combination of online machine learning algorithms together with the message-passing algorithm has not been used before. The proposed methodology outperforms the state-of-the-art online machine learning algorithms.

***Index Terms*—Online learning, RLF prediction, UAV.**

I. Introduction

UNMANNED aerial vehicles (UAVs), also known as drones or flying robots, have gained significant attention in various real-life applications. The flying ad hoc network (FANET) is established to leverage high-speed communications. However, due to the high mobility of UAVs in FANETs, the network topology may continuously change, making it challenging to establish end-to-end connections. Radio Link Failure (RLF) prediction can help UAVs handle this issue and improve FANET performance to ensure continuous service availability. Accurate prediction of radio link failures (RLF) is critical for ensuring reliable communications in flying ad hoc networks (FANETs). However, link prediction remains challenging due to the dynamic topology and unpredictable mobility patterns in FANETs. Nodes can move in and out of communication range rapidly, leading to frequent link disruptions. The ability to accurately predict impending link failures can enable proactive mitigation strategies. For example, drones could switch to more reliable links ahead of time to prevent packet loss and service interruptions. Link prediction also allows optimizing routing by avoiding unstable links that are about to fail. Furthermore, timely knowledge of upcoming link losses enables adapting transmission parameters to maintain connectivity. The development of link prediction techniques tailored to the FANET environment is therefore essential for efficient network operation and robust aeronautical communications. Machine learning holds promise for developing adaptive, data-driven predictors that can operate in real-time based on local interactions. This motivates our exploration of online learning combined with message passing for high-accuracy RLF prediction in FANETs.

Two essential characteristics should be present in RLF predictors. First, adaptivity is critical since an RLF predictor must cope with quality fluctuations over time, especially for FANET. Second, plug-and-play is crucial since RLF predictors should be applied without requiring any predeployment effort, which might not be feasible for all deployment scenarios, even if the effort is reduced. Online machine learning algorithms can be used to build RLF predictors that are adaptive without requiring predeployment effort.

Most current machine learning approaches are limited to the traditional batch setting, where data is provided in advance to the training process. Model selection and meta-parameter optimization can rely on the full set of data, and training can assume that the data and its underlying structure are static. In contrast, online machine learning involves continuous model adaptation based on constantly arriving data. The underlying distribution of the data, which changes over time, presents some primary challenges and difficulties in the dynamic environment. Old data can become irrelevant or even detrimental to model the current concept. Online machine learning [3] exhibits great potential for performance improvement with the sequential arrival of data and superiority over offline learning, including real-time predictions and lower memory requirements.

This paper explores the possibility of using online machine learning algorithms together with local information gathering by messages to predict RLF. Each drone uses only its local information and information gathered from its neighbor. Specifically, the main contributions of this paper are as follows.

1) Motivated by the characteristics of RLF predictors mentioned above, we propose a link failure prediction model *ML-Net*, which combines the online machine learning algorithm with a message-passing algorithm.
2) To the best of our knowledge, we pioneer the use of the online machine learning method combined with the message-passing algorithm for RLF prediction.
3) We conduct simulations to analyze the performance of the *ML-Net*. We compare the proposed solution with state-of-the-art techniques and perform an analysis of simulation results. The analysis shows that the proposed approach *ML-Net* achieves better performance than the state-of-the-art methods.

The remainder of this paper is organized as follows. Section II discusses recent related studies. Section III describes the theoretical model used to define our problem. The details of the proposed approach are described in Section IV. In Section V, we describe all aspects of the evaluation setup and simulation results. Finally, we summarize our paper with conclusions and suggestions for future research in Section VI.

II. Related Work

In this study, we propose the use of online machine learning algorithms combined with message-passing approach for predicting Radio Link Failure (RLF) in communication networks. Several recent studies on RLF suggest using online machine learning algorithms for prediction.

In [1], the authors studied the link quality prediction for wireless mesh networks. They performed a performance analysis of four state-of-the-art algorithms for link quality prediction and proposed a new hybrid online algorithm for link quality prediction based on this analysis.

The authors of [7] presented an adaptive link estimator (TALENT) that uses online learning. They argued that TALENT adapts to network dynamics better than statically trained models without the need for advance data collection for training the model before deployment in a real system. The solutions follows the performance of the autoencoders very closely with a tiny margin on very bad, very good and intermediate link quality classes.

The study [8] introduced a framework to reduce energy and network capacity overhead expenses by incorporating active learning to selectively label only a portion of the samples from the data stream. The framework also uses incremental training batches to conserve labeling resources and updates the batches using change detection and forgetting mechanisms to mitigate concept drift. Experimental results showed that the framework reduces label queries by up to 21.5% and prediction error by up to 9% after periods of concept drift.

As a part of their work in [12], the authors looked at the problem of predicting channel quality between vehicles in terms of path loss, which shows strong fluctuations over time due to the highly dynamic nature of vehicular environments. They proposed a framework for a data-driven path loss prediction model that combines the changepoint detection method and online learning. The evaluation of the proposed framework was done using real-world datasets.

Machine learning methods were used in [10] to predict the short-term evolution of link quality for switching to a better link for data transmission. The problem was modeled as a game of prediction based on experts' advice, using the Link-Quality Indicator (LQI) metric. A decision-maker predicts the LQI values, called a forecaster, who receives advice from several experts. To predict values close to actual LQI values, the forecaster can learn how to adapt its strategy. As a general model, the proposed learning and prediction model can be easily adapted to different link-quality metrics or prediction methods.

In summary, some recent works have used online machine-learning algorithms to predict RLF and took into consideration the adaptability issue, see [2]. Based on these previous studies, we propose the use of a novel method based on online machine learning with a combination of neighbor information gathering by a message-passing solution.

III. System Model and Problem Formulation

We consider a set of n UAVs (denoted as S) that are deployed arbitrarily in a given area and may be located in any position in $\mathbb{R}^3$. The network imposed on S is connected.

We assume that the transmission power of all nodes is fixed. We denote the transmission power of node i at time t as P_i^t. The set of nodes i's neighbors at time t depends on the current positions and channel gain. Based on [13], we can approximate the path loss on link ij as follows:

$$L_{ij}^t = -10\log_{10} G_l \left(\frac{\lambda}{4\pi d^t}\right)^2, \tag{1}$$

where G_l is the product of the transmitter and receiver antenna field radiation patterns of LOS transmissions, d^t is the distance between a transmitter and its corresponding receiver at time t, and λ is the operating wavelength. Thus, the path loss is related to the transmission distance when transmitting radio signals over a specific channel through a specific antenna.

Let N_i^t be the set of neighbors of node i at time t. Formally, N_i^t is defined as:

$$N_i^t = \{j \in S \setminus i | L_{ij}^t \geq \theta\}, \tag{2}$$

where θ is a predefined parameter of the network.

A network is defined as a directed graph $G^t = (S, E^t)$, where node $i \in S$ has a directed edge (link) to each of its neighbors according to Equation 2. The edge set E^t is the union of all directed edges among the nodes in time t, $E^t = \bigcup_{i \in S} E_i^t$.

At each timestamp t, when node i receives a message from node j, node i collects a set x_t of metrics (features) that describe the communication with node j at time t. Node i adds x_t to a multivariate time series that represents features of communication with node j, denoting this multivariate time series as $X_{ij}^t = \{x_1, \ldots, x_t\}$.

Thus, each edge $e^t_{ij} \in E^t$ has a multivariate time series associated with it, X^t_{ij}. In addition, e^t_{ij} has an associated binary class variable $v^t_{ij} \in \{1, -1\}$, representing the existence (1) or failure (-1) of the edge.

Now we can formally define our problem. Given the current X^t_{ij}, we wish to determine whether the edge e_{ij} will fail or not in the following timestamp (namely, at $t + 1$).

IV. *ML-Net*

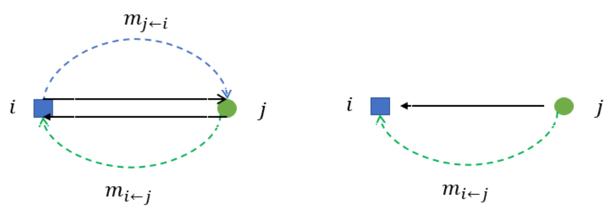
In this section, we describe the proposed *ML-Net* approach. We combine the machine online learning algorithm with the network structure algorithm to cope with RLF prediction in the next timestamp. We suggest using the expression $S(X^t_{ij}, m^t_{i\leftarrow j})$ which approximates the probability that the edge e^t_{ij} will fall in the next timestamp.

$$S(X^t_{ij}, m^t_{i\leftarrow j}) = \begin{cases} \alpha \hat{Pr}(v^{t+1}_{ij} = -1|X^t_{ij}) + (1-\alpha) m^t_{i\leftarrow j}, & \text{Message from node } j \text{ was received} \\ \hat{Pr}(v^{t+1}_{ij} = -1|X^t_{ij}), & \text{otherwise} \end{cases} \tag{3}$$

where $0 \leq \alpha \leq 1$, $m^t_{i\leftarrow j}$ is the message sent by j to i including the belief of node j about v^t_{ji}, and $\hat{Pr}(v^{t+1}_{ij} = -1|X^t_{ij})$ is the probability that edge e_{ij} will fail in timestamp $t+1$ given multivariate time series X^t_{ij}. We use on the shelf online machine learning algorithm to calculate $\hat{Pr}(v^{t+1}_{ij} = -1|X^t_{ij})$ (in Section V-A we will broadly describe the proposed algorithms).

$$m^t_{i\leftarrow j} = \begin{cases} \alpha \hat{Pr}(v^{t+1}_{ji} = -1|X^t_{ji}) + (1-\alpha) m^t_{j\leftarrow i}, & \text{Message from node } j \text{ was received} \\ \hat{Pr}(v^{t+1}_{ji} = -1|X^t_{ji}), & \text{otherwise} \end{cases} \tag{4}$$

Due to the fact that the $G^t = (S, E^t)$ is a directed graph, then the following scenario may occur: the edge $e^t_{i,j}$ exists but the edge $e^t_{j,i}$ does not exist. In this scenario, $m^t_{i\leftarrow j}$ may arrive but $m^t_{j\leftarrow i}$ cannot arrive. In Figure 1 we can see an example of this scenario.



a) Edges e_{ij} and e_{ji} exist

b) Edge e_{ij} exists

Fig. 1: Directed edges.

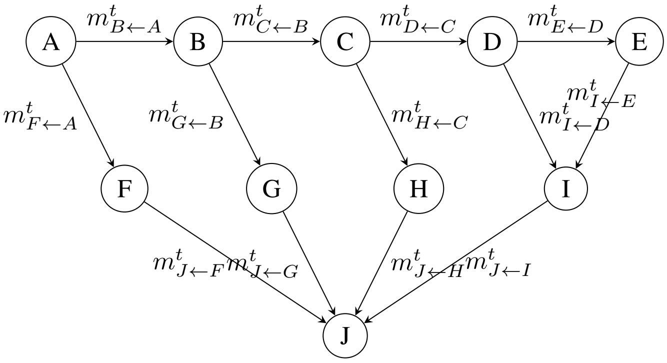


Fig. 2: Illustration of Message-Passing Algorithm

Figure 2 showcases a random graph with 10 nodes (A-J), representing entities in a system. The directed edges between nodes symbolize the flow of messages, demonstrating the operation of the message-passing algorithm. The labels on the arrows, m^t_{ij}, represent the messages passed from node i to node j at time t. This graph serves as a visual representation of the algorithm's functioning in a system with multiple interacting components.

Algorithm 1 ML-Net Algorithm

```
Input: Node $i$, current time $t$, set of neighbors $N^t_i$, link features $X^t_{ij}$
Output: Link failure prediction $S(X^t_{ij}, m^{t\leftarrow j}_i)$
Hyperparameters: $\alpha$
for $j \in N^t_i$ do
    if $msg_received(j, t)$ then
        $m^{t\leftarrow j}_i = \alpha \hat{P}(v^{t+1}_{ij} = -1|X^t_{ij}) + (1-\alpha) m^{t\leftarrow i}_j$
    else
        $m^{t\leftarrow j}_i = \hat{P}(v^{t+1}_{ij} = -1|X^t_{ij})$
    end if
    $S(X^t_{ij}, m^{t\leftarrow j}_i) = \alpha \hat{P}(v^{t+1}_{ij} = -1|X^t_{ij}) + (1-\alpha) m^{t\leftarrow j}_i$
end for
return $S(X^t_{ij}, m^{t\leftarrow j}_i)$
```

The ML-Net algorithm shows the key steps for combining online machine learning with message passing to predict link failures. For each neighbor j, node i first checks if a message was received from j at time t. If so, i updates its belief about the $i \rightarrow j$ link failing using both its own prediction and j's belief from the message. This allows propagating information through the network. If no message is received, i relies only on its own link prediction. Finally, the algorithm returns the combined prediction score S for each link $i \rightarrow j$ based on the updated beliefs. This demonstrates how ML-Net leverages both local online learning models and information exchange with neighbors to achieve accurate real-time RLF prediction in dynamic FANET environments.

A. *Notations*

The notations used in this paper are summarized in Table I. Also, Table I contains further notation defined below in the paper.

Symbol	Meaning
S	Set of drones
L_{ij}^t	Path loss between receiver j and transmitter i
n	Size of set S, $n = \|S\|$
$\mathcal{K}$	The set of sources and destination in the network
$G^t = (S, E^t)$	Directed graph that represents the network in timestamp t
X_{ij}^t	multivariate time series associated with measures from link ij
P_i	The transmission power
v_{ij}^t	Binary variable represented the link failure between node i to j in time t
e_{ij}^t	Edge between node i to j in time t
$m_{i \leftarrow j}^t$	Message sent by j to i (contains v_{ji}^t)

TABLE I: Summary of notations used in this paper

V. Experimental Setup

Simulation is the most reliable and cost-effective approach to examining the performance of real-world problems. For performance evaluation of the proposed technique, OMNET++ [15] is used as the primary tool for FANET environment generation and routing protocol implementation. This section describes the data generation process. Let $\mathcal{K}$ represent the set of the source and destination pairs within the network that should communicate.

We simulated the network for $|S| = 20$ and for $|\mathcal{K}| = 3$. In Figure 3 we can see an example of a network with 20 UAV's and $|\mathcal{K}| = 3$.

The routing protocol we chose to use is AODV (Ad-Hoc On-Demand Distance Vector) [14]. This protocol is designed for wireless and mobile ad hoc networks and has been broadly adopted for FANETs. Next, we present the schemes for comparison and the evaluation metrics.

A. Schemes for Comparison

In the following, we briefly describe the state-of-the-art online machine learning algorithms and their hyperparameters used in this paper to compare with *ML-Net*. These algorithms were selected as the comparison baseline solutions due to their advanced nature.

1) The Extremely Fast Decision Tree (EFDT) [9]: EFDT constructs a tree incrementally. The EFDT seeks to select and deploy a split as soon as it is confident the split is useful and then revisits that decision, replacing the split if it becomes evident that a better split is available. The EFDT learns rapidly from a stationary distribution, and eventually, it learns the asymptotic batch tree if the distribution from which the data are drawn is stationary. We chose a EFDT with the following hyperparameters: grace period of 200 and split confidence of $1e^{-7}$.

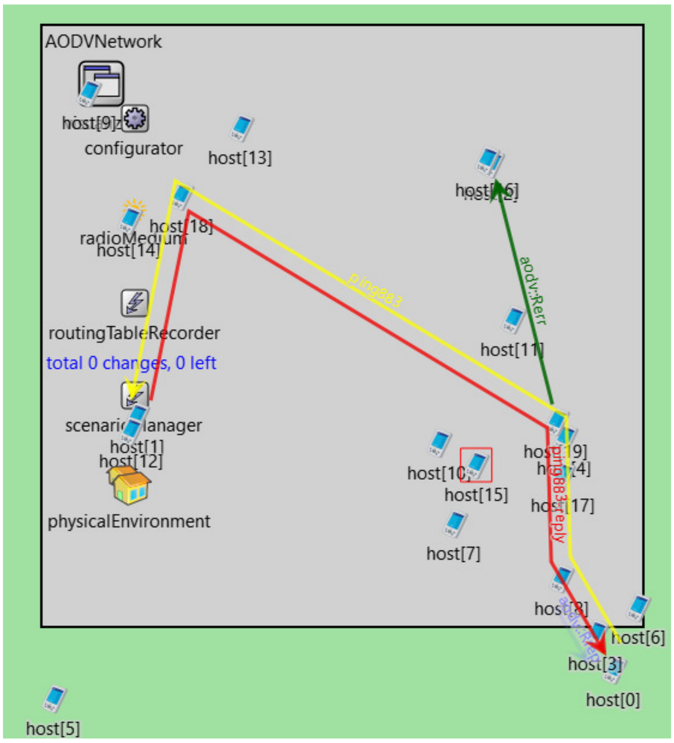


Fig. 3: Example of network with 20 nodes and 3 pairs of source and destination.

2) Adaptive Random Forest (ARF) [4]: ARF algorithm for the classification of evolving data streams enables the Random Forests algorithm for evolving data stream learning. There are effective resampling methods and flexible operators in ARF that can cope with different types of concept drifts without requiring complex optimizations for different data sets. We chose a ARF with the following hyperparameters: grace period of 50, lambda is 6, max features are 3, split confidence of 0.01, and tie threshold of 0.05.

3) Hoeffding Tree Classifier (HTC) [6]: This method involves determining an upper bound for the learner's loss based on the number of examples used in each step of the algorithm. In this algorithm, the number of examples required for each step is minimized while ensuring the model performance is not significantly different from the one obtained from an infinite dataset. We chose a HTC with the following hyperparameters: grace period of 200, min samples reevaluate of 20, split confidence of $1e^{-7}$, and tie threshold of 0.05.

4) Streaming Random Patches Classifier (SRPC) [5]: The Streaming Random Patches algorithm is a machine learning technique that involves randomly selecting subsets or ”patches” of data from a continuous stream of input. These patches are used to train and update a model iteratively over time, allowing the model to adapt and learn from new incoming data.

B. Evaluation Metric

We use the following performance metrics to evaluate the performance of different machine learning algorithms:

1) Precision: The ratio of the number of true positives over the total number classified as positive. In the context of our problem, it is the ratio of correctly predicted link failures versus the total number of link failures predicted. The precision value is computed as follows:

$$P = \frac{T_P}{T_P + F_P}, \tag{5}$$

where P is the precision value, T_P is the number of "true positives," and F_P is the number of "false positives."

2) Recall: The ratio of the number of data points associated with link faults correctly classified over the total the number of data points associated with link faults that have occurred. The recall value is given by

$$R = \frac{T_P}{T_P + F_N} \tag{6}$$

3) F_1-Score: The F_1-Score is the harmonic average of the precision and recall values. It takes a value in $[0, 1]$. Higher the value of F_1-Score, the better the performance of the machine learning technique. It is computed as follows:

$$F_1 = \frac{2P \cdot R}{P + R} \tag{7}$$

4) Cohen's kappa- κ : Cohen's kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories the Cohen's kappa formula can be written as:

$$\kappa = \frac{2(T_P \cdot T_N - F_N \cdot F_P)}{(T_P + F_P) \cdot (F_P + T_N) + (T_P + F_N) \cdot (F_N + T_N)} \tag{8}$$

C. Features

This section provides an exploratory data analysis of key features in the communication system. Visualizations are utilized to understand feature distributions and relationships in the data.

1) **Time**: This feature represents the elapsed time since the start of the simulation or experiment. It captures the dynamics of the communication system over time.
2) **IdTransmitter**: This categorical feature identifies the transmitter for each signal received. It allows the model to capture any specific characteristics related to individual transmitters.
3) **64-QAM SNR**: These features represent the Signal-to-Noise Ratio (SNR) for the 64-level Quadrature Amplitude Modulation (64-QAM) scheme. SNR is a measure of signal quality, with higher values generally indicating better quality.
4) **64-QAM BER**: These features represent the Bit Error Rate (BER) for 64-QAM. BER is another measure of signal quality, with lower values indicating fewer errors and therefore better quality.

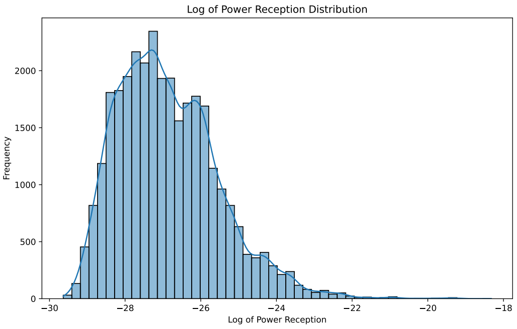


Fig. 4: Distribution of Log Received Power

5) **powR**: This feature represents the power of the received signal. It is a critical factor in communication systems, as a stronger received signal usually implies a better communication link.
6) **angle_between**: This feature represents the angle between the transmitter and receiver. The relative orientation between these devices can affect the signal strength and thus the link quality.
7) **Link**: This is the target variable that we want to predict. It is a binary variable representing the quality of the communication link, with '1' indicating a good link and '0' indicating a bad link. This allows us to frame the problem as a binary classification task.

These features provide a comprehensive description of each signal reception event, capturing information about the transmitter, signal quality, received signal power, and relative device orientation. This enables construction complex models to predict the communication link quality.

Figure 4 shows the histogram that presents the distribution of the logarithm of the received Power Reception. The x-axis shows the log power values, and the y-axis shows the frequency. The kernel density estimate smooth line approximates the probability density function.

The log transformation handles skewed data by making the distribution more symmetric. The log-transformed power distribution appears Gaussian, concentrated around -26.

The KDE peak indicates the most frequent log power value. The distribution spread provides insights into power variability. This transformation can help modeling techniques assume normality.

Figure 5 is a histogram that shows the distribution of angles between transmitters and receivers. The x-axis represents the angle from 0 to 180 degrees. The y-axis represents the frequency. This reveals common orientations in the simulation.

The overlaid Kernel Density Estimate (KDE) approximates the distribution. It assists in identifying shape characteristics like peaks. These insights are key to understanding angle variability and its potential impact on system performance.

The heatmap presented in Figure 6 illustrates feature correlations. Each cell shows the correlation coefficient between feature pairs. Cool colors indicate negative correlations. Warm

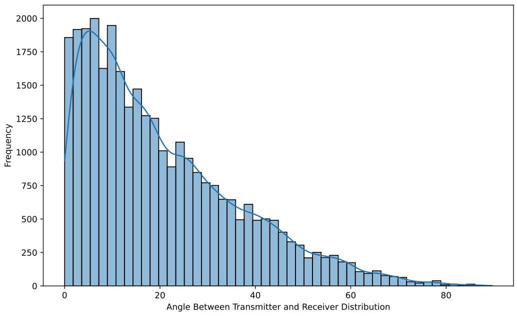


Fig. 5: Distribution of Transmitter-Receiver Angle

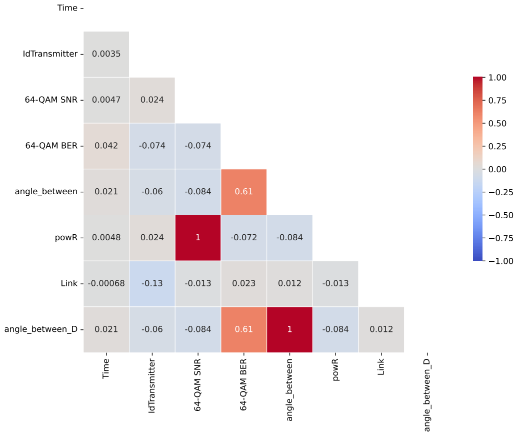


Fig. 6: Correlation Matrix

colors indicate positive correlations. Neutral colors indicate minimal correlation.

This visualization identifies relationships between features. It assists in feature selection by highlighting potential redundancies.

D. Evaluation and Numerical Results

We evaluate the proposed method using data collected from simulations implemented in OMNET++ [15]. As part of our simulation, we randomly deployed 20 drones and randomly selected three pairs of source destinations from these drones as shown in Figure 3 for an example. The drones run the AODV routing protocol.

Each node participating in communication extracts the following parameters (features) when receiving a message from its neighbor: reception power, transmission power, SINR, modulation, location, and orientation. Features collection does not require particular messages. Simulation runs for 3000 seconds, during which the link failure is collected.

Table II summarizes the parameters used by us in the simulation.

We first used the schemes from Section V-A to predict RLF so that we could compare our method with their performance. The compression mechanism is implemented using [11]. We calculate $\hat{Pr}(v_{ij}^{t+1} = -1|X_{ij}^{t})$ using all the schemes for comparison, estimate the RLF by Eq. 3, and provide the performance for each of the cases.

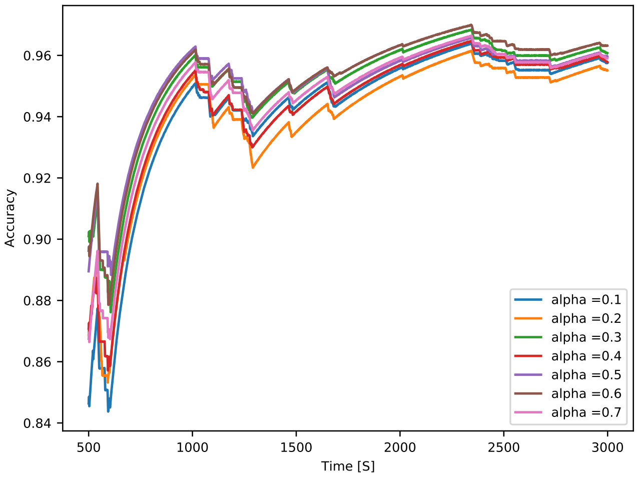


Fig. 7: α selection for *ML-Net*

Figure 7 shows the performance of *ML-Net* as a function of α. The x-axis represents the time of the simulation, and the y-axis represents the accuracy of the RLF prediction. As we can learn from the figure, the accuracy of the RLF prediction generally increases as we increase the value of α. This is because a larger value of α gives more weight to the node's own information and less to its neighbor's information. We can also see that when α is very small, the algorithm is overly influenced by the neighbors' information, and the predictions may not be accurate enough. On the other hand, when α is very large, the algorithm may not be able to adapt quickly to changes in the network topology due to a lack of influence from the neighbors' information. Based on Figure 7, we chose the alpha value to be 0.6, as it strikes a balance between the node's own information and its neighbor's information and achieves the best accuracy value.

Parameter	**Value**
Number of drones	20
Simulation Playground Size	800 $\times$ 800 m^2
Bandwidth	2MHz
σ	10^{-3}
Power	1.4mW
λ	10dB
SINR threshold	4dB
α	0.6
Speed	$\mathcal{N}(200, 0.001) mps$

TABLE II: Simulation Configuration

We provide a detailed explanation of the performance analysis of *ML-Net* with different online learning algorithms in calculating $\hat{Pr}(v^{t+1}ji = -1|X^{t}ji)$. To evaluate the performance of our proposed approach, we compare it with other

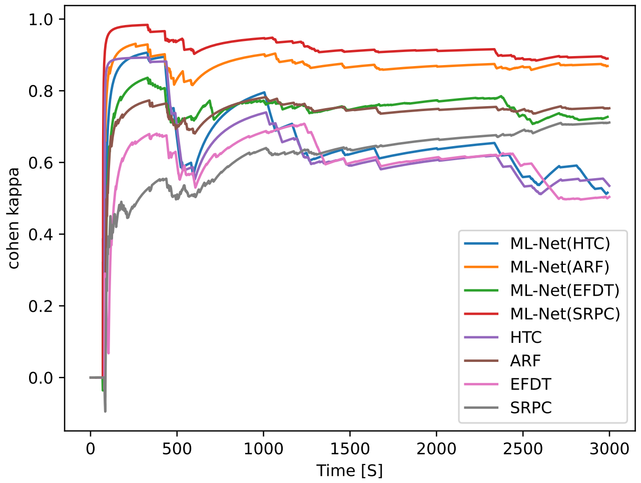


Fig. 8: κ achieved by schemes for comparison versus *ML-Net*

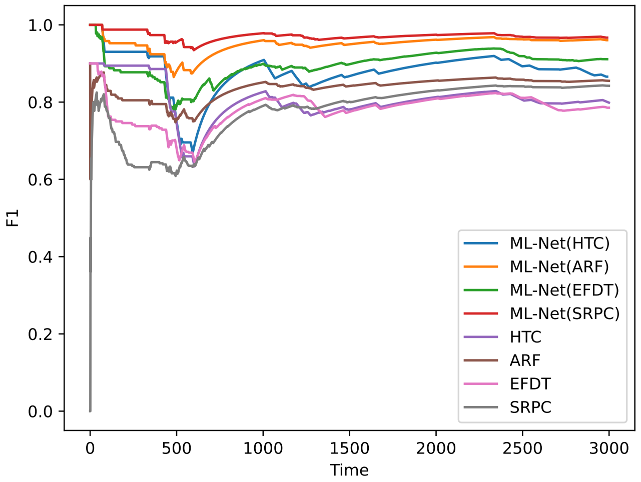


Fig. 9: F_1

existing schemes (see subsection V-A). The evaluation is based on several performance metrics, including precision, recall, accuracy, Cohen's kappa, and F_1.

Figures 9, 10, 11 and 12 demonstrate the performance of *ML-Net* with different online learning algorithms in calculating $\hat{Pr}(v^{t+1}ji = -1|X^tji)$. We observe that the proposed method outperforms the comparison schemes, regardless of the online learning algorithm used. This indicates that our approach is not dependent on the specific online learning algorithm utilized to calculate $\hat{Pr}(v^{t+1}ji = -1|X^tji)$.

Specifically, the evaluation was carried out using several performance metrics. Precision refers to the fraction of correctly identified negative instances among all the predicted negative instances. The recall is the fraction of correctly identified negative instances among all the actual negative instances. Accuracy refers to the fraction of correctly predicted instances among all the instances. Cohen's kappa is a statistical measure of inter-rater agreement between two raters for categorical items. Variable F_1 is the harmonic mean of precision and recall.

The results of the performance analysis are depicted in the figures mentioned above. The results show that the proposed approach outperforms the comparison schemes in terms of all the performance metrics evaluated. Therefore, we can conclude that *ML-Net* is an effective method for calculating $\hat{Pr}(v^{t+1}ji = -1|X^tji)$ in an online learning setting.

To summarize, we see that *ML-Net* outperforms, for each of the evaluation metrics, the previously known scheme that achieves the best performance for this metric.

VI. Conclusions

This paper addresses the critical issue of Radio Link Failures in FANETs, which stands as a significant challenge requiring innovative solution. We propose a novel approach that combines online machine learning algorithms with message-passing, a technique that has not been explored in this context before. Our proposed solution, known as *ML-Net*, significantly outperforms the best existing competitors across all evaluation metrics, indicating the immense potential of this approach in addressing real-time prediction challenges in communication networks.

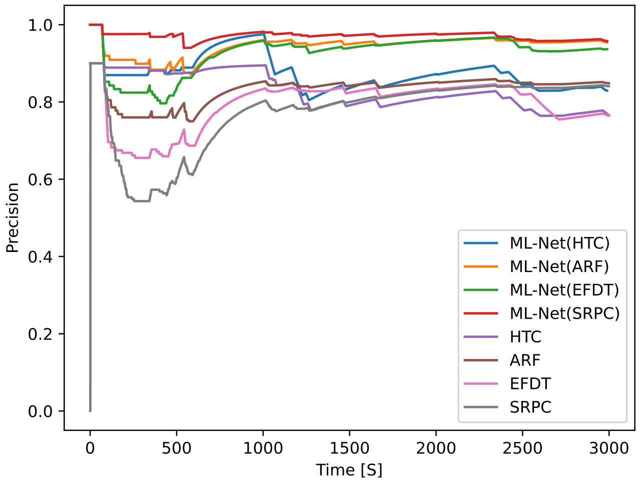


Fig. 10: Precision achieved by schemes for comparison vs. *ML-Net*

Our obtained results are highly promising and suggest that this novel combination of online learning algorithms and message-passing algorithms has the potential to revolutionize the field of FANETs and pave the way for more effective and efficient communication networks in the future. Also, it would be interesting to analyze analytically the influence of parameter α.

Acknowledgments

This research was (partially) funded by the Israeli Science Foundation (Grant No. 465/22), Israeli Ministry of Science (Grant No. 0005355), and by the Army Research Office under Grant Number W911NF-22-1-0225. The views and conclu-

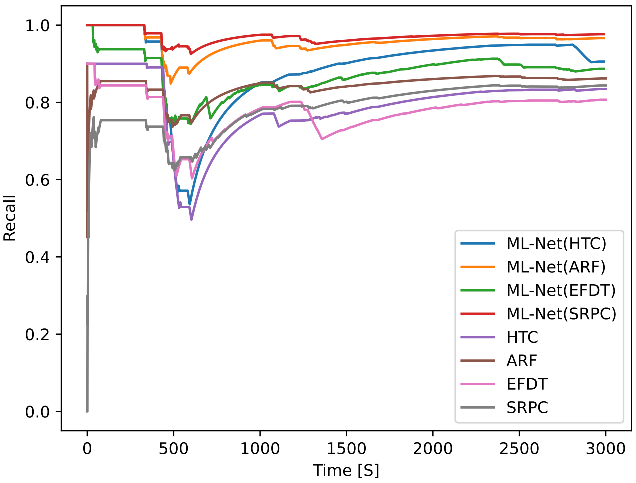


Fig. 11: Recall achieved by schemes for comparison vs. *ML-Net*

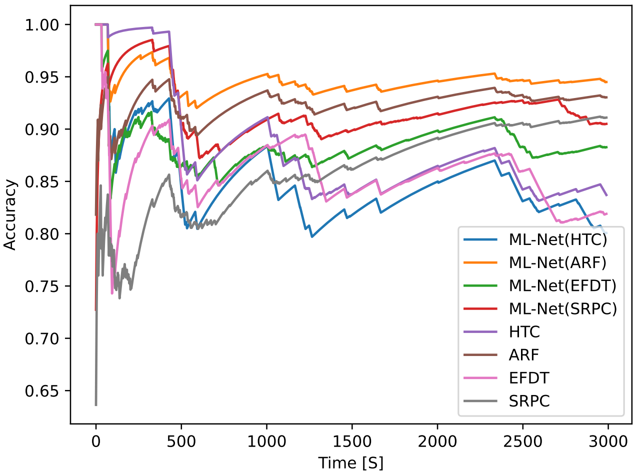


Fig. 12: Accuracy achieved by schemes for comparison vs. *ML-Net*

sions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Miguel L Bote-Lorenzo, Eduardo Gómez-Sánchez, Carlos Mediavilla-Pastor, and Juan I Asensio-Pérez. Online machine learning algorithms to predict link quality in community wireless mesh networks. *Computer Networks*, 132:68–80, 2018.

[2] Gregor Cerar, Halil Yetgin, Mihael Mohorcic, and Carolina Fortuna. Machine learning for wireless link quality estimation: A survey. *IEEE Commun. Surv. Tutorials*, 23(2):696–728, 2021.

[3] Óscar Fontenla-Romero, Bertha Guijarro-Berdiñas, David Martinez-Rego, Beatriz Pérez-Sánchez, and Diego Peteiro-Barral. Online machine learning. In *Efficiency and Scalability Methods for Computational Intellect*, pages 27–54. IGI Global, 2013.

[4] Heitor M Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9):1469–1495, 2017.

[5] Heitor Murilo Gomes, Jesse Read, and Albert Bifet. Streaming random patches for evolving data stream classification. In *2019 IEEE international conference on data mining (ICDM)*, pages 240–249. IEEE, 2019.

[6] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106, 2001.

[7] Tao Liu and Alberto E. Cerpa. Temporal adaptive link quality prediction with online learning. *ACM Trans. Sen. Netw.*, 10(3), may 2014.

[8] Christopher J Lowrance and Adrian P Lauf. An active and incremental learning framework for the online prediction of link quality in robot networks. *Engineering Applications of Artificial Intelligence*, 77:197–211, 2019.

[9] Chaitanya Manapragada, Geoffrey I. Webb, and Mahsa Salehi. Extremely fast decision tree. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, page 1953–1962, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Dana Marinca and Pascale Minet. On-line learning and prediction of link quality in wireless sensor networks. In *2014 IEEE Global Communications Conference*, pages 1245–1251, 2014.

[11] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdessalem. Scikit-multiflow: A multi-output streaming framework. *The Journal of Machine Learning Research*, 19(1):2915–2914, 2018.

[12] Ramya Panthangi M., Mate Boban, Chan Zhou, and Slawomir Stanczak. Online learning framework for v2v link quality prediction. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.

[13] John David Parsons and Prof J David Parsons. *The mobile radio propagation channel*, volume 2. Wiley New York, 2000.

[14] Charles E Perkins and Elizabeth M Royer. Ad-hoc on-demand distance vector routing. In *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100. IEEE, 1999.

[15] Andras Varga. Omnet++. In *Modeling and tools for network simulation*, pages 35–59. Springer, 2010.

DOI: 10.15439/2023F9832
ISSN 2300-5963 ACSIS, Vol. 35

A Survey on Congestion Control and Scheduling for Multipath TCP: Machine Learning vs Classical Approaches

Maisha Maliha
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
Email: maisha.maliha-1@ou.edu

Golnaz Habibi
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
Email: golnaz@ou.edu

Mohammed Atiquzzaman
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA
Email: atiq@ou.edu

***Abstract*—Multipath TCP (MPTCP) has been widely used as an efficient way for communication in many applications. Data centers, smartphones, and network operators use MPTCP to balance the traffic in a network efficiently. MPTCP is an extension of TCP (Transmission Control Protocol), which provides multiple paths, leading to higher throughput and low latency. Although MPTCP has shown better performance than TCP in many applications, it has its own challenges. The network can become congested due to heavy traffic in the multiple paths (subflows) if the subflow rates are not determined correctly. Moreover, communication latency can occur if the packets are not scheduled correctly between the subflows. This paper reviews techniques to solve the above-mentioned problems based on two main approaches; non data-driven (classical) and data-driven (Machine Learning) approaches. This paper compares these two approaches and highlights their strengths and weaknesses with a view to motivating future researchers in this exciting area of machine learning for communications. This paper also provides details on the simulation of MPTCP and its implementations in real environments.**

***Index Terms*—Multipath TCP, congestion control, scheduling, deep reinforcement learning, machine learning**

I. Introduction

COMMUNICATION is key in many domains, such as defense, hospitality, technology, and space. In telecommunications, packet switching is a method of grouping data in smaller packets for faster communication [1]. One of the earliest packet-switched networks started with the Advanced Research Projects Agency Network (ARPANET) [2] in the United States, which is also called the forerunner of the Internet. Today, the Internet has expanded and now consists of a set of protocols for global communications. Basically, a protocol is a set of rules that the sender and receiver must agree on to communicate with each other. The two well-known Transport layer protocols are User Datagram Protocol (UDP) [3] and Transmission Control Protocol (TCP). UDP does not need any handshaking, which means the receiver does not send any acknowledgment to the sender when it receives a message. UDP thus leads to faster communication.

TCP [4] provides more consistent communication by considering handshaking between the sender and receiver. With millions of devices connected to the Internet, there is a demand for faster communication, but TCP fails to meet that need. It's because of TCP's congestion control algorithm, which decreases the throughput (message delivery rate) in response to the loss of packets in the network. Also, the handshaking of TCP increases the time necessary for a packet to travel, resulting in higher latency. Keeping in mind these problems, Multipath TCP (MPTCP) has been introduced by the Internet Engineering Task Force (IETF) [5] to use multiple paths effectively and efficiently between the sender and receiver.

TCP connections can experience packet losses or connection drops, resulting in a poor user experience [6]. MPTCP can use multiple TCP connections, known as subflows, in parallel to overcome TCP's limitations. One of the main goals of MPTCP is to control congestion and maintain traffic flows. Another focus of MPTCP is scheduling the packets over different subflows to send packets with the smallest round-trip time (RTT) [7], which is the time taken to send a data packet to the destination and receive an acknowledgment from the receiver. There are a set of traditional techniques such as Dynamic-Window Coupling (DWC) [8], Opportunistic Linked Increases Algorithm (OLIA) [9], Balanced linked adaptation (BALIA) [10], and Adaptive and Efficient Packet Scheduler (AEPS) [11] that control congestion or simultaneously schedule packets over multiple paths in MPTCP.

Since the standardization of MPTCP, a lot of classical approaches have been proposed to improve the performance of the network in terms of throughput and latency, but most of them perform poorly in highly dynamic networks. Recently, *data-driven* approaches, which are mostly based on Deep Reinforcement Learning, perform much better in dynamic networks because of their ability to learn the network conditions. To the best of our knowledge, there have been only a few proposals on controlling congestion while scheduling packets using deep reinforcement learning-based approaches. Although some methods have been proposed to control congestion, those works have not focused on reducing RTT. Some researchers have proposed schedulers to reduce latency, but they did not consider achieving high throughput.

One of the biggest benefits of using MPTCP is its capacity to use all the available subflows and boost the network's throughput. However, to achieve high goodput, a scheduling strategy is important. Scheduling in MPTCP distributes packets over different subflows based on the smallest RTT. Schedulers using classical approaches, like [12]–[14], increased the throughput but failed to adapt to the dynamic nature of a real-world network. They were tested using simulation, which, of course, does not emulate the real-world scenario. Machine Learning and Reinforcement Learning models have improved the above drawbacks as they can learn from past experience and they are more robust to the dynamic nature of real-world networks; However, machine learning techniques are usually slower than classical approaches and need a huge dataset to train the models [15]–[17].

A. Contributions

The *objective* of this paper is to provide a brief overview of the existing work on MPTCP, including both classical and machine learning-based approaches. We discuss how previous researchers have addressed MPTCP challenges and summarized their solutions. Previous works have reviewed the existing congestion control and scheduling techniques for MPTCP [7], [18]–[21]. Among those works, some review papers have focused on either congestion control of MPTCP or only the scheduling of MPTCP, while others have reviewed only the existing work on MPTCP establishment. There are also some works that have mentioned both congestion control and scheduling but have not focused much on the scheduling-based works of MPTCP. The *contributions* of this paper are as follows:

- Discuss the difference between the traditional TCP and MPTCP communication protocols.
- Discuss in detail both the congestion control and the packet scheduling problems separately.
- Comparison between the performance of ML-based and classical algorithms in terms of controlling congestion and packet scheduling.
- Summarize basic concepts in MPTCP, including the establishment of MPTCP in real-world platforms and simulators.
- Highlights advantages and limitations of previous works that can help the readers investigate future improvements on MPTCP.

The rest of our paper is organized as follows; Section II describes the terminologies in communication and deep reinforcement learning. Section III compares TCP and MPTCP. Sections IV and V describe previous works on congestion control and scheduling of MPTCP, respectively. Section VI focuses on the performance of both congestion control and packet scheduling. MPTCP implementations in the kernel and NS-3 are described in Section VII. Lastly, in Section VIII, we conclude our survey by discussing future works in MPTCP congestion control and scheduling.

II. Background & Terminologies

A. Overview of TCP

TCP is a connection-oriented communication standard that computer applications use to communicate over a network. It is a packet transfer protocol in the Transport Layer [22] of the TCP model. TCP uses only one dedicated path for packet transfer. Though TCP guarantees data integrity of the packets, it has to face packet loss, delay and other problems which are discussed in Section III. Also, network congestion is another major problem in TCP which is discussed briefly later. Section II-B summarizes some concepts in TCP which would be in common with MPTCP and they are also used in congestion control.

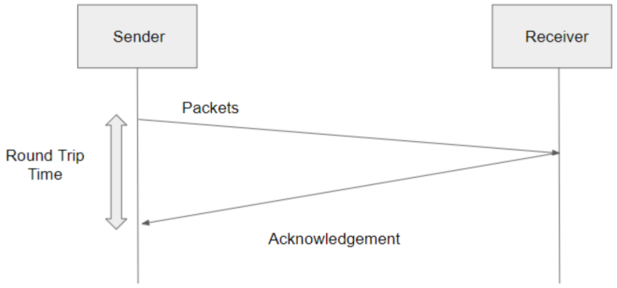


Fig. 1. Illustration of the RTT of a packet from the sender to the receiver.

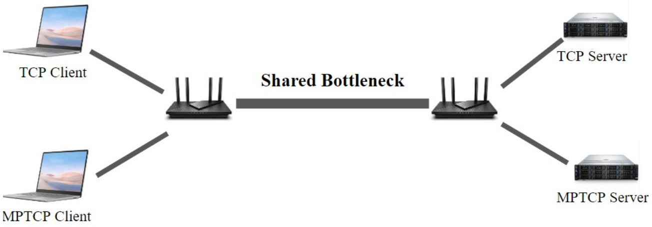


Fig. 2. Illustration of shared bottleneck scenario in a network [23].

B. Some Concepts in TCP

- **Round Trip Time (RTT):** The time required to send a packet from the client to the server, and the time it takes for the server to receive an acknowledgment about receiving the packet is known as the round trip time (RTT). Reducing the round trip time is a primary focus of MPTCP. Figure 1 illustrates the meaning of RTT.
- **Throughput vs Goodput:** Throughput refers to the total number of packets transferred to the destination within a fixed time frame. On the other hand, Goodput is the number of meaningful packets that are delivered to the destination within a given time frame.
- **Low Latency vs High Latency:** Latency refers to the amount of time required to send a packet from source to destination and back again. Low latency is always preferable.

- **Congestion Window (CWND):** The congestion window decides the number of bytes or how many packets will be sent at a given time. Depending on the larger congestion window size, the throughput will also be maximized. Congestion window size has been determined by the slow start and congestion avoidance phases of TCP which will be discussed in the next part.
- **Bottleneck vs Shared Bottleneck:** Bottleneck occurs when there is not enough network capacity in a connection to handle the current volume of traffic. On the other hand, when a bottleneck link is shared between multiple subflows, it is referred as a shared bottleneck which is useful for maximizing the throughput. Figure 2 illustrates the concept of a shared bottleneck scenario.

C. Congestion Control in TCP

TCP's congestion control mechanism has three phases; (1) slow start phase; (2) congestion avoidance phase; and (3) congestion detection phase. The basic difference between these three phases is the rate of increase in the congestion window size.

- **Slow Start Phase:** Slow start phase works as a part of the congestion control algorithm in TCP by controlling the amount of data flow in a network. When a network becomes congested from excessive data in the network, the slow start phase chokes the traffic by limiting the congestion window size. In the slow start phase, the sender sends a packet that contains its initial congestion window, and the client responds with its maximum buffer size after receiving the packet. If the sender gets the acknowledgment from the receiver, the number of packets to be sent to the receiver is doubled. This procedure continues until no acknowledgment is received. The acknowledgment may not be received for two reasons: if congestion occurs or the window limit of the client is reached.
- **Congestion Avoidance Phase- Additive Increase:** The congestion avoidance phase starts when the congestion window size of the TCP reaches a threshold in the slow start phase. In this phase, the size of the congestion window increases linearly. To elaborate, assume the congestion window size at time t is 20 and all the packets have been transmitted successfully, then the congestion window size at time $t+1$ will be 21.
- **Congestion Detection Phase- Multiplicative Decrease:** If congestion occurs in the slow start phase or congestion avoidance phase, the congestion window size is decreased. This is called the multiplicative decrease phase, where TCP follows an exponential reduction of the congestion window. The additive increase and multiplicative decrease phases of the congestion avoidance and detection phases are referred to as Additive Increase Multiplicative Decrease (AIMD). An example of AIMD is shown in Figure 3.

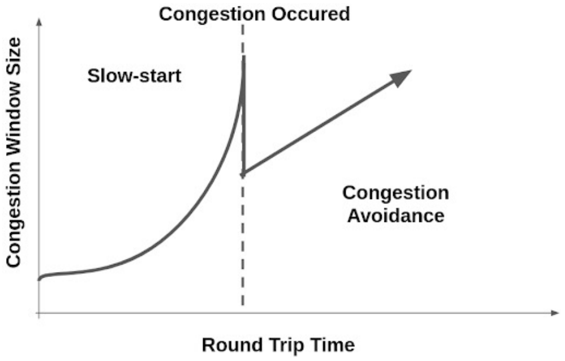


Fig. 3. Change in the congestion window in AIMD algorithm when a packet loss is encountered.

D. Overview of MPTCP

As opposed to TCP, which solely considers one path to transfer data, MPTCP is a transport layer protocol that allows the transfer of packets along multiple paths between the sender and the receiver. This helps the network increase its load capacity, thereby transferring a larger number of packets compared to TCP. The paths in MPTCP are called subflows. When one or more of the subflows fails to send a packet, it can flow through other subflows, leading to a fault-tolerant network. MPTCP is used in several areas of communication where there is a need for high throughput and very low latency during packet transfer. MPTCP is used in different applications such as online streaming, networking, gaming industries, VPNs. Figure 4 depicts the use of MPTCP where a cellphone may use either one of the subflows from two subflows to get connected to the server: one is the Wifi, and another subflow is the 5G network. The following section explains the procedure for establishing the MPTCP communication.

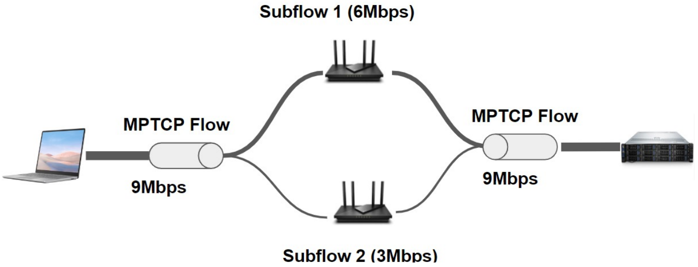


Fig. 4. Overview of MPTCP

.

1) Establishment of MPTCP Connection: The establishment of MPTCP between a sender and the receiver has two stages; In the first step a single flow is established. This phase is similar to TCP. Then, subsequent subflows are created. In the first stage, the sender and the receiver use one subflow to set up the MPTCP connection between them by sharing randomly generated keys. This lays the foundation for creating further paths between the sender and the receiver. Figure 5 shows the establishment of an MPTCP connection using all subflows.

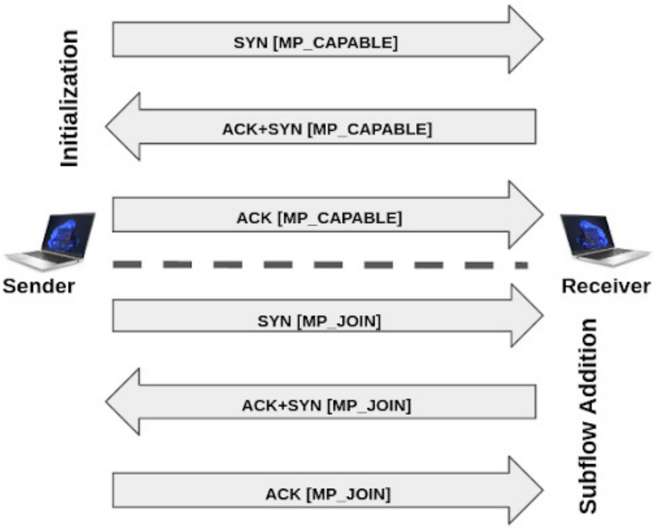


Fig. 5. Establishment of an MPTCP connection.

In MPTCP, after the establishment of the initial handshake as in TCP, the subsequent subflows are also handshaked [24]. MPTCP follows three-way handshaking consisting of SYN (synchronize), ACK+SYN and ACK (acknowledge). In the SYN packet, the sender shares its own token and a random nonce (number). Here the token is the hash value of the key using some cryptographic function which can be calculated in the initial phase with the keys exchanged in that phase. Subsequently, in the SYN+ACK packet, the receiver creates an HMAC (Hash-based message authentication code), the receiver's token and its nonce.

The option MP_CAPABLE is used to check whether the remote host is MPTCP enabled in the initial subflow. The option MP_JOIN is used in the additional subflow establishment to associate with MPTCP connections. Lastly, the sender responds with its HMAC in the ACK packet. [24]

E. Overview of Machine Learning Concepts used in Congestion Control in MPTCP

1) RNN and LSTM: RNN (Recurrent Neural Network) is one kind of neural network that passes the output from the previous steps to the next steps. RNN consists of the input layer, hidden layers, and the output layer. Its hidden state remembers the previous information to predict the next output. RNN works well in terms of correlation, while LSTM (Long Short-Term Memory) not only connects the correlation but also focuses on the context of the information. LSTM is an artificial neural network that follows feedback connections and stores previous information to predict the next one. It is a modified version of the RNN that solves the vanishing gradient problem [25] and can easily process longer sequences. LSTM has an input gate, an output gate and a forget gate. The Input Gate takes an input and vectorizes the input value. The forget Gate is responsible for forgetting unnecessary information, while the output gate generates the output. This helps the framework keep the necessary information and forget the unnecessary ones. Figure 6 shows the different components of an LSTM and compares with RNN. The LSTM-based framework is very popular to create Deep Reinforcement Learning-based congestion control system for MPTCP [26], [27].

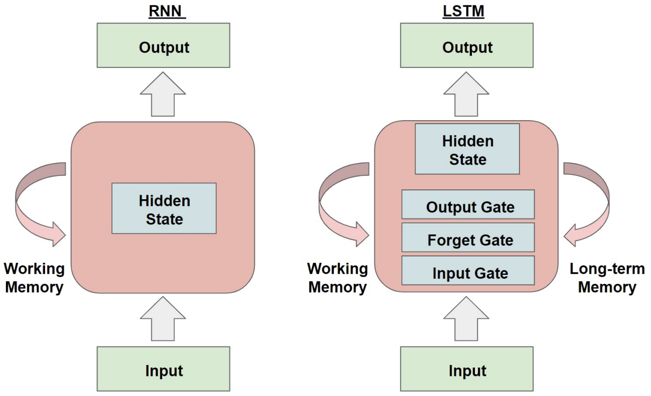


Fig. 6. Architecture of RNN (left) vs LSTM (right).

2) Reinforcement Learning and Deep Q-Learning (DQN): A Markov model [28] consists of a tuple $(s, a, r(s, a))$, where s is the current state vector, a is the vector of actions an agent takes, and R is the reward or the feedback the environment provides for the agent, given the current state and action. In a Markov Decision Process, the agent makes a set of actions, called policy, to maximize its expected reward. The function $Q(s, a)$ defines the optimal (*i.e.*, maximum) expected reward the agent can get, given the current state s and action a. Evaluating Q is important to plan for the optimal policy. Due to the stochastic nature of many environments and agents, calculating the exact value of Q is usually not possible. Instead, the agent learns Q values from its experience; this is called Q-learning, a type of reinforcement learning.

DQN Reinforcement Learning (DRL) uses a deep neural network (DNN) to learn/estimate the Q function. The deep neural network could be RNN, LSTM, CNN etc. In contrast, a traditional Q estimation (e.g., Temporal Difference (TD) learning [29]), is usually a memory table that stores all the previous records of the different steps that have been taken, along with their rewards which is not scalable for an environment with large size of space and actions, or when the action/state space is continuous. Deep-RL has been used in communication for learning the communication strategy between multiple agents. DRL is also used in MPTCP implementation to control congestion and schedule packets to maximize throughput and get low latency. In MPTCP communication network, the state can be throughput, sending rate, RTT, and loss of packets; actions should be increasing/decreasing congestion window size and reward is the measurement of either good or bad performance of the network, based on the state and action.

3) Actor-critic model: An actor-critic has two major components: an actor and a critic. An actor takes the state of the current environment and determines the best action that needs to be taken, depending on that state. Whereas the critic works for the evaluation role by taking the environment state with actions and returning a score/reward (*e.g.,* Q) to decide how good is that action for that state. Determining the best action depends on the Q score, which is calculated separately in the critic network. An actor-critic learning technique learns both a policy function and a value function at the same time. The value function aids in enhancing the value function's training procedures, while the policy function instructs on how to make decisions.

4) Transformers and Self-Attention: The transformer is an encoder-decoder model and has been used in many applications [30]. Self attention follows the encoder part of transformers and can be used in many communication applications such as congestion control [27], or scheduling packets for MPTCP. In natural language processing or machine translation, self-attention for a particular word measures the dependency of that word on the other words; more relevant words have a higher value in self-attention, and loosely connected words have lower values [31], [32]. This concept is brought into the area of communication and subflows. Here the words have been replaced by the subflows. In MPTCP, self-attention checks the dependency of a subflow with the other subflows by assigning different weights to the states of the other subflows. Here the states of the subflows may refer as RTT, packet loss, packet delay, and throughput [27].

III. Traditional TCP vs Multipath TCP

In this section, we discuss some major challenges in network communication, how MPTCP and TCP address them in different situations, including the comparison of MPTCP and TCP performance.

A. Packet Loss

Packet loss happens in a network when a packet fails to reach the receiver. Packet loss in a network indicates network congestion, disruption, or even a complete loss of connection. When a TCP connection encounters a packet loss, it considers a sign of network congestion and, therefore, halves the size of the congestion window and the threshold value; hence the throughput decreases. However, MPTCP is more robust in such cases; whenever it sees one of the subflows having packet loss, it reduces the congestion window of that subflow, while the other subflows remain intact. This was experimentally tested [33] where TCP and MPTCP transmission performances were compared using WiFi and LTE. In this real-world experiment, 750 individuals from 16 nations utilized a crowd-sourced smartphone application for 180 days. Deng et al. [33] compared MPTCP and TCP performance via WiFi or LTE from 20 different places across seven cities in the United States. The fastest link of normal TCP exceeds MPTCP performance for short flows. In this scenario, MPTCP fails to select an appropriate communication path to reduce transmission time for a small amount of data; hence, the packet loss increases for MPTCP. However, TCP experiences greater packet loss than MPTCP in log-distance communication experiments.

B. Packet Delay

Packet delay in a network is the time taken to transfer a packet of data from sender to receiver. The packet delay is highly affected by the number of routers or switches in the route, the nature of the path the packet follows, and the congestion in that path. The vulnerability of packet delay in both TCP and MPTCP scenarios can be described by one communication example. If a computer is connected via both wireless and wired connections, it may communicate with servers using TCP or MPTCP. In terms of TCP, if the connection is established over the wireless connection, it will have a greater overall delay than MPTCP. Wireless connection experiences high packet loss as opposed to the wired connection which leads to higher latency [34].

MPTCP has the choice to choose either a wired or wireless connection, it may transfer the packets via the wired connection if it experiences major packet delay through a wireless connection. Raiciu et al. [35] have compared the performance of MPTCP and TCP-based on the packet delays in different network topologies. Their findings reveal that MPTCP can achieve 90 percent bandwidth utilization and low overall packet delay when the number of subflows is two and eight in the VL2 [36] and FatTree network [37] topologies, respectively. The result also showed that MPTCP not only increases the bandwidth but also increases the robustness to network changes by lowering the packet delay.

C. Out-of-Order Packets

In TCP, a message is divided into multiple parts, known as packets. Each of these packets is given a unique number known as the sequence number. When these packets reach the receiver, it uses the sequence number to put them in order to retrieve the message. If these orders are not maintained during transmission, the delivery is called out-of-order delivery of packets. UDP is notorious for these deliveries as it does not consider any handshaking when each packet is received. But in TCP, the next packet is not sent until it gets the acknowledgment from the previous packet. If it gets a time out or negative acknowledgement (NACK) [38], the packets are sent again; therefore, out-of-order delivery is rare in TCP. However, in MPTCP, there is a high chance of out-of-order delivery of packets as they use different subflows with different delays. Therefore, scheduling is one of the most challenging tasks in MPTCP, and a lot of work has been done to address this issue. Yang et al. [39] have mentioned a situation where the jitter can happen for transferring data in MPTCP. They tackled this issue using an innovative traditional scheduling process named Delay Aware Packet Scheduling technique to remove the jitter in the packets. Han et al. [17] used a queue to keep redundant packets that may get lost.

D. Round Trip Time

Round Trip Time (RTT) has been discussed in section II. Chen et al. [40] have compared the performances of TCP and MPTCP over WiFi and cellular networks where the authors compared the RTTs of the transmission protocols. They conducted two sets of experiments; in the first experiment, they used small-sized files, and in the second, large-sized files were used. In the first experiment, the file size varied from 8KB to 32MB. When WiFi is the default route, there is no discernible gain in MPTCP download performance over TCP. For small file downloads (such as 64 KB), the single route via WiFi delivers the optimum speed. However, a single LTE (Long-Term Evolution) [1] channel becomes the optimum option for relatively longer traffic flows. MPTCP outperforms TCP for larger files.

IV. Congestion Control of MPTCP

Congestion control is a concept of controlling congestion in a network and could happen in both TCP and MPTCP. Congestion occurs when there is too much data that needs to be sent through a network. Congestion control regulates the flow of data packets into the network, allowing for efficient use of a shared network infrastructure and preventing congestion collapse. In TCP, where there is only one subflow, the network is easily congested.

Different types of algorithms have been proposed to improve the congestion in TCP networks, such as TCP Cubic [41], TCP Vegas [42], and TCP Reno [42]. MPTCP provides several subflows, which results in a reduction of congestion. MPTCP has been designed to address the congestion issue, while still having the traffic flowing like a single-path TCP. A naive implementation of CC in a multipath setting would be using regular TCP congestion control for each subflow; however, it is not efficient as MPTCP which uses multiple concurrent TCP connections. Having a congestion control that manages the packet flows on subflows concurrently seems more efficient. For this purpose, many methods have been proposed to improve congestion in MPTCP. Congestion control algorithms for MPTCP are classified as classical and machine learning approaches which will be discussed in the following subsections.

A. Classical Congestion Control Approaches

Most of the existing congestion control algorithms in MPTCP setting focus on the Congestion-Avoidance (CA) phase that solely considers long flow transmissions and does not focus much on the slow start phase. The congestion avoidance phase prevents a network from being overflooded by data such that it discards packets with low priority to be delivered, and the rate of transmission rises linearly over time. Another approach is to focus on Slow Start. The Slow Start phase limits the quantity of data to be sent over a network to avoid congestion. However, it causes exponential growth of the congestion window in the uncoupled Slow-Start (SS) phase, leading to buffer overflow from burst data. In terms of solving the mentioned problems, Yang et al. [43] have proposed a Throughput Consistency Congestion Control (TCCC) algorithm which consists of both Coupled Slow-Start (CSS) and Aggressive Congestion Avoidance (ACA). The usage of CSS prevents packet loss brought on by large data bursts and ACA works on getting fair bandwidth which is shared in congestion avoidance. Their proposed framework enhances transmission efficiency. However, the CSS algorithm only plays a part in the initial slow start phase of MPTCP. As the subflows of MPTCP belong to different phases in congestion control (see Section II-C), the CSS algorithm needs much extra consideration, which makes congestion control more challenging in MPTCP [43].

Traditional AIMD used in TCP shows poor performance adaptation in terms of network state-changing situations in MPTCP. Gilad et al. [44], have presented a method named MPCC that uses online learning. Their implementation has been performed in Linux kernel and the method has been tested on different network conditions and many different network topologies. In terms of improving the implementation, their analysis needs to be reached beyond parallel link networks. As further research, they have mentioned about boosting the performance for short flows and solving the bandwidth mismatch problems on network paths.

An energy-aware based congestion control algorithm (ecMTCP) has been developed by Le et al. [12] where the method distributes traffic between the most crowded and least crowded paths, as well as across paths with different energy costs, to achieve load balancing and energy savings. For simulation purposes, they used NS-2 simulator [45], and their design mechanisms can work on getting higher throughput in terms of both TCP and MPTCP flows. The main goal was to shift the traffic to less energy-intensive and less crowded paths. Cao et al. [13] proposed weighted Vegas (wVegas), a delay-based congestion control scheme for MPTCP. This algorithm has detected the packet queuing delay of each path and ultimately has decreased the packet load of congested subflows by increasing the load of the less congested one. This framework performed traffic shifting which can cause less packet losses and provide better traffic balance in subflows. Cao et al. used NS-3 simulator [46] to conduct the simulation and build a Network Utility Maximization Model by proposing an approximate iterative algorithm to reach their aim of controlling congestion.

Ji et. al [14] mentioned that existing multipath congestion control algorithms are unable to quickly adjust to dynamic traffic due to the heterogeneous Quality of Service (QoS). QoS refers the technologies that work on a network to manage traffic and enhance performance by reducing packet loss, delay, and latency in a network. It may lead to poor performance in certain network environments. To mitigate these issues, firstly, the authors have noticed the performance constraints of the most recent multipath congestion control algorithms through vast experimentation. Then, they used a unique control policy optimization phase referred to an adaptive QoS-aware multipath congestion management system that can quickly adapt to network changes. Their method uses the Random Forest Regressing (RFR) [47] method to carry out QoS-specific utility function optimization to adapt and encourage the improvement of the selected performance metric. They

[1] wireless broadband communication standard came before 4G network

conducted the implementation in Linux kernel and showed their work outperformed most of the multipath congestion control methods such as wVegas [13], Opportunistic Linked Increases Algorithm (OLIA) [9], Balanced Linked Adaptation (BALIA) [10].

Singh et al. [48] improved the Opportunistic Linked Increases Algorithm [9] and Dynamic-Window Coupling (DWC) [8]. The authors provided a mechanism to reduce the overall packet reordering delay and focused on the buffer size in the receiver side. Their proposed work showed good performance in terms of various bottleneck scenarios.

Hassayoun et al. [8] proposed a multipath congestion control scheme called Dynamic-Window Coupling (DWC) to obtain higher throughput to each end-to-end multiple paths. The authors also detected shared bottlenecks by monitoring loss and delay signals, and then grouped their congestion control mechanism over all subflows that shared a common bottleneck. Detecting the bottleneck for network conditions and regrouping the subflows in terms of the same bottlenecks leads to higher throughput. They also introduced subflow sets, a concept for enabling subflows to smoothly switch between independent and shared bottleneck-based congestion control. The algorithm has been implemented in NS-2 simulator. As future research, the authors introduced the possibility of including ”memory” into the detection method for detecting the previous subflow groupings.

Ferlin et al. [49] have mentioned that increasing the bandwidth of multiple links and getting higher throughput will be impossible if two or more paths do not share a bottleneck. They found out the nonshared bottleneck paths of the coupled congestion control for links, they referred to it as a penalty, and to overcome it, they implemented shared bottleneck detection (SBD) algorithm for MPTCP. This work can balance congestion and throughput. Their observation has shown that in the case of non-shared bottleneck scenario, the maximum throughput can be achieved up to 40% with two subflows. Also, the throughput gain increased by above 100% when the number of subflows increased to five. Their implementation has been performed in Linux kernel and for emulation purposes, they have also used CORE network emulator [50].

B. Machine Learning Approaches for Congestion Control in MPTCP

Though lots of work have been done in classical-based approaches for congestion control of MPTCP. But for controlling congestion, classical-based approaches focus solely on different types of congestion indicators (*i.e.*, packet loss or RTT). In the case of classical-based approaches, the decision-making process totally depends on these unpredictable factors, which leads to poor performance. Whereas ML-based approaches aim to provide decisions based on experience, and can adapt well to any network situation. Thus, ML-based approaches outperform classical-based methods [51].

In reality, networks are dynamic, and the state of the network changes frequently. Due to that, MPTCP performs poorly in many practical situations as MPTCP has to adapt in new network states. Zhuang et al. [52], introduced a Reinforcement Learning technology that can learn the best route to send TCP packets such that the throughput has been maximized. They proposed a simple algorithm for controlling multipath congestion, where congestion control has been approached as a multi-armed bandit [53] issue based on online learning (MP-OL), which allowed flexible and adaptive transmission rate adjustments for each subflow with good performance.

In [26], the authors proposed a Deep Reinforcement Learning (DRL)-based framework to control congestion where a single DRL agent has been utilized to perform congestion control for all MPTCP flows to maximize the total utility. Figure 7 illustrates the concept of DRL for MPTCP congestion control. They implemented the MPTCP in the Linux kernel and used an LSTM-based neural network under a DRL framework to develop a representation for all active flows. Their work was the first work where the authors incorporated the LSTM-based representation network into an actor-critic architecture for controlling congestion which used the deterministic policy gradient [54] to train the critic, actor, and LSTM networks.

He et al. [27] worked on increasing/decreasing the sending rates of packets in response to congestion, where each DRL agent can control the congestion window size of each subflow. Their proposed DRL-based MPTCP framework also included self-attention, which has been used to check the dependencies of one subflow with the weighted sum of other subflows. They compared their work with DRL-CC [26] and showed their method outperforms DRL-CC.

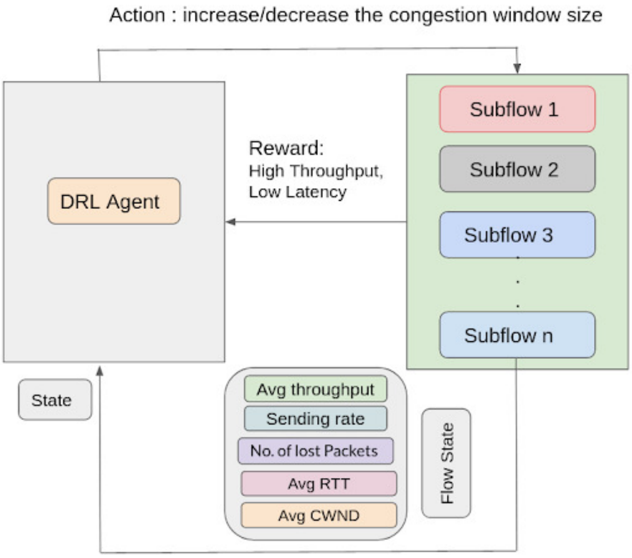


Fig. 7. Subflows are controlled by a DRL agent.

Li et al. [55] proposed a method called SmartCC which can learn a set of congestion rules for observing the environments and taking actions to adjust the congestion window size of each subflow. For the MPTCP implementation task, the authors used the NS-3 simulator.

The Internet of Deep Space Things, or IoDST, offered communication services for mission spacecraft that send video data. To improve TCP throughput and stream playback, Ha et al. [15] designed a congestion control framework for MPTCP, which can be used for data streaming transmission. Their proposed Q-learning and Deep Q-Network (DQN)-based congestion control scheme calculated the ideal congestion window for data transfer in IoDST conversations.

Xu et al. [56] proposed the SGIN-based High-Speed Railway (HSR) scenario with MPTCP. Space-ground integrated networks (SGINs) has been referred to as promising network architecture that provides seamless, high-rate, and reliable data transmission with incredibly wide coverage. By utilizing MPTCP in the SGIN, simultaneous data transfer over terrestrial and satellite networks has been made possible. However, due to MPTCP's current congestion control (CC) mechanisms, it's difficult to know the difference between negative effects (like packet loss and/or increased round-trip time) brought on by congestion and those brought on by handovers. This may lead to severe performance degradation in the SGIN-based HSR scenario, where handover may occur frequently. To solve it, a DRL-based novel approach has been proposed to improve the goodput which outperformed other state-of-the-art algorithms.

Xu et al. [57] presented a DRL-based novel framework for traffic engineering that can make decisions under the guidance of actor-critic networks. In their work, the state consisted of two components, such as the throughput and the delay of each communication session. On the other hand, the action has been defined as the solution to Traffic Engineering (TE) problems. The authors used the NS-3 simulator, and the reward of the model was the sum of the output from the utility function for an entire communication session. The utility function was a function of the throughput and delay of the network, which depicted how the network can perform. In the paper, each session had 20 iterations. In each iteration, the agent sent its actions to the environment and recorded the value from the utility function before updating the reward value. While they considered only one DRL agent (*i.e.*, decision maker) in their framework, adding multiple agents can be considered to further improve the performance.

Pokhrel et al. [58] introduced a transfer learning-based MPTCP framework for Industrial IoT, where the neighboring machines can collaborate to learn from each other. In their approach, when a new DRL system controlling the IoT network joins the environment, it can use the idea of transfer learning. [2] NS-3 was used to simulate the algorithm. Their model has been proven theoretically and needs further research to determine its performance in a real-world situation.

V. Scheduling of MPTCP

Scheduling of MPTCP decides the amount of data that needs to be scheduled to different subflows based on getting the higher performance (high throughput, low latency, less packet loss) in MPTCP. In this section, different classical and ML-based approaches have been discussed which can be used to schedule packets in MPTCP.

A. Classical Approaches for Scheduling in MPTCP

Hwang et al. [59] dealt with the problem of scheduling small-length packets. However, the authors mentioned MPTCP is usually advantageous for long-lived flows, and it performs worse than single-path TCP when the flow size is tiny (*e.g.*, hundreds of KiloBytes). In this scenario, the quickest method is preferable since latency is far more critical than network bandwidth with such tiny data deliveries. The regular MPTCP packet scheduler may pick a slow path if the fast path's congestion window is unavailable, resulting in a delayed flow completion time.

To address this issue, Hwang et al. [59] suggested a novel MPTCP packet scheduler that momentarily blocks the slow path when the latency difference between the slow and fast paths is considerable, allowing the tiny quantity of data to be delivered swiftly via the fast path. The authors used the method to find the subflow with the lowest RTT regardless of the availability of the congestion window, and then they used the existing Lowest-RTT-First policy [60] to choose the optimal subflow. They then returned the best one if the difference between the best subflow RTT and the minimum RTT is less than a certain threshold. They picked 100ms for threshold delay when testing 3G and WiFi networks in this paper.

Chaturvedi et al. [11] analyzed different existing schedulers and identified some current outstanding concerns, such as head-of-line (HoL) blocking and out-of-order packet delivery. HoL blocking may occur when a single data packets queue may wait to be transmitted and the packet at the head of the line may not be able to move ahead due to congestion [61]. These problems reduce MPTCP performance and to mitigate the issues, the authors have presented an adaptive and efficient packet scheduler (AEPS). This novel MPTCP packet scheduler not only addresses these concerns but also offers high throughput with a short completion time by using the capacity of all available pathways. AEPS can send data packets to the receiver in the order they were received, and its performance is unaffected by the size of the receiver buffer, or the size of the data being transmitted. The AEPS has been developed with three objectives: (1) packets should arrive to the receiver buffer; (2) all pathways' bandwidth should be used; and (3) completion time should be as short as possible. According to the authors, the first condition assisted AEPS in resolving the HoL blocking and received window-limiting issues by sending packets to the receiver buffer in sequence. The second condition summed the bandwidth of each interface (path) by using all accessible pathways to the MPTCP source, which also helped to enhance throughput. The third criterion aided in choosing the routing for each packet so that the total network completion time can be minimized.

Dong et al. [62] thoroughly compared existing scheduling algorithms and guided the development of new scheduling algorithms in 5G. The authors examined the influence of several

[2] Transfer learning uses a previously trained model as the foundation for a new model on a different task.

network parameters, such as RTT, buffer size, and file size, on the performance of current extensively used scheduling algorithms over a wide range of network circumstances. The paper compares the Lowest-RTT-First [60], Delay-aware packet scheduler (DAPS) [63], Out-of-order transmission for in-order arrival scheduler (OTIAS) [64] and Blocking estimation-based MPTCP scheduler (BLEST) scheduling [65] algorithms. The number of timeouts and flow completion time are compared in the path heterogeneity test. The results showed Lowest-RTT-First has the most timeouts, while the BLEST has the fewest. BLEST surpasses other algorithms in varying buffer size outcomes, followed by OTIAS and DAPS. Since BLEST can dynamically predict whether head-of-line blocking will occur and hence minimizes the quantity of out-of-order packets. In the different file size tests, BLEST and LowRTT perform better than DAPS, and OTIAS outperforms BLEST.

Le et al. [66] tackled the problem of out-of-order delivery in MPTCP. Because of the diverse nature of latency and bandwidth on each channel, the out-of-order packet issue becomes severe for MPTCP. To solve this issue, the authors presented the forward-delay-based packet scheduling (FDPS) method for MPTCP. The technique is divided into two parts: predicting the forward delay differences across pathways and picking data to send through a path when the congestion window is available.

B. Machine Learning Approaches for scheduling in MPTCP

In recent times, many ML-based approaches have been proposed to improve the scheduling mechanism of MPTCP. Though classical approaches achieve good performance in terms of scheduling in MPTCP, ML-based approaches also show promising result and becomes popular in terms of getting higher throughput with lower latency than non-ML methods.

Wu et al. [16] applied a learning-based technique to schedule packets in the different paths of an MPTCP. The authors have presented FALCON, a learning-based multipath scheduler that can adapt to changing network circumstances quickly and correctly using meta-learning. The meta-learning algorithm comprises two parts: offline training and online training parts. The online learning module captures the network-changing conditions whereas the offline learning module takes the experience(data) from the online module and divides the experience into different groups depending on the network conditions.

Han et al. [17] used the technique of redundancy of packets to reduce packet loss by suggesting EdAR (Experience-driven Adaptive Redundant packet scheduler). In the face of dramatic network environment changes, EdAR enables dynamically scheduling redundant packets using an experience-driven learning-based strategy for multipath performance enhancement. To allow accurate learning and prediction, a Deep Reinforcement Learning (DRL) agent-based framework has been created that learns both the network environment and the optimal course of action. EdAR has two transmission modes: standard transmission and redundant transmission. Standard transmission follows the regular data transition. Regarding the redundancy transmission, there is a buffer called redundant buffer. The redundant buffer holds packets that have already been transmitted but have yet to be acknowledged. If a new packet is transmitted from the send buffer on a subflow, it is copied to the redundant buffer. If a packet in the redundant buffer is not sent out or acknowledged, it is deleted from the redundant buffer. Silva et al. [67], used linear regression [68] to predict throughput and latency in MPTCP subflows, and proposed Artificial Neural Network [69]-based linear classifier to choose the best subflow which can provide better performance in MPTCP scheduler. They implemented their work in NS-3 simulator.

VI. Congestion Control and Scheduling of MPTCP

Few works have been done focusing on congestion control and scheduling the packets of MPTCP at the same time. Those works get higher throughput and lower latency in terms of performance evaluation. Though further research regarding congestion control with packet scheduling is needed to be done. This section reviews classical and machine learning approaches that have been done in this domain.

A. Classical Approaches on both Congestion Control and Scheduling of MPTCP

Wei et al. [23] proposed a model that gets higher throughput when the networks do not go through a shared bottleneck. Their work had two outcomes: (1) When no congestion would occur, their method has been able to get higher throughput than a single TCP. (2) When there is congestion in the network, their method has at least the same throughput as TCP. Their method also measured how severe or minor the congestion is in the network. They have introduced both SB-CC (Shared Bottleneck-based Congestion Control Scheme) and SB-FPS (Shared Bottleneck-based Forward Prediction packet Scheduling scheme), where SB-CC can detect shared bottlenecks and estimate the congestion degree of all subflows. SB-FPS can perfectly schedule data in shared bottleneck and can also distribute data according to the congestion window size of each subflows. For implementing MPTCP, they used the Linux kernel and achieved higher throughput.

B. Machine Learning Approaches on both Congestion Control and Scheduling of MPTCP

Pokhrel et al. [71] have introduced the Deep Q learning (DQL)-based method to control congestion and schedule packets for MPTCP. Their proposed DQL framework has utilized the LSTM-based recurrent neural network where in their framework the Q function provided the logarithm value of goodput for the previous iteration. Here, the policy function was the actor-critic of two LSTMs and the value function was the reward. They considered RTT, throughput, and sending rate as the state. Depending on the state, their model provided action on whether window size needed to be increased or decreased and what changes can be taken in the schedule of packets for the subflows. In their work, the reward was the

TABLE I
Performance measurement of reviewed papers for Congestion Control and Packet Scheduling in MPTCP.

Approaches	**Paper**	***Feature***	***Strength***	***Limitations***	***Implementation***
ML	DeepCC [27]	Congestion Control	increasing/decreasing the sending rates of packets	increased computational time	Linux Kernel
	DRL-CC [26]	Congestion Control	high throughput	complexity by large state space	Linux Kernel
	SmartCC [55]	Congestion Control	dealt with multiple communication path in heterogeneous networks	did not consider TCP-friendliness issues	NS-3
	IoDST [15]	Congestion Control	calculated the ideal congestion window	did not use real experiments or emulated tests	computer simulations
	DRL for Handover-Aware MPTCP CC [56]	Congestion Control	higher goodput than other state-of-the-art algorithms	training time becomes longer	Linux Kernel
	MPTCP Meets Transfer Learning [58]	Congestion Control & Packet Scheduling	improved the efficiency of newly deployed machines	not proven in practical situations	NS-3
	FALCON [16]	Packet Scheduling	can adapt in network changing conditions	needs to understand the learning outcome	Multipath QUIC [70]
	EDAR [17]	Packet Scheduling	enabled dynamically scheduling redundant packets	increased computational time	NS-3
Classical	TCCC [43]	Congestion Control	improved efficiency in nonshared bottleneck scenario	needs extra consideration for different phases of MPTCP	NS-3 and Linux kernel
	MPCC [44]	Congestion Control	tested on different network conditions	bandwidth mismatch problems on network paths	Linux kernel
	ACCeSS [14]	Congestion Control	quickly adjust to dynamic traffic	performance can be improved	Linux kernel
	CC MPTCP with shared bottleneck detection [49]	Congestion Control	balancing congestion with improving the throughput	needs to improve it's robustness	Linux kernel and CORE
	Packet scheduling for multipath TCP [59]	Packet Scheduling	decreased the completion time of short flows	needs to improve overall transmission rate	Linux kernel
	AEPS [11]	Packet Scheduling	high throughput and low completion time	performed poorly in heterogeneous networks	Linux kernel and NS-3
	wVegas [13]	Congestion Control	less packet losses	needs to effectively handle multiple extended high-speed paths	NS-3
	DWC for MPTCP-CC [8]	Congestion Control	high throughput	did not mention their model performed better than others	NS-2

summation of all the Q functions for all subflows. Similar to other RL algorithms, the optimal decision was learned to maximize the reward. They have made their MPTCP implementation in the Linux kernel and achieved low delays with maximum goodput.

VII. Implementation of MPTCP

In this section, we describe different ways of implementing MPTCP either in real hardware (kernel) or in simulator and list some of the works for each type of implementation as the reader's reference. Previously, NS-2 simulator has been used for implementing MPTCP [8], [72]. Now, most of the recent works have focused on implementing MPTCP in the Linux kernel after enabling MPTCP in the operating system or using the NS-3 simulator. Very few works implemented MPTCP on the CORE emulator.

A. Simulation

Chihani et al. [73] implemented MPTCP in the NS-3 simulator and introduced a new protocol that worked better in various network conditions. They compared different packet reordering systems and analyzed that their implementations will be necessary for further MPTCP performance analysis in terms of controlling congestion. Nadeem et al. [74] worked on introducing three path managers; default, ndiffports, and fullmesh to create an MPTCP patch for implementing MPTCP in the NS-3 development version. While the default patch has not made any new subflows, fullmesh made a mesh of whole new subflows towards the feasible pairs of IP addresses; ndiffports introduced subflows in between the same IP pair with the help of distinct source and destination. It showed better results in terms of getting higher throughput and less flow completion time than prior works. Coudron et al. [75] proposed MPTCP implementations in NS-3 to handle network traffic. They also compared their algorithm with previous work implemented in NS-3 and Kernel. Table I lists some other MPTCP implementations in NS-2, NS-3 and CORE simulator with the aim of congestion control or schedule packets or perform both congestion control and packet scheduling for MPTCP.

B. Real Hardware (kernel)

Network simulators sometimes fail to depict the original network conditions, as the real-world network is highly dynamic; the breaking of links and the creation of new links are spontaneous. Therefore, the evaluation of MPTCP on real-world networks using Linux kernels shows a much bigger picture of its strengths and weaknesses. In the work [76], the authors have implemented MPTCP in a Linux kernel to study if each subflow has a different scheduler and then how the different subflows of an MPTCP may dispute bottleneck links with conventional single-path TCP. They tested LIA, OLIA, BALIA and wVegas on Linux kernel implementation of MPTCP and evaluated the throughput, latency, etc., on real-world networks. Zannettou et al. [24] used the kernel implementation of MPTCP to show their MPTCP-aware scheduling performs better than random hashing of packets to subflows which is generally used. They used the FatTree [77] and Jellyfish [78] topologies to conduct their experiments. FatTree is a highly structured topology used in data centers to obtain the highest throughput cost-effectively, while Jellyfish is the most commonly used randomly structured topology which can support more hosts than the FatTree, while keeping almost the same throughput. The commercial application for MPTCP support is available online [79].

VIII. Conclusion

This paper focuses on two crucial concepts in MPTCP - congestion control and scheduling. The study shows how the most recent works fulfill the previous work gaps and mitigate the above two MPTCP issues using different classical and ML-based approaches. Our study also presents the advantages and limitations of current works and encourages the researchers to continue further improvements in this domain. As in every communication sector MPTCP establishes a tremendous role, it is necessary to improve the performance of MPTCP, and our paper can be beneficial for the readers to have an extensive knowledge of MPTCP performance issues and can use it for proposing new algorithms.

References

[1] L. G. Roberts, "The evolution of packet switching," *Proceedings of the IEEE*, vol. 66, no. 11, pp. 1307–1313, 1978.

[2] M. Hauben, "History of ARPANET," *Site de l'Instituto Superior de Engenharia do Porto*, vol. 17, pp. 1–20, 2007.

[3] J. Postel, "Rfc0768: User Datagram Protocol," 1980.

[4] J. Postel, "Transmission Control Protocol," tech. rep., Information Sciences Institute, University of Southern California, 1981.

[5] S. H. Baidya and R. Prakash, "Improving the performance of multipath TCP over heterogeneous paths using slow path adaptation," in *2014 IEEE International Conference on Communications (ICC)*, pp. 3222–3227, IEEE, 2014.

[6] G. Huston, "TCP in a wireless world," *IEEE Internet Computing*, vol. 5, no. 2, pp. 82–84, 2001.

[7] L. Chao, C. Wu, T. Yoshinaga, W. Bao, and Y. Ji, "A brief review of multipath TCP for vehicular networks," *Sensors*, vol. 21, no. 8, p. 2793, 2021.

[8] S. Hassayoun, J. Iyengar, and D. Ros, "Dynamic window coupling for multipath congestion control," in *2011 19th IEEE International Conference on Network Protocols*, pp. 341–352, IEEE, 2011.

[9] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "MPTCP is not Pareto-optimal: Performance issues and a possible solution," *IEEE/ACM Transactions On Networking*, vol. 21, no. 5, pp. 1651–1665, 2013.

[10] A. Walid, Q. Peng, J. Hwang, and S. Low, "Balanced linked adaptation congestion control algorithm for MPTCP," *Internet Engineering Task Force, Internet-Draft draft-walid-mptcp-congestion-control-04*, 2016.

[11] R. K. Chaturvedi and S. Chand, "An adaptive and efficient packet scheduler for multipath TCP," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, pp. 349–365, 2021.

[12] T. A. Le, C. S. Hong, M. A. Razzaque, S. Lee, and H. Jung, "ecMTCP: An energy-aware congestion control algorithm for multipath TCP," *IEEE Communications Letters*, vol. 16, no. 2, pp. 275–277, 2011.

[13] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10, IEEE, 2012.

[14] X. Ji, B. Han, R. Li, C. Xu, Y. Li, and J. Su, "ACCeSS: adaptive QoS-aware congestion control for multipath TCP," in *2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, IEEE, 2022.

[15] T. Ha, A. Masood, W. Na, and S. Cho, "Intelligent multi-path TCP congestion control for video streaming in internet of deep space things communication," *ICT Express*, 2023.

[16] H. Wu, O. Alay, A. Brunstrom, G. Caso, and S. Ferlin, "Falcon: Fast and accurate multipath scheduling using offline and online learning," *arXiv preprint arXiv:2201.08969*, 2022.

[17] J. Han, K. Xue, J. Li, R. Zhuang, R. Li, R. Yu, G. Xue, and Q. Sun, "EdAR: An experience-driven multipath scheduler for seamless handoff in mobile networks," *IEEE Transactions on Wireless Communications*, 2023.

[18] S. J. Siddiqi, F. Naeem, S. Khan, K. S. Khan, and M. Tariq, "Towards AI-enabled traffic management in multipath TCP: A survey," *Computer Communications*, vol. 181, pp. 412–427, 2022.

[19] M. Y. Asiri, "A survey of multipath TCP scheduling schemes: Open challenges and potential enablers." https://www.techrxiv.org/, 2021.

[20] P. Tomar, G. Kumar, L. P. Verma, V. K. Sharma, D. Kanellopoulos, S. S. Rawat, and Y. Alotaibi, "CMT-SCTP and MPTCP multipath transport protocols: A comprehensive review," *Electronics*, vol. 11, no. 15, p. 2384, 2022.

[21] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion control design for multipath transport protocols: A survey," *IEEE communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2948–2969, 2016.

[22] A. Jasin, R. Alsaqour, M. S. Abdelhaq, O. Alsukour, and R. Saeed, "Review on current transport layer protocols for TCP/IP model," *International Journal of Digital Content Technology and its Applications*, vol. 6, no. 14, pp. 495–503, 2012.

[23] W. Wei, K. Xue, J. Han, D. S. Wei, and P. Hong, "Shared bottleneck-based congestion control and packet scheduling for multipath TCP," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 653–666, 2020.

[24] S. Zannettou, M. Sirivianos, and F. Papadopoulos, "Exploiting path diversity in datacenters using MPTCP-aware SDN," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 539–546, IEEE, 2016.

[25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[26] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, "Experience-driven congestion control: When multi-path TCP meets deep reinforcement

learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1325–1336, 2019.

[27] B. He, J. Wang, Q. Qi, H. Sun, J. Liao, C. Du, X. Yang, and Z. Han, "DeepCC: Multi-agent deep reinforcement learning congestion control for multi-path TCP based on self-attention," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4770–4788, 2021.

[28] M. L. Puterman, "Markov decision processes," *Handbooks in Operations Research and Management Science*, vol. 2, pp. 331–434, 1990.

[29] G. Tesauro *et al.*, "Temporal difference learning and TD-Gammon," *Communications of the ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[30] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[32] V. Pramanik and M. Maliha, "Analyzing sentiment towards a product using DistilBERT and LSTM," in *2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 811–816, IEEE, 2022.

[33] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, "WiFi, LTE, or both? Measuring multi-homed wireless internet performance," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, pp. 181–194, 2014.

[34] S. Singh, P. Mudgal, P. Chaudhary, and A. K. Tripathi, "Comparative analysis of packet loss in extended wired LAN environment," *International Journal of Computer Applications*, vol. 117, no. 2, 2015.

[35] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 266–277, 2011.

[36] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, pp. 51–62, 2009.

[37] "Fat-Tree Design." https://clusterdesign.org/fat-trees/. [Accessed 19-Jul-2023].

[38] K. T. Hanna and P. Loshin, "NACK (NAK, negative acknowledgment, not acknowledged)," *TechTarget*, Aug 2021.

[39] F. Yang, Q. Wang, and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 749–752, IEEE, 2014.

[40] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," in *Proceedings of the 2013 cConference on Internet Measurement Conference*, pp. 455–468, 2013.

[41] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[42] O. Ait-Hellal and E. Altman, "Analysis of TCP vegas and TCP reno," *Telecommunication Systems*, vol. 15, no. 3-4, pp. 381–404, 2000.

[43] J. Yang, J. Han, K. Xue, Y. Wang, J. Li, Y. Xing, H. Yue, and D. S. Wei, "TCCC: a throughput consistency congestion control algorithm for MPTCP in mixed transmission of long and short flows," *IEEE Transactions on Network and Service Management*, 2023.

[44] T. Gilad, N. Rozen-Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, "MPCC: Online learning multipath transport," in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, pp. 121–135, 2020.

[45] M. H. Rehmani and Y. Saleem, "Network simulator NS-2," in *Encyclopedia of Information Science and Technology, Third Edition*, pp. 6249–6258, IGI Global, 2015.

[46] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the NS-3 simulator," *SIGCOMM Demonstration*, vol. 14, no. 14, p. 527, 2008.

[47] X. Li *et al.*, "Using" random forest" for classification and regression.," *Chinese Journal of Applied Entomology*, vol. 50, no. 4, pp. 1190–1197, 2013.

[48] A. Singh, M. Xiang, A. Konsgen, C. Goerg, and Y. Zaki, "Enhancing fairness and congestion control in multipath TCP," in *6th joint IFIP Wireless and Mobile Networking Conference (WMNC)*, pp. 1–8, IEEE, 2013.

[49] S. Ferlin, Ö. Alay, T. Dreibholz, D. A. Hayes, and M. Welzl, "Revisiting congestion control for multipath TCP with shared bottleneck detection," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, IEEE, 2016.

[50] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "CORE: A real-time network emulator," in *MILCOM 2008-2008 IEEE Military Communications Conference*, pp. 1–7, IEEE, 2008.

[51] T. Zhang and S. Mao, "Machine learning for end-to-end congestion control," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 52–57, 2020.

[52] R. Zhuang, J. Han, K. Xue, J. Li, D. S. Wei, R. Li, Q. Sun, and J. Lu, "Achieving flexible and lightweight multipath congestion control through online learning," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 46–59, 2022.

[53] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[54] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning*, pp. 387–395, Pmlr, 2014.

[55] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2621–2633, 2019.

[56] J. Xu and B. Ai, "Deep reinforcement learning for handover-aware MPTCP congestion control in space-ground integrated network of railways," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 200–207, 2021.

[57] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 1871–1879, IEEE, 2018.

[58] S. R. Pokhrel, L. Pan, N. Kumar, R. Doss, and H. L. Vu, "Multipath TCP meets transfer learning: A novel edge-based learning for industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10299–10307, 2021.

[59] J. Hwang and J. Yoo, "Packet scheduling for multipath TCP," in *2015 Seventh International Conference on Ubiquitous and Future Networks*, pp. 177–179, IEEE, 2015.

[60] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, pp. 27–32, 2014.

[61] M. Scharf and S. Kiesel, "NXG03-5: Head-of-line blocking in TCP and SCTP: Analysis and measurements," in *IEEE Globecom 2006*, pp. 1–5, IEEE, 2006.

[62] P. Dong, J. Xie, W. Tang, N. Xiong, H. Zhong, and A. V. Vasilakos, "Performance evaluation of multipath TCP scheduling algorithms," *IEEE Access*, vol. 7, pp. 29818–29825, 2019.

[63] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, "DAPS: Intelligent delay-aware packet scheduling for multipath transport," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1222–1227, IEEE, 2014.

[64] F. Yang, Q. Wang, and P. D. Amer, "Out-of-order transmission for in-order arrival scheduling for multipath TCP," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 749–752, IEEE, 2014.

[65] S. Ferlin, Ö. Alay, O. Mehani, and R. Boreli, "BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pp. 431–439, IEEE, 2016.

[66] T.-A. Le and L. X. Bui, "Forward delay-based packet scheduling algorithm for multipath TCP," *Mobile Networks and Applications*, vol. 23, no. 1, pp. 4–12, 2018.

[67] F. Silva, M. Togou, and G.-M. Muntean, "An innovative machine learning approach to improve MPTCP performance," in *2020 International Conference on High Performance Computing and Simulation*, IEEE, 2020.

[68] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 3, pp. 275–294, 2012.

[69] T. M. Mitchell, "Artificial neural networks," *Machine Learning*, vol. 45, no. 81, p. 127, 1997.

[70] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2018.

[71] S. R. Pokhrel and A. Walid, “Learning to harness bandwidth with multipath congestion control and scheduling,” *IEEE Transactions on Mobile Computing*, 2021.

[72] Q. Peng, A. Walid, and S. H. Low, “Multipath TCP algorithms: Theory and design,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 305–316, 2013.

[73] B. Chihani and C. Denis, “A Multipath TCP model for ns-3 simulator,” *arXiv preprint arXiv:1112.1932*, 2011.

[74] K. Nadeem and T. M. Jadoon, “An NS-3 MPTCP implementation,” in *Quality, Reliability, Security and Robustness in Heterogeneous Systems: 14th EAI International Conference, Qshine 2018, Ho Chi Minh City, Vietnam, December 3–4, 2018, Proceedings 14*, pp. 48–60, Springer, 2019.

[75] M. Coudron and S. Secci, “An implementation of multipath TCP in NS3,” *Computer Networks*, vol. 116, pp. 1–11, 2017.

[76] B. Y. L. Kimura and A. A. F. Loureiro, “MPTCP linux kernel congestion controls,” *arXiv preprint arXiv:1812.03210*, 2018.

[77] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63–74, 2008.

[78] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, “Jellyfish: Networking data centers randomly,” in *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pp. 225–238, 2012.

[79] C. Paasch and S. Barre, “Multipath TCP in the linux kernel.” available from http://www.multipath-tcp.org.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 63–64

DOI: 10.15439/2023F0004
ISSN 2300-5963 ACSIS, Vol. 35

Abstract Approach to Entropy and Co-Entropy in Measurable and Probability Spaces Invited Lecture—Extended Abstract

Gianpiero Cattaneo*
* Retired from Department of Informatics, Systems and Communications
University of Milano–Bicocca, Italy

THE talk is divided in following parts.

(Pa1) First of all, the talk is based on the *Universe* $\mathcal{U} := (\mathbb{N}, \mathcal{P}(\mathbb{N}))$, consisting of the set of natural integers $\mathbb{N}$ and its power set $\mathcal{P}(\mathbb{N})$. Elements $a \in \mathbb{N}$ are interpreted as *micro-states* of the system and subsets $A \subseteq \mathcal{P}(\mathbb{N})$ of the universe are interpreted as *events* which can be tested on it. The first part concerns the abstract approach to *measure distributions of total measure* $M > 0$, as non negative sequences of integer numbers $\widehat{m} := (m_1, m_2, \dots, m_M)$, i.e., with any $0 \leq m_j \in \mathbb{N}$, and satisfying the condition $\sum_{j=1}^{\infty} m_j = M$. These measure distributions can also be interpreted as *physical macro-states* of the system and their collection will be denoted as $\mathcal{MD}_M$.

From any M-measure distribution $\vec{m}$ it is induced, in a standard way, the *probability distribution* $\vec{p} := (p_1 = m_1/M, p_2 = m_2/M, \dots, p_M = p_M/M)$, where any $p_j := m_j/M \geq 0$ and $\sum_{j=1}^{M} p_j = 1$ (from which it follows that $p_j \in [0, 1]$, according to the usual definition of probability). Based on these basic notions, the following important definitions are then introduced:

a) The *entropy* associate to a M-measure distribution $\vec{m}$ is defined as $H(\vec{m}) := \log M - \frac{1}{M} \sum_j m_j \log m_j$ (from which it follows the standard Shannon entropy $H(\vec{p}) = -\sum_j p_j \log p_j$).

b) The *co-entropy* (i.e., *complementary* entropy) of $\vec{m}$ is defined as $K(\vec{m}) := \frac{1}{M} \sum_j m_j \log m_j$ (from which it follows $K(\vec{p}) = \sum_j p_j \log(M \cdot p_j)$).

Now, once introduced the *total entropy* as $E(\vec{m}) := H(\vec{m}) + K(\vec{m})$, it is easy to prove the *total entropy conservation principle*: $\forall \vec{m}$, $E(\vec{m}) = \log M$ (resp., $\forall \vec{p}$, $E(\vec{p}) := H(\vec{p}) + K(\vec{p}) = \log M$), quantity *always* equal to the constant $\log M$, i.e., invariant with respect to the considered measure distribution $\vec{m}$ (resp., to the considered probability distribution $\vec{p}$).

(Pa2) In the context of the *concrete Universe*, the second part is dedicated to the partition $\pi(\mathbb{N}) = \{A_1, A_2, \dots, A_K\}$ ($K \leq M$) of the universe $\mathbb{N}$, formed by the so-called *blocks* A_j, and generating the equivalence relation $\equiv$ on $\mathbb{N}$:

let $a_r, a_s \in \mathbb{N}$, then $a_r \equiv a_s$ iff $\exists A \in \pi(\mathbb{N})$: $a_r \in A$ and $a_s \in A$.

The partition $\pi(\mathbb{N})$ determines the measure distribution:

$$\vec{m}(\pi(\mathbb{N})) = \big(m(A_1) = |A_1|, \\ m(A_2) = |A_2|, \dots, m(A_K) = |A_K|\big),$$

of total measure $m(\pi(X)) := \sum_{j=1}^{K} m(A_j) = \sum_{j=1}^{K} |A_j| = M$; with corresponding entropy and co-entropy.

But in some sense, the present part (Pa2) constitutes a point of connection between the abstract content of point (Pa1) and the concrete part which constitutes the fundamental content of the next point (Pa3).

(Pa3) The collection $\mathcal{MD}_M$ of all measure distributions was the subject of a long research conducted together with French collaborators many years ago entitled *Ice Pile Models*. Brylawski in a paper published od Discrete Mathematics (1973) investigated from a pure algebraic point of view the collection $\mathcal{IP}_M$ of M-uples of non-negative integers $\alpha := (a_1, a_2, \dots, a_M)$, called *integer partitions*, satisfying the further condition of being *non-increasing*: $\forall i$, $a_{i-1} \geq a_i$. Trivially, $\mathcal{IP}_M \subseteq \mathcal{MD}_M$. This collection has a lattice structure with respect to a lattice order $\leq$, called *dominance*, with minimum element $\mathbf{0} := (M, 0, \dots, 0)$ and maximum element $\mathbf{1} := (1, 1, \dots, 1)$, i.e., $\forall \alpha \in \mathcal{IP}_M$, $\mathbf{0} \leq \alpha \leq \mathbf{1}$. On the lattice $\mathcal{IP}_M$ Brylawski introduces the definition of *elementary transition* as the binary relationship: $\alpha \to \beta$ iff $\alpha \lneqq \beta$ and $\exists \gamma$: $\alpha \leq \gamma \leq \beta$, then either $\gamma = \alpha$ or $\gamma = \beta$. On the standard configuration space $\mathcal{IP}_M$ of Brylawski, we interpret the component of place i of the configuration α as a column of a_i grains that can move from left to right. The most interesting result is the following

Theorem – *The configuration transition $\alpha \to \alpha'$ in the space $\mathcal{IP}_M$ occurs iff one of the only two cases occurs: (1) The* vertical *evolution rule (VR) which solves jumps in the configuration of two units. Formally,*

$$\textit{let } \; a_{j-1} \geq a_j + 2, \quad \textit{then}$$
$$(a_{j-1}, a_j, a_{j+1}) \xrightarrow{(VR)} (a_{j-1} - 1, a_j + 1, a_{j+1})$$

(2) the horizontal evolution rule (HR) in which a grain can slip from column $j-1$ to column j according to the following local behaviour

$$\textit{let } a_{j-1}-1=a_j, \quad \textit{then}$$
$$(a_{j-1},a_j,a_{j+1}) \xrightarrow{(HR)} (a_{j-1}-1,a_j+1,a_{j+1})$$

But, the Ice Pile model has been defined by Goles–Kiwi (1993) just as the structure based on the Brylawski lattice equipped with the two local evolution rules (VR) and (HR). Furthermore, from another point of view, since the two local rules involve neighborhoods of the center a_i of radius 1, they define a one-dimensional *Elementary Cellular Automata*. In any case, whatever the point of view adopted, the application of the two rules (HR) and (VR) determine a discrete time dynamical system since it is possible to consider all the dynamic evolutions $\gamma : \mathbb{N} \mapsto \mathcal{IP}_M$ of initial state $\mathbf{0}$ of the kind $\mathbf{0} \to \alpha(1) \to \alpha(2) \to \ldots \to \alpha(t) \to \ldots$, which by a Theorem converges after a finite number t_f of time steps to the unique final equilibrium configuration $\ldots \to \alpha(t_f) = \mathbf{1}$.

The important point is that if to any configuration $\alpha(t)$ of a trajectory γ one calculates the corresponding entropy $H(\alpha(t))$ one obtains the strictly increasing chain of positive numbers $0 = H(\mathbf{0}) < H(\alpha(1)) < H(\alpha(2)) < \ldots < H(\alpha(t)) < \ldots < H(\mathbf{1})$ and, also important result, if any transition $\alpha(t) \to \alpha(t+1)$ is the result of a parallel application of the (VR) rule to $\alpha(t)$, then the dynamical evolution is unique. It is an open problem to prove some similar result in the case of the Ice Pile parallel dynamical evolution.

(Pa4) The forth part is dedicated to the exposition of those theories which are a concrete application of the abstract treatment made in point (Pa1) of entropy in measurable spaces:

(1) the *Pawlak rough set theory* as mathematical approach to imperfect knowledge, with its application to information systems, based on a non-empty set X of objects forming the Universe of the discourse, a non-empty set Att of attributes, and a function $F : X \times Att \mapsto val$ assigning to any pair $(x,a) \in X \times Att$ consisting of an object x and an attribute a a value $F(x,a) \in val$. Fixed a collection $\mathcal{A} := \{a_1, a_2, \ldots, a_N\}$ from the set of all attributes Att, two objects x_1 and x_2 are considered *indistinguishable* with respect to $\mathcal{A}$ iff $F(x_1,a) = F(x_2,a)$ for every attribute $a \in \mathcal{A}$. Trivially, this binary relation of indistinguishability is an equivalence relation on the set of all objects X, inducing a partition $\pi_{\mathcal{A}}(X)$ of the universe X, and so all the results of part (Pa2) can be applied to the present case of rough set theory;

(2) the *Zadeh fuzzy set theory*, as distributive lattice equipped with a (unique) Kleene negation connective. The De Luca–Termini approach to fuzzy set as distributive lattice equipped with a (unique) Brouwer, or intuitionistic, negation connective has been successively introduced. Based on these two (non intersecting) approaches, it is then defined the Brouwer–Zadeh (BZ) distributive lattice structure equipped with both the Zadeh and the Brouwer negation connectives, where rough sets constitute its crisp part (without however capturing all the applicative richness exposed in point (1)).

(3) the *conservative self–reversible logical gates*. Conservative logic is a model of computation whose principal aim is to compute with zero dissipation of internal Shannon entropy. This goal is reached by basing the model upon reversible and conservative primitives, for example Fredkin and Toffoli gates, which reflect physical principles such as the *reversibility* of microscopic dynamical laws and the *conservation* of certain physical quantities, such as the entropy of the physical system used to perform the computations.

Finally, a standard procedure for embedding non-reversible logical gates into reversible–conservative logical ones is exposed.

Towards reliable rule mining about code smells: The McPython approach

Maciej Ziobrowski, Mirosław Ochodek, Jerzy Nawrocki, Bartosz Walter
Poznan University of Technology, Poznań, Poland
maciej.ziobrowski@student.put.poznan.pl; {miroslaw.ochodek, jerzy.nawrocki, bartosz.walter}@put.poznan.pl

***Index Terms*—Rule mining, code smell, McPython, Python, domain specific languages**

Problem

CODE smell is a risky pattern in code that can lead, in the future, to problems with code maintenance. One of the approaches to identifying smells in the code is metric-based smell detection. A classic example is the *God Class* smell which can be detected by using three metrics (see, e.g., [1], [2], [3]):

- Weighted Method Count (WMC – sum of McCabe's complexity of all methods in the analysed class),
- Tight Class Cohesion (TCC – relative number of directly connected methods within the analysed class), and
- Access to Foreign Data (ATFD – number of classes containing attributes referenced by the analysed class directly or via get/set methods).

To make a decision (smelly / not smelly), computed metrics are compared against predefined thresholds. So, the quality of smell detection depends not only on a set of chosen metrics, but also on their thresholds.

Unfortunately, the quality of the existing smell detectors is still not satisfactory (cf. [4]) and there is a need for more research in the area. One of the issues worth investigation is the impact of a set of code smells on severity of the detriment caused by them. To conduct this research in a clear and reproducible way one needs an appropriate workbench (a critical review of the literature in the area is presented in [5]).

The proposed workbench

In this paper, it is postulated that empirical research on smell detectors should be based on (1) precise definitions of the analysed smells, and (2) smell detection rules (including metric thresholds) should be mined from software repositories using machine learning (ML).

The overall architecture of the proposed workbench is illustrated in Fig. 1. Given a code repository, a code smell detector identifies all smelly classes while the issue detector identifies troublesome classes (e.g., defective classes - here one can use an idea proposed by Śliwerski *et al.* in [6]). The reports generated by both detectors are consolidated to produce a decision table (the decision table of Fig. 1 refers to the *God Class* smell with three thresholds, _WMC, _TCC, and _ATFD, corresponding to the three metrics mentioned earlier). Given a decision table, one can use e.g. C4.5 algorithm to get a decision tree (see [7] or [8]). Another option is to apply rough-set approach (see e.g., [9]).

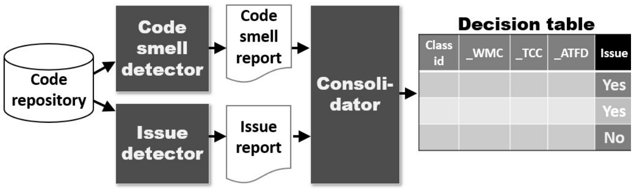


Fig. 1. Architecture of the proposed workbench.

The McPython language

For defining metric-based code smells we propose a domain-specific language, McPython (Meta Code in Python). Its notation is based on Python. Description of a smell detector consists of three parts: code model, smell definitions, and query.

Code model defines all the code attributes needed for detection of a given smell (it corresponds to view model in the 3-layer model of code proposed in [10]). Those attributes are provided by another program, code modeller, and code model just defines what is needed from the code modeller. As McPython is focused on object-oriented languages, there are four categories of entities represented in each model, namely: classes, their attributes, methods, and their parameters. An example of code model is presented in the first part of Listing 1. Each code entity has a number of attributes along with their JSON types (*nat* is an extra type denoting natural numbers and it is a subset of *int*). Each description of entity category starts with the `ent` keyword and ends with a double colon (::).

A smell definition is a Python-like function returning a Boolean value. It is accompanied by a set of auxiliary functions (some of them can be imported). The second part of Listing 1 contains a function named `GodClass` defining the *God Class* smell and an auxiliary function `WMC`. The `GodClass` function uses three special parameters called *thresholds*: `_WMC`, `_TCC`, and `_ATFD`. A threshold represents an upper/lower bound on some metric. It is declared in a separate line, its name begins with an underscore ('_') and is preceded with the `thr` keyword.

Smell definitions can refer to attributes specified in the code model and they can contain mathematical symbols such as summation ($\sum$) or quantifiers($\forall$, $\exists$). On the other hand there are some restrictions imposed on McPython code:

Listing 1. God Class detector in McPython.

```
--- Model:
ent class:
    name:    string,   # class’s name
    methods: list,     # method ids
ent method:
    name: string,  # method’s name
    McCABE: nat :: # cyclomatic complx.
--- Smells:
import TCC, ATFD
def WMC(c: class):
    return ∑ m ∈ c.methods: m.McCABE::
thr _WMC
thr _TCC
thr _ATFD
def GodClass(c: class):
    return WMC(c) ⩾ _WMC ∧
           TCC(c) < _TCC ∧ ATFD > _ATFD
--- Query:
_WMC  ∈ [45, 47]
_TCC  ∈ [0.4, 0.3, 0.2]
_ATFD ∈ [4, 6]
select c.name for c ∈ class \
              where GodClass(c)
```

- each variable is assigned a value only once;
- there are no compound statements like `while` or `if`.

Parameters of McPython functions can have types assigned to them. Those types are categories of code entities, e.g., `class` or `method`.

The third part of code in McPython is a `query`. It starts with specifying the values of the thresholds one is interested in. Then comes the `select` clause which resembles the one known from SQL. The result of the query is a report showing the requested attributes of all the code entities matching the query for all the possible combinations of the values of thresholds.

Implementation remarks

McPython definition of a smell detector is encoded in Unicode what makes all the mathematical symbols easily available. When McPython code is ready one has:

- to translate it to Python 3, and
- to generate a model of the analysed code (smell detector expects on the input a code model, not the code itself).

The process is illustrated in Fig. 2. An advantage of running smell detector on a code model instead of the code itself is possibility of using the same definition of a code smell on repositories written in different programming language, provided that one has a code modeller for a given language.

Current version of McPython translator is written in Python 3 (Python accepts Unicode as an input). Model of the analysed code is implemented as a list of all its entities (position of an entity on the list serves as its identifier) and it is read with the library function `json.loads`. McPython constructs concerning operations over sets, e.g., a universal quantifier ($\forall$) or summation ($\sum$), are translated as calls to an appropriate function (definitions of those functions are added to the generated code). Those functions have two parameters: a set of code entities (represented by their identifiers) and a condition or expression that is evaluated for each element of a given set (here lambda expressions of Python proved very useful).

Code modeller for the Python language (cf. Fig. 2) is built with the help of Python’s `ast` module and the `NodeVisitor` class contained in it. First all class nodes of a given abstract syntax tree are visited and then their method are analysed. The collected data are stored as an array of dictionaries and transformed into JSON with the `dumps` function of the `json` module.

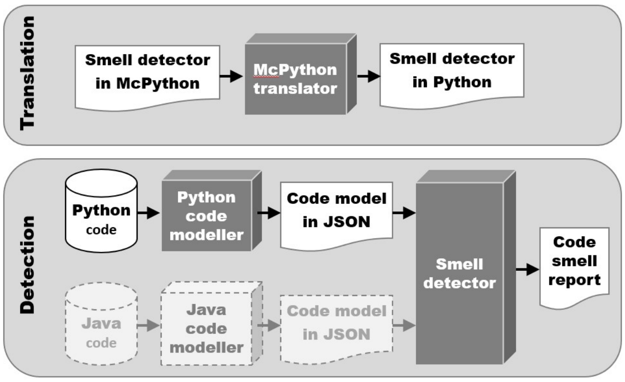


Fig. 2. Translation and detection phase.

References

[1] S. M. Olbrich, D. S. Cruzes, and D. I. Sjøberg, “Are all code smells harmful? a study of god classes and brain classes in the evolution of three open source systems,” in *2010 IEEE International Conference on Software Maintenance*, 2010, pp. 1–10.

[2] Ł. Puławski, “Improvement of design anti-pattern detection with spatio-temporal rules in the software development process,” in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 521–528.

[3] Łukasz Puławski, “Methods of detecting spatio-temporal patterns in software development processes,” Ph.D. dissertation, University of Warsaw, 2022. [Online]. Available: https://ornak.icm.edu.pl/bitstream/handle/item/4533/0000-DR-1827-praca.pdf?sequence=1

[4] T. Sharma, G. Suryanarayana, and G. Samarthyam, “Challenges to and solutions for refactoring adoption: An industrial perspective,” *IEEE Software*, vol. 32, no. 6, pp. 44–51, 2015.

[5] T. Lewowski and L. Madeyski, “How far are we from reproducible research on code smell detection? a systematic literature review,” *Information and Software Technology*, vol. 144, p. 106783, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095058492100224X

[6] J. Sliwerski, T. Zimmermann, and A. Zeller, “When do changes induce fixes?” *SIGSOFT Softw. Eng. Notes*, vol. 30, no. 4, p. 1–5, may 2005. [Online]. Available: https://doi.org/10.1145/1082983.1083147

[7] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[8] “Weka 3: Machine learning software in java,” https://www.cs.waikato.ac.nz/ml/weka/.

[9] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko, “Rough sets,” *Commun. ACM*, vol. 38, no. 11, p. 88–95, nov 1995. [Online]. Available: https://doi.org/10.1145/219717.219791

[10] D. Strein, R. Lincke, J. Lundberg, and W. Löwe, “An extensible meta-model for program analysis,” *IEEE Transactions on Software Engineering*, vol. 33, no. 9, pp. 592–607, 2007.

DOI: 10.15439/2023F0001
ISSN 2300-5963 ACSIS, Vol. 35

Combination of Fuzzy Sets and Rough Sets for Machine Learning Purposes (Tutorial – Extended Abstract)

Chris Cornelis, Henri Bollaert
Computational Web Intelligence,
Department of Applied Mathematics, Computer Science and Statistics,
Ghent University, Belgium

FUZZY set theory (Zadeh [22], 1965) is a popular AI tool designed to model and process vague information. Specifically, it is based on the idea that membership to a given concept, or logical truthhood of a given proposition, can be a matter of degree. On the other hand, rough set theory (Pawlak [14], 1982) was proposed as a way to handle potentially inconsistent data inside information systems. In Pawlak's original proposal, this is achieved by providing a lower and upper approximation of a concept, using the equivalence classes of an indiscernibility relation as building blocks.

Noting the highly complementary characteristics of fuzzy sets and rough sets, Dubois and Prade [7] proposed the first working definition of a fuzzy rough set, and thus paved the way for a flourishing hybrid theory with numerous theoretical [8] and practical [18] advances.

In this tutorial, we will explain how fuzzy rough sets may be successfully applied to a variety of machine learning problems. After a brief discussion of how the hybridization between fuzzy sets and rough sets may be achieved, including an extension based on ordered weighted average operators (see e.g. [1], [4]–[6]), we will focus on the following practical applications:

1) Fuzzy-rough nearest neighbor (FRNN) classification [10], [11], [21], along with its adaptations for imbalanced datasets [15], [19] and multi-label datasets [20]
2) Fuzzy-rough feature selection (FRFS) [2], [3]
3) Fuzzy-rough instance selection (FRIS) [9] and Fuzzy-rough prototype selection (FRPS) [16], [17]

We will also demonstrate software implementations of all of these algorithms in the Python library fuzzy-rough-learn [12], [13].

References

[1] C. Cornelis, M. De Cock, A.M. Radzikowska, Fuzzy rough sets: from theory into practice, in: Handbook of Granular Computing (W. Pedrycz, A. Skowron, V. Kreinovich, eds.), Wiley, 2008, pp. 533–552.

[2] C. Cornelis, R. Jensen, A noise-tolerant approach to fuzzy-rough feature selection, in: Proc. 2008 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008), 2008, pp. 1598–1605.

[3] C. Cornelis, R. Jensen, G. Hurtado Martín, and D. Ślęzak, Feature selection with fuzzy decision reducts, in: Proc. Third International Conference on Rough Sets and Knowledge Technology (RSKT 2008), 2008, pp. 284–291.

[4] C. Cornelis, N. Verbiest, and R. Jensen, Ordered weighted average based fuzzy rough sets, in: Proc. 5th International Conference on Rough Sets and Knowledge Technology (RSKT 2010), 2010, pp. 78–85.

[5] M. De Cock, C. Cornelis, E.E. Kerre, Fuzzy rough sets: beyond the obvious, in: Proc. 2004 IEEE Int. Conf. on Fuzzy Systems, FUZZ-IEEE'04, Volume 1, 2004, pp. 103-108.

[6] L. D'eer, N. Verbiest, C. Cornelis, L. Godo, A comprehensive study of implicator-conjunctor based and noise-tolerant fuzzy rough sets: definitions, properties and robustness analysis, Fuzzy Sets and Systems **275**, 2015, pp. 1–38.

[7] D. Dubois and H. Prade, Rough fuzzy sets and fuzzy rough sets, International Journal of General Systems **17**, 1990, pp. 91–209.

[8] M. Inuiguchi, W.Z. Wu, C. Cornelis, N. Verbiest, Fuzzy-rough hybridization, in: Springer Handbook of Computational Intelligence, 2015, pp. 425–451.

[9] R. Jensen, C. Cornelis, Fuzzy-rough instance selection, in: Proc. 19th International Conference on Fuzzy Systems (FUZZ-IEEE 2010), 2010, pp. 1776–1782.

[10] R. Jensen and C. Cornelis, Fuzzy-rough nearest neighbour classification, Transactions on rough sets, vol. XIII, 2011, pp. 56–72.

[11] O.U. Lenz, D. Peralta, C. Cornelis, Scalable approximate FRNN-OWA classification, IEEE Transactions on Fuzzy Systems **28**(5), 2020, pp. 929–938.

[12] O.U. Lenz, D. Peralta, C. Cornelis, Fuzzy-rough-learn 0.1: A Python library for machine learning with fuzzy rough sets, in: Proc. International Joint Conference on Rough Sets, 2020, pp. 491–499.

[13] O.U. Lenz, D. Peralta, C. Cornelis, Fuzzy-rough-learn 0.2: a Python library for fuzzy rough set algorithms and one-class classification, in: Proc. 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2022, pp. 1–8.

[14] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences **11**(5), 1982, pp. 341–356.

[15] E. Ramentol, S. Vluymans, N. Verbiest, Y. Caballero, R. Bello, C. Cornelis, F. Herrera, IFROWANN: imbalanced fuzzy-rough ordered weighted average nearest neighbor classification, IEEE Transactions on Fuzzy Systems **23**(5), 2015, pp. 1622–1637.

[16] N. Verbiest, C. Cornelis, F. Herrera, FRPS: a fuzzy rough prototype selection method, Pattern Recognition **46**(10), 2013, pp. 2770–2782.

[17] N. Verbiest, C. Cornelis, F. Herrera, OWA-FRPS: a prototype selection method based on ordered weighted average fuzzy rough set theory, in: Proc. 14th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC 2013), 2013, pp. 180–190.

[18] S. Vluymans, L. D'eer, Y. Saeys, C. Cornelis, Applications of fuzzy rough set theory in machine learning: a survey, Fundamenta Informaticae **142**(1-4), 2015, pp. 53–86.

[19] S. Vluymans, A. Fernández, Y. Saeys, C. Cornelis, F. Herrera, Dynamic affinity-based classification of multi-class imbalanced data with one-vs-one decomposition: a fuzzy rough approach, Knowledge and Information Systems **6**(1), 2018, pp. 55–84.

[20] S. Vluymans, C. Cornelis, F. Herrera, Y. Saeys, Multi-label classification using a fuzzy rough neighborhood consensus, Information Sciences **433-434**, 2018, pp. 96–114.

[21] S. Vluymans, N. Mac Parthalain, C. Cornelis, Y. Saeys, Weight selection strategies for ordered weighted average based fuzzy rough sets, Information Sciences **501**, 2019, pp. 155–171.

[22] L.A. Zadeh, Fuzzy sets, Information and Control **8**, 1965, pp. 338–353.

DOI: 10.15439/2023F0003
ISSN 2300-5963 ACSIS, Vol. 35

Rough Sets: An Introductory Overview Tutorial – Extended Abstract

Soma Dutta
University of Warmia and Mazury in Olsztyn
Słoneczna 54, Olsztyn, Poland
soma.dutta@matman.uwm.edu.pl

Davide Ciucci
University of Milano-Bicocca
viale Sarca 336/14, 20126, Milano, Italy
davide.ciucci@unimib.it

***Abstract*—The aim of this tutorial is to present a brief overview of the theory of rough sets from the perspective of its mathematical foundations, history of development as well as connections with other branches of mathematics and informatics. The content concerns both the theoretical and practical aspects of applications. The above mentioned target of the tutorial will be covered in two parts. In the first part we would aim to present the introduction to rough sets and the second part will focus on the connections with other branches of mathematics and informatics.**

I. Introduction to Rough Sets

THE THEORY of rough sets, pioneered by Z. Pawlak in 1982 [1], [2], provides a way to formalize imprecise concepts with respect to a given set of attributes. Let us think of a data table, where the rows represent descriptions of the objects from a universe U with respect to a set of (conditional) attributes A, and each column of the data table corresponds to an attribute of A. Each such description is usually known as *information signature* of an object with respect to the set of attributes A, and it can be represented as a vector of values over A. Formally, one can think of a pair (U, A) where for each $a \in A$ and $u \in U$, $a(u)$ denotes the value of the object u for the attribute a in a relevant value set. In rough set literature such a pair is known as *information system* or *information table*. Moreover, an information system along with a designated attribute d for decision, say $(U, A \cup \{d\})$ is called a *decision system*.

Now, given an information system, with respect to any subset X of A the whole data table can be clustered into equivalence classes of objects where each equivalence class contains all those objects from the universe which have the same values with respect to each attribute of X. Thus from (U, A) we obtain a pair (U, R) where R is the respective equivalence relation, usually known as *indistinguishability relation*. In rough set literature (U, R) is known as *approximation space*. A subset $S \subseteq U$, may be called a concept, can be either definable in terms of the union of some equivalence classes, or may fall in the overlapping zone of a few equivalence classes; in the latter case the concept is regarded as imprecise with respect to the considered set of attributes. So, given any set of objects S one can describe the intention of S in terms of rough approximation operators. The lower approximation of S picks up those equivalence classes which are completely contained within S, and thus gives a flavour of 'certainty zone for the concept'. On the other hand, the upper approximation of S selects those equivalence classes which have non-empty intersection with S, and this corresponds to a 'possibility zone for the concept'.

With this mathematical foundation the development of rough set theory goes further to address many useful aspects of data mining. For example, suppose there are two decision classes representing positive and negative decision for the attribute d (i.e., subsets of U representing $d = 1$ and $d = 0$ respectively). Now if these decision classes are not definable in a straightforward manner, one may require to characterize them using rough set approximations. In this presentation, we will try to present some of such aspects which have practical uses in the context of data mining. A few such issues are listed below.

1) To describe a data table in terms of comprehensible rules so that using those rules unseen test examples can be effectively classified.
2) To find out a smaller set of attributes, a *reduct* in RST terminology, that can faithfully classify the decision classes as it is presented with respect to the whole set of attributes.
3) To handle a decision system where two indistinguishable objects may have different decision values.
4) To design decision valuations describing different aspects of decision making by aggregating available information of the training objects (i.e., already available objects).
5) To check similar aspects of decision making when the available data is not clustered into disjoint equivalence classes as the underlying notion of indistinguishability can be based on a relation which is weaker than an equivalence relation, such as a similarity relation.

II. Connection with other branches of mathematics and informatics

Due to its simplicity and effectiveness, the concept of approximations has found applications in various fields, establishing connections with several branches of mathematics and computer science right from its inception. Over the years, we have observed a substantial accumulation of results, which we can only provide a high-level summary of.

A. Mathematics

The main contributions relate to logic and topology, with multiple connections also existing in algebra and graph theory.

a) Logics: Rough sets are associated with modal logics. Indeed, given the standard syntax of the modal system S5, a semantics can be provided by the indiscernibility relation used as the modal accessibility relation. In this manner, the lower and upper approximations coincide with the logic operators of necessity and possibility [3]. It is evident that for rough sets based on weaker forms of relations, there corresponds weaker modal logics. Additionally, by interpreting the lower approximation as positive (or true), the complement of the upper approximation as negative (or false) and the remaining elements of the universe as unknown a correspondence with three-valued logics can be established [4]. Finally, a complete logical framework based on a distinct notion of truth, namely *rough truth*, has been defined where both syntax and semantics are "rough" [5].

b) Topology and Algebra: The lower and upper approximations behave like a topological interior and closure operator on a Boolean algebra. Several links between various types of topological operators and models of rough sets have been established [6]. Moving to a more abstract context, a hierarchy of topological operators can be defined on a lattice structure, each corresponding to a different model based on various generalizations of rough sets [7]. Many authors have taken further steps toward abstraction by defining the approximations in different types of algebraic structures, such as rings and groups.

c) Graph Theory: The connection with graph theory can be interpreted in at least two ways. Firstly, by relating ideas from rough sets to those on graph theory. One significant result in this setting is the equivalence between computing reducts and computing minimal transversal on hypergraphs. The latter is a well-known problem and algorithms that solve it in incremental polynomial time exist [8]. Another approach, consists in applying rough-set ideas to graph theory, such as attempting to define approximations or reducts on graphs.

B. Computer Science

We highlight the main contributions of rough sets to Artificial Intelligence and Theoretical Computer Science.

a) Knowledge Representation: Of course, the first link with computer science and artificial intelligence concerns the ability of rough sets to represent and handle uncertainty due to the granularity of the universe. Particularly fruitful in this domain has been the connection with other tools to manage different forms of uncertainty, mainly fuzzy sets [9] and belief functions [10].

b) Machine Learning and Data Mining: From an application stanpoint, the main contribution of Rough sets is in Machine Learning and Data Mining, where they have been used in several tasks [11]. In particular, in Machine Learning their success can be seen in feature selection and classification by means of reducts and decision rules and in clustering where the idea of approximations has been applied to obtain new soft clustering methods. In Data Mining, a major contribution is the use of rough sets to perform *approximate queries* in relational databases by means of standard SQL statements, this approach also lead to a successful industrial application [12].

c) Theoretical Computer Science: The concept of entropy has been used to evaluate the uncertainty of a given information table with an equivalence relation. Extended approaches to missing values and generalized relation has also been provided [13] as well as applications in computing approximate reducts [14]. Another connection with TCS concerns the use of rough sets in dealing with uncertainty in discrete dynamical systems, such as cellular automata, reaction systems and Petri nets [15].

References

[1] Z. Pawlak, "Rough sets," *Int. J. Inform. Comput. Sci.*, vol. 11, pp. 341–356, 1982.

[2] ——, *Rough sets - theoretical aspects of reasoning about data*, ser. Theory and decision library : series D. Kluwer, 1991, vol. 9.

[3] E. Orlowska, "A logic of indiscernibility relations," ser. Lecture Notes in Computer Sciences. Berlin: Springer-Verlag, 1985, no. 208, pp. 177–186.

[4] D. Ciucci and D. Dubois, "Three-valued logics, uncertainty management and rough sets," *Trans. Rough Sets*, vol. 17, pp. 1–32, 2014. [Online]. Available: https://doi.org/10.1007/978-3-642-54756-0_1

[5] M. Banerjee, "Logic for rough truth," *Fundam. Informaticae*, vol. 71, no. 2-3, pp. 139–151, 2006.

[6] P. K. Singh and S. Tiwari, "Topological structures in rough set theory: A survey," *Hacettepe Journal of Mathematics and Statistics*, vol. 49, no. 4, pp. 1270 – 1294, 2020.

[7] G. Cattaneo and D. Ciucci, "Lattices with interior and closure operators and abstract approximation spaces," *Trans. Rough Sets*, vol. 10, pp. 67–116, 2009.

[8] G. Chiaselotti, D. Ciucci, T. Gentile, and F. Infusino, "Generalizations of rough set tools inspired by graph theory," *Fundam. Informaticae*, vol. 148, no. 1-2, pp. 207–227, 2016.

[9] M. Inuiguchi, W.-Z. Wu, C. Cornelis, and N. Verbiest, *Fuzzy-Rough Hybridization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 425–451.

[10] A. Campagner, D. Ciucci, and T. Denoeux, "Belief functions and rough sets: Survey and new insights," *Int. J. Approx. Reason.*, vol. 143, pp. 192–215, 2022.

[11] R. Bello and R. Falcon, *Rough Sets in Machine Learning: A Review*. Cham: Springer International Publishing, 2017, pp. 87–118.

[12] D. Slezak, P. Synak, A. Wojna, and J. Wroblewski, "Two database related interpretations of rough approximations: Data organization and query execution," *Fundam. Informaticae*, vol. 127, no. 1-4, pp. 445–459, 2013.

[13] D. Bianucci and G. Cattaneo, "Information entropy and granulation co-entropy of partitions and coverings: A summary," *Trans. Rough Sets*, vol. 10, pp. 15–66, 2009.

[14] D. Slezak, "Approximate entropy reducts," *Fundam. Informaticae*, vol. 53, no. 3-4, pp. 365–390, 2002.

[15] A. Campagner, D. Ciucci, and V. Dorigatti, "Uncertainty representation in dynamical systems using rough set theory," *Theor. Comput. Sci.*, vol. 908, pp. 28–42, 2022.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 71–71
DOI: 10.15439/2023F0002
ISSN 2300-5963 ACSIS, Vol. 35

Reducts in Rough Sets: Algorithmic Insights, Open Source Libraries and Applications (Tutorial – Extended Abstract)

Andrzej Janusz*†, Sebastian Stawicki†
*Faculty of Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
Email: ajanusz@mimuw.edu.pl
†QED Software, Mazowiecka 11/49, 00-052 Warsaw, Poland
Email: sebastian.stawicki@qed.pl

THE theory of rough sets [1] is a powerful mathematical framework for handling imprecise or uncertain information in data analysis and decision-making. At its core, rough set theory introduces the concept of decision reducts [2], which are subsets of attributes or features that preserve the essential information needed to make accurate decisions while eliminating redundant or irrelevant information. By identifying ensembles of decision reducts [3], analysts can simplify complex datasets, improve classification accuracy, and gain valuable insights from noisy or incomplete data [4]. These appealing characteristics make rough sets a valuable tool in various fields, including machine learning, data mining, and expert systems [5].

There have been proposed many extensions to the notion of decision reducts, such as approximate decision reducts [6], dynamic decision reducts [7], DAARs [8], decision bireducts [9], and many others. The key objective of most of them was to prevent the inclusion of illusionary dependencies between attributes and decision values to the reducts. A lot of research was also committed to the problem of algorithms for the efficient computation of diverse reduct sets [10]. This topic is particularly important from the perspective of practical applications of the rough set theory [11].

In this tutorial, we focus on the latter aspect of the decision reduct-related research. We discuss various, both, well-known and relatively new algorithms, and consider their specific advantages. We explain in detail selected implementation aspects that are crucial for the efficient computation of many types of decision reducts. We also overview and demonstrate libraries in popular programming languages that allow easy computation of reducts on real-world datasets, including *RoughSets* library for R [12] and a novel Python language library *scikit-rough*[1] [11]. Finally, we share the results of a study aiming at the comparison of the computational efficiency of various reduct algorithms.

References

[1] Z. Pawlak and A. Skowron, "Rudiments of rough sets," *Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland*, vol. Information Sciences 177 (2007), pp. 3–27, 2006.

[2] M. Grzegorowski, "Governance of the redundancy in the feature selection based on rough sets' reducts," in *Rough Sets*, V. Flores, F. Gomide, A. Janusz, C. Meneses, D. Miao, G. Peters, D. Ślęzak, G. Wang, R. Weber, and Y. Yao, Eds. Cham: Springer International Publishing, 2016, pp. 548–557.

[3] J.-H. Zhai, X.-Z. Wang, and H.-C. Wang, "Dynamic ensemble of rough set reducts for data classification," in *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings*. Berlin, Heidelberg: Springer-Verlag, 2023, p. 642–649. [Online]. Available: https://doi.org/10.1007/978-3-319-11740-9_59

[4] A. Janusz and S. Stawicki, "Applications of approximate reducts to the feature selection problem," in *Rough Sets and Knowledge Technology - 6th International Conference, RSKT 2011, Banff, Canada, October 9-12, 2011. Proceedings*, ser. Lecture Notes in Computer Science, J. Yao, S. Ramanna, G. Wang, and Z. Suraj, Eds., vol. 6954. Springer, 2011, pp. 45–50. [Online]. Available: https://doi.org/10.1007/978-3-642-24425-4_8

[5] A. Skowron and S. Dutta, "Rough sets: Past, present, and future," *Natural Computing: An International Journal*, vol. 17, no. 4, p. 855–876, dec 2018. [Online]. Available: https://doi.org/10.1007/s11047-018-9700-3

[6] H. S. Nguyen and D. Ślęzak, "Approximate reducts and association rules," in *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, N. Zhong, A. Skowron, and S. Ohsuga, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 137–145.

[7] J. G. Bazan, A. Skowron, and P. Synak, "Dynamic reducts as a tool for extracting laws from decisions tables," in *Methodologies for Intelligent Systems*, Z. W. Raś and M. Zemankova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 346–355.

[8] A. Janusz and D. Ślęzak, "Computation of approximate reducts with dynamically adjusted approximation threshold," in *Foundations of Intelligent Systems - 22nd International Symposium, ISMIS 2015, Lyon, France, October 21-23, 2015, Proceedings*, ser. Lecture Notes in Computer Science, F. Esposito, O. Pivert, M. Hacid, Z. W. Ras, and S. Ferilli, Eds., vol. 9384. Springer, 2015, pp. 19–28. [Online]. Available: https://doi.org/10.1007/978-3-319-25252-0_3

[9] S. Stawicki, D. Ślęzak, A. Janusz, and S. Widz, "Decision bireducts and decision reducts - a comparison," *Int. J. Approx. Reason.*, vol. 84, pp. 75–109, 2017. [Online]. Available: https://doi.org/10.1016/j.ijar.2017.02.007

[10] A. Janusz and D. Ślęzak, "Rough set methods for attribute clustering and selection," *Appl. Artif. Intell.*, vol. 28, no. 3, pp. 220–242, 2014. [Online]. Available: https://doi.org/10.1080/08839514.2014.883902

[11] A. Janusz, D. Ślęzak, S. Stawicki, and K. Stencel, "A practical study of methods for deriving insightful attribute importance rankings using decision bireducts," *Inf. Sci.*, vol. 645, p. 119354, 2023. [Online]. Available: https://doi.org/10.1016/j.ins.2023.119354

[12] L. S. Riza, A. Janusz, C. Bergmeir, C. Cornelis, F. Herrera, D. Ślęzak, and J. M. Benítez, "Implementing algorithms of rough set theory and fuzzy rough set theory in the R package "roughsets"," *Inf. Sci.*, vol. 287, pp. 68–89, 2014. [Online]. Available: https://doi.org/10.1016/j.ins.2014.07.029

[1] https://github.com/sebov/scikit-rough

DOI: 10.15439/2023F4851
ISSN 2300-5963 ACSIS, Vol. 35

Type 1 Diabetes Mellitus Saudi Patients' Perspective on the Adopting IoT-Enabled CGM: Validation of Critical Factors in the IAI-CGM A Framework

Hamad Almansour
The Applied College
Najran University
University of Sussex
Informatics Department
Brighton, UK
Email: ha432@sussex.ac.uk

Natalia Beloff
University of Sussex
Informatics Department
Brighton, UK
Email: N.Beloff@sussex.ac.uk

Martin White
University of Sussex
Informatics Department
Brighton, UK
Email: M.White@sussex.ac.uk

***Abstract*—The increasing prevalence of diabetes, particularly in Saudi Arabia, calls for effective self-management tools to monitor blood sugar levels, such as Continuous Glucose Monitors. These are medical devices that can be used to track the glucose levels of people without a fingerstick blood sample. However, the adoption of IoT-enabled Continuous Glucose Monitors (IoT-CGM) can be challenging due to the use of new technology. This study proposes the Intention to Adopt IoT-enabled Continuous Glucose Monitors (IAI-CGM) a framework, which incorporates practical, technological, and user behaviour considerations based on the Technology Acceptance Model (TAM). The study defines 8 hypotheses that are analysed using structural equation modelling. Data was collected; from 873 type 1 diabetes patients (T1DM) from Saudi Arabia. The model predicts the significant impact of all factors on adoption intent except technology -related self-efficacy (TRSE), enabling the assessment of Saudi T1DM patients for IoT-CGM readiness. Furthermore, the framework's novelty may serve as inspiration for developing comparable frameworks for wearable or attached health monitoring devices in patients with other illnesses and in other geographical locations.**

***Index Terms*—IAI-CGM, TIDM, Internet of Things, Continuous Glucose Monitors, Adoption Intention, Technology Acceptance Model.**

I. INTRODUCTION

THE RISING prevalence of diabetes in Saudi Arabia can be attributed to the Westernized food and sedentary lifestyle prevalent in the country. A quarter of Saudi adults are expected to develop diabetes by 2030, with over half of Saudis over the age of 30 having diabetes or being at risk [1]. The solution to this issue lies in patients taking control of their healthcare and self-managing their blood sugar levels [2]. However, patients often lack the ability to follow physicians' advice on self-care management, necessitating the use of sophisticated technology. The emergence of innovative wearable technology has created an opportunity for the creation of a wireless body area network that could monitor healthcare delivery, including diabetes self-management [3]. To standardize IoT-enabled Continuous Glucose Monitoring (CGM) devices in Saudi Arabian primary care institutions, the country must undertake an in-depth analysis and publication on IoT in its healthcare system.

This study aims to measure patient preparation and willingness to use a CGM in primary diabetes care settings in Saudi Arabia to promote patient autonomy. The research will involve conducting a literature review to determine the extent to which intelligent technologies have permeated the Saudi healthcare system. It will also use the Technology Acceptance Model (TAM) to measure patient autonomy readiness towards IoT adoption by surveying people with type-1 diabetes. The study's scope and objectives include a quantitative analysis of diabetic patients in Saudi Arabia regarding their preparedness for autonomy. With resource constraints and significant health management risks facing emerging nations, research in this field is vital. In addition, the study aims to fill a research gap by examining Saudi Arabia's digital transformation efforts and readiness for IoT in healthcare [4]. The study's results will contribute to advancing IoT-enabled CGM adoption and primary diabetes self-management in Saudi Arabia, helping the country meet its digital transformation goals by 2030 [5].

II. THEORETICAL BACKGROUND

A. *Role of CGM in T1DM Primary Care*

Continuous Glucose Monitoring (CGM) is crucial for primary care in managing Type 1 diabetes. It enables patients to respond to their readings and maintain safe blood sugar levels, leading to increased self-management care [6]. CGMs provide continual feedback on blood sugar levels, encouraging patients to take their medication as recommended, thereby improving glucose levels.

However, in Saudi Arabia, poor drug compliance is a significant issue, with inadequate education and urban lifestyles being the main factors [7].

To improve technology tolerance among low-income and urban patients, IoT-CGM deployment requires patient health literacy improvement [8]. When primary care institutions provide self-management instruction and follow-up, it can help reduce patients' blood glucose levels. Self-management increases patients' health-related motivation, autonomy, and competence, but its effectiveness requires educating patients about their responsibilities in controlling T1DM. Encouraging patient autonomy helps doctors regulate blood glucose levels. This implies that with the help of health education provided to the patients could increase the sense of responsibilities to control T1DM disease themselves with self-empowerment [9]. The study concludes that IoT-CGMs might improve patient autonomy, but their adoption is uncertain, and adoption issues need to be addressed [10].

B. *IoT-CGM Adoption Concerns*

The use of Internet of Things continuous glucose monitoring (IoT-CGM) devices has been shown to provide numerous benefits to diabetics, such as improved long-term complication management [11]. However, despite the advancements in accuracy and dependability, these devices have not been widely adopted, with most diabetics still relying on the traditional method of drawing blood from a fingertip to measure their blood glucose levels. The slow uptake of IoT-CGM can be attributed to various factors, such as initial concerns about the devices' lack of accuracy [12], patients' unawareness of the advantages they offer, and sociopsychological and economic barriers to adoption.

Studies have shown that proper use of CGM devices, advice from medical professionals, and acting on CGM alarms can significantly reduce long-term difficulties in diabetic patients [13], [14]. The study by Gia et al. [15] provided evidence that proper guidance from health professional can help to reduce the long-term complications from 75 percent to 40 percent in the accuracy of CGM.

However, without a technology acceptance model (TAM) study, it is difficult to determine whether patients nationwide will adhere to the recommendations and continue to benefit from long-term complication management. Furthermore, wearable smart gadgets, including IoT-CGM, are still in their infancy, and persuading developing nations to adopt them will require work.

However, according to Rodbard [16], still there are only 8 percent to 17 percent of T1DM patients using CGM devices. Although, it is evidenced by the study [17], that the overall size of IoT healthcare sector expected to reach up as a 2 trillion US dollar industry by 2025. Despite this prediction fancy prediction still the adoption rate is very low.

Another study by Ayanlande et al. [18] elaborates that acceptability of CMG devices is also hinge on the patient's socio-psychological aspect, along with the affordability of patients and the healthcare system of the region where they reside. Another study provides that the availability is dependent on the scope of the service to meet minimum requirements [19]. Furthermore, another study [20], discussed the development of HIT enabled patient care that empowers patient and provides them with self-management capabilities, with the end goal that health matters such as diabetes patient care does not go poorly managed.

Saudi Arabia, a wealthy nation with an unequal income distribution, has only recently started using IoT-CGM. Its patient population is diverse, including those who have already adopted IoT-CGM, those who have heard of it but are hesitant to use it, and those who have never heard of it. Therefore, research on IoT-CGM adoption in Saudi Arabia includes many controversies.

To address the slow uptake of IoT-CGM, a new paradigm for adoption using the theoretical foundations of the technology acceptance model is proposed. The TAM considers factors such as sociopsychological, economic, and healthcare infrastructure determinants of acceptance, affordability, accessibility, and satisfaction with the service's breadth. However, there is a dearth of knowledge on the usage of healthcare technology that is unique to type 1 diabetes mellitus (T1DM), which is a health issue that should not be handled carelessly [21].

Making IoT-CGM more widely used in Saudi Arabia and other nations is a major issue, and the first stage in achieving the criteria set for improving healthcare quality in Saudi Arabia is understanding the viability of adoption. The literature on continuous glucose monitoring for diabetes with the Internet of Things is voluminous, but most studies have focused on the advantages of self-management, its effectiveness, and an analysis of the variables that influence the slow uptake of medical technology.

Therefore, determining the adoption characteristics of T1DM patients and providing a framework for adoption specific to them constitute the initial stages. Wearable smart devices offer the most appropriate solutions, but patients are not required to utilize them, and health factors and technological adoption scales are not always taken into consideration. Thus, providing a framework for IoT-CGM adoption that is specific to T1DM patients is crucial. The proposed structure provides theoretical underpinnings for this framework and a clear picture of a nation's present adaptability.

The authors proposed the Intention to Adopt IoT-enabled Continuous Glucose Monitors (IAI-CGM) framework in [4]. Based on this framework, the study aims to identify factors affecting adoption to analyse the readiness and willingness of primary diabetes T1DM patients, particularly in Saudi Arabia, to use (IoT-CGM) technology. The framework has strong predictive capabilities for assessing the adoption of Internet of Things-enabled Continuous Glucose Monitors (IoT-CGMs) used for monitoring blood glucose levels. The factors of the proposed framework include Perceived Reliability (PR), Perceived Usefulness (PU), Ease of Use (EU), Information Overload (IO), Technology Related Self-Efficacy (TRSE), Attitude (AT), Intention to Use (IT), and

Visibility of Body Change (VBC) as designed by Borges and Kubiak's [22].

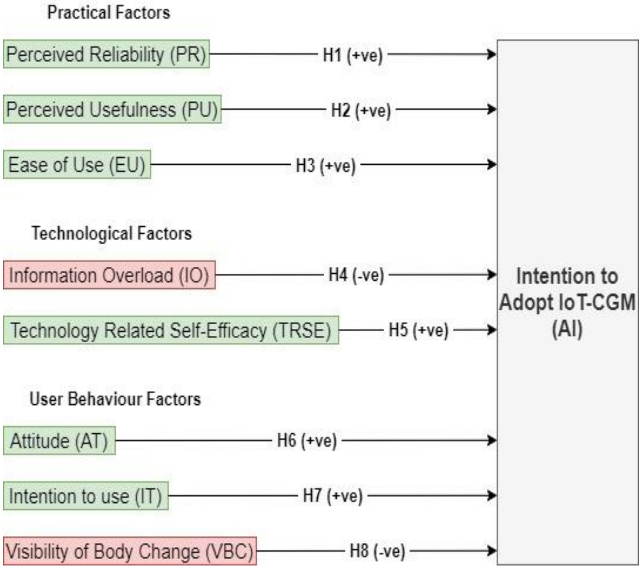

Practical Factors
Perceived Reliability (PR)
H1 (+ve)
Perceived Usefulness (PU)
H2 (+ve)
Ease of Use (EU)
H3 (+ve)
Technological Factors
Information Overload (IO)
H4 (-ve)
Technology Related Self-Efficacy (TRSE)
H5 (+ve)
User Behaviour Factors
Attitude (AT)
H6 (+ve)
Intention to use (IT)
H7 (+ve)
Visibility of Body Change (VBC)
H8 (-ve)
Intention to Adopt IoT-CGM (AI)

Fig 1: Proposed IAI-CGM Framework [4]

II. RESEARCH METHODOLOGY

This study aims to investigate the factors that influence the adoption of IoT-CGM devices by T1DM patients in Saudi Arabia. The study used an online survey to collect quantitative data from participants. Qualtrics was used for survey creation and administration, while Microsoft OneDrive was utilized for data storage. In addition, IBM SPSS was used for data analysis.

The survey questions were grounded in the theoretical framework of the study, which identified usability, technology, and user behaviour as the primary drivers of adoption intention. The study sample size was calculated based on a confidence level of 95% and a confidence interval of 5%; the total population is 11,662 of T1DM, and based on the statistics provided by [2], at least 372 responses we are required. Therefore, an accurate total sample of 873 was collected from the T1DM patients from King Khalid Hospital Saudi Arabia and Najran region.

There are three categories of human factors that influence patients' decisions to embrace continuous glucose monitoring devices that are enabled by the Internet of Things: those that are purely theoretical, those that pertain to technology, and those that pertain to user behaviour. Perceived reliability, perceived utility, and simplicity of use are three of the metrics used to assess the practical factors of a product's performance. The technological factors are measured by the innovation orientation and the technology-related self-efficacy measures, while the user behavioural factors are measured by the attitude, intention to use, and the visibility of body change factors.

The study took ethical measures to protect the interests of all those involved, making sure participants knew that taking the survey was completely optional and that they could stop at any time without consequences or explanations. Participants were also made aware that their data would not be sold or otherwise distributed to outside parties and were assured complete privacy. The study did not seek individual information in the questionnaires, and participants were instructed not to provide any identifying information in the survey. In the case of a participant's comment warranting a quotation, a unique numeric identity would be issued instead. Data collection methods and research instruments were approved by Institutional Review Board (IRB) in Saudi Arabia, Ministry of Health; King Khaled Najran Hospital (KKNH) by assigning IRB registration number with KACST, KSA: H-11-N-081; IRB Log Number 2022-01E.

The survey questionnaire was developed based on the findings of a previous study [22]. The theoretical framework of this study provides the foundation for the survey. The survey consists of questions that allow for assigning values to each variable. This is done using a 5-point Likert scale system, where 1 indicates "Strongly disagree", and 5 indicates "strongly agree".

IV. RESULTS

In a quantitative study, selecting an appropriate sample size is critical to ensure reliable and transferable results while eliminating bias. The sampling strategy employed in this study followed scientific principles and focused on patients with Type 1 diabetes at King Khaled Najran Hospital and the Najran region in Saudi Arabia. The survey was administered online using Qualtrics, with data immediately entered into a database. Outliers are exceptional data points that fall outside of the norm and can result from typos, poor wording of questionnaires, or data input errors. Univariate outliers stand out in only one way, while multivariate outliers have multiple scores that deviate from the norm [23]. The frequency distribution test and other normality tests can help detect outliers [24]. The research variables in this study had z-scores of less than 3.29 and a standard deviation range of 0.708 to 1.302, indicating no extreme values among the outliers.

A. *Data Analysis and Confirmatory Factor Analysis (CFA)*

Confirming the theoretical structure of variables is a crucial part of CFA analysis. It involves conducting one-dimensionality, reliability, divergent, and convergent validity tests. Researchers can determine whether to adopt a theory based on these results. To establish relationships in SEM, researchers should perform CFA on all latent variables, with a minimum latent concept loading of 0.60 for the assessment items measured by one-dimensionality [25]. CFA findings are founded on nine different latent constructs. A few examples of these constructs are technology-related self-efficacy (TRSE), information overload (IO), perceived reliability (PR), perceived usefulness (PU), ease of use (EU), attitude (AT), intention to use (IT), visibility of body change (VBC), and intention to adopt (AI).

TABLE I
RELIABILITY AND CONSTRUCT VALIDITY

Constructs	Cronbach (above 0.7)	CR	AVE	MSV
TRSE	.756	0.763	0.521	0.274
IO	.772	0.776	0.537	0.383
PR	.842	0.844	0.643	0.127
PU	.821	0.848	0.652	0.475
EU	.821	0.835	0.629	0.241
AT	.773	0.785	0.553	0.441
IT	.786	0.790	0.557	0.475
VBC	.856	.0.868	0.623	0.397

B. *Measures of the Model Validity*

To ensure construct validity and composite reliability, Cronbach's alpha and Composite Reliability (C.R.) values were used before hypothesis testing. The present study used Cronbach's alpha with a cut-off value of 0.70, and another study suggests cut-off values of 0.60 for C.R. and 0.70 for Cronbach's alpha [26]. The reliability scores in Table I are greater than 0.70. Both Cronbach's alpha and C.R. were used to examine internal consistency, showing the convergent and discriminant validity and reliability of findings and comparing it with cut-off-points [27]. The results indicate that all variables are free from measurement error. Assessment for consistency refers to measurements on the same point on two different scales, assuming that an instrument can assess a similar context over time. Cronbach's alpha is considered the initial step to ensure reliability. The present study uses Cronbach's alpha and Composite reliability to test instrument reliability. The study used average variance extracted (AVE) to identify variation in latent variables caused by random measurement errors, with a cut-off value of 0.50 or greater. AVE values were between 0.537 to 0.644. Discriminant validity was achieved through the maximum shared squared variance (MSV), Fornell-Larcker test, and average shared squared variance (ASV), with AVE for each construct higher than MSV [28]. The CFA assessment of the model showed that both convergent and discriminant validity met the fitness criteria.

TABLE II
DESCRIPTIVE ANALYSIS AND FACTORS LOADINGS FOR ITEMS

Measure	Chi-square (CMIN) / Degrees of freedom (D.F.)	Comparative fit index (CFI)	Standardized root mean square residual (SRMR)	Root mean square error of approximation (RMSEA)	Normed Fit Index (NFI)
Estimate	4.380	0.913	0.054	0.062	0.901
Threshold	Between 1 and 5	> 0.90	< 0.08	< 0.08	> 0.90
Interpretation	Good fit	Good fit	Good fit	Good fit	Good fit

C. Model Good Fit

The study involved 873 participants, and the observed results for the fitness of the model indicated that all observed statistical values were under the cut-off threshold as given (CMIN/DF = 4.380, CFI = 0.913, SRMR = 0.054, RMSEA = 0.062, NFI = 0.901). Therefore, no issue was observed about the good fit of the model. The results are shown in Table II.

D. Discriminant Validity

Table III illustrated that there were no problems regarding discriminant validity, as the square roots of the AVEs' "diagonal line" showed values greater than the values below the diagonal, as demonstrated below. The findings suggest that the CFA model's evaluation was valid with respect to both convergent and discriminant validity criteria.

TABLE III
DISCRIMINANT VALIDITY

	TRSE	IO	PR	PU	EU	AT	IT	VBC	AI
TRSE	**0.722**								
IO	-0.249	**0.733**							
PR	-0.066	0.356	**0.802**						
PU	0.523	-0.294	-0.034	**0.807**					
EU	0.270	-0.400	-0.230	0.375	**0.793**				
AT	0.430	-0.274	0.023	0.664	0.377	**0.744**			
IT	0.429	-0.431	-0.062	0.689	0.458	0.617	**0.747**		
VBC	-0.350	0.619	0.313	-0.494	-0.491	-0.428	-0.610	**0.789**	
AI	0.421	-0.584	0.059	0.713	0.484	0.669	0.725	-0.630	**0.725**

E. Structural Equation Model

In order to test the hypotheses of the study Structural Equation Modelling (SEM) was used to investigate the interrelationship between latent and observed variables via the software AMOS version 28. The SEM approach is widely used in various research fields, such as psychology, behavioural studies and education. In order to ensure the accuracy of the data, a confirmatory factor analysis was conducted to assess the validity of the collected data. The SEM model includes both structural and measurement models. Therefore, the AVE score in the convergent validity test should be greater than 0.5, indicating a good model fit [27].

All factors caused a significant change in the dependent factors except hypothesis 5, where TRSE does not have a significant effect on the intention to adopt IoT-CGM. The present study discusses critical factors related to the intention to adopt IoT-CGM.

Results clearly indicate significant differences from previous findings, which may be attributable to regional or cultural differences in corporate climate or at the individual level. After getting empirical evidence based on structural equation modelling, it was observed in Table IV that Perceived Reliability (PR) has a significant positive influence on intention to adopt IoT-CGM (β = 0.154 along with p-value < 0.001). Similarly, perceived usefulness (PU) has a significant positive influence on the intention to adopt IoT-CGM (β = 0.251 along with p-value < 0.001). Next, it was observed that Ease of Use (EU) has a significant positive influence on intention to adopt IoT-CGM (β = 0.077 along with p-value < 0.001). Similarly, information overload (IO) has a significant negative influence on the intention to adopt

TABLE IV
REGRESSION WEIGHTS FOR PATH COEFFICIENTS AND ITS SIGNIFICANCE

Structural Relation			**Regression Weight**	**Standard Error (S.E.)**	**Critical ratio (C.R.)**	**P value**	**Result**
AI	<---	TRSE	-0.017	0.035	-0.484	> 0.629	Rejected
AI	<---	IO	-0.128	0.032	-3.986	< .001***	Supported
AI	<---	PR	0.154	0.024	6.433	< .001***	Supported
AI	<---	PU	0.251	0.050	5.045	< .001***	Supported
AI	<---	EU	0.077	0.028	2.737	< 0.006**	Supported
AI	<---	AT	0.194	0.049	3.929	< .001***	Supported
AI	<---	VBC	-0.153	0.036	-4.209	< .001***	Supported
AI	<---	IT	0.171	0.058	2.928	<0.003**	Supported

IoT-CGM (β = -0.128, along with p < 0.001). On the one hand, Technology Related Self-Efficacy (TRSE) does not have a significant influence on the intention to adopt IoT-CGM (beta = -0.017 along with p-value > 0.05). Furthermore, attitude (AT) positively influences Adoption Intention (AI), β value =.194 along with p < 0.05). Next, results show that intention to use (IT) positively influences intention to adopt IoT-CGM, with β value = 0.171 along with p < 0.05). Results further indicate that visibility of body change (VBC) has a significant negative impact on intention to adopt IoT-CGM along with β = -0.153 and p-value < 0.05. Based on overall results, it is observed that all independent variables cause a significant change in intention to adopt IoT-CGM except TRSE, as it did not cause a significant change in the intention to adopt IoT-CGM. One reason could be the mistrust that makes people reject IoT-CGM. So, the intention to not use IoT-CGM is correlated with a lack of knowledge, little desire to learn and doubt related to the technology or its adoption.

F. Descriptive Statistics

Descriptive results Table V shows that in the survey, there were 440 Male (50.4%), and 433 females (49.6%). 36.7% of participants were within the age group of 18-25, 37.1 % of participants were within the age group 26-35, participants from the age group of 36 to 45 were 22.0 %; and participants from the age group of 46 to 60 were only 3.9 %, and there were only 3 participants or (0.30 %) older than 60. The majority, 43.5%, were reported to have a bachelor's degree, 29.1% had secondary school education and Ph.D. degree holders only accounted for 0.5%. Furthermore, only 2.1% had only completed primary school, while 18.3% had diplomas, 2.7% had master's degrees, and 3.8% had only completed elementary school education.

TABLE V
SAMPLE CHARACTERISTICS (N = 873)

		Frequency	**Percentage**			**Frequency**	**Percentage**
Gender	**Male**	440	50.4%	**Education level**	**Primary School**	18	2.1%
	Female	433	49.6%		**Elementary School**	33	3.8%
Ages	**18-25**	320	36.7%		**Secondary school or less**	254	29.1%
	26-35	324	37.1%		**Diploma**	160	18.3%
	36-45	192	22.0%		**Bachelor degree**	380	43.5%
	46-60	34	3.9%		**Master degree**	24	2.7%
	60+	3	0.3%		**Doctorate degree**	4	0.5%

V. DISCUSSION

The study provided an updated framework based on empirical findings on Internet of Things-enabled Continuous Glucose Monitoring (IoT-CGM), which empowers type 1 diabetics. The results from Table IV. The approach illustrated how several factors affect patients' inclinations to use internet-enabled continuous glucose monitoring. The dimensions of human factors that affect the patients' intentions to adopt IoT-enabled continuous glucose monitoring devices are grouped into three factors.

Practical factors: These are the first set of factors that influences the adoption of IoT-CGM.

The study found that perceived reliability was a significant factor that affected the adoption of IoT-CGM [29]. Similarly, empirical results from the present study show that perceived reliability (PR) has a significant positive influence on intention to adopt IoT-CGM along with β = 0.154 and p-value < 0.001. Therefore, there are 0.154 units of positive change in IAI-CGM when PR changes by 1 unit. Another study revealed that customers' feelings regarding new technology were positively correlated with their faith in the gadget's accuracy [30]. Trust was found to be the most crucial attribute while communicating with doctors.

The perceived usefulness of IoT-CGMs improves behavioural intention. User satisfaction with continuous glucose monitors (CGMs) may be affected by aspects, including the availability of trend and graph glucose readings and the ability of CGMs to compensate automatically for glucose level swings in real-time [31]. The present study also observed that perceived usefulness (PU) is a significant factor that positively influences intention to adopt IoT-CGM along β = 0.251 and p-value < 0.001. Furthermore, it shows 0.251 units change in IAI-CGM due to PU.

Ease of use (EU) is an essential practical factor in the practical factor when adopting a product. Furthermore, the simplicity of use is a critical factor that influences early computer adopters' behavioural intentions [31]. Similarly, the present study also observes that Ease of Use (EU) has a significant positive influence on IAI-CGM with β = 0.077 along with a p-value < 0.001. However, there is a positive influence of the EU on IAI-CGM.

Technological factors: These are the second set of factors that affects the adoption of IoT-CGM.

Information overload is one of the critical technological factors. Tansey et al. [31] have suggested that an excessive number of features in CGMs can lead to information overload for users. Although real-time glucose readings in IoT-CGMs may be an attractive feature but a constant influx of information can also be burdensome for users, potentially resulting in a negative response to the technology. Similarly, it was observed in the empirical results that information overload (IO) has a significant negative influence on intention to adopt IoT-CGM along with β = -0.128 and $p < 0.001$. However, the presence of an overload feature in IAI-CGM can have a negative impact on user adoption, as users may become overwhelmed by the excess information provided.

Technological self-efficacy [4] is characterized as type 1 diabetes patients' perception of their capability to utilize IoT-enabled continuous glucose monitors and their ability to trust new technology. Apart from expectations and the ability to trust, there could also be the issue of technology and the lack of knowledge and know-how to use the technology, which might be the reason for the low adoption rate causing the insignificant relationship of TRSE with the intention to adopt [32] Similarly, results show that Technology Related Self-Efficacy (TRSE) does not have a significant influence on the intention to adopt IoT-CGM beta = - 0.017 along with p-value > 0.05.

User behavioural factors: These are the third set of factors that affects the adoption of IoT-CGM.

Users' views on technology are measured in this factor. Technology perception is also reflected in consumers' mental processes, which shows intention to use [33]. Empirical results also show that intention to use positively influences IAI-CGM, β value =.171, along with $p < 0.05$. Patients' perspectives of how others view them also influences their technology choices. The patients' scheduled activities were based on their attitudes and personal values [34]. A person's behavioural intention also depend on how much they value their own attitudes and the societal norms around them. Similarly, empirical results show that attitude (AT) positively influences Adoption Intention (AI), β value =.194 along with $p < 0.05$. Furthermore, visibility of body change (VBC) is found to have negative influence on the Adoption Intention (AI), β value =-.153 along with $p < 0.05$.

The present study discusses critical factors related to intention to adopt IoT-CGM. Empirical findings with some differences may be attributable to regional or cultural perspectives. However, adoption of IAI-CGM framework may subject to cultural and individual differences that need to be addressed before adopting this framework.

The Updated framework that is presented below In Fig.2, that can be used to determine the adoption intention of IoT-CGMs and has been termed the Intention to Adopt IoT-enabled CGM (IAI-CGM) framework. This framework has been proposed to determine the adoption intention based on the practical factors of using IoT-CGMs, the technological factors of using IoT-CGMs, and factors regarding user behavioural factors. PR, PU, and EU are all practical factors that increases the adoption intention among users. IO and TRSE are technological factors. IO decreases adoption intention, and TRSE has an insignificant effect on adoption intention. User behavioural factors are AT, IT, and VBC. AT and IT factors increase adoption intention, while VBC decreases adoption intention.

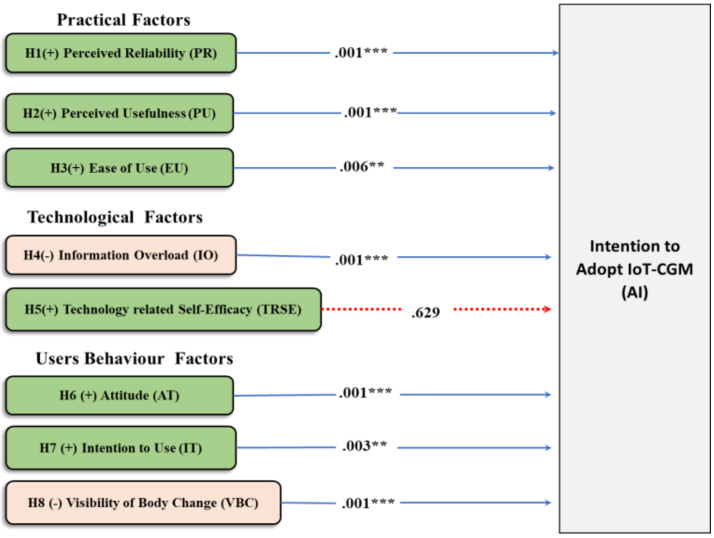

Practical Factors
H1(+) Perceived Reliability (PR)
.001***
H2(+) Perceived Usefulness (PU)
.001***
H3(+) Ease of Use (EU)
.006**
Technological Factors
H4(-) Information Overload (IO)
.001***
H5(+) Technology related Self-Efficacy (TRSE)
.629
Users Behaviour Factors
H6 (+) Attitude (AT)
.001***
H7 (+) Intention to Use (IT)
.003**
H8 (-) Visibility of Body Change (VBC)
.001***
Intention to Adopt IoT-CGM (AI)

Fig2: The Updated IAI-CGM Framework

A. *Implications*

This study analyzed the factors affecting the adoption of Internet of things-enabled continuous glucose monitors among Saudi users. The authors used an integrated research methodology that incorporated the TAM and other factors to examine the perceived reliability and utility of the monitors, as well as user behavioural and technological factors (see Fig 2 The Updated IAI-CGM Framework). The study found that an individual's perceived reliability of continuous glucose monitoring and its utility were the most important factors affecting adoption. Physical changes in the body negatively impacted adoption, and technology-related self-efficacy did not affect adoption intention. The study also revealed that Saudi Arabians were skeptical of government hospitals and their ability to train them in wearable technologies. This study provides valuable insights into the adoption of IoT-enabled continuous glucose monitors and can aid in future research in this area.

VI. CONCLUSION

In this study, we constructed the Updated framework (IAI-CGM) to identify critical factors that influence the adoption of (IoT-CGM). We present the theoretical background by way of a literature review that outlines previous empirical findings related to type 1 diabetic patients using IoT-CGM from King Khalid Hospital Saudi Arabia and the Najran region. Then, our updated framework (IAI-CGM) is based on investigating three main categories of factors, including practical factors, technological factors, and user behavioural factors.

Among the practical factors, PR, PU, and EU are identified as drivers that increase adoption intention among users. However, with the technological factors, only IO influenced adoption intention, while TRSE was found to have an insignificant impact on adoption intention. Finally, user behavioural factors AT, IT, and VBC are also relevant. Both AT and IT increase adoption intention, while VBC has the opposite effect. Results are intended to provide valuable insight into the main factors that influence type 1 diabetic patients to adopt IoT-CGM in Saudi Arabia.

The study was conducted on Saudi Arabian citizens with type 1 diabetes, and the findings revealed new avenues for future research. The study recommends that future research should focus on specific areas, such as software development models and process structure models, to better understand the factors that contribute to acceptance and the hurdles that must be overcome. This research emphasizes the importance of addressing adoption challenges to improve healthcare delivery.

The future work will involve testing the IAI-CGM framework using qualitative approach based on 15 semi-structured interviews. The study will be conducted on T1DM patients admitted in diabetes primary care stage. The patients will be recruited from King Khalid Hospital, located in Najran, Saudi Arabia. Based on the qualitative results, the model will have the potential to be further improved. Qualitative results will help to identify other critical factors for adoption intention (AI) for IAI-CGM framework.

ACKNOWLEDGEMENT

The authors extend their appreciation to the Directorate General of Health Affairs in Najran. We would like to express our gratitude to Najran Research Department, King Khalid Hospital Najran and Diabetes and Endocrine Centre. Finally, we would like to express our heartfelt appreciation to all participants for their time and effort.

REFERENCES

[1] K. Al-Rubeaan *et al.*, “Epidemiology of abnormal glucose metabolism in a country facing its epidemic: <scp>SAUDI-DM</scp> study,” *J. Diabetes*, vol. 7, no. 5, Sep. 2015, doi: 10.1111/1753-0407.12224.

[2] M. A. Al Dawish and A. A. Robert, “Diabetes Mellitus in Saudi Arabia: Challenges and possible Solutions,” in *Handbook of Healthcare in the Arab World* , Cham: Springer Nature Switzerland, 2019, pp. 1–18.

[3] A. Solanas *et al.*, “Smart health: A context-aware health paradigm within smart cities,” *IEEE Commun. Mag.*, vol. 52, no. 8, Aug. 2014, doi: 10.1109/MCOM.2014.6871673.

[4] H. Almansour, N. Beloff, and M. White, “IAI-CGM: A Framework for Intention to Adopt IoT-Enabled Continuous Glucose Monitors,” 2023, pp. 637–660. doi: 10.1007/978-3-031-16072-1_46.

[5] SaudiVision2030, “National Transformation Program Delivery Plan 2018-2020,” *Vision 2030*, 2019. https://vision2030.gov.sa/sites/default/files/attachments/NTP English Public Document_2810.pdf (accessed May 12, 2021).

[6] M. K. Rhee *et al.*, “Patient adherence improves glycemic control,” *Diabetes Educ.*, vol. 31, no. 2, pp. 240–250, Mar. 2005, doi: 10.1177/0145721705274927.

[7] A. Khan, Z. Al-Abdul Lateef, M. Al Aithan, M. Bu-Khamseen, I. Al Ibrahim, and S. Khan, “Factors contributing to non-compliance among diabetics attending primary health centers in the Al Hasa district of Saudi Arabia,” *J. Fam. Community Med.*, vol. 19, no. 1, p. 26, 2012, doi: 10.4103/2230-8229.94008.

[8] S. L. Norris, J. Lau, S. J. Smith, C. H. Schmid, and M. M. Engelgau, “Self-management education for adults with type 2 diabetes. A meta-analysis of the effect on glycemic control,” *Diabetes Care*, vol. 25, no. 7, pp. 1159–1171, Jul. 2002, doi: 10.2337/diacare.25.7.1159.

[9] M. Heisler, D. M. Smith, R. A. Hayward, S. L. Krein, and E. A. Kerr, “How well do patients’ assessments of their diabetes self-management correlate with actual glycemic control and receipt of recommended diabetes services?,” *Diabetes Care*, vol. 26, no. 3, pp. 738–743, Mar. 2003, doi: 10.2337/diacare.26.3.738.

[10] G. C. Williams, H. A. McGregor, A. Zeldman, Z. R. Freedman, and E. L. Deci, “Testing a Self-Determination Theory Process Model for Promoting Glycemic Control Through Diabetes Self-

Management," *Heal. Psychol.*, vol. 23, no. 1, pp. 58–66, Jan. 2004, doi: 10.1037/0278-6133.23.1.58.

[11] D. Olczuk and R. Priefer, "A history of continuous glucose monitors (CGMs) in self-monitoring of diabetes mellitus," *Diabetes and Metabolic Syndrome: Clinical Research and Reviews*, vol. 12, no. 2. Elsevier Ltd, pp. 181–187, Apr. 01, 2018. doi: 10.1016/j.dsx.2017.09.005.

[12] R. Ajjan, D. Slattery, and E. Wright, "Continuous Glucose Monitoring: A Brief Review for Primary Care Practitioners," *Adv. Ther.*, vol. 36, no. 3, Mar. 2019, doi: 10.1007/s12325-019-0870-x.

[13] H. S. Chang, S. C. Lee, and Y. G. Ji, "Wearable device adoption model with TAM and TTF," *Int. J. Mob. Commun.*, vol. 14, no. 5, p. 518, Jan. 2016, doi: 10.1504/IJMC.2016.078726.

[14] Y. J. Kim, K. R. Saviers, T. S. Fisher, and P. P. Irazoqui, "Continuous glucose monitoring with a flexible biosensor and wireless data acquisition system," *Sensors Actuators, B Chem.*, vol. 275, pp. 237–243, Dec. 2018, doi: 10.1016/j.snb.2018.08.028.

[15] T. N. Gia *et al.*, "IoT-based continuous glucose monitoring system: A feasibility study," in *Procedia Computer Science*, Jan. 2017, vol. 109, pp. 327–334. doi: 10.1016/j.procs.2017.05.359.

[16] D. Rodbard, "Continuous Glucose Monitoring: A Review of Successes, Challenges, and Opportunities," *Diabetes Technology and Therapeutics*, vol. 18, no. S2. Mary Ann Liebert Inc., pp. S23–S213, Feb. 01, 2016. doi: 10.1089/dia.2015.0417.

[17] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3. Chongqing University of Posts and Telecommunications, pp. 161–175, Aug. 01, 2018. doi: 10.1016/j.dcan.2017.10.002.

[18] O. S. Ayanlade, T. O. Oyebisi, and B. A. Kolawole, "Health Information Technology Acceptance Framework for diabetes management," *Heliyon*, vol. 5, no. 5, May 2019, doi: 10.1016/j.heliyon.2019.e01735.

[19] N. Davoody, S. Koch, I. Krakau, and M. Hägglund, "Post-discharge stroke patients' information needs as input to proposing patient-centred eHealth services," *BMC Med. Inform. Decis. Mak.*, vol. 16, no. 1, pp. 1–13, Jun. 2016, doi: 10.1186/s12911-016-0307-2.

[20] K. Gray and C. Gilbert, "Digital health research methods and tools: Suggestions and selected resources for researchers," in *Intelligent Systems Reference Library*, vol. 137, Springer Science and Business Media Deutschland GmbH, 2018, pp. 5–34. doi: 10.1007/978-3-319-67513-8_2.

[21] A. H. Krist, D. E. Nease, G. L. Kreps, L. Overholser, and M. McKenzie, "Engaging Patients in Primary and Specialty Care," in *Oncology Informatics*, Elsevier, 2016, pp. 55–79. doi: 10.1016/b978-0-12-802115-6.00004-5.

[22] U. Borges and T. Kubiak, "Continuous Glucose Monitoring in Type 1 Diabetes: Human Factors and Usage," *J. Diabetes Sci. Technol.*, vol. 10, no. 3, pp. 633–639, May 2016, doi: 10.1177/1932296816634736.

[23] G. Domino and M. L. Domino, *Psychological testing: An introduction*. Cambridge University Press, 2006.

[24] T. A. Brown, *Confirmatory factor analysis for applied research*. Guilford publications, 2015.

[25] Z. Awang, A. Afthanorhan, and M. A. M. Asri, "Parametric and non parametric approach in structural equation modeling (SEM): The application of bootstrapping," *Mod. Appl. Sci.*, vol. 9, no. 9, p. 58, 2015.

[26] J. A. Gliem and R. R. Gliem, "Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales," 2003.

[27] Hair J F, Black W C, Anderson R E, and Babin B J, *Multivariate data analysis: A global perspective*, 7th ed. Upper Saddle River NJ: Prentice Hall, 2009.

[28] J. F. Hair Jr, L. M. Matthews, R. L. Matthews, and M. Sarstedt, "PLS-SEM or CB-SEM: updated guidelines on which method to use," *Int. J. Multivar. Data Anal.*, vol. 1, no. 2, pp. 107–123, 2017.

[29] H. Yildirim and A. M. T. Ali-Eldin, "A model for predicting user intention to use wearable IoT devices at the workplace," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, no. 4, pp. 497–505, Oct. 2019, doi: 10.1016/j.jksuci.2018.03.001.

[30] N. Wang, H. Tian, S. Zhu, Y. Li, and L. Yuan, "Analysis of public acceptance of electric vehicle charging scheduling based on the technology acceptance model," *Energy*, vol. 258, 2022, doi: 10.1016/j.energy.2022.124804.

[31] M. Tansey *et al.*, "Satisfaction with continuous glucose monitoring in adults and youths with Type1 diabetes," *Diabet. Med.*, vol. 28, no. 9, pp. 1118–1122, Sep. 2011, doi: 10.1111/j.1464-5491.2011.03368.x.

[32] K. D. Barnard, K. K. Hood, J. Weissberg-Benchell, C. Aldred, N. Oliver, and L. Laffel, *Psychosocial Assessment of Artificial Pancreas (AP): Commentary and Review of Existing Measures and Their Applicability in AP Research*, vol. 17, no. 4. Mary Ann Liebert Inc., 2015, pp. 295–300. doi: 10.1089/dia.2014.0305.

[33] F. D. Davis and R. P. Bagozzi, "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Manage. Sci.*, vol. 35, no. 8, pp. 982–1003, 1989.

[34] M. Taylor and A. Taylor, "The technology life cycle: Conceptualization and managerial implications," *Int. J. Prod. Econ.*, vol. 140, no. 1, pp. 541–553, Nov. 2012, doi: 10.1016/j.ijpe.2012.07.006.

DOI: 10.15439/2023F5474
ISSN 2300-5963 ACSIS, Vol. 35

Defect Backlog Size Prediction for Open-Source Projects with the Autoregressive Moving Average and Exponential Smoothing Models

Paulina Anioła (Sielicka)
Email: paulina.aniola23@gmail.com

Sushant Kumar Pandey, Miroslaw Staron
0000-0003-1882-2435
0000-0002-9052-0864
Dept. of CSE Chalmers |
University of Gothenburg, Sweden

Mirosław Ochodek
0000-0002-9103-717X
Poznan University of Technology,
ul. Piotrowo 2, 60-695 Poznan, Poland
Email: miroslaw.ochodek@put.poznan.pl

***Abstract*—Context: predicting the number of defects in a defect backlog in a given time horizon can help allocate project resources and organize software development. Goal: to compare the accuracy of three defect backlog prediction methods in the context of large open-source (OSS) projects, i.e., ARIMA, Exponential Smoothing (ETS), and the state-of-the-art method developed at Ericsson AB (MS). Method: we perform a simulation study on a sample of 20 open-source projects to compare the prediction accuracy of the methods. Also, we use the Naïve prediction method as a baseline for sanity check. We use statistical inference tests and effect size coefficients to compare the prediction errors. Results: ARIMA, ETS, and MS were more accurate than the Naïve method. Also, the prediction errors were statistically lower for ETS than for MS (however, the effect size was negligible). Conclusions: ETS seems slightly more accurate than MS when predicting defect backlog size of OSS projects.**

I. Introduction

DEFECT backlog is the collection of all project defect reports that need to be handled. The size of this collection changes over time. The problem of monitoring defect backlogs is important in all modern software development organizations. In agile software development, it is important to correctly prioritize defects to continuously deliver business value. Also, especially in large organizations, the assignment of developers and testers to projects is often done dynamically, on demand. When a situation in a project demands more human resources for quality improvements, developers shift their focus from feature implementations to defect removal [2]. Because of these dynamic changes, knowing in advance that a project may require more human resources in the following week is valuable information for the managers, developers, testers, and other project stakeholders.

In particular, the managers need to know defect backlogs for the coming weeks. Therefore defect prediction models that forecast the number of defects that will need to be handled in a given time horizon are needed. There have been multiple studies on designing such models in industrial contexts [3], [4], [5], [6]. One of the successful studies on defect backlog prediction was conducted at Ericsson by Staron and Meding [3]. They proposed an autoregressive model (MS) that is based on the moving average and predicts defect backlog size within a weekly horizon. Although the MS model turned out to be very accurate at Ericsson, there are several other state-of-the-art autoregressive models for time series forecasting that have not been tried out for defect-backlog predictions. Two of them are Autoregressive integrated moving average (ARIMA) [7] and Exponential Smoothing (ETS) [8], [9], which are the two most widely used approaches to time series forecasting [10].

Although most of the previous studies on defect prediction were based on open-source software (OSS) datasets, the studies on defect backlog predictions in Ericsson have not been replicated in the OSS context. The flow of OSS projects differs from the flow of their industrial counterparts, however, they are often larger in terms of the number of involved contributors. Predicting the number of defects in a backlog is not easy due to uncertainties in identifying all the defects. The dynamic nature of software development, with its changing requirements and iterative cycles, adds to the complexity. Moreover, the accuracy of predictions can be affected by the quality and relevance of historical data used for analysis. Most existing methods rely on data from classical repositories like NASA and PROMISE, which may have limitations. Lastly, current predictive models may not consider all the contextual factors and unique project characteristics that impact defect discovery.

The goal of this study is to design defect-backlog prediction models based on the ARIMA and ETS methods and validate their accuracy in the context of large OSS projects. We use the state-of-the-art MS model developed at Ericsson [3] as a baseline for comparison since it has been reported as an accurate defect backlog prediction model validated in an industrial setting.

The structure of this paper is as follows. Section II provides a brief overview of the ARIMA and ETS methods, while Section III discusses the related work. Section IV describes the research methodology of our study. The results are presented

This work was supported by the Poznan University of Technology within the project 0311/SBAD/0738. Some of the paper's contents come from the corresponding author's master thesis [1].

and discussed in Section V. Finally, Section VI summarizes the main findings of our study.

II. Background

A. Defect Backlog

In many software projects, defects that need to be resolved are collected in defect backlogs. There are many defect-tracking tools on the market (e.g., Bugzilla, Jira, ReQtest). This kind of software helps the entire team and managers to get a view of how many defects remain in the software and what they are. To help developers work on the project, the defects in the backlog can be ordered according to priority. Often each issue includes additional information which differs between projects. If the software is regularly tested the size of the defect backlog changes over time. The number of defects that have been reported in a specific period of time is called *defect inflow*. Similarly, the number of defects that have been resolved in that period is referred to as *defect outflow*. The defect backlog size change within a given period of time (for the sake of this study, a week) is the difference between its inflow and outflow.

B. Autoregressive Integrated Moving Average

ARIMA stands for Autoregressive Integrated Moving Average. As the name suggests, it combines two time-series techniques, namely, the Autoregressive model and Moving Average.

ARIMA requires the time series to be stationary. The values of stationary time series do not depend on time. Thus, if we can see a trend or seasonality in time series, it means that it is non-stationary—its value depends on the time. Non-stationary time series have to be first transformed into stationary time series by using the differencing operation. The differenced time series is calculated as changes between subsequent observations [10]—see Equation 1. The differencing operation can be repeated multiple times if the obtained time series is still non-stationary.

$$y_t^{'} = y_t - y_{t-1} \tag{1}$$

where:
$y_t^{'}$ - value of the differenced series at time t,
y_t - value of the original series at time t,
y_{t-1} - value of the original series at time $t-1$.

The Autoregressive model is based on multiple linear regression. What distinguished it from other linear regression models is that it predicts the outcome variable (y) using past values as predictor variables (x). This approach assumes that there is some correlation between subsequent values in a time series (autocorrelation). The Autoregressive model is defined by Equation 2 [10].

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + \epsilon_t \tag{2}$$

where:
c - constant value,
ϕ - model parameter,
ϵ_t - error,
p - order of the model.

The p value in Equation 2 is called *the order* of the Autoregressive model. It determines how many past values will be considered to calculate the outcome. The autoregressive model of order p can be referred to as AR(p).

The Moving Average model calculates the outcome variable as a linear combination of past forecast errors. The formula of the model is presented in Equation 3 [10].

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} \tag{3}$$

where:
c - constant value,
θ - model parameter,
ϵ - forecast error,
q - order of the model.

It shows that the value of y_t can be considered as a weighted moving average of the past forecast errors. The model order value q determines how many past forecast errors will influence the outcome. The moving average model of order q can be referenced as MA(q).

The equation of a non-seasonal ARIMA model presented in Equation 4 shows that it combines components of autoregressive and moving average models, which are lagged values and lagged errors.

$$y_t^{'} = c + \phi_1 y_{t-1}^{'} + ... + \phi_p y_{t-p}^{'} + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q} + \epsilon_t \tag{4}$$

where:
$y_t^{'}$ - differenced series.
The outcome of the model is a differenced series. To get the actual predicted values time series need to be integrated. Integrating is the reverse of differencing. The transformation aims to add the trend or seasonality which were previously removed.

The non-seasonal ARIMA model is characterized by 3 parameters:

- p - order of autoregression part,
- d - degree of involved differencing,
- q - order of moving average part.

The model of some specific parameters can be referenced as ARIMA(p, d, q).

We use the ARIMA implementation provided by the R forecast package [11]. The function `auto.arima()` estimates the model parameters by analyzing the training data.

C. Exponential Smoothing

The general idea behind Exponential Smoothing (ETS) forecasting methods is that predicted values are weighted averages of past observations. The weight which is associated with the observation depends on how old the observation is. Thus, the oldest observations will have a smaller impact on the outcome than the recent ones.

The simplest version of the exponential smoothing method, called Simple Exponential Smoothing, is expressed by Equation 5 [10]. The application of this version of the method is limited to the data with no clear trend or seasonality.

$$\hat{y}_{T+1} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + ... \quad (5)$$

where:

- $0 \geq \alpha \leq 1$ is smoothing parameter.

The smoothing parameter regulates how the weights change with the change in the distance of observation. If α is small, more weight is given to the observations from the past. If it is large, more weight is associated with the recent observations.

Equation 5 [10] can be described in the component form as presented in Equation 6.

$$\begin{aligned} \text{Forecast Equation} \quad & \hat{y}_{t+h} = l_t \\ \text{Smoothing Equation} \quad & l_t = \alpha y_t + (1-\alpha)l_{t-1} \end{aligned} \quad (6)$$

where:

- h - number of steps to forecast,
- l - level component.

A single component is called level (smoothed value) l_t of the series at time t. From the forecast equation, we can see that the predicted value at time $t+1$ is the level of the time series at time t. Replacing the level component in the smoothing equation according to the relation $\hat{y}_{t+h} = l_t$ leads to the exponential smoothing form presented in Equation 5 [10].

To extend the application of the simple exponential smoothing method for data with trend, an additional component has been added to the equations. The extended method's name is Holt's linear trend method and is expressed by 3 equations presented in Equation 7 [10].

$$\begin{aligned} \text{Forecast Equation} \quad & \hat{y}_{t+h} = l_t + hb_t \\ \text{Level Equation} \quad & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \end{aligned} \quad (7)$$

where:

- b - the estimate of the trend,
- β^* - smoothing parameter for the trend, $0 \geq \beta^* \leq 1$.

The trend (slope) forecast function is no longer flat as it was in the case of Simple Exponential Smoothing. However, this method is still not especially useful because of the fact that the trend is constant. The method assumes that the outcome values always increase or decrease in the same way. Thus, an additional parameter called damping parameter has been introduced to deal with that. Because of this modification, the trend can be flattened in the future. The form of the method which includes the damping parameter is expressed by Equation 8. As we can see with damping parameter $\phi = 1$, the method is the same as Holt's linear method presented in Formula 7.

$$\begin{aligned} \text{Forecast Equation} \quad & \hat{y}_{t+h} = l_t + (\phi + \phi^2 + ... + \phi^h)b_t \\ \text{Level Equation} \quad & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \phi b_{t-1}) \\ \text{Trend Equation} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1} \end{aligned} \quad (8)$$

where:

- ϕ - damping parameter, $0 \geq \phi \leq 1$.

Holt's method can be extended with the seasonal component. This version is called Holt-Winters' seasonal method. There are two versions of the method: additive and multiplicative. The additive method is suitable for the series with constant seasonal variations. On the other hand, the multiplicative method is preferred when the variations change in proportion to the series. The component form of Holt-Winters' additive method is expressed by Formula 9 [10].

$$\begin{aligned} \text{Forecast Equation} \quad & \hat{y}_{t+h} = l_t + hb_t + s_{t+h-m(k+1)} \\ \text{Level Equation} \quad & l_t = \alpha(y_t - s_{t-m}) \\ & \quad + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\ \text{Seasonal Equation} \quad & s_t = \gamma(y_t - l_{t-1} - b_{t-1}) \\ & \quad + (1-\gamma)s_{t-m} \end{aligned} \quad (9)$$

where:

- m - number of seasons in a year,
- k - integer part of $(h-1)/m$,
- γ - smoothing parameter for the seasonality, $0 \geq \gamma \leq 1-\alpha$.

  The component form of Holt's-Winters' multiplicative method is expressed by formulas 10.

$$\begin{aligned} \text{Forecast Equation} \quad & \hat{y}_{t+h} = l_t + hb_t + s_{t+h-m(k+1)} \\ \text{Level Equation} \quad & l_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation} \quad & b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)b_{t-1} \\ \text{Seasonal Equation} \quad & s_t = \gamma\frac{y_t}{l_{t-1} - b_{t-1}} + (1-\gamma)s_{t-m} \end{aligned} \quad (10)$$

The difference between those two versions of Holt-Winters' method is how they express the seasonal component and then take it into account. In the additive method, it is expressed in absolute terms and then is seasonally subtracted from the series. In contrast to this in the multiplicative method, the seasonal component is expressed in relative terms and then the series is seasonally divided by it.

There are 9 different exponential smoothing methods. Those presented so far are examples of different combinations of components. Each method is defined by the type of trend and seasonal components.

The types of trend components are:

- None (N),
- Additive (A),
- Additive damped (A_d) .

The types of seasonal components are:

- None (N),
- Additive (A_d),
- Multiplicative (M).

Each exponential smoothing method can be labeled with two letters which refer to the type of trend and seasonal components. Table I presents the classification of exponential smoothing methods.

TABLE I: Classification of exponential smoothing methods.

Trend component	Seasonal Component		
	None (N)	Additive (A)	Multiplicative (M)
None (N)	(N, N)	(N, A)	(N, M)
Additive (A)	(A, N)	(A, A)	(A, M)
Additive damped (A_d)	(A_d, N)	(A_d, A)	(A_d, M)

For each of the 9 presented methods, there are two models differing in the way of expressing the errors. The first model with additive errors and the second one with multiplicative errors. For each method, the forecast points of two different models are the same. However, they generate different prediction intervals. To make the distinction the classification in Table I is extended by the third letter. Every exponential smoothing model is labeled with three letters as ETS(Error, Trend, Seasonal). Thus, the model that includes additive error, none trend component, and multiplicative seasonal component would be denoted as ETS(A, N, M).

We use the ETS implementation provided by the R forecast package [11]. The function `ets()` estimates the model parameters by analyzing the training data.

III. Related Work

A. Software Reliability Growth Models

Software Reliability Growth Models are equations used to model the growth of software reliability using defect inflow data gathered during the development process. Researchers use SRGMs to make defects forecasting. The side effect of predicting defects themselves is the knowledge about the number of defects in defect backlog. Using this relationship and applying SRGMs to predict the size of the defect backlog is a popular technique [12]. There is no standard way of selecting the most appropriate SRGMs for given defect data. There are studies that reveal the best-fitted models for reliability in some types of projects. In [6] researchers investigated the distribution of defect inflow in automotive domain projects which could aid in finding the best-fitting SRGMs. This work presents that selecting the appropriate model is the most challenging part of the forecast. There are more than 100 SRGMs.

B. Linear Regression

Linear regression modeling was also applied to the problem of defect backlog prediction. The examples of independent variables which are used in the regression models are [13]:

- program metrics (such as program size, number of variables),
- number of defects found in the earlier phase,
- testing time,
- design methodology.

Yu, Shen, and Dunsmore [13] investigated the correlation between those variables and the number of defects that remain in the software. They discovered the strongest relationship between a number of defects identified during earlier phases of development and those discovered later.

C. Defect Backlog Prediction at Ericsson AB

A research program executed at Ericsson AB company resulted in a few studies on defect backlog predictions. In the first study [4], Software Reliability Growth Models were designed to defect inflow prediction after release. The results were not satisfying. Defects profile described by the model significantly deviated from the profile of defects in the studied project.

In the follow-up study on defect inflow predictions in a large-size software project [5], the prediction accuracy of different methods (e.g., multivariate linear regression or method which used the moving average of defect inflow) was compared. Table II presents average prediction errors depending on number of predictor variables and the used method. To evaluate the accuracy of methods they used the Mean Magnitude of Relative Error (MMRE). The error of none of the evaluated methods was good enough to reach the required accuracy level by the organization.

TABLE II: Extract of prediction accuracy in a large-size project for 1-week interval [5].

Model	Type of model	MMRE (%)
Project milestone progress	Multivariate linear regression + PCA over milestone progress	52
2 weeks moving average	Moving average	34
3 weeks moving average	Moving average	38
Test progress – best statistical models	Multivariate linear regression + PCA over test progress	58
Test progress – statistics and expert combined	Multivariate linear regression over test progress (variables chosen by experts)	28
Expert estimations	Expert estimates based on historical data	375

To improve prediction accuracy researchers from Ericsson decided to conduct a more detailed study on medium-size project [3]. This time they evaluated the prediction accuracy of three methods:

- Multivariate linear regression,
- Analogy-based prediction,
- Expert estimations.

Seven variables identified as most influential on defect inflow were chosen from a set of over 50 and used to construct a multivariate linear regression model. For analogy-based prediction, researchers collected an analogy database (projects that they found the most similar to the one that they were working on). The variables used for calculating similarity were [3]:

- the number of test cases planned in integration testing 4 weeks before the predicted week,
- the number of test cases executed in integration testing 4 weeks before the predicted week.

Also, they decided to enrich analogy-based predictions by involving experts and asking them to choose variables that they found the most influential on defect inflow and assign them weights.

In the following study at Ericsson AB [3], the problem was reframed to predict defect backlog size instead of predicting defect inflow. A new method was proposed by Meding and Staron (MS) that relied on the moving average of defect inflow and defect outflow and the previous backlog size (see Equation 11). The proposed model allowed for predicting defect backlog size with the highest accuracy (MMRE of 16%) compared to the previous studies.

$$db(i) = db(i-1)\frac{di(i-1)+di(i-2)+di(i-3)}{3} - \frac{do(i-1)+do(i-2)+do(i-3)}{3} \qquad (11)$$

where:

- db(x) - defect backlog in week x,
- di(x) - predicted defect inflow in week x,
- do(x) - predicted defect outflow in week x.

IV. Research Methodology

A. Research Goal and Questions

We perform a Simulation-Based-Study (SBS) [14] using the data from OSS projects to *compare the accuracy* of two new defect-backlog prediction models based on Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS) with the state-of-the-art Meding-Staron model (MS). We formulate the following research questions:

- RQ1: *Are the MS, ARIMA, and ETS models more accurate than the Naïve prediction method?*
- RQ2: *Are ARIMA and/or ETS more accurate than the MS model when predicting the number of defects in defects backlogs of OSS projects?*

The question RQ1 could be considered a sanity test for the models. Shepperd and MacDonell [15] recommend performing such a test against "random guessing," however, we decided to use the so-called Naïve method instead of guessing. This method uses the actual observed values from the last week as the forecast for the next week. Although very simple, the Naïve method is reported to "work remarkably well for many economic and financial time series" [10]. Therefore, our sanity test is more demanding than the one proposed by Shepperd and MacDonell. However, for practical reasons, if a given model does not outperform the Naïve method, there is no point in considering it for real-life applications. The latter question (RQ2) is the central research question of this study. We compare the accuracy of the MS model, which according to the literature is the most accurate model for defect-backlog predictions with two models which are state-of-the-art in time-series predictions.

The replication package for this study is available on GitHub.[1]

B. Dataset

We collected defect backlogs from 20 Bugzilla instances of OSS projects managed by Apache Foundation, Eclipse, Mozilla Foundation, Linux, Open Office, and Libre Office. We selected only the projects that had a sufficiently long defect reporting period. The shortest defect-tracking period was 8 years (Libre Office Draw), while the longest was 22 years (Mozilla Core).

In the first step, we fetched defect reports from the *Bugzilla* service instances of OSS projects. The reports were grouped based on the dates when they were submitted or resolved and their severity level. By counting the number of defects submitted, resolved, or remaining in the backlog, we calculated defect backlog level (number of defect reports still opened at the end of the week), defect inflow (number of defects reported in a given week), and defect outflow (number of defects resolved in a given week) for every *week*.

The resulting dataset consisted of 20 defect backlogs presented in Table III. The beginning of each defect backlog is determined by the date of the first reports submitted to Bugzilla. The end of the bug tracking period is the same for all projects (01-01-2019). The average defect backlog size presented in Table IV ranges from 112 defects/week for Kernel Networking to 29,050 defects/week for Mozilla Core.

TABLE III: Dataset of OSS projects under study.

Project	Defect-tracking period
Eclipse Platform	10-10-2001 - 01-01-2019
Eclipse Birt	15-03-2005 - 01-01-2019
Eclipse Jdt	03-03-2005 - 01-01-2019
Eclipse Data tools	03-03-2005 - 01-01-2019
Eclipse PDE	20-11-2001 - 01-01-2019
Mozilla Firefox	30-07-1999 - 01-01-2019
Mozilla Core	28-03-1997 - 01-01-2019
Mozilla Thunderbird	02-01-2000 - 01-01-2019
Mozilla Calendar	09-11-2000 - 01-01-2019
Kernel File System	18-11-2002 - 01-01-2019
Kernel Networking	15-11-2005 - 01-01-2019
Kernel IO Storage	14-11-2002 - 01-01-2019
Open Office Writer	30-10-2000 - 01-01-2019
Open Office Calc	23-10-2000 - 01-01-2019
Open Office Draw	30-10-2000 - 01-01-2019
Apache Ant	11-09-2000 - 01-01-2019
Apache Apache2	15-01-2001 - 01-01-2019
Libre Office Writer	15-01-2001 - 01-01-2019
Libre Office Calc	08-10-2010 - 01-01-2019
Libre Office Draw	15-01-2011 - 01-01-2019

While visualizing the change in defect backlogs over time, we observed a suspicious phenomenon of rapid, significant drops in the number of defects in the backlog. We perceive them as anomalies that could result from "cleaning" processes of Bugzilla instances from irrelevant defect reports. Figure 1 presents an example of defect backlog level change over time in the Open Office Draw project with at least two sudden major drops in the number of defects around weeks 650 and 860, which are unlikely to be caused by the real defect fixing activities. Unfortunately, such drops are unexpected and poorly predicted by the considered prediction methods. On

[1] https://github.com/paulinaaniola/DefectBacklogPrediction.

TABLE IV: Average defect backlog sizes in OSS projects (defects/week).

Product	Mean backlog size
Kernel Networking	112.38
Eclipse Platform	7,240.95
Eclipse Data Tools	159.05
Eclipse Birt	1,000.21
Eclipse JDT	3,329.15
Eclipse PDE	850.86
Mozilla Calendar	1,219.78
Mozilla Firefox	11,354.33
Mozilla Core	29,050.79
Mozilla Thunderbird	3,708.5
Kernel IO Storage	161.42
Kernel File System	190.59
Open Office Writer	8,123.17
Open Office Calc	3,245.73
Open Office Draw	810.53
Apache Ant	1,213.95
Apache Apache 2	976.48
Libre Office Writer	3,245.32
Libre Office Calc	1,816.79
Libre Office Draw	370.36

the contrary, they have a less visible impact on the prediction errors made by the Naïve method since the error is present only for a single week following such a drop.

To mitigate the effect of sudden drops in the level of the backlog, we decided to split the defect backlogs based on the presence of unexpected drops in the number of defects. The process of dividing backlogs into fragments was the same for all projects and started with differencing the defect backlog level which result is presented in Figure 2. The peaks in the differenced time-series plot correspond to the sudden falls in backlog level from Figure 1. To consistently determine the weeks for which there are sudden decreases in the backlog level and to set the boundaries between fragments in those weeks, the 99.5 percentile of the difference in the backlog between successive weeks was calculated. For instance, Open Office Draw backlog was divided into 5 fragments presented in Figure 3. The final prediction error of a given method for a divided backlog is counted as the average of errors for individual fragments.

C. Predictions And Accuracy Evaluation

We performed training and accuracy evaluation for each individual defect backlog. We used data from all the previous weeks to train the ARIMA and ETS models and predict the number of defects in the backlog for the following week. We based the accuracy evaluation on Absolute Error calculated according to Equation 12 and calculated Mean Absolute Error (MAE) for each backlog.

$$AE = |actual\, value - predicted\, value| \tag{12}$$

We also calculated a variant of standardized accuracy measure (SA_m) [15] that shows a relative improvement in accuracy in comparison to the Naïve method, which was calculated according to Equation 13.

$$SA_m = (1 - \frac{MAE_m}{MAE_n}) * 100\% \tag{13}$$

where:
– $\mathrm{MAE_m}$ - Mean Absolute Error for the method m,
– $\mathrm{MAE_n}$ - Mean Absolute Error for Naïve method.

We used a non-parametric Wilcoxon signed-rank test to compare AE between prediction models with significance level $\alpha = 0.05$ and Cliff's δ effect-size coefficient to quantify the strength of the observed difference. Cliff's δ evaluates how often the values of one set are larger than the ones from the second set. The thresholds used for Cliff's δ coefficient interpretation proposed by Kitchenham et al. [16] are as follows: δ < 0.112 – negligible, 0.112 $\geq \delta$ < 0.276 – small, 0.276 $\geq \delta$ < 0.428 – medium, and $\delta \geq$ 0.428 – large.

We also calculate the number needed to treat (NNT = δ^{-1}) measure, which is commonly used in the field of medical science. NNT indicates how many patients need to be treated with a drug to heal one patient and is the measure of the medicine's effectiveness. The lowest the NNT value the fastest we achieve the improvement. In the context of this study, NNT could be interpreted as the number of weeks one would have to use a given prediction method A instead of method B to observe improvement in the accuracy for at least one week. We also calculated the average NNT to aggregate information from all the projects.

V. Results And Discussion

Mean prediction errors for the three considered methods and the Naïve method are presented in Table V, while the mean errors transformed to standardized accuracy measures (SA_m) are presented in Table VI.

ARIMA, ETS, and MS predicted backlog sizes with mean errors lower than the Naïve method for most of the projects (mean SA equal to ca. 17.1%, 17.7%, and 10.3%, respectively). However, there were three projects for which the Naïve method performed better than all three other methods. The effect size and the results of Wilcoxon signed-rank tests for comparison between AE (intra-project level) for the considered models and the naive one are presented in Table VII. For every of the considered models that were at least 10 projects for which a statistically significant difference in the central tendency of AE was detected. The effect size was at least "small" for nearly half of the projects, i.e., 10/20 (ARIMA vs. Naïve), 9/20 (ETS vs. Naïve), and 11/20 (MS vs. Naïve). That translates to NNT at the levels of 8 weeks for ARIMA, 14 weeks for ETS, and 36 weeks for MS. Finally, we performed Wilcoxon signed-rank test at the level of MAE (dataset level). In all three cases, the difference in the central tendency for MAE between considered models and the Naïve method turned out to be statically significant. **Therefore, we conclude that all of the considered methods outperform the Naïve method (RQ1).**

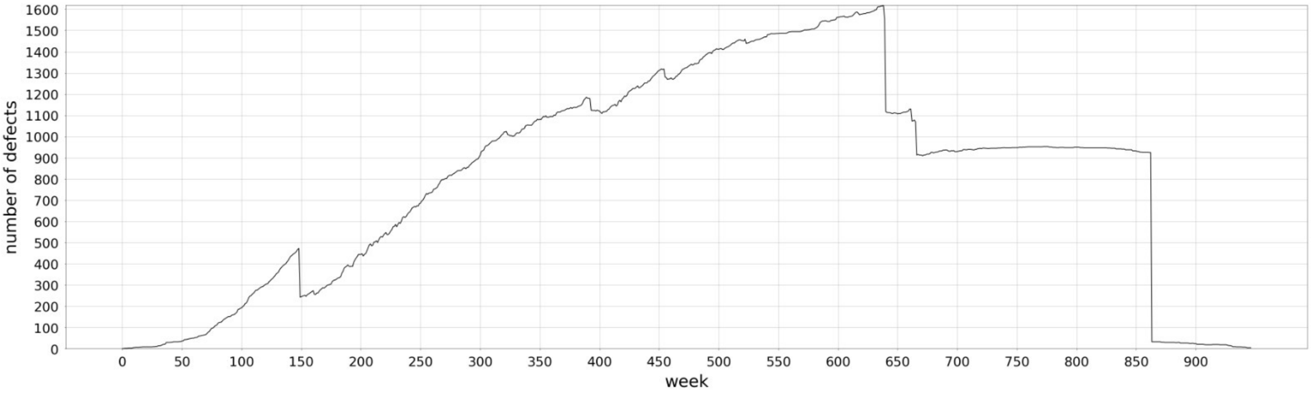


Fig. 1: Defect backlog level changes in Open Office Draw.

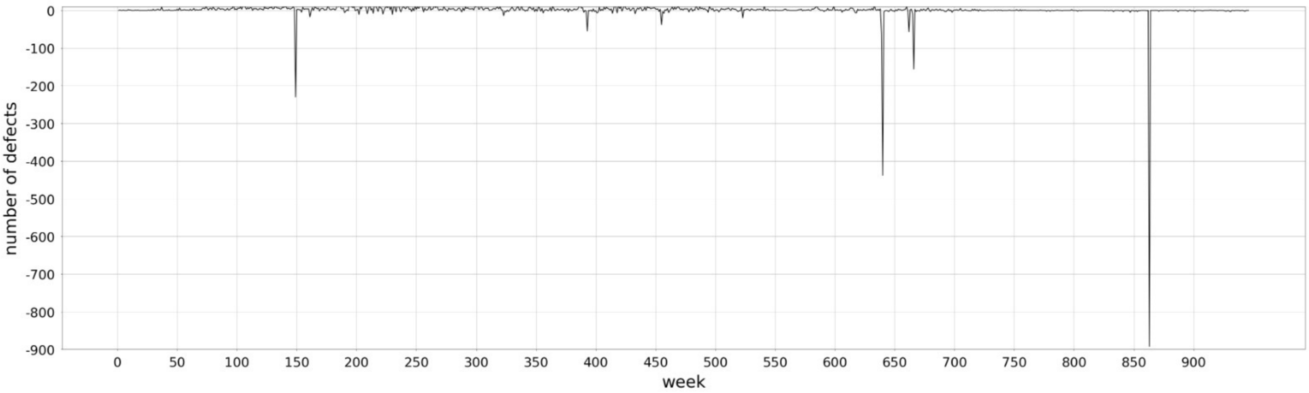


Fig. 2: The differenced time series of defect backlog level in the Open Office Draw project.

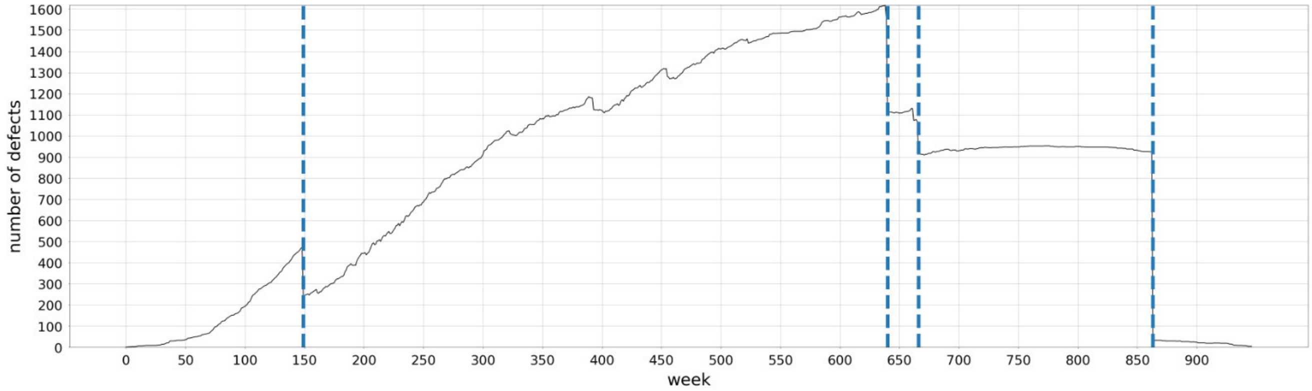


Fig. 3: Open Office Draw defect backlog divided into 5 fragments.

TABLE V: Defect backlog size prediction errors.

Product	MAE_{ARIMA}	MAE_{ETS}	MAE_{MS}	MAE_{Naive}
Kernel Networking	1.97	2.02	2.3	1.81
Eclipse Platform	44.99	42.5	43.39	47.6
Eclipse Data Tools	3.43	3.63	3.23	3.63
Eclipse Birt	15.17	14.82	17.74	16.36
Eclipse Jdt	21.07	20.36	25.19	22.62
Eclipse Pde	7.61	7.6	8.63	7.88
Mozilla Calendar	7.92	7.46	7.78	9.85
Mozilla Firefox	56.81	56.45	57.44	70.11
Mozilla Core	87.04	84.59	85.89	110.63
Mozilla Thunderbird	17.4	18	17.62	21.05
Kernel IO Storage	2.61	2.58	2.88	2.55
Kernel File System	2.95	2.88	2.86	2.79
Open Office Writer	11.73	10.54	14.65	23.6
Open Office Calc	5.74	5.67	7.77	10.69
Open Office Draw	2.23	2.16	2.89	2.71
Apache Ant	3.35	3.01	3.42	6.27
Apache Apache 2	4.53	4.59	5.82	6.23
Libre Office Writer	20.88	21.73	20.76	27.97
Libre Office Calc	13.31	12.94	12.1	17.57
Libre Office Draw	3.24	3.63	3.32	3.81
mean MAE	**16.70**	**16.36**	**17.28**	**20.79**
standard deviation MAE	**22.02**	**21.44**	**21.68**	**27.20**

TABLE VI: Defect backlog prediction improvement (SA) compared to the Naïve method predictions.

Product	SA_{ARIMA} [%]	SA_{ETS} [%]	SA_{MS} [%]
Kernel Networking	-8.84	-11.6	-27.07
Eclipse Platform	5.48	10.71	8.84
Eclipse Data Tools	5.51	0	11.02
Eclipse Birt	7.27	9.41	-8.44
Eclipse Jdt	6.85	9.99	-11.36
Eclipse Pde	3.43	3.55	-9.52
Mozilla Calendar	19.59	24.26	21.02
Mozilla Firefox	18.97	19.48	18.07
Mozilla Core	21.32	23.54	22.36
Mozilla Thunderbird	17.34	14.49	16.29
Kernel IO Storage	-2.35	-1.18	-12.94
Kernel File System	-5.73	-3.23	-2.51
Open Office Writer	50.3	55.34	37.92
Open Office Calc	46.3	46.96	27.32
Open Office Draw	17.71	20.3	-6.64
Apache Ant	46.57	51.99	45.45
Apache Apache 2	27.29	26.32	6.58
Libre Office Writer	25.35	22.31	25.78
Libre Office Calc	24.25	26.35	31.13
Libre Office Draw	14.96	4.72	12.86
mean SA	**17.08**	**17.69**	**10.31**
standard deviation	**16.72**	**18.11**	**19.11**

TABLE VII: Effect size (n-negligible, s-small, m-medium) and Wilcoxon singed-rank tests result for comparison between ARIMA, ETS, MS, and the Naïve method (T – null hypothesis rejected with $\alpha = 0.05$).

	AE_{ARIMA} AE_{Naive}		AE_{ETS} AE_{Naive}		AE_{MS} AE_{Naive}	
Product	δ	**diff. test**	δ	**diff. test**	δ	**diff. test**
Kernel Networking	n	F	n	F	n	T
Eclipse Platform	n	F	n	T	n	F
Eclipse Data Tools	n	F	n	F	n	F
Eclipse Birt	n	F	n	T	n	F
Eclipse Jdt	n	F	n	T	n	F
Eclipse Pde	n	F	n	F	n	T
Mozilla Calendar	s	T	s	T	s	T
Mozilla Firefox	s	T	s	T	s	T
Mozilla Core	s	T	s	T	s	T
Mozilla Thunderbird	s	T	n	T	s	T
Kernel IO Storage	n	F	n	F	n	F
Kernel File System	n	F	n	F	n	F
Open Office Writer	m	T	m	T	s	T
Open Office Calc	m	T	m	T	m	T
Open Office Draw	n	T	n	T	n	F
Apache Ant	s	T	s	T	s	T
Apache Apache 2	s	T	n	T	n	F
Libre Office Writer	m	T	m	T	m	T
Libre Office Calc	m	T	m	T	m	T
Libre Office Draw	n	T	n	F	s	F

In the next step, we compared the accuracy of the state-of-the-art MS method and two methods proposed in this paper that are based on ARIMA and ETS. As it follows from Table V the lowest mean MAE was observed for ETS (16.36) and ARIMA (16.70), however, the mean MAE for MS was only ca. 5% higher (17.28) than the one observed for ETS. It is also visible that the methods perform consistently for all projects, i.e., there are no methods that would visibly outperform other methods on a single project. Also, as it follows from Table VIII, only for 3-4 projects the observed difference in the central tendency for AE (intra-project level) could be considered statistically significant. Also, the effect size could be interpreted as "negligible" for comparing AE for all the projects that translated to NNT at the level of 68 for ARIMA vs. MS and 32 ETS vs. MS. However, statistical inference at the dataset level regarding central tendency in MAE resulted in rejecting the null hypothesis for comparison between ETS and MS methods. **Therefore, we conclude that**

all the considered methods are good candidates to be used for predicting defect backlog size for OSS projects—with a slight preference towards ETS (RQ2). Taking into account NNT, statistically, one shall see improvement in defect prediction for at least one week after applying ETS for 32 weeks instead of MS.

TABLE VIII: Effect size (n-negligible, s-small, m-medium) and Wilcoxon singed-rank tests result for comparison between ARIMA, ETS and MS (T – null hypothesis rejected with $\alpha = 0.05$).

	AE_{ARIMA} AE_{MS}		AE_{ETS} AE_{MS}	
Product	δ	**diff. test**	δ	**diff. test**
Kernel Networking	n	T	n	F
Eclipse Platform	n	F	n	F
Eclipse Data Tools	n	F	n	F
Eclipse Birt	n	F	n	F
Eclipse Jdt	n	T	n	T
Eclipse Pde	n	T	n	T
Mozilla Calendar	n	F	n	F
Mozilla Firefox	n	F	n	F
Mozilla Core	n	F	n	F
Mozilla Thunderbird	n	F	n	F
Kernel IO Storage	n	F	n	F
Kernel File System	n	F	n	F
Open Office Writer	n	F	n	F
Open Office Calc	n	F	n	F
Open Office Draw	n	F	n	F
Apache Ant	n	F	n	F
Apache Apache 2	n	T	n	T
Libre Office Writer	n	F	n	F
Libre Office Calc	n	F	n	F
Libre Office Draw	n	F	n	F

A. Threats to Validity

We address the threats to validity in the manner as described by Wohlin et al. [17] and de França et al. [14].

a) Construct validity: The main construct validity threat concerns the process of cleaning the data. Each backlog was divided into fragments in places of sudden falls in the number of defects. The rule of determining the sudden falls was the same for all backlogs. It assumes a division point between weeks for which the difference in the number of defects in the backlog was more than 99.5 percentile of the differences between successive weeks from the backlog. It resulted that all backlogs being divided, even those in which the aggressive declines have not really taken place.

b) Internal validity: There exists a threat to the internal validity of this study regarding the validity of reported defects. For OSS projects, all users can report defects to Bugzilla. The new reports may be duplicates or not be real defects. We cannot control who makes reports and what they are. Because of that, the size of some defect backlogs is extremely large.

c) External validity: The main threat to the external validity of our results is the fact that we applied the defect backlog prediction methods only to a selected sample of well-established OSS projects that maintain public Bugzilla instances. We do not know whether our results would also apply to smaller OSS projects, however, there is a question of whether such projects would benefit from defect backlog predictions. Also, we limited our study to OSS projects only, therefore, we would be careful in generalizing the findings to industrial projects since the ways of working differ visibly between the OSS and industrial settings. Even when it comes to OSS projects themselves, we have to be aware that the process could be less stable in time than it is for the industry (e.g., the number of contributors involved, the number of commits they produce, or the number of defects they fix can vary in time). Also, we used the 1-week prediction horizon after the previous studies in Ericsson AB, however, we cannot claim based on our results that the methods will behave the same way if a longer prediction horizon is needed by a given OSS community.

d) Conclusion validity: The main threat to conclusion validity regards performing multiple statistical inference tests while drawing some of the conclusions (statistical inference tests at intra-project and dataset levels). We set the local significance level α to 0.05, however, the true, global significance level would be much higher. Still, the outcomes of the statistical inference tests were only one of a few sources of information that we used to draw the conclusions, therefore, the impact of rejecting a true null hypothesis would have a minor impact on the final conclusions.

VI. Conclusions and Future direction

In this paper, we evaluated three defect backlog prediction methods in the context of open-source projects, i.e., the state-of-the-art Meding-Staron model (MS) and two new models based on Autoregressive Integrated Moving Average (ARIMA) and Exponential Smoothing (ETS) time-series forecasting methods.

We compared the accuracy of these methods on the dataset consisting of defect backlog histories of 20 large open-source projects (ranging from 8 to 22 years). In the first step, we performed a sanity check by comparing the accuracy of these methods with the accuracy of the so-called Naïve method, which predicts a defect backlog size in the following week to be the same as the one observed in the current week. All three methods outperformed the Naïve method by ca. 10% to 17.7% (the largest improvement was observed for the ETS method). The observed differences in mean absolute errors (MAE) were statistically significant for all the methods. With respect to effect size, one would have to use these methods instead the Naïve one for 8 to 36 weeks to, statistically, notice improvement in defect backlog predictions for at least one week.

Exponential Smoothing (ETS) provided slightly more accurate defect backlog size predictions than the state-of-the-art MS method (by ca. 5%) and the prediction method based on ARIMA (by ca. 2%). If one decides to use the ETS-based defect backlog prediction method instead of the MS method should statistically, notice an improvement in defect backlog predictions for at least one week after using it for 32 weeks.

Therefore, the main contributions of our study to the body of knowledge are as follows:

- we provided some new observations regarding the state-of-the-art Meding-Staron model (MS) by evaluating its accuracy in the open source context (in addition to the previous studies in Ericsson). Based on the results of this and the previous studies on that method, we can conclude that it is suitable for defect backlog size prediction in both industrial and open-source settings.
- we proposed two new defect backlog size prediction methods based on the state-of-the-art time series forecasting methods ARIMA and ETS, which perform slightly better (especially ETS) than the MS method in the open source context.
- all the considered defect backlog prediction models based on autoregression (ARIMA, ETS, and MS) are very accurate when estimating defect backlog levels in OSS. Taking into account that the mean size of the defect backlogs for the considered OSS project ranged from ca. 112 to 29,000 defect reports and the mean absolute error for the ETS model ranged from ca. 2 to 85 defects, the relative error for that model was at the level of 0.13% to 2.28% (with respect to the average backlog size of a project).

As future research directions, we plan to investigate the accuracy of artificial neural network-based time-series prediction models (e.g., LSTMs, GRU) for defect backlog predictions. We would also like to validate the proposed models based on ARIMA and ETS based on historical data from industrial projects.

REFERENCES

[1] P. Sielicka, "Defect backlog size prediction with the autoregressive moving average and exponential smoothing models," Master's thesis, Poznan University of Technology, Poland, 2019.

[2] W. Meding, "Effective monitoring of progress of agile software development teams in modern software companies: an industrial case study," in *Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement*, 2017, pp. 23–32.

[3] M. Staron and W. Meding, "A method for forecasting defect backlog in large streamline software development projects and its industrial evaluation," *Information and Software Technology*, vol. 52, no. 10, pp. 1069–1079, 2010.

[4] ——, "Defect inflow prediction in large software project," *e-Informatica Software Engineering Journal*, vol. 4, no. 1, pp. 89–107, 2010.

[5] ——, "Predicting weekly defect inflow in large software projects based on project planning and test status," *Information and Software Technology*, vol. 50, no. 7, pp. 782–796, 2008.

[6] R. Rana, M. Staron, C. Berger, and J. Hansson, "Analysing defect inflow distribution of automotive software projects." PROMISE '14' - 10th International Conference on Predictive Models in Software Engineering, 2013.

[7] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[8] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[9] P. R. Winters, "Forecasting sales by exponentially weighted moving averages," *Management science*, vol. 6, no. 3, pp. 324–342, 1960.

[10] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice, 2nd edition*. OTexts: Melbourne, Australia, 2013, oTexts.com/fpp2. Accessed on 03.06.2019.

[11] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: the forecast package for R," *Journal of Statistical Software*, vol. 26, no. 3, pp. 1–22, 2008.

[12] R. Rana, "Software defect prediction techniques in automotive domain: Evaluation, selection and adoption," Ph.D. dissertation, University of Gothenburg, 2015.

[13] H. D. T.J. Yu, V.Y. Shen, "An analysis of several software defect models," *IEEE Transactions on Software Engineering*, vol. 14, no. 9, pp. 1261 – 1270, 1998.

[14] B. Bernard Nicolau de França and G. Horta Travassos, "Simulation based studies in software engineering: A matter of validity," *CLEI electronic journal*, vol. 18, no. 1, pp. 5–5, 2015.

[15] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Information and Software Technology*, vol. 54, no. 8, pp. 820–827, 2012.

[16] B. Kitchenham, L. Madeyski, D. Budgen, J. Keung, P. Brereton, S. Charters, S. Gibbs, and A. Pohthong, "Robust statistical methods for empirical software engineering," *Empirical Software Engineering*, vol. 22, no. 2, 2017.

[17] C. Wohlin and et al, *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publisher, Boston, MA, 2000.

DOI: 10.15439/2023F8371
ISSN 2300-5963 ACSIS, Vol. 35

Can Unlabelled Data Improve AI Applications? A Comparative Study on Self-Supervised Learning in Computer Vision.

Markus Bauer
Center for Scalable Data Analytics and Artificial Intelligence
Humboldtstraße 25, Leipzig, 04105 Germany
Email: bauer@wifa.uni-leipzig.de

Christoph Augenstein
Center for Scalable Data Analytics and Artificial Intelligence
Humboldtstraße 25, Leipzig, 04105 Germany
Email: augenstein@wifa.uni-leipzig.de

***Abstract*—Artificial Intelligence (AI) represents a highly investigated area of study at present and has already become an indispensable component within an extensive range of business models and applications. One major downside of current supervised AI approaches lies in the need of numerous annotated data points to train the models. Self-supervised learning (SSL) circumvents the need for annotation, by creating supervision signals such as labels from the data itself, rather than requiring experts for this task. Current approaches mainly include the use of generative methods such as autoencoders and joint embedding architectures to fulfil this task. Recent works present comparable results to supervised learning in downstream scenarios such as classification after SSL-pretraining. To achieve this, typically modifications are required to suit the approach for the exact downstream task. Yet, current review works haven't paid too much attention to the practical implications of using SSL. Thus, we investigated and implemented popular SSL approaches, suitable for downstream tasks such as classification, from an initial collection of more than 400 papers. We evaluate a selection of these approaches under real-world dataset conditions, and in direct comparison to the supervised learning scenario. We discuss SSL's potential to take up with supervised learning, as well as the influence of the right training methods. Furthermore, we also introduce future directions for SSL research, as well as current limitations in real-world applications.**

I. Introduction

SELF-supervised learning (SSL) has recently gained massive attention as a promising new learning paradigm in the machine learning world. The main advantage over supervised learning lies in SSL's capability to reduce the amount of required prior work of data scientists by avoiding manual annotation. Recent work has shown that state-of-the-art results can be achieved in downstream tasks, when using SSL as a pretraining method. This includes classification scenarios, as well as clustering.

Especially for medical and industry applications, where annotations can only be created by highly trained, rarely available experts, this can possibly be a game-changer in bringing Artificial Intelligence (AI) to a wider mass of companies. Currently, however, SSL methods are typically evaluated on large image datasets such as ImageNet [1], which compare poorly to real-world data. The problem is that ImageNet and comparable datasets consist of well-balanced, well-curated and giant data collections. Real-world datasets, especially in the industry and medical sector, on the other hand, typically only feature a few 1000 to 10000 samples which contain systematic label noise, defective images and imbalanced data. Additionally, such datasets often contain obvious data properties and fine-granular ones, whereas the latter ones need to be represented by the SSL-extracted features. Popular SSL methods currently are only proven to work correctly if balanced data [2] and large batch sizes are used [3]. For datasets and tasks under real-world conditions, this is often unfeasible and thus, the applicability and required modifications of SSL remain unclear.

We thus reviewed the literature to find the most important approaches for SSL in computer vision and summarize their basic functions, as well as possible modifications. We focused on practical implications of the presented approaches and investigated possible application fields of different algorithmic groups and their modifications. Additionally, we compared the performance of SSL models and their supervised pendants. For this, we evaluated the models using two datasets composed of MNIST and Oxford-Flower+IIT-Pet data [4], [5], [6] with little dataset size and different complexities of problems to address, thus, with conditions that hold for real-world applications as well.

To retrieve an extensive and representative collection of SSL-approaches, various databases were queried, using the system as described in Bramer et al. [7]. We identified appropriate journals and conferences for the topic and selected appropriate search engines that contain these. Several well-recognized journals such as the "IEEE Transactions on Pattern Analysis and Machine Intelligence" cover the scope of our work. We selected 20 of the best-ranked journals that cover our scope and picked an initial database according to the best availability of these journals. Among other choices, such as Cabi, Inspec and Web of Science, EBSCO had the best availability and thus was selected. The EBSCO academic search premier contains more than 3000 peer reviewed journals. In addition, we used the IEEE database directly, as well as Google Scholar, to also include popular conference proceedings such as the MICCAI, CVPR and NeurIPS, as well as publishers like ACM Digital Library and less popular journals. EBSCO is also preferred because it offers the use of thesauri terms, which are useful for filtering the search results. From an initial collection of 1792 samples, we finally extracted 42 articles to be considered for our comparative study. The detailed literature selection process is shown in Fig. 1.

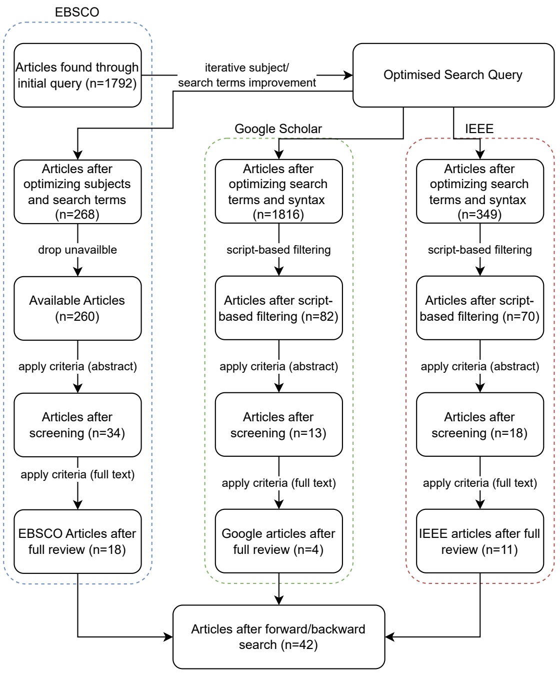

EBSCO
Articles found through initial query (n=1792)
iterative subject/ search terms improvement
Optimised Search Query
Google Scholar
IEEE
Articles after optimizing subjects and search terms (n=268)
Articles after optimizing search terms and syntax (n=1816)
Articles after optimizing search terms and syntax (n=349)
drop unavailble
script-based filtering
script-based filtering
Available Articles (n=260)
Articles after script-based filtering (n=82)
Articles after script-based filtering (n=70)
apply criteria (abstract)
apply criteria (abstract)
apply criteria (abstract)
Articles after screening (n=34)
Articles after screening (n=13)
Articles after screening (n=18)
apply criteria (full text)
apply criteria (full text)
apply criteria (full text)
EBSCO Articles after full review (n=18)
Google articles after full review (n=4)
IEEE articles after full review (n=11)
Articles after forward/backward search (n=42)

Fig. 1: Overview of the split search using EBSCO, IEEE Explore and Google Scholar.

II. Related Work

Various works have been proposed in the past that demonstrate impressively the value of SSL for clustering and classification tasks. Generally, two major approaches with five possible modification types can be found in the literature, among preliminary, less generalizable works. Fig. 2 provides an overview of SSL and its configuration possibilities.

Autoencoders (AEs) are the possibly most prominent architecture, and already gained attention for several years. The core principle is simple: Input data (i.e., images) is processed using a CNN encoder and a feature vector (:= embedding or latent) is created. Afterwards, the inverted CNN is used as a decoder to reconstruct the input. The supervision signal is then generated from the distance of the generated and input image. Thus, an optimization can be done, whereas reconstruction will improve with more meaningful latents that capture significant properties of the input data. A first application of this architecture was denoising, as proposed by Vincent et al. [8], followed by variants that make use of Gaussian inference [9] and additional classification tasks, called adversarial learning [10]. Even though AEs can achieve remarkable results for, e.g., clustering with smaller and simple image datasets (such as the MNIST dataset), they have known limitations when working with more complex data.

Thus, another class of SSL algorithms, joint embedding architectures (JEA), have gained attention recently. In JEA, the input will be processed using at least two CNNs (whereas weights may be shared), sent through a bottleneck to reduce dimensions, and then passed through a projection head, which is typically a fully connected network. As for the AE, backpropagation is the last step in JEA training. JEA uses a contrastive loss function [11] or an entropy-based function [12] that keeps the vectors of augmented and non-augmented image versions consistent. Very early works of Noroozi et al. [13] show that suitable SSL signals for JEA may be easy and intuitive to implement, e.g., by splitting or solving jigsaw puzzles. Even though such works didn't directly combine latents, and thus technically aren't JEAs, they share the idea and can be seen as close relatives of JEA. As the model needs to understand contextual information before solving the task, representative features will be learned as a side effect. Similar results can be achieved using image rotation [14], pseudo-classification [15] or combining multiple augmentations [11]. JEA's representative feature extraction capabilities can even improve if historical examples are used from a memory bank [16].

The major groups of AE and JEA can be divided further according to the following six modification types:

- Input variations: Input data is directly modified before being processed.
- Backbone variations: Other architectures than a single CNN are used.
- Latent variations: The features created by the bottleneck are processed in a parallel or preceding path to projection/decoding.
- Projection variations: A more sophisticated model than, e.g., an MLP is used as projection head.
- Temporal component: Time-dependent information is captured.

The following sections contain information about AE and JEA basic and modified approaches.

A. AE with Input Variations

The proposed basic principle of the AE as of Vincent et al. [8], as well as the ones of Kingma et al. and Makhzani et al. [9], [10], may be modified such that the input is preprocessed in a certain way, before feeding it to the AE. Such modifications may include augmentations, such as stretching, rotation and jitter, that will or will not be part of the target image. That way, the training can be guided to pay particular attention, or ignore certain properties of the input data, and thus to learn more robust features. This can be of value, e.g., in single-cell cytometric imaging, where AEs may be vulnerable to trivial features such as cell rotation [17]. Generally, an AE approach with input variations is seemingly mostly feasible, if obvious data properties should be ignored, as the typical application of this architecture often comes with little to no knowledge about important input data properties. This is not

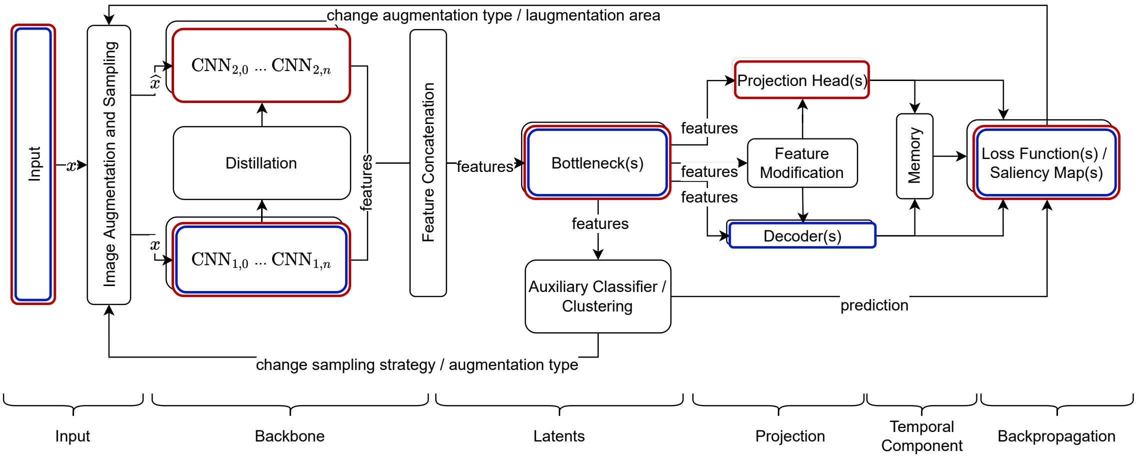


Fig. 2: Overview of the discovered SSL training elements and their respective order. The default configurations for JEA (red) and AE (blue) are marked. Note that not all elements are necessarily used.

only limited to examples in the research field of life sciences. Another application in the industry context could, e.g., be the extraction of different features than build-part size in 3D-print classification. As the build-parts are printed layer after layer and hence their size differs throughout the print job, the most simple property is to learn exactly this varying size. This will, however, have almost no implications for, e.g., quality measurements. Using augmentations such as scaling and stretching, particular attention for anomalies can likely be created that enables extracting more meaningful latents that, e.g., correlate with the print quality.

B. AE with Backbone Variations

Another approach to increase the AE's performance lies in changing the encoder backbone from a simple CNN to a more sophisticated architecture. This includes using multiple backbone branches, or an encoder ensemble, whereas each model receives different input (i.e., augmented versions of the original). Furthermore, features from different model depths can be concatenated as the latent, to consider different morphological complexities in the vector representations. By modifying the backbone that way, a better weighting of global and local features may be achieved [18], as well as multiple views that carry different information may be considered for feature extraction, e.g., as in the case of analysing spectral bands in hyperspectral imaging (HSI) classification [19]. The concept can also be adapted to applications that leverage information, e.g., from audio and text data. Even though this variant may cause the trained models to be significantly more memory-intensive or slower, their feature extraction capabilities also may improve remarkably, when compared to a basic AE.

C. AE with Latent / Loss Variations

Similar to the approach of Makhzani et al. [10] the learning signal can be supported by not only formulating the cost functions as a reconstruction problem, but rather adding an auxiliary learning signal, such as the adversarial problem. Recent work shows that different learning signals than the adversarial discriminator, such as clustering, may be of value to assist the training process. They are included as a multi-loss cost function. Possible applications include the direct consideration of the extracted features clustering probabilities, e.g., in tasks where clustering of different cell lines is the goal [20]. Popular approaches to achieve better clustering capabilities include the use of Gaussian mixtures rather than a single Gaussian for the feature space [21]. The advantage of this modification is that it successfully extracts more robust features and thus is more suited against data distribution drift and noise. Typical application fields could also lie in life-science imaging, where artefacts are likely to arise and confuse trained models.

D. JEA with Backbone Variations

In contrast to the generative AE models, JEA enables one to train without the need for image generation. State-of-the-art results have recently been achieved in different downstream tasks of models pretrained with JEA. There are, however, applications, where the standard architecture needs to be modified analogous to the generative AE methods.

A first approach is to change the backbone or combine multiple architectures. The influence of different backbone models has been shown by Guerin et al. [22], who conclude that different architectures have different strengths/target domains and thus combining them helps to increase model performance. They use multiple pretrained architectures to perform clustering using JULE [23].

Another variant of backbone modifications is the use of Vision Transformers (ViTs) that could be shown to achieve comparable or even better results than CNNs that process images as a whole, while also, e.g., offering better properties when it comes to interpretability (c.f. [24]). They are thus a promising alternative to, e.g., residual networks [25]. In contrast to convolutional networks, ViTs show better properties regarding the weighting of local and global features, which can be advantageous in imaging domains that require context for each object, such as polarimetric synthetic aperture radar image classification [26]. For applications such as remote sensing imaging, where images are inherently very similar even though they may show different objects (e.g., a town house vs. a barn), ViTs also can be combined with knowledge distillation [27]. The imposed teacher vs. student asymmetry then improves fine-granular feature extraction capabilities. The distillation approach furthermore offers the possibility to reduce the network's parameters within the student model, and thus to decrease memory usage and processing times. JEAs backbone may also be modified, when multiple domains need to be unified, e.g., 2D and 3D images [28], or if special downstream tasks such as object detection are the training target [29].

E. JEA with Latent Variations

Directly modifying the latents may be of value to enable the model to, e.g., perform fine-grained clustering. This includes rearranging, normalization and regularization of the latents before processing them with the projection head and triggering the backpropagation. Additionally, created latents may also be used, to generate an auxiliary training signal that improves properties such as drift-resistance. This can be of interest, e.g., where slight changes between images have no meaning as in the case of scene classification, and thus the model needs to be robust against changes in poses, part configurations, as well as to relative motion between objects, and scene structures. Possible methods to implement such latent variations are spatial assembly [30] or the use of adversarial training for resistance against perturbations [31]. The use of an adversaria task also allows adapting JEA for image hashing problems [32].

Recent works as Masked Siamese Networks (MSN) or Joint Embedding Predictive Architectures (JEPA) randomly mask parts of the latents [33], [34], to simulate masking of the input images while maintaining asymmetry between, e.g., a student and teacher network. That way, the JEA scenario can be used with significantly less computation time, as no computation expensive augmentations need to be performed.

Modifications to, and further auxiliary processing of latents is advantageous where slight image changes have a significant impact on the data domain they refer to. This can, e.g., be the case in quality control, where small deviations from the standard can have a significant impact on product quality, in representation learning scenarios or in security applications.

F. JEA with Projection Variations

After latents have been extracted by the backbone, features typically are projected to a fixed-size vector, e.g., using fully connected layer and average pooling. Different methods allow for improving this projection compatible to later downstream applications such as clustering. For medical imaging, typically images need to be split into patches, and thus separate modules need to be created, that keep features consistent through the global image scope [35]. Furthermore, clustering capabilities of the extracted features may be improved, by assigning the features to cluster prototypes according to an optimal transport problem [36], which advantageously also avoids trivial solutions (mode collapse). Thus, using different projections heads, one can not only control the level of granularity in information extraction, but also adapt the model to special application scenarios, as in the case of histopathological whole-slide-imaging.

G. JEA with Temporal Component

In many scenarios, important information is time-dependent and thus can't be extracted from a single data point. Therefore, additional components need to be integrated, such that the model can capture the time-information. This can also enable a model to be suitable for streaming scenarios. Including temporal information can be done using distillation procedures [37], or memory bank approaches [38]. While distillation has the advantage of imposing asymmetry, which helps to find more robust features, memory banks can reduce the calculation efforts in terms of GPU-usage. Integrating a temporal component into the JEA learning setup further extends the possible application scope, e.g., to include quality monitoring, where slight data distribution changes are expected and need to be captured by the model to represent all data classes correctly [39]. This modification is, however, more of interest for time-series or video-based scenarios, than for imaging.

H. Preliminary Works

In addition to the presented groups and subgroups, further approaches have been discovered that show preliminary results or, as of now, fail to achieve state-of-the-art results on real-world data. They are presented for completeness here. Modifications to the default JEA scenario may also include recoupling of the model's output, e.g., by leveraging the Grad-Cam approach for sharpening the area of interest in the input image [40]. To ensure a higher probability of finding a global minimum during CNN parameter optimization, dropout is known as a useful measure, especially as it helps to avoid over-fitting. Thus, advanced dropout methods such as biologically inspired ones [41] may be a good choice to add to the JEA or AE model. The right choice of pre-text tasks also heavily influences the training. Thus, popular works often investigate how different augmentations contribute to the training results [42]. Other approaches show that methods typically used in JEA, such as the one of He et al. [43], are also of value for AEs. Examples include the mapping of AE features to dictionaries rather than simple keeping them for further processing [44] or combining AE and JEA architectures [45]. In special cases, selecting specific training variants, such as evolutionary [46] or sparse kernel network training, as well as solutions inspired by biological processes such as associative learning [47] may be of value to the JEA pipeline as well. A promising approach for future SSL directions lies in the usage of energy-based models (EBMs). These models may be capable of more fine-granular analysis of data distributions, and are thus in the focus of various visionary works, such as the one of Yann LeCun

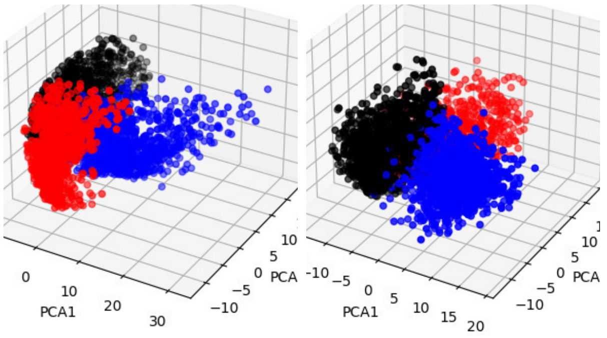
[48]. Restricted Boltzmann Machines are one very rudimentary implementation of such EBMs and have been studied in the literature, even though their capabilities in computer vision are very limited, as they model the data distribution, which is rarely feasible for images of a real-world size [49], [50]. Similar as with RBMs, self-organizing maps [51] are a popular choice, especially in life sciences and genetics, but have come unfashionable due to their poor scalability.

III. Qualitative and Quantitative Model Evaluation

To get a more profound understanding of the similarities and differences of SSL vs. supervised training, we analysed the observations and findings, and collected implications for SSL's applicability to real-world problems. We compared eleven different models, as depicted in Table I using three of the classes taken from the MNIST dataset, as well as various thousands images of cats, dogs, and flowers taken from the Oxford IIT-PET and Oxford IIT-Flower dataset. Note that each custom data set contains an obvious task and a more challenging task, as well as a slightly imbalanced data amount between the classes (e.g., more flowers than pets). For the MNIST data set, these are the separation of the digit four from eight and zero, and the separation of the similar digits zero and eight itself. Analogous, the Oxford-based data needs to be categorized into flowers and pets, as well as cats and dogs. The parameters for each training have been optimized in a grid search. For the qualitative analysis, the extracted latents were further reduced using a principal component analysis (PCA) and inspected. Similar results could be achieved for all models, except for the model using spatial transformations, as it struggled in finding a suitable solution. Fig. 3 shows the results of the highest accuracy models, as referred in Table II, for the MNIST dataset. It's evident that both models captured expressive filters and can successfully separate all the three digit types. The features extracted with the baseline model (SimCLR), however, seemed to use the latent space more efficiently, as they are denser. For the remaining models except the one with spatial transforms, distinguishable classes could be observed as well, while clusters were more entangled than in the presented examples.

TABLE I: Overview of implemented approaches and the subcategories. We implemented the particular methods as suggested in the referenced literature.

Approach	Group	Subgroup	References
AE Baseline	AE	-	[9]
AE + clustering	AE	Latent Variations	[21]
AE + ensemble	AE	Backbone Variations	[17], [19]
AE + input variations	AE	Input Variations	[17]
JEA baseline (SimCLR)	JEA	-	[11]
JEA + distillation	JEA	Backbone Variations	[27] w.o ViT
JEA + spatial transforms	JEA	Input Variations	[30]
JEPA	JEA	Latent Variations	[34]
MSN	JEA	Latent Variations	[34]
SwAV	JEA	Projection Variations	[52], [36]
Dino	JEA	Temporal Component	[53]

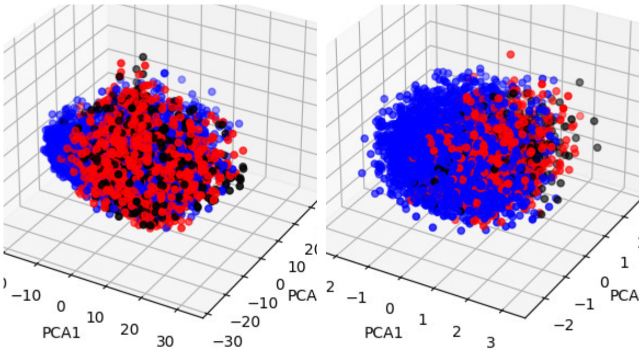


(a) AE with clustering. (b) SimCLR.

Fig. 3: Features of a selection of MNIST samples (red – 0, black – 4, blue – 8), when converted to a three-dimensional space using the PCA.

For the Oxford datasets, qualitative results could be found to be worse than the ones for the MNIST dataset. This was not surprising, given the fact that the data contains far more complex objects. As shown in Fig. 4, the significant difference of flowers vs. pets could generally be recognized in the latent space's clusters, while the pets' features themselves yield only strongly entangled clusters. It's noteworthy to say that further compression through PCA possibly amplified the entangling, especially in the case of SimCLR. The remaining approaches showed similar behaviour for the Oxford datasets' qualitative analysis. The main difference was the amount of entangling.

From the qualitative analysis, three major observations could be made about SSL's capabilities. First, more complex datasets very likely require a higher number of samples, to guarantee finding a minimum on the error surface during training. Furthermore, fine-granular distinction of different data point classes, such as "cat" and "dog", may be challenging for both the AE and JEA setup. In addition, no significant difference between AE and SSL models could be observed in qualitative analysis, opposing to the results presented in further sections of this paper. The differences among the reviewed



(a) AE with clustering. (b) SimCLR.

Fig. 4: Features of a selection of Oxford-Pets+Oxford-Flowers samples (red – cat, black – dog, blue – flower). In contrast to the MNIST data, the fine-grained task couldn't be solved.

modification types were investigated deeper, by evaluating model linear classification performance on the dataset's test cohorts, after pretraining with SSL. For the MNIST dataset, similar to the qualitative results, all approaches, except the spatial transforms, achieved results comparable to state-of-the-art models. The spatial transforms likely fail to achieve a good result, as the learned augmentations do not contribute to finding similar vector representations in the JEA setups. This effect is similar to the one observed by Li et al. [42] who conclude that only certain augmentation techniques are suited for their training task.

Overall, the AE with auxiliary (i.e., clustering) task performed best, in terms of absolute accuracy, with 99.87% (c.f. Table II). The JEA approaches achieved similarly good results, whereas SwAV even had the highest improvement compared to the supervised pendant of 1.37%. The results of the quantitative performance analysis thus matched the ones of the qualitative analysis. As the MNIST dataset could even be learned by linear machine learning models such as support vector machines, this finding can only be seen as proof of concept for the implemented approaches. For the Oxford data, more differences could be observed among different models. The previously top-performing AE with clustering only achieved a poor accuracy of 49.44% (c.f. Table II). Further investigation strongly suggested that the model failed to learn a meaningful distribution to generate data, thus the clustering loss was blocking the training, rather than helping, by creating confusing samples. Generally, the generative AE approaches performed worse than the JEA ones. This is likely because data complexity was too high for the small number of samples. As this is an unavoidable condition for numerous practical scenarios, generative AE approaches showed a systematic weakness here.

TABLE II: ΔAcc. (Accuracy – Supervised Accuracy) of different models on the MNIST and Oxford datasets, within a 5-fold cross-validation downstream test, when trained using SSL and in a supervised fashion.

Dataset	Approach	Acc. (%)	ΔAcc. (ppt.)	$\frac{t \cdot s^{-1}}{epoch}$
MNIST	AE baseline	99.46	0.03	29
	AE + clustering	**99.87**	1.29	36
	AE + ensemble	91.89	-6.87	87
	AE + input variations	99.31	1.21	153
	JEA + distillation	87.82	-8.65	47
	JEA + spatial transforms	50.44	-46.43	66
	SimCLR	99.72	1.36	58
	JEPA	91.45	-6.22	30
	MSN	93.21	-5.12	103
	SwAV	97.40	**1.37**	90
	Dino	90.36	-9.20	260
Oxford	AE Baseline	49.44	-20.26	17
	AE + clustering	55.82	-16.25	17
	AE + ensemble	56.27	-24.98	32
	AE + input variations	59.01	-21.1	46
	JEA + distillation	62.02	**-7.43**	21
	JEA + spatial transforms	34.32	-44.48	25
	SimCLR	**72.80**	-8.44	36
	JEPA	57.50	-24.19	17
	MSN	56.30	-25.39	58
	SwAV	55.40	-26.29	39
	Dino	61.90	-8.00	65

Among the JEA approaches, surprisingly, the baseline model performed best, with 72.80% accuracy. It's noteworthy to say that JEPA and MSN, however, took the shortest training time, as they already showed stable results after a few ten epochs. JEPA also had the fastest processing time per epoch. The distillation scenario showed the least difference to the supervised pendant, but similar to the generative AE approaches, the dataset size was possibly too small. The ViT-based Dino model likely suffers from the same problem. For the SwAV model, loss froze at an early stage of the training, at around epoch 50. This indicates that only a trivial solution was found in swapped prediction problem. Thus, training loss failed to converge to the global minimum. The effect could be compensated slightly by using a buffer to collect multiple batches, which increased the batches' variability, before solving the swapped prediction problem. The results were, however, inferior. The SimCLR approach, overall, could be found to work reasonably well, when compared to the supervised pendant. This also aligns with the findings of Zhong et al. [54].

The results on the Oxford datasets, opposing to the ones of the MNIST data, showed that systematic problems occurred for all the presented approaches, given a higher complexity of the data. Thus, the confusion matrices of the models we're examined, to see what the limitations are. Table III shows the result for the AE with clustering task, which was performing best on the MNIST dataset, and the JEA baseline. The AE with clustering model achieved a correct classification of flowers against the pets of 91%. For the more challenging part of distinguishing the pets themselves as cats, or respectively dogs, the model, however, performed only slightly better than a random classifier. The JEA Baseline provides better results with TP rates of 47, 77 and 95% for cats, dogs, and flowers. Similar to the AE, the less challenging task was solved remarkably well, while the dog vs. cat problem was characterized by a strong overfitting to the dog class.

TABLE III: Confusion matrix of the AE with latent variations and the JEA baseline on the Oxford-Flower+IIIT-Pet dataset.

	AE with clustering			SimCLR		
Pred / True	Cat	Dog	Flower	Cat	Dog	Flower
Cat	0.23	**0.45**	0.31	**0.47**	0.24	0.29
Dog	0.16	**0.53**	0.31	0.07	**0.77**	0.17
Flower	0.03	0.06	**0.91**	0.01	0.04	**0.95**

As the training data only contained a few thousand images per class, the low data amount, in contrast to results created on the ImageNet dataset, seemed to be a limitation for all approaches. It's also noticeable that even more sophisticated JEA approaches generated worse solutions, as the results of the remaining models aligned with the ones of the clustering AE. An explanation for this may be that the dataset provided too few inputs to solve the optimization problem for a sufficient number of parameters, given the relatively small dataset size. This was especially the case for multi-loss training such as Dino, SwAV and JEA with distillation. The problem is likely caused by the non-contrastive loss-term. For example, in the clustering-based approaches (clustering AE and SwAV), the optimization of the clustering problem is only possible, if the batches are large enough at each step and have enough variability between the epochs. As this is not necessarily the case using a small dataset, the clustering-loss part will hinder training and eventually cause convergence to a suboptimal or trivial solution. Analogously, the second loss term of Dino and plain distillation do not contribute positively to the training. The AEs with ensemble and input variations showed improvements, when compared to the AE baseline, but couldn't keep up with the more capable JEA.

IV. Future Perspectives for SSL

From our results, we conclude that SSL can open new perspectives for future AI research. Even though the validated architectures failed to solve the most fine-grained task without specific modifications (i.e., separating pets and flowers of the Oxford data), various practical applications can be found in the literature that serve as a proof-of-concept for adapting SSL in industry and medical applications. In most of the cases identified in the literature, at least SSL-pretraining was helpful to extract more robust features or even such ones, that supervised learning was missing.

Future work should concentrate on facilitating the implementation of SSL in practice. Additionally, it should be investigated, how SSL can help in curating and understanding data, rather than simply using it as a tool to pretrain a model. Figure 5 shows such a scenario. After initially processing the unlabelled data, latents get annotated by an expert. To decrease the annotation burden, suggestions for annotation can be made by group selection of latents or labelling closest neighbours. That way, a larger dataset could be annotated using only a few examples. For uncertain latents, e.g., those that show similar logits for all pseudo classes, concepts such as active learning may be used for sampling. This way, experts could focus on conceptual work and result validation, rather than on searching for significant/anomalous data. Such a tool could bring major improvements to scenarios such as clinical applications, where initial annotation will likely result in subjective bias, or in scenarios where a-priori annotation is unfeasible due to the data set size.

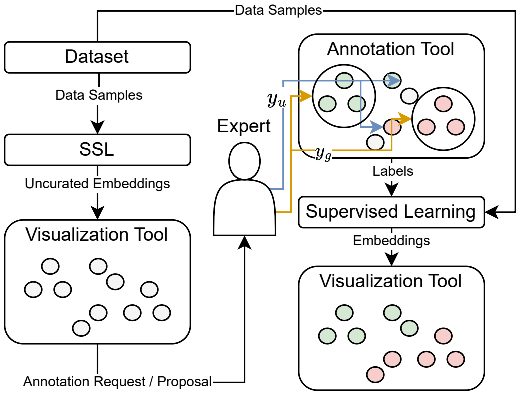


Fig. 5: SSL as a tool for data annotation.

V. Conclusions

In this work, we presented a comparative overview of current SSL methods. We conclude that SSL is a promising method to change the paradigms of machine learning, even though none of the approaches yet achieves identical or better performance than the supervised pendant on more complex datasets. The models solved less challenging tasks without problems, and showed promising initial results for more challenging tasks. Overall, they provided good baseline results that suggest SSL may be capable of achieving or surpassing performance of supervised training.

Regarding the training setups, we acknowledge that grid search may be a suboptimal choice to get optimal accuracies, especially as models with fewer hyperparameters benefit more from grid search than more complex ones due to search complexity. The performance results thus can only be seen as an indicator for certain characteristics of SSL approaches.

In addition to presenting the reviewed work, we performed an experimental validation using a selection of approaches. From the qualitative analysis, we conclude that all models generally capture important information from the data. The models, however, failed to solve the more challenging task. The latter finding is supported by the results of the performance analysis. Among all approaches, JEA methods outperformed generative (AE) ones. The SimCLR approach even showed a rudimentary solution to the challenging task of separating cats from dogs.

The biggest problem for all the models still seemed to be related to small dataset size. This may be the most significant weakness of SSL, as this condition typically can't be compensated. Additionally, many hyperparameters such as temperature [11], patch size (in case of using ViTs) or batch size need to be examined when using SSL (c.f. [3]). Especially the fact that approaches such as SimCLR, SwAV, MSN and JEPA require gigantic batch sizes of more than 1000 images, unnecessarily limits SSL applicability as multi-GPU clusters will be needed for calculation, with a data set of according size. In practice, this means a high technical burden to implement such a solution. Therefore, future research should focus on small-dataset SSL, that also works under real-world conditions, rather than focusing on ImageNet benchmarks. Additionally, more effort should be spent to understand structural differences between supervised and SSL models, and the exact effects leading to this behaviour.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 248–255. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2009.5206848

[2] M. Assran, R. Balestriero, Q. Duval, F. Bordes, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, and N. Ballas, "The hidden uniform cluster prior in self-supervised learning," *CoRR*, vol. abs/2210.07277, 2022.

[3] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," *CoRR*, vol. abs/2304.12210, 2023.

[4] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012. [Online]. Available: http://dx.doi.org/10.1109/MSP.2012.2211477

[5] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.42

[6] P. Omkar, M., V. Andrea, Z. Andrew, and J. C., V., "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2012.6248092

[7] W. M. Bramer, G. B. D. Jonge, M. L. Rethlefsen, F. Mast, and J. Kleijnen, "A systematic approach to searching: an efficient and complete method to develop literature searches," *Journal of the Medical Library Association*, vol. 106, no. 4, Oct. 2018. [Online]. Available: http://dx.doi.org/10.5195/jmla.2018.283

[8] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, dec 2010. [Online]. Available: http://dx.doi.org/10.5555/1756006.1953039

[9] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[10] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *CoRR*, vol. abs/1511.05644, 2015.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020.

[12] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," *CoRR*, vol. abs/2103.03230, 2021.

[13] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," *CoRR*, vol. abs/1603.09246, 2016.

[14] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," *CoRR*, vol. abs/1803.07728, 2018.

[15] Z. Wu, Y. Xiong, S. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance-level discrimination," *CoRR*, vol. abs/1805.01978, 2018.

[16] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," *CoRR*, vol. abs/1912.01991, 2019.

[17] L. Ternes, M. Dane, S. Gross, M. Labrie, G. Mills, J. Gray, L. Heiser, and Y. H. Chang, "A multi-encoder variational autoencoder controls multiple transformational features in single-cell image analysis," *Communications Biology*, vol. 5, no. 1, 2022. [Online]. Available: http://dx.doi.org/10.1038/s42003-022-03218-x

[18] W. Xiong, L. Zhang, B. Du, and D. Tao, "Combining local and global: Rich and robust feature pooling for visual recognition," *Pattern Recognition*, vol. 62, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2016.08.006

[19] S. Zhang, M. Xu, J. Zhou, and S. Jia, "Unsupervised spatial-spectral cnn-based feature learning for hyperspectral image classification," *IEEE Transactions on Geoscience & Remote Sensing*, 2022. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2022.3153673

[20] C. Vununu, S.-H. Lee, and K.-R. Kwon, "A strictly unsupervised deep learning method for hep-2 cell image classification," *Sensors (14248220)*, vol. 20, no. 9, 2020. [Online]. Available: http://dx.doi.org/10.3390/s20092717

[21] V. Prasad, D. Das, and B. Bhowmick, "Variational clustering: Leveraging variational autoencoders for image clustering," *CoRR*, vol. abs/2005.046132, 2020.

[22] J. Guérin, S. Thiery, E. Nyiri, O. Gibaru, and B. Boots, "Combining pretrained cnn feature extractors to enhance clustering of complex natural images," *Neurocomputing*, vol. 423, 2021. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2020.10.068

[23] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.556

[24] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pre-training or strong data augmentations," *CoRR*, vol. abs/2106.01548, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[26] H. Dong, L. Zhang, and B. Zou, "Exploring vision transformers for polarimetric sar image classification," *IEEE Transactions on Geoscience & Remote Sensing*, 2022. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2021.3137383

[27] X. Wang, J. Zhu, Z. Yan, Z. Zhang, Y. Zhang, Y. Chen, and H. Li, "Last: Label-free self-distillation contrastive learning with transformer architecture for remote sensing image scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022. [Online]. Available: http://dx.doi.org/10.1109/LGRS.2022.3185088

[28] W. Zhou, Y. Hou, K. Ouyang, and S. Zhou, "Exploring complementary information of self–supervised pretext tasks for unsupervised video pre–training," *IET Computer Vision (Wiley-Blackwell)*, vol. 16, no. 3, 2022. [Online]. Available: http://dx.doi.org/10.1049/cvi2.12084

[29] J. Ding, E. Xie, H. Xu, C. Jiang, Z. Li, P. Luo, and G.-S. Xia, "Deeply unsupervised patch re-identification for pre-training object detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2022.3164911

[30] Y. Li, S. Kan, J. Yuan, W. Cao, and Z. He, "Spatial assembly networks for image representation learning," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 871–13 880. [Online]. Available: http://dx.doi.org/10.1109/CVPR46437.2021.01366

[31] L. Fan, S. Liu, P.-Y. Chen, G. Zhang, and C. Gan, "When does contrastive learning preserve adversarial robustness from pretraining to finetuning?" *CoRR*, vol. abs/2111.01124, 2021.

[32] P. Feng and H. Zhang, "Self-supervised image hash retrieval based on adversarial distillation," in *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, 2022, pp. 732–737. [Online]. Available: http://dx.doi.org/10.1109/CACML55074.2022.00127

[33] M. Assran, M. Caron, I. Misra, P. Bojanowski, F. Bordes, P. Vincent, A. Joulin, M. Rabbat, and N. Ballas, "Masked siamese networks for label-efficient learning," *CoRR*, vol. abs/2204.07141, 2022.

[34] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," *CoRR*, vol. abs/2301.08243, 2023.

[35] J. Yan, H. Chen, X. Li, and J. Yao, "Deep contrastive learning based tissue clustering for annotation-free histopathology image analysis," *Computerized Medical Imaging & Graphics*, vol. 97, pp. N.PAG–N.PAG, 2022. [Online]. Available: http://dx.doi.org/10.1016/j.compmedimag.2022.102053

[36] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *CoRR*, vol. abs/2006.09882, 2020.

[37] A. Gomez-Villa, B. Twardowski, L. Yu, A. D. Bagdanov, and J. van de Weijer, "Continually learning self-supervised representations with projected functional regularization," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2022, pp. 3866–3876. [Online]. Available: http://dx.doi.org/10.1109/CVPRW56347.2022.00432

[38] H. Kahng and S. B. Kim, "Self-supervised representation learning for wafer bin map defect pattern classification," *IEEE Transactions on Semiconductor Manufacturing*, vol. 34, no. 1, 2021. [Online]. Available: http://dx.doi.org/10.1109/TSM.2020.3038165

[39] W. Dai, M. Erdt, and A. Sourin, "Self-supervised pairing image clustering for automated quality control," *Visual Computer*, vol. 38, no. 4, 2022. [Online]. Available: http://dx.doi.org/10.1007/s00371-021-02137-y

[40] C.-H. Yeh, C.-Y. Hong, Y.-C. Hsu, and T.-L. Liu, "Saga: Self-augmentation with guided attention for representation learning," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3463–3467. [Online]. Available: http://dx.doi.org/10.1109/ICASSP43922.2022.9747302

[41] P. Yin, L. Qi, X. Xi, B. Zhang, and H. Qiao, "Nflb dropout: Improve generalization ability by dropping out the best -a biologically inspired adaptive dropout method for unsupervised learning," in *2016*

International Joint Conference on Neural Networks (IJCNN), 2016, pp. 1180–1186. [Online]. Available: http://dx.doi.org/10.1109/IJCNN.2016.7727331

[42] X. Li, X. Hu, X. Qi, L. Yu, W. Zhao, P.-A. Heng, and L. Xing, "Rotation-oriented collaborative self-supervised learning for retinal disease diagnosis," *IEEE Transactions on Medical Imaging*, vol. 40, no. 9, 2021. [Online]. Available: http://dx.doi.org/10.1109/TMI.2021.3075244

[43] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019.

[44] K. Pang, K. Li, Y. Yang, H. Zhang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Generalising fine-grained sketch-based image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2019.00077

[45] J. Lu, L. Li, and C. Zhang, "Self-reinforcing unsupervised matching," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 44, no. 8, 2022. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2021.3061945

[46] X. Fang, Y. Cai, Z. Cai, X. Jiang, and Z. Chen, "Sparse feature learning of hyperspectral imagery via multiobjective-based extreme learning machine," *Sensors (14248220)*, vol. 20, no. 5, 2020. [Online]. Available: http://dx.doi.org/10.3390/s20051262

[47] J. Liu, M. Gong, and H. He, "Deep associative neural network for associative memory based on unsupervised representation learning," *Neural Networks*, vol. 113, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2019.01.004

[48] Y. LeCun, "A path towards autonomous machine intelligence," *under review*, 2022.

[49] J. Zhang, H. Wang, J. Chu, S. Huang, T. Li, and Q. Zhao, "Improved gaussian–bernoulli restricted boltzmann machine for learning discriminative representations," *Knowledge-Based Systems*, vol. 185, pp. N.PAG–N.PAG, 2019. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2019.104911

[50] B. Xiaojun and W. Haibo, "Contractive slab and spike convolutional deep boltzmann machine," *Neurocomputing*, vol. 290, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2018.02.048

[51] M. Sakkari, M. Hamdi, H. Elmannai, A. AlGarni, and M. Zaied, "Feature extraction-based deep self-organizing map," *Circuits, Systems & Signal Processing*, vol. 41, no. 5, 2022. [Online]. Available: http://dx.doi.org/10.1007/s00034-021-01914-3

[52] P. Goyal, Q. Duval, J. Reizenstein, M. Leavitt, M. Xu, B. Lefaudeux, M. Singh, V. Reis, M. Caron, P. Bojanowski, A. Joulin, and I. Misra, "VISSL," https://github.com/facebookresearch/vissl, 2021.

[53] S. H. Lee, S. Lee, and B. C. Song, "Vision transformer for small-size datasets," *CoRR*, vol. abs/2112.13492, 2021.

[54] Y. Zhong, H. Tang, J. Chen, J. Peng, and Y.-X. Wang, "Is self-supervised learning more robust than supervised learning?" *CoRR*, vol. abs/2206.05259, 2022.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 103–110
DOI: 10.15439/2023F3888
ISSN 2300-5963 ACSIS, Vol. 35

CADM: Big Data to Limit Creative Accounting in Saudi-Listed Companies

Maysoon Bineid
0009-0007-4447-3706
Department of IS and Technology University of Jeddah, SA Department of Informatics
University of Sussex, Brighton.
Falmer, BN1 9RH, UK
Email: M.bineid@sussex.ac.uk

Natalia Beloff
0000-0002-8872-7786
Department of Informatics University of Sussex, Brighton.
Falmer, BN1 9RH, UK
Email: N.Beloff@sussex.ac.uk

Anastasia Khanina
School of Business and Law
University of Brighton
Brighton, BN2 4NU, UK
Email: A.Khanina@brighton.ac.uk

Martin White
0000-0001-8686-2274
Department of Informatics
University of Sussex, Brighton.
Falmer, BN1 9RH, UK
Email: M.White@sussex.ac.uk

***Abstract*—Global financial scandals have demonstrated the harmful impact of creative accounting, a practice where managers creatively manipulate financial reports to conceal a company's actual performance and influence stakeholders' decision-making. Studies showed that Saudi-listed companies use it in preparing financial statements. Despite posing a significant risk to the Saudi financial market, detecting it using ordinary auditing procedures remains challenging. Big data analytics has provided practical applications in auditing, and recently, the employment of Deep Learning in fraud detection has delivered remarkably accurate results. Still, limited research has considered it in detecting creative accounting. This study proposes a novel framework using a hybrid learning approach. It suggests training on a simulated dataset of financial statements prepared (i.e., deliberately manipulated) based on financial statements available in the literature for supervised learning. It is then tested on real-world financial reports from the Saudi Open Data and Saudi Statistics. Our framework contributes to the literature with a new governing approach to limit creative accounting and improve financial reporting quality.**

***Index Terms*—Creative Accounting, Big Data, Deep Learning.**

I. Introduction

CREATIVE accounting (CA) practices have negatively affected the financial reporting quality and disturbed the trust in the information extracted from financial statements. The issue with CA is that it does not necessarily violate the International Financial Reporting Standards (IFRS), yet it has the same severe consequences as Financial Statement Fraud (FSF) [1]. Besides, CA is more enigmatic and almost impossible to detect using traditional auditing techniques. Some studies consider CA and Earnings Management (EM) as FSF, while others identify the thin line between them [1]. Another harmful practice that could be identified along CA and FSF is Window Dressing (WD), where managers invest in the freedom of interpretation area to manipulate the presentation of reports. The term 'creative' gives a positive impression about the practice whereas, in reality, this practice has been considered to be the primary cause behind many financial scandals such as Enron, WorldCom, in the U.S., and Parmalat, Royal Ahold, and Vivendi Universal in Europe [2][3][4][5][6][7][8]. These incidents confirm that account manipulation is designed to gain a temporary benefit, eventually leading to financial scandals and substantial losses. In Saudi Arabia, cases of CA exist and, according to the literature, employ the same accounting techniques for similar reasons. Considering the proposition that less efficient markets tend to have greater tolerance to manipulations, the weak-form efficiency of the Saudi stock market Tadawul, as proved by [9], indicates the high possibility of manipulations. Many studies have investigated the practice in the region, but non include real-time case studies [10][11] [12][13]. The results of these studies agreed that financial statements do not represent the true and fair position of a company, although being approved by auditing procedures.

However, big data analytics and AI models are currently employed in different business sectors. Many models and techniques have been developed and validated to replace or supplement traditional accounting and auditing procedures. In our context, the literature is rich with significant contributions in FSF detection using data mining and machine learning ML. The availability of data types like financial (FIN) and non-financial (N-FIN) and the possibility to include these data types in advanced intelligent models motivated researchers to develop many applications that meet business needs. Indeed, detecting CA using traditional techniques (e.g., accrual-based detection) requires non-public, inaccessible, and time-consuming data to reach [14]. On the other hand, the literature is rich with examples of Machine Learning (ML) and Deep Learning (DL) models that can learn from publicly available FIN and N-FIN data and produce predictions and results with high accuracy.

The use of big data analytics made it possible to include N-FIN data and aggregate accounting data with other sectors' data. Consequently, accounting results are now more accurate and credible [15]. Moreover, big accounting firms are adopting server-based platforms that support auditors by implementing real-time financial data collected directly from clients. For example, in April 2022, *PwC* announced the use of a new cloud-based auditing platform named *Aura* that provides many services powered by advanced analytics. These capabilities motivate us to further add to the capacity

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

and efficiency of these analytical tools for enhanced detection procedures.

This study aims to overcome the misrepresentation of information in financial reports in Saudi Arabia and improve financial reporting quality by proposing a framework for the Creative Accounting Detection Model (CADM). The model suggests the employment of a Hybrid Deep Learning (HDL) that implements Artificial Neural Network (ANN), Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM). It uses FIN and N-FIN data related to selected Saudi-listed companies in different sectors. CADM introduces a new approach to evaluate published financial information more accurately and limit CA. It will also create new insights into the quality of financial statements and indicate the extent to which these practices are incorporated.

II. Research Background

A. Overview of CA

CA is a term to describe the accounting procedures used to present an enhanced image of an enterprise that misleads users [16]. The word '*creative*' means using new expertly invented ways of preparing accounts [6] to make the company more appealing to stakeholders without committing fraud. The practice is considered legal within IFRS but deviates from its goal and spirit as it operates in the grey area between legitimacy (in the context of IFRS) and fraud. As in *Fig 1.*, CA can exceed the regulations and become fraudulent, yet in this case easier to detect.

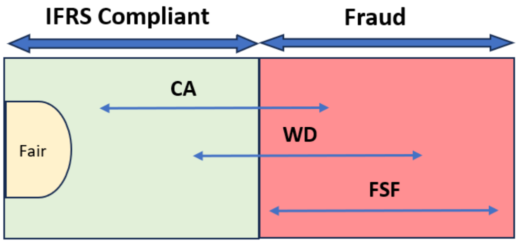


Fig 1. Financial Accounting and Its Relationship with IFRS

The quality of financial reporting (measured by different methods such as discretionary accruals, accounting conservatism, and asymmetry of information [17]) is significantly affected by CA practices [18]. On the international scale, more financial problems have been discussed in the literature [19], such as low liquidity, tightening credit conditions, and price volatility. However, the literature has no particular definition for the practice [20]. Yet the most recent definition by [6] described these practices as follows:" *They are the methods which deviate from the rules and regulations, it is an excessive complication and use of innovative ways to visualise income, assets, and liabilities, it is an innovative and aggressive way of reporting financial statements, it is a systematic misrepresentation of the true and fair financial statements*." It can be concluded that any accounting procedure meant to present non-realistic financial information is a form of CA.

B. CA Incentives

Incentives differ according to the business type and size. The reason for a manager in a private company to engage in this practice is different from the reason for a manager in a public company since the practice has different impacts on various parties, and sometimes more than one party in the same incident. Regardless of all the different impacts, incentives have been initiated from the Agency Problem (AP). Accordingly, we classify the incentives based on three main areas of primary impacts: internal affairs, stock market, and third-party's decisions., as shown in Table I. Whether these incentives existed from personal, organisational, or political backgrounds, they ultimately motivated report preparers individually or in groups to engage in the CA practice.

Table I.
Classification of Creative Accounting Incentives

Incentive impact	Examples	Studies
Internal affairs	Share options, personal satisfaction, job security	[1]
	Remuneration and management compensation (bonuses)	[56], [57], [58],
	Political costs (tax and zakat), lawsuits	[59], [56]
Stock market	Stock prices, Information asymmetries conflict	[60], [61], [62]
	Debt covenants	[58], [63]
Third party's decision	Financial forecasts	[64], [61], [65]
	Political costs (tax and zakat), lawsuits	[59], [56]

C. CA Techniques

The financial report preparation involves many techniques that can be innovatively different from one firm to another. There are rules and policies for these techniques, yet managers apply the approach that fulfils their personal or institutional interests. However, CA techniques can be categorised regarding the report type being prepared [21], the accounting area operating in [12][22], the accounting items being used [23], and the type of creativity being applied [24]. This study categorises CA techniques using two layers: the relative accounting area, as in [12], and the accounting items, as in [23]. Table II shows accounting techniques categorised by both approaches and highlights the techniques applied in this study.

D. CA Detection

The recognition of CA practices as a crime is quite an argument [19] [2] [6] [21]. Therefore, big data analytics studies were limited to FSF detection [25][26] and FSF prediction models [27]. Earlier FSF detection models, like the M-score model [28], were mainly quantitative depending on numerical FIN data. In contrast, recent models are qualitative intelligent models (i.e., ML models) that have proved to outperform the

TABLE II.
CA TECHNIQUES

Area	Accounting items	Technique
Classification	Extraordinary items	Including ordinary items
	Contingent liabilities	Misclassifying contingent liabilities
	Cost	Capitalization of costs
Disclosure	Financing	Off-balance-sheet financing
		Pre-acquisition write-down
	Cashflow	Reporting for cashflow
	Special Purpose Entities	Deferred considerations
	Accounting choice	Failure to justify accounting choice
	Pension fund	Failure to disclose
Timing	Revenue	Revenue recognition
	Accruals	Overvaluing/ undervaluing
Estimation	Intangible assets	Overestimating intangible assets
	Currency	Currency mismatching
	Assets and liabilities	Misreported assets and liabilities

earlier models [29][30][31] [32]. As revealed by [33], fraud detection has the highest percentage (39.4%) of published studies in the Journal of Emerging Technologies in Accounting *JETA*.

Since the speed of uncovering FSF can limit its consequences [8], the need to find faster and more accurate models is becoming essential. Unfortunately, the literature has no research on detecting CA as an IFSR practice using big data analytics as far as this study. Due to the ambiguous nature of the practice and the sophisticated historical actions involved, no specific financial ratios or traditional mathematical model can accurately detect it. However, FSF prediction scenarios can be considered in our proposal for many reasons. First, FSF prediction models use historical FSs (i.e., time-series dataset) labelled as fraudulent to learn from and reflect which variables can be used to classify the case. This can help predict future fraud or financial distress. Presuming that CA practices eventually lead to FSF (e.g., Enron started using SPEs legally, then it became 'increasingly doubtful' over time [1]), CA detection has similar domain characteristics of predicting FSF. It can be adapted to detect further activities that don't exceed IFRS limits. Another reason is that both procedures aim to prevent future fraud and provide alerting flags that don't normally appear to stakeholders in their usual financial information reviews.

E. The case of Saudi Arabia

According to the ACEF occupational fraud report in 2022, Saudi Arabia was the second-highest number of occupational fraud cases in the middle east [34]. Many studies investigated the problem in Saudi-listed companies. Most of these studies were quantitative, as opinions from accounting academics and professionals were gathered and analysed for a better understanding of the incentives and techniques [35][36][4][37][13]. Still, some studies were empirical papers focused on one business type [11][35] or one business size [36][13], applying a detection model with successful results. However, the application of big data and ML in the Saudi business research domain was focused on marketing[38] and finance[39] but never on accounting. On the other hand, accounting professionals have been using ML models, especially in some audit procedures has been employed in accounting firms. Apart from The Big Four[1], although these platforms guarantee efficiency for both the client and the auditing firm itself, they are not used due to cost and technical training limitations.

Recently, Saudi Arabia has been through several economic transformation steps that influenced the governing materials of accounting. An instance of these steps is joining the World Trade Organization (WTO) and adopting the International Financial Reporting Standards (IFRS) [40]. In addition, during the last 20 years, the Kingdom has established new institutions to regulate businesses and control the Saudi stock market. Some of these institutions were specifically designed to fulfil an essential Saudi vision for the future: *Vision 2030*[2] [41]. An instance of these institutions is the Saudi Data and Artificial Intelligence Authority (SADAIA), which has provided many valuable services and facilities for market and academic research. It is a helpful attempt to support the effort to improve financial reporting and leverage the Saudi business environment with innovative technologies through adequate investment in the country's prospects.

III. BIG DATA IN ACCOUNTING AND AUDITING

The accounting literature is rich with innovative analytical models that have the potential to enrich the accounting environment, develop accounting regulations and reduce the profession's defects [42]. The JETA has 51.5% of its publications between 2005-2015 on data analytics [33]. Moreover, 13% of published research on emerging technologies in the accounting domain was about big data analytics [43]. Although studies address the limitation in the literature regarding the use of big data in accounting [44][45], there exists a consensus that

[1] A term used to refer to the 4 big accounting firms: PwC, KPMG, EY, and Deloitte.

[2] https://www.vision2030.gov.sa/

ML algorithms in big data used in accounting research are growing remarkably [46], providing new services to the business environment that are never possible before. Further, standardising accounting data through new unified formats like XBRL and secure data structures like Blockchain added promising opportunities for efficient research and improved accounting outcomes.

By professional means, accounting and auditing embrace big data analytics in different procedures. The Big 4 are investing heavily in data analytics and artificial intelligence [47] and promoting embracing big data technologies. For instance, the recent adaptation of the *Halo* online platform by *PwC* implemented the inclusion of whole population analysis, which outperforms the sampling techniques that are usually used in auditing procedures along with many recalculations and risk assessment tools (e.g., journal entry testing and general ledger analysis) that become possible by its enhanced connectivity and high server-based processing capabilities. Moreover, regulatory agencies' results have been enhanced by incorporating non-financial data as a supplement to the traditional financial data in their systems (e.g., the UK government's tax authority uses different sources of data from the internet, social media, land registry records, international tax authorities, and banks [48]).

The research on the application of big data in accounting (summarised in TABLE III) is more focused on auditing. The analytical nature of auditing procedures made it more likely to benefit from these applications. Many ML frameworks were used in the research giving insightful results. Each framework used different models, datasets, and features for multiple objectives. However, recent studies in this domain used a subfield of ML, namely DL. It is recently trending in every field, particularly in accounting research, because of its potential to learn from massive amounts of data. ML and DL models can be used parallelly to build a model with improved capabilities, as in Hybrid Learning (HL). They can also be combined, and the output of one model is the input for the second, and that is called Ensembled Learning (EL) as in [49][29]. Accordingly, the training process of CADM uses an HL approach for enhanced performance. Training is performed in two stages: the feature extraction stage and the CA detection stage. Therefore, the CADM framework is designed to apply an HL approach for its suitability to our proposal. The following section will briefly describe the CADM framework and models used.

TABLE III.
BIG DATA IN FSF RESEARCH

Method	Model	Dataset/ sources	Objective	Study	Year
Data Mining	LR NN (BP) DT	Data related to fraud factors based on the fraud triangle (incentive, opportunity, attitude)	Detecting FSF	[51]	2015
	DT, SVM, K-NN, RS	FSs	Predict audit opinion	[66]	2021
ML	BERT	TXT	Analysis of AD	[67]	2021
	GLRT	Audio data (Conference calls)	Detecting FSF	[68]	2015
Hybrid ML	Neural Networks MLP, PNN	ARs, FRs, RRs FRs,	Predict audit opinion	[52]	2016
	MLogit, SVM, BN, CSL	FS, Market Variables, and Governance measures	Detect Financial misstatement with fraud intention	[69]	2016
	Meta-Learning (SG + AL)	FS + context-based data	Detect FSF	[14]	2012
	BOW + SVM	FS + TXT	Detecting FSF	[70]	2010
	BOW + HAN	FIN + TXT	Detecting FSF	[30]	2020
Ensemble ML	XGBoost	FS	Detect FSF	[29]	2023
	ADABoost, XGBoost, CUSBoosr, RUSBoost	FS	Predict FSF	[49]	2023
Hybrid DL	RNN, CNN, LSTM, GRU	Selected financial features	Detecting corporate financial fraud	[50]	2023
	RNN	FIN +non-FIN features	Detecting FSF	[71]	2021
	NN	FS	Detection of accrued EM	[72]	2023

LR: Logistic Regression, **DT**: Decision Trees, **BOW**: Bag Of Words, **EM**: Earning Management, **FR**: Financial Reports, **AR**: Auditor's Report, **RR**: Regulatory Report, **FS**: Financial Statement**, NN**: Neural Networks, **RNN**: Recurrent Neural Network**, CNN**: Convolutional Neural Network, **GRU**: Gate Recurrent Unit, **LSTM**: Long Short-Term Memory, **HAN**: Hierarchical Attention Network, **SG**: Stack Generalization, **AL**: Adaptive Learning, **BERT**: Bidirectional Encoder Representation from Transformation, **AD**: Accounting Disclosure, **SVM**: Support Vector Machine, **K-NN**: K-Nearest Neighbor, **RS**: Rough Sets, **MLogit**: Multinomial Logistic Regression, **BN**: Bayesian Network**, CSL**: Cost-Sensitive Learning.

IV. CADM Proposed Framework

This study considers previous FSF detection and prediction research in designing the CADM framework. The peculiarity of our detecting model is found in the DL models' hybrid nature and the datasets originality. Another critical point of distinction is our innovative learning process design. FSF prediction models are built using real publicly available FRs that regulators have recognised as fraudulent, whereas no labelled CA incidents are open to learning from. Consequently, this study intends to simulate manipulated accounting datasets to learn from and then test the model on real-time Saudi FSs that are publicly available.

A. *Datasets and Data Sources*

Datasets are divided into two groups: training data and test data. Training data is a simulated dataset of FSs prepared (i.e., deliberately manipulated) based on pre-FSF statements available in the literature for supervised learning. The second data set is a real-world FRs chosen from the Saudi Open Data Portal and Saudi Statistics Authority (for price history and market information), companies' websites (for reports on corporate governance information, marketing information, discretionary disclosures, and investments), corporation social media accounts, Saudi Press Agency, Zakat, Tax and Customs Authority (ZATCA) (for historical levels of compliance), and Capital Market Authority (CMA) (for observations and statistics about the company). Both training and testing datasets consist of two types of data: FIN and N-FIN, from relevant sources, as shown in Table IV. As CA evolves, the datasets will be time-series datasets of Saudi-listed companies for 2012-2022. Finally, we know the usefulness of including audio and video data types like GPS and CCTV recordings, yet we postpone their inclusion for future work.

Table IV.
Data Types and Sources

Source	Data (examples)	
	FIN	**N-FIN**
Annual Report	Financial statements, Enumerations.	Changes in Corporate Structure, Enumerations, Board Meetings, Auditor reports.
CMA	Market Values, Performance, Asset Quality Index.	Insider dealing reports, Actual Assurance.
ZATCA	Tax Compliance, Zakat Compliance.	Regulations Changes, Tax Compliance, Zakat Compliance.
Media	Business Analytics.	Lawsuits, Mergers and Acquisitions.

B. *Variables*

The data types and sources described above mean that features will be scattered between multiple types of datasets. We will select variables from prior FSF studies and test them to validate their inclusion. Researchers have tested many FIN features, such as indices and financial ratios, and N-FIN features, such as changes in the board, market-adjusted stock returns, governance measures, economic changes, and changes in regulations. Other N-FIN features can be included according to their availability in Saudi sources, such as board meetings, meeting minutes, and meeting content, as tested in [26]. As shown in Table IV., FIN and N-FIN features selection will follow the criteria in [50], [49], and [51] accordingly. For any AI model to be accurate, variables should be limited[52], though this study proposes to test extended obtainable variables and exclude some in the exploration process if needed.

C. *Modeling and Testing Methodology*

As shown in *Fig 2*, our proposed framework consists of multiple steps. First, data collection starts by accessing open Saudi databases to gather information about listed companies and then gathering FRs from available sources. Second, in the data pre-processing stage, we set up a wide selection of variables to be included (based on available datasets), and then they will be limited while training. Third, feature extraction will be performed using ML models. Then, the feature selection process that gives the best results will be chosen. Finally, we will use some DL models, such as RNN and LSTM, for training. Finally, we test the model on a real-time group dataset described in the following section.

D. *Deep Learning DL*

DL is a subfield of ML that can learn patterns and structures that reside in data and find relations in data. Many models are used in DL, but the most used models are deep belief networks DBN and convolutional neural network CNN [53], but they perform successfully for image recognition problems. Since our dataset design needs a model that can handle sequential interconnected data, we propose using ANN that can take such features as the LSTM can. In the CADM framework, we suggest using ANN and RNN to train our model using the simulated dataset. We will also recommend using LSTM for the improved performance of CADM.

E. *Long short-Term Memory LSTM*

LSTM is a type of RNN that can handle time-series datasets and include previous states in the calculation. It is commonly used in DL application research in finance, like stock price predictions and portfolio management [54]. It can remember short-term and long-term values, which makes it useful in sequential data [55]. *Fig 3* shows LSTM unit has three types of gates that regulate input. The unit receives an input and a previous state, then calculates the output and updates the memory state. This type of network is needed to train the model due to the nature of the time-series datasets we use. We argue that

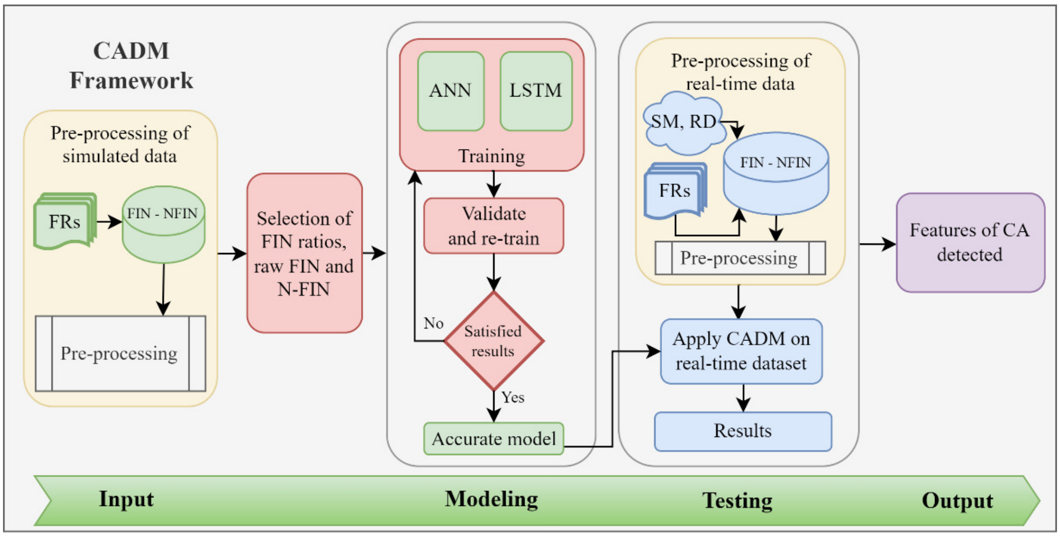


Fig 2. CADM Proposed Framework

CA techniques evolve, and the probability of detecting them increases significantly when considering changes over time.

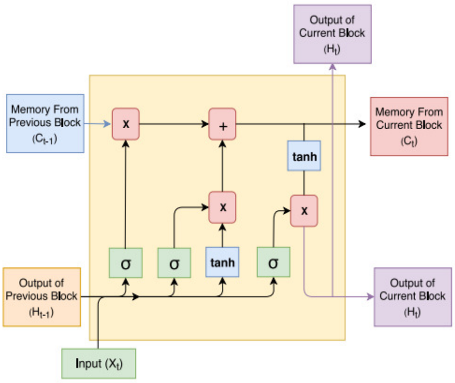

Output of Current Block (H_t)
Memory From Previous Block (C_{t-1})
x
+
Memory From Current Block (C_t)
tanh
x
x
σ
σ
tanh
σ
Output of Previous Block (H_{t-1})
Output of Current Block (H_t)
Input (X_t)

Fig 3. LSTM Structure

F. *Python and Hadoop*

Python has many efficient packages to use when dealing with big data. Our framework suggests the use of *Python* on the *Hadoop* platform. *Hadoop* is a distributed file system that scales up thousands of computers to store, process, and control big data operations. Since we plan to train a rather complex model, *Hadoop* with *Apache Spark* integration will be used as it can run distributed programs (e.g., using the *MapReduce* method) in the most reliable, error-tolerable, scalable, and portable way.

V. Conclusion

CA is proven to be harmful, and the consequences exceed the long-lasting financial and reputational damage for corporations and their CEOs, to national-level consequences like a country-wide lower level of financial market participation [8]. This paper aims to protect the Saudi financial market from the negative impacts of CA and offer stakeholders an opportunity to depend on enhanced sources of financial information for better decision-making. We introduced the framework of a new analytical model CADM that inspects the features available to detect CA used in preparing FRs. We suggested the inclusion of FIN data and N-FIN in the learning and testing stages through two DL models: ANN and LSTM. We also suggested two sources of the dataset that represent a sample of Saudi-listed companies from different sectors for a period of 10 years.

VI. Limitations and Future Work

This research is part of ongoing research on big data applications in CA. Accessibility and availability of applicable datasets remain a challenge. However, open-source Saudi platforms like SDAIA, CMA Open Data, and ZATCA Open Data are expected to provide the required dataset size to train and test our model. Additionally, the implementation and testing stages of the CADM framework may encounter some pre-processing restrictions in the translation and standardisation stages. Yet, there is a reasonable chance that most FIN data are already standardised, given that Saudi companies are obligated to publish their FSs in XBRL format. Moreover, we are preparing a short survey designed for independent auditors in Saudi Arabia to validate the suitability and accessibility of our selected variables before starting the data collection process. The next stage's outcomes are expected to unveil new insights and contribute to the existing accounting and big data literature.

References

[1] Jones Michael, *Creative accounting, fraud, and international accounting scandals*. Chichester, West Sussex, England; John Wiley and Sons, 2011.

[2] F. Tassadaq and Q. A. Malik, "Creative accounting and financial reporting: Model development and empirical testing," *International Journal of Economics and Financial Issues*, vol. 5, no. 2, pp. 544–551, 2015.

[3] R. S. Al-Shabeeb and K. R. Al-Adeem, "The Ethics of Earnings Management : A Survey Study," *Global Journal of Economics and Business*, vol. 6, no. 1, pp. 62–80, 2019.

[4] M. B. Baajajah, S. and Khalifah, "The Effect of Creative Accounting Practices on Investments Decision Makers in Saudi Stock Market," *King Abdulaziz University, Journal of Economics and Administration*, vol. 29, pp. 3–64, 2015.

[5] C. E. Rabin, "Determinants of auditors' attitudes towards creative accounting," *Meditari Accountancy Research*, vol. 13, no. 2, pp. 67–88, 2005, doi: 10.1108/10222529200500013.

[6] C. M. Gupta and D. Kumar, "Creative accounting a tool for financial crime: a review of the techniques and its effects," *J Financ Crime*, vol. 27, no. 2, pp. 397–411, 2020, doi: 10.1108/JFC-06-2019-0075.

[7] B. Yadav, "Creative Accounting: An Empirical Study from Professional Prospective," *International Journal of Management and Social Sciences Research*, vol. 3, no. 1, pp. 38–53, 2014, [Online]. Available: http://www.irjcjournals.org/ijmssr/Jan2014/8.pdf

[8] R. Cole, S. Johan, and D. Schweizer, "Corporate failures: Declines, collapses, and scandals," *Journal of Corporate Finance*, vol. 67, no. December 2020, p. 101872, 2021, doi: 10.1016/j.jcorpfin.2020.101872.

[9] B. Asiri and H. Alzeera, "Is the Saudi Stock Market Efficient? A case of weak-form efficiency," Online, 2013. [Online]. Available: https://ssrn.com/abstract=2276520

[10] M. S. Alsehli, "Earnings Management in Saudi Arabia," *Institution of Public Administration*, vol. 46, no. 3, pp. 511–546, 2006.

[11] M. Al Shetwi, "Earnings Management in Saudi Nonfinancial Listed Companies," *International Journal of Business and Social Science*, vol. 11, no. 1, 2020, doi: 10.30845/ijbss.v11n1p3.

[12] M. Bineid and A. Asiri, "Creative Accounting Incentives and Techniques in Saudi Public Companies," *KAU Journal of Administration and Economics*, vol. 2, no. 27, 2013, doi: 10.4197/Eco.

[13] A. A. Alhebri and S. D. Al-Duais, "Family businesses restrict accrual and real earnings management: Case study in Saudi Arabia," *Cogent Business and Management*, vol. 7, no. 1, Jan. 2020, doi: 10.1080/23311975.2020.1806669.

[14] A. Abbasi, C. Albrecht, A. Vance, and J. Hansen, "Metafraud: A Meta-Learning Framework for Detecting Financial Fraud," *MIS Quarterly*, vol. 36, no. 4, pp. 1293–1327, 2012, [Online]. Available: http://www.misq.org

[15] J. D. Warren, K. C. Moffitt, and P. Byrnes, "How Big Data Will Change Accounting," vol. 29, no. 2, pp. 397–407, 2015, doi: 10.2308/acch-51069.

[16] H. R. Al-bayati, "Creative Accounting and Its Role in Misleading Decision Makers," *University of Iraq*, no. 50, pp. 423–431, 2021.

[17] A. M. Malo-Alain, M. M. A. H. Melegy, and M. R. Y. Ghoneim, "The effects of sustainability disclosure on the quality of financial reports in Saudi business environment," *Academy of Accounting and Financial Studies Journal*, vol. 23, no. 5, pp. 1–13, 2019.

[18] R. A. A. Ahmed, "The Role of Corporate Governance to Limit creative Accounting Practices and Its Effect on The Qulity of Accounting Information," *Global Journal of Economics*, vol. 17, no. 1, pp. 1–16, 2019.

[19] D. S. Gherai and D. E. Balaciu, "From Creative Accounting Practices And Enron Phenomenon To The Current Financial Crisis," *Annales Universitatis Apulensis Series Oeconomica*, vol. 1, no. 13, pp. 34–41, 2011, doi: 10.29302/oeconomica.2011.13.1.3.

[20] A. Malik, N. I. Abumustafa, and H. Shah, "Revisiting Creative Accounting in the Context of Islamic Economic and Finance System," *Asian Soc Sci*, vol. 15, no. 2, p. 80, 2019, doi: 10.5539/ass.v15n2p80.

[21] H. A. Almustawfiy, "Creative Accounting Applications, Opportunistic Behavior, and Integrity of Accounting Information System: The Case of Iraq," vol. 24, no. 6, pp. 1–12, 2021.

[22] I. de la T. Torre, *Creative Accounting Exposed*. Palgrave Macmillan; 2008th edition (27 Nov. 2008), 2008.

[23] T. Smith, *Accounting for Growth*. UK: Random House Business, 1992.

[24] I. Griffiths and D. Griffiths, *New Creative Accounting: How to Make Your Profits What You Want Them to Be*. Palgrave , Print UK; 1995th edition, 1995.

[25] A. Maniatis, "Detecting the probability of financial fraud due to earnings manipulation in companies listed in Athens Stock Exchange Market," *J Financ Crime*, vol. 29, no. 2, pp. 603–619, 2022, doi: 10.1108/JFC-04-2021-0083.

[26] J. Tang, W. Hartford, and K. E. Karim, "Financial fraud detection and big data analytics – implications on auditors ' use of fraud brainstorming session," no. 1973, 2018, doi: 10.1108/MAJ-01-2018-1767.

[27] J. L. Perols, R. M. Bowen, C. Zimmermann, and B. Samba, "Finding Needles in a Haystack: Using Data Analytics to Improve Fraud Prediction," *The Accounting Review*, vol. 92, no. 2, pp. 221–245, 2017, doi: 10.2308/accr-51562.

[28] Messod D. Beneish and P.J. Vorst, "The Cost of Fraud Prediction Errors," *The Accounting. Review*, no. 6, pp. 91–121, 2022.

[29] A. Al Ali, A. M. Khedr, M. El-Bannany, and S. Kanakkayil, "A Powerful Predicting Model for Financial Statement Fraud Based on Optimized XGBoost Ensemble Learning Technique," *Applied Sciences (Switzerland)*, vol. 13, no. 4, Feb. 2023, doi: 10.3390/app13042272.

[30] P. Craja, A. Kim, and S. Lessmann, "Deep learning for detecting financial statement fraud," *Decis Support Syst*, vol. 139, Dec. 2020, doi: 10.1016/j.dss.2020.113421.

[31] J. Li and Z. Sun, "Application of deep learning in recognition of accrued earnings management," *Heliyon*, vol. 9, no. 3, Mar. 2023, doi: 10.1016/j.heliyon.2023.e13664.

[32] R. Liu, F. Mai, Z. Shan, and Y. Wu, "Predicting shareholder litigation on insider trading from financial text: An interpretable deep learning approach," *Information and Management*, vol. 57, no. 8, Dec. 2020, doi: 10.1016/j.im.2020.103387.

[33] B. W. Muehlmann, V. Chiu, and Q. Liu, "Emerging technologies research in accounting: JETA's first decade," *Journal of Emerging Technologies in Accounting*, vol. 12, no. 1, pp. 17–50, Dec. 2015, doi: 10.2308/jeta-51245.

[34] ACEF, "Occupational Fraud 2022: A Report To The Nations," 2022.

[35] A. F. Al-Hassan, "Earnings Management using Accruals: Empirical Study on Saudi Companies," *Arabic Journal of Administration*, 2018, [Online]. Available: http://search.mandumah.com/record/940867

[36] A. A. Alhebri and S. D. Al-Duais, "Family businesses restrict accrual and real earnings management: Case study in Saudi Arabia," *Cogent Business and Management*, vol. 7, no. 1, Jan. 2020, doi: 10.1080/23311975.2020.1806669.

[37] K. Baatour, H. Ben Othman, and K. Hussainey, "The effect of multiple directorships on real and accrualbased earnings management: Evidence from Saudi listed firms," *Accounting Research Journal*, vol. 30, no. 4, pp. 395–412, 2017, doi: 10.1108/ARJ-06-2015-0081.

[38] A. Alghamdi, "A Hybrid Method for Big Data Analysis Using Fuzzy Clustering, Feature Selection and Adaptive Neuro-Fuzzy Inferences System Techniques: Case of Mecca and Medina Hotels in Saudi Arabia," *Arab J Sci Eng*, 2022, doi: 10.1007/s13369-022-06978-0.

[39] Y. E. Park and Y. Javed, "Insights discovery through hidden sentiment in big data: Evidence from Saudi Arabia's financial sector," *Journal of Asian Finance, Economics and Business*, vol. 7, no. 6, pp. 457–464, 2020, doi: 10.13106/JAFEB.2020.VOL7.NO6.457.

[40] M. Nurunnabi, E. K. Jermakowicz, and H. Donker, "Implementing IFRS in Saudi Arabia: evidence from publicly traded companies," *International Journal of Accounting and Information Management*, vol. 28, no. 2, pp. 243–273, Apr. 2020, doi: 10.1108/IJAIM-04-2019-0049.

[41] D. Moshashai, A. M. Leber, and J. D. Savage, "Saudi Arabia plans for its economic future: Vision 2030, the National Transformation Plan and Saudi fiscal reform," *British Journal of Middle Eastern Studies*, vol. 47, no. 3, pp. 381–401, May 2020, doi: 10.1080/13530194.2018.1500269.

[42] J. M. Cainas, W. M. Tietz, and T. Miller-Nobles, "Kat insurance: data analytics cases for introductory accounting using excel, power bi, and/or tableau," *Journal of Emerging Technologies in Accounting*, vol. 18, no. 1, pp. 77–85, 2021, doi: 10.2308/JETA-2020-039.

[43] V. Chiu, Q. Liu, B. Muehlmann, and A. A. Baldwin, "A bibliometric analysis of accounting information systems journals and their emerging technologies contributions," *International Journal of Accounting Information Systems*, vol. 32, pp. 24–43, Mar. 2019, doi: 10.1016/j.accinf.2018.11.003.

[44] A. E. A. Ibrahim, A. A. Elamer, and A. N. Ezat, "The convergence of big data and accounting: innovative research opportunities," *Technol Forecast Soc Change*, vol. 173, no. June, p. 121171, 2021, doi: 10.1016/j.techfore.2021.121171.

[45] S. Cockcroft and M. Russell, "Big Data Opportunities for Accounting and Finance Practice and Research," *Australian Accounting Review*, vol. 28, no. 3, pp. 323–333, 2018, doi: 10.1111/auar.12218.

[46] M. G. Alles, "Drivers of the use and facilitators and obstacles of the evolution of big data by the audit profession," *Accounting Horizons*, vol. 29, no. 2, pp. 439–449, 2015, doi: 10.2308/acch-51067.

[47] M. Alles and G. L. Gray, "Incorporating big data in audits: Identifying inhibitors and a research agenda to address those inhibitors," *International Journal of Accounting Information Systems*, vol. 22, pp. 44–59, 2016, doi: 10.1016/j.accinf.2016.07.004.

[48] ICAEW, "Big data and analytics: the impact on the accountancy profession," *Institute of Chartered Accountants, England and Wales (ICAEW), London, UK*, pp. 1–20, 2019.

[49] M. J. Rahman and H. Zhu, "Predicting accounting fraud using imbalanced ensemble learning classifiers – evidence from China," *Accounting and Finance*, 2023, doi: 10.1111/acfi.13044.

[50] Z. Y. Chen and D. Han, "Detecting corporate financial fraud via two-stage mapping in joint temporal and financial feature domain," *Expert Syst Appl*, vol. 217, May 2023, doi: 10.1016/j.eswa.2023.119559.

[51] C. C. Lin, A. A. Chiu, S. Y. Huang, and D. C. Yen, "Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments," *Knowl Based Syst*, vol. 89, pp. 459–470, Nov. 2015, doi: 10.1016/j.knosys.2015.08.011.

[52] M. A. Fernández-Gámez, F. García-Lagos, and J. R. Sánchez-Serrano, "Integrating corporate governance and financial variables for the identification of qualified audit opinions with neural networks," *Neural Comput Appl*, vol. 27, no. 5, pp. 1427–1444, Jul. 2016, doi: 10.1007/s00521-015-1944-6.

[53] M. Talha, S. Ali, S. Shah, F. G. Khan, and J. Iqbal, "Integration of Big Data and Deep Learning," in *Springer Briefs in Computer Science*, Springer, 2019, pp. 43–52. doi: 10.1007/978-981-13-3459-7_4.

[54] S. Nosratabadi *et al.*, "Data science in economics: Comprehensive review of advanced machine learning and deep learning methods," *Mathematics*, vol. 8, no. 10. MDPI AG, pp. 1–25, Oct. 01, 2020. doi: 10.3390/math8101799.

[55] A. M. Ozbayoglu, M. U. Gudelek, and O. B. Sezer, "Deep learning for financial applications: A survey," *Applied Soft Computing Journal*, vol. 93. Elsevier Ltd, Aug. 01, 2020. doi: 10.1016/j.asoc.2020.106384.

[56] T. D. Fields, T. Z. Lys, and L. Vincent, "Empirical research on accounting choice," *Journal of Accounting and Economics*, vol. 31, no. 1–3, pp. 255–307, 2001, doi: 10.1016/S0165-4101(01)00028-3.

[57] A. M. Goel and A. V. Thakor, "Why Do Firms Smooth Earnings ? Published by : The University of Chicago Press," *The Journal of Business*, vol. 76, no. 1, pp. 151–192, 2003.

[58] H. Stolowy and G. Breton, "Accounts Manipulation: A Literature Review and Proposed Conceptual Framework," *Review of Accounting and Finance*, vol. 3, no. 1, pp. 5–92, 2004, doi: 10.1108/eb043395.

[59] W. D. Northcut and C. C. Vines, "Earning management in response to political scrutiny of effective tax rates," *Journal of the American Taxation Association*, vol. 20, no. 2, pp. 22–36, 1998.

[60] D. Balaciu, V. Bogdan, and A. B. Vladu, "a Brief Review of Creative Accounting Literature and Its," *Annales Universitatis Apulensis Series Oeconomica*, vol. 11, no. 1, 2009.

[61] E. E. Akpanuko and N. J. Umoren, "The influence of creative accounting on the credibility of accounting reports," *Journal of Financial Reporting and Accounting*, vol. 16, no. 2, pp. 292–310, 2018, doi: 10.1108/JFRA-08-2016-0064.

[62] C. Gowthorpe and O. Amat, "Creative accounting: Some ethical issues of macro- and micro-manipulation," *Journal of Business Ethics*, vol. 57, no. 1, pp. 55–64, 2005, doi: 10.1007/s10551-004-3822-5.

[63] G. Iatridis and G. Kadorinis, "Earnings management and firm financial motives: A financial investigation of UK listed firms," *International Review of Financial Analysis*, vol. 18, no. 4, pp. 164–173, 2009, doi: 10.1016/j.irfa.2009.06.001.

[64] B. A. Badertscher, "Overvaluation and the choice of alternative earnings management mechanisms," *Accounting Review*, vol. 86, no. 5, pp. 1491–1518, 2011, doi: 10.2308/accr-10092.

[65] C. P. , S. S. Marcus André Melo, "Why do some governments resort to ' creative accounting ' but not others ? Fiscal governance in the Brazilian federation Souza Stable," *International Political Science Review /*, vol. 35, no. 5, pp. 595–612, 2014, doi: 10.1177/01925121.

[66] A. Saeedi, "Audit Opinion Prediction: A Comparison of Data Mining Techniques," *Journal of Emerging Technologies in Accounting*, vol. 18, no. 2, pp. 125–147, Sep. 2021, doi: 10.2308/JETA-19-10-02-40.

[67] F. Siano and P. Wysocki, "Transfer learning and textual analysis of accounting disclosures: Applying big data methods to small(er) datasets," *Accounting Horizons*, vol. 35, no. 3, pp. 217–244, Sep. 2021, doi: 10.2308/HORIZONS-19-161.

[68] C. S. Throckmorton, W. J. Mayew, M. Venkatachalam, and L. M. Collins, "Financial fraud detection using vocal, linguistic and financial cues," *Decis Support Syst*, vol. 74, pp. 78–87, Jun. 2015, doi: 10.1016/j.dss.2015.04.006.

[69] Y. J. Kim, B. Baik, and S. Cho, "Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning," *Expert Syst Appl*, vol. 62, pp. 32–43, Nov. 2016, doi: 10.1016/j.eswa.2016.06.016.

[70] S. Goel, J. Gangolly, S. R. Faerman, and O. Uzuner, "Can linguistic predictors detect fraudulent financial filings?," *Journal of Emerging Technologies in Accounting*, vol. 7, no. 1, pp. 25–46, 2010, doi: 10.2308/jeta.2010.7.1.25.

[71] C. L. Jan, "Detection of financial statement fraud using deep learning for sustainable development of capital markets under information asymmetry," *Sustainability (Switzerland)*, vol. 13, no. 17, Sep. 2021, doi: 10.3390/su13179879.

[72] J. Li and Z. Sun, "Application of deep learning in recognition of accrued earnings management," *Heliyon*, vol. 9, no. 3, Mar. 2023, doi: 10.1016/j.heliyon.2023.e13664.

DOI: 10.15439/2023F2832
ISSN 2300-5963 ACSIS, Vol. 35

Toward an Optimal Solution to the Network Partitioning Problem

Arman Ferdowsi *
K. N. Toosi University of Technology
Department of Mathematics
Email: armanferdowsi@email.kntu.ac.ir

Maryam Dehghan Chenary
University of Vienna
Institute of Business Decisions and Analytics
Email: maryam.dehghan.chenary@univie.ac.at

***Abstract*—This paper delves into the realm of community detection in network science and graph theory with the overarching objective of unraveling the underlying structures between nodes within a network. In this pursuit, we put forth a novel and comprehensive approach to ascertain the optimal solution to maximizing the renowned community quality metric known as Max-Min Modularity. Through a series of experiments encompassing diverse case studies, we substantiate the efficacy and validity of our proposed approach, further bolstering its credibility.**

***Index Terms*—Graph partitioning, community detection, mathematical programming, exact method**

I. Introduction and Related Work

In COMPLEX networks, nodes usually divide into several subsets sharing common characteristics and relationships, forming *communities*. The discovery and analysis of these structures hold paramount importance in computer science, particularly within the network domain and graph theory. Identifying cohesive and interconnected groups of nodes enables a deeper understanding of complex systems, facilitating insights into structural patterns, functional modules, and underlying relationships. The ability to detect network communities owes immense value and applications in networked systems, such as Social Network Analysis [1], [2], Biological Networks [3], Cosmological Networks [4], WEB analysis [5], Distributed Computing [6], Signal Processing [7], and Data Clustering [8]. It enables us to uncover hierarchical structures, predict missing links, and enhance network resilience.

More concretely, a network can be represented as a graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. A community within the network can be seen as a subset of vertices $C \subseteq V$, characterized by a dense connection of edges among the nodes within the subset and a sparse connection of edges with other subsets; see Fig. 1. In this regard, the community detection problem can be defined as partitioning V into a set of communities $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ that often entails optimizing a specific *quality measure* that quantifies the excellence of a community. A wide array of quality measures has been proposed in the literature, encompassing both *connectivity-based* and *topology-based* metrics [9].

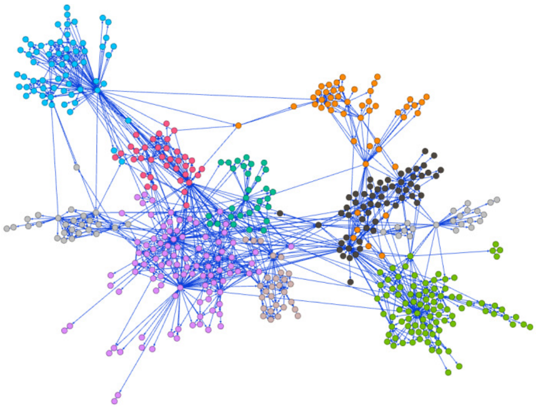


Fig. 1: Illustration of a network and its communities as subsets of vertices with densely connected nodes within each subset and sparser connections to nodes outside the batch.

For example, in [10], a dynamic connectivity-based metric is introduced to assess the quality of a community C by computing the ratio between the sum of the radius of C and the number of edges exiting C, divided by the number of edges with both endpoints belonging to C. The radius, a measure showcasing the size and compactness of the function, plays a crucial role in this evaluation. The authors of [10] also devised a two-stage heuristic algorithm to identify high-quality communities by minimizing the proposed metric. The initial stage of the algorithm, which is particularly crucial for our research, intelligently identifies an initial set of remarkably high-quality communities. This will be followed by a revising phase aimed at refining and enhancing the quality of the communities. The contributions in [10] properly highlight the significance of connectivity-based metrics in assessing community quality.

On the other hand, topological metrics have also gained considerable attention in the field of community detection. Notably, *Modularity*, introduced by Newman [11], stands out as one of the most widely recognized and extensively utilized measures in this regard. For a network G containing n vertices and m edges, the Modularity (Q) of a given partitioning $\mathbf{C}$ is mathematically defined as follows:

This research was supported by the Austrian Science Fund (FWF) project DMAC (grant no. P32431).

* Also affiliated with TU Wien, Embedded Computing Systems Group. *Email: aferdowsi@ecs.tuwien.ac.at.*

$$Q(\mathbf{C}) = \frac{1}{2m} \sum_{i,j \in V} [a_{ij} - \frac{d_i d_j}{2m}] \sigma(i,j) \tag{1}$$

where $A = (a_{ij})$ is the adjacency matrix of G with a_{ij} sets to one if an edge exists between node i and node j, and zero otherwise. d_i represents the degree of node i and is defined as the sum of all entries in the i-th row of the adjacency matrix. Moreover, $\sigma(i,j)$ is one if i and j are in the same community and zero otherwise.

Simply put, Modularity quantifies the number of edges within a community minus the expected number of such edges leading to the fact that communities with higher Modularity values have better quality. Therefore, maximizing Modularity results in identifying high-quality communities within a network.

Nevertheless, despite the widespread use of Modularity, it has been known to have certain limitations (see [12], [9] for more details). Notably, Modularity only takes into account the existing edges of the network, meaning it solely evaluates the goodness of a community based on its fit with the observed edges, while it fails to consider disconnected nodes (absent edges) within the same community. This is indeed a drawback since the disconnection of nodes does not inherently imply an absence of underlying relations between them. To overcome this limitation, an extension of Modularity called *Max-Min Modularity* [13] has been developed, which improves the accuracy of the measure by penalizing Modularity when disconnected nodes are present in the same community. In Max-Min Modularity, an additional zero-one relation matrix $U = (u_{ij})$ is introduced, which defines the relationship between pairs of disconnected nodes in the network. The value of u_{ij} is one if disconnected nodes i and j are *related* and zero otherwise. This extension acknowledges the significance of indirect connections between disconnected nodes by penalizing the Modularity measure only when unrelated nodes coexist within a community. In a more abstract sense, consider a complemented graph $G' = (V, E')$, where E' contains an edge between every pair of disconnected nodes in G that are unrelated. In other words, an edge exists between nodes i and j in G' if there is no such edge in G, and u_{ij} is zero. Let $A' = (a'_{ij})$ be the adjacency matrix of G', and d'_i be the degree of node i in G'. Additionally, let m' be the number of edges in G'. The Max-Min Modularity (Q_{MM}) of a given partition $\mathbf{C}$ of V is defined as follows:

$$Q_{MM}(\mathbf{C}) = \sum_{i,j \in V} [\frac{1}{2m}(a_{ij} - \frac{d_i d_j}{2m}) - \frac{1}{2m'}(a'_{ij} - \frac{d'_i d'_j}{2m'})] \sigma(i,j) \tag{2}$$

We refer to the problem of partitioning a network with respect to maximizing the Max-Min Modularity as the *Max-Min Modularity Maximization* problem.

Chen et al. [13] proposed a hierarchical clustering algorithm, similar to what Newman [11] had offered for the classical Modularity Maximization Problem, that approximately greedily optimizes Max-Min Modularity. In addition to the suboptimal precision of the final community detection results obtained through the heuristic approach, the primary drawback of their method lies in its reliance on a user-defined relation matrix rather than a systematic approach. This dependency on subjective input introduces the potential for misinterpretations, biases, and erroneous human decisions, which can lead to significant issues. Relying on a user-defined matrix not only increases the likelihood of errors but also lacks the robustness and objectivity provided by a systematic and automated procedure.

Furthermore, since it is known that solving the Max-Min Modularity Maximization problem is computationally challenging, as it is proven to be NP-hard, most of the approaches for solving it rely on heuristic methods. [13]. There exist, of course, methods focusing more on exact techniques, such as the approach presented in [14], in which the authors successfully formulated the Max-Min Modularity Maximization problem as an *integer programming* model and proposed an equivalent sub-problem that simplified the overall formulation. This streamlined approach facilitated the efficient solving of the model's linear relaxation and provided a systematic means of defining the relation matrix. They further employed a local search strategy to convert the fractional solutions to integer ones that led to obtaining a set of communities. Nevertheless, despite their outstanding breakthrough in the modeling and the solution approach, the error caused by the rounding approach prevents us from obtaining an optimal solution, which is indeed crucial in scenarios where precise analysis is required. This underscores the need for alternative methods that can provide more accurate results.

Main contribution:

(1) Building upon the integer programming modeling discussed in [14], we present an alternative integer model for the Max-Min Modularity Maximization problem that offers a significant reduction in the number of variables and constraints. This streamlined model enhances computational efficiency while maintaining the same optimization capabilities. The equivalence of the proposed model and the original formulation proved in Theorem 1, affirms that both models yield the same set of optimal solutions.

(2) Inspired by the prominent algorithm proposed in [10], we devise a methodology to generate an initial feasible solution for the formulated model. By employing a row generation technique in combination with the branch and bound method and leveraging the powerful CPLEX[1] solver, we efficiently and optimally solve the model. This comprehensive approach enables the detection of a set of (near) optimal communities whose efficacy and effectiveness become apparent in the experiments done.

The structure of this paper unfolds as follows: In Section II, we delve into modeling an effective integer programming formulation for the Max-Min Modularity Maximization problem, followed by an insightful theoretical exploration of how

[1] CPLEX is a powerful optimization software package developed by IBM that employs advanced algorithms to solve linear programming, mixed-integer programming, and quadratic programming problems efficiently.

to simplify the model. Subsequently, Section III outlines our devised approach for acquiring an intelligent initial feasible solution and employing a row/column generation technique to solve the model optimally. Finally, Section IV is dedicated to showcasing the experimental results, providing a comprehensive analysis of the outcomes.

II. Mathematical modeling

Let the binary variable x_{ij} indicate if nodes i and j belong to the same community or not; the value of x_{ij} is zero if nodes i and j belong to the same community, and one otherwise. Let $I_{all} = \{(i,j) \in V^2 \mid i < j\}$; and $q_{ij} = a_{ij} - \frac{d_i d_j}{2m}$, for each $(i,j) \in I_{all}$. As described in [15], the Modularity Maximization problem can be formulated in terms of the following integer linear program.

$$\max \quad \frac{1}{m} \sum_{(i,j)\in I_{all}} q_{ij}(1-x_{ij}) \qquad \text{(IP-M)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i<j<k \qquad (3)$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall i<j<k \qquad (4)$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall i<j<k \qquad (5)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in I_{all} \qquad (6)$$

Constraints (3)-(5) guarantee that if i and j are in the same community and j and k are in the same community, then so are i and k. We refer to the relaxation of (IP-M), obtained by replacing the constraints $x_{ij} \in \{0,1\}$ by $x_{ij} \in [0,1]$, as (LP-M).

Now to turn our attention to the Max-Min Modularity Maximization problem, we first recap the systematic and precise approach provided in [14] for defining the relation matrix governed by an optimal fractional solution x^* to the linear programming problem (LP-M). Considerably crucial is that x^* can be efficiently obtained in polynomial time using various algorithms such as the row and column generation algorithm introduced by [16]. It is also important to note that x^* gives rise to a metric known as the LP distance on the graph G. In this context, x^*_{ij} can be interpreted as the "distance" between nodes i and j, and notably, the constraints (3)-(5) guarantee the fulfillment of the triangle inequality for any nodes $i,j,k \in V$ in the induced metric. Evidently, the larger the LP distance between two nodes, the less related those nodes are. This observation, along with the fact that the Modularity Maximization problem can be effectively formulated for weighted graphs as demonstrated by [17], serves as motivation to define the relation matrix and the corresponding complemented (weighted) graph G' utilizing the LP distance rather than using user knowledge.

In this framework, we define the relation matrix $A' = (a'_{ij})$ (and consequently G', with (a'_{ij}) representing the weight of the edge between nodes i and j in G'), as follows:

$$a'_{ij} = \begin{cases} x^*_{ij} & \text{if } a_{ij} = 0 \text{ and } j > i \\ x^*_{ji} & \text{if } a_{ij} = 0 \text{ and } i > j \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

Consequently, given a matrix $A' = (a'_{ij})$, the Max-Min Modularity Maximization problem can be formulated as the following integer programming problem. Let $c_{ij} = \frac{q_{ij}}{m} - \frac{q'_{ij}}{m'}$, where $q'_{ij} = a'_{ij} - \frac{d'_i d'_j}{2m'}$, $d'_i = \sum_{l=1}^{n} a'_{il}$, and $m' = \sum_{(i,j)\in I_{all}} a'_{ij}$ for each $(i,j) \in I_{all}$.

$$\max \quad \sum_{(i,j)\in I_{all}} c_{ij}(1-x_{ij}) \qquad \text{(IP-MM)}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i<j<k \qquad (8)$$

$$x_{ij} - x_{jk} + x_{ik} \geq 0 \qquad \forall i<j<k \qquad (9)$$

$$-x_{ij} + x_{jk} + x_{ik} \geq 0 \qquad \forall i<j<k \qquad (10)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in I_{all} \qquad (11)$$

The very first thing to note, however, is that solving (IP-MM) falls in the class of NP-hard problems, making it challenging to be optimally solved. Consequently, we came to investigate whether it is possible to simplify (IP-MM) while maintaining the same set of optimal solutions. To achieve this objective, we have obtained the subsequent conceptual insights derived from the notion of row generation: The following lemma demonstrates that the optimal solution remains unaffected when only focusing on variables x_{ij} with $c_{ij} > 0$, and the subsequent theorem establishes that the optimal solution to (IPs-MM) without constraints involving x_{ij} where $c_{ij} \leq 0$ is equivalent to the optimal solution to (IP-MM).

Lemma 1: If a binary variable x_{ij} satisfies the constraints (8), (9), and (10), then it is sufficient to consider only the variables x_{ij} for which $c_{ij} > 0$ in the objective function (IP-MM).

Proof. Suppose we have a binary variable x_{ij} that satisfies the constraints (8), (9), and (10). We will show that if $c_{ij} \leq 0$, then x_{ij} will not affect the optimal solution of the objective function (IP-MM). Let us consider the term $c_{ij}(1-x_{ij})$ in the objective function (IP-MM). If $c_{ij} \leq 0$, then regardless of the value of x_{ij} (0 or 1), the term $c_{ij}(1-x_{ij})$ will be non-positive. Thus, including x_{ij} in the objective function with $c_{ij} \leq 0$ does not contribute to maximizing the objective. On the other hand, if $c_{ij} > 0$, including x_{ij} in the objective function can potentially increase the objective value by setting x_{ij} to 0 (i.e., nodes i and j belong to the same community) since $c_{ij}(1-0) = c_{ij}$. Therefore, it is sufficient to consider only the variables x_{ij} for which $c_{ij} > 0$ in the objective function (IP-MM). □

Theorem 1: For the given integer programming problem, if we exclude the constraints involving variables x_{ij} where $c_{ij} \leq 0$, the optimal solution of the modified problem remains the same as the original problem.

Proof. Let us assume that we have an optimal solution to the original (IP-MM) model, which satisfies all the constraints including those involving variables x_{ij} where $c_{ij} \leq 0$. We will show that by excluding these constraints, we can still obtain the same optimal solution. If x_{ij} satisfies the constraints (8), (9), and (10), its value will not change when we exclude the constraints involving $c_{ij} \leq 0$. The reason is that these constraints do not impose any restrictions on x_{ij}; they only provide additional information. Removing them does not alter the feasible region. Furthermore, since $c_{ij} \leq 0$, excluding these constraints means that the corresponding term $c_{ij}(1 - x_{ij})$ is non-positive and does not contribute to the objective function. Therefore, the objective value remains unchanged. Consequently, the optimal solution for the modified problem without constraints involving x_{ij} where $c_{ij} \leq 0$ is the same as the optimal solution for the original problem. This concludes the proof of the theorem. □

Having these considered, one can simplify (IP-MM) by considering only the variables x_{ij} where $c_{ij} > 0$ in the objective function. We can express this modified model as follows:

$$\max \sum_{(i,j)\in I_{pos}} c_{ij}(1 - x_{ij}), \tag{12}$$

where I_{pos} is the set of all pairs $(i, j) \in I_{all}$ for which $c_{ij} > 0$. We still need to ensure that the constraints (8), (9), and (10) hold for the selected variables. To achieve this, we introduce new binary variables y_{ijk} for all $i < j < k$ such that:

$$y_{ijk} = \begin{cases} 1 & \text{if } x_{ij} + x_{jk} - x_{ik} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

By introducing these variables, we can replace the constraints (8), (9), and (10) with the following constraint:

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \quad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{14}$$

Finally, we include the binary variable definitions:

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in I_{pos} \tag{15}$$

$$y_{ijk} \in \{0, 1\} \qquad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{16}$$

Gathering all together, we get an equivalent sparse model for (IP-MM) as follows:

$$\max \sum_{(i,j)\in I_{pos}} c_{ij}(1 - x_{ij}) \tag{IPs-MM}$$

$$x_{ij} + x_{jk} - x_{ik} \geq 0 \qquad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{17}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j) \in I_{pos} \tag{18}$$

$$y_{ijk} \in \{0, 1\} \qquad \forall i < j < k \text{ s.t. } (i, j) \in I_{pos} \tag{19}$$

Highly encouraging is that (IPs-MM) has considerably fewer constraints and variables compared to the original (IP-MM) model but preserves the same set of optimal solutions.

III. Solution Approach

Despite (IPs-MM) providing us with a considerably simpler integer modeling than (IP-MM), it could still remain unlikely to obtain the optimal solution, particularly when dealing with average to large-scale networks. Undoubtedly, one possible way to speed up the branch and bound technique using the CPLEX solver is to feed it with a reasonably good initial solution. Starting with a smart choice among the feasible solution space has turned out to improve performance immensely. In this vein, we came to take advantage of the heuristic two-stage community detection algorithm introduced in [10]. Considerably relevant to our research is the first stage of the algorithm's authority to swiftly find a collection of initial communities with excellent quality w.r.t. optimizing a connectivity-based criterion they established.

In this procedure, the degree of a node and its set of neighbors are defined naturally. Additionally, the inner and outer edges of a community C, denoted as E_C^{in} and E_C^{out}, respectively, represent the edges with both endpoints and one endpoint within C. Let $\alpha \in \{1, \ldots, diameter(G)\}$ be an integer and define $P_\alpha \subseteq V$ as a sequence of nodes arranged in descending order according to their degrees, provided they are all at least α steps away from each other. P_α is referred to as the *influential nodes* of G with respect to α. The distance between a vertex $j \notin C$ and the community C is then determined as the length of the shortest path from j to the influential node of C. The radius of a community C, denoted by $r(C)$, is the maximum distance from its influential node to other vertices within the community.

Subsequently, the authors of [10] proposed a measure to compute the quality of a community C:

$$q(C) = \frac{|E_C^{out}| + r(C)}{|E_C^{in}|}. \tag{20}$$

Furthermore, given a set of communities $\mathbb{C} = \{C_1, \ldots, C_k\}$, the quality of the partition $\mathbb{C}$ is defined as follow:

$$q_G^{\mathbb{C}} = \sum_{C\in\mathbb{C}} q(C). \tag{21}$$

Having everything considered, we can now summarize the procedure of determining an appropriate set of initial communities in the following manner:

- Repeat the procedure below for all α between 2 and $diameter(G)$ and pick $\mathbb{C}_\alpha$ with the minimum $q_G^{\mathbb{C}_\alpha}$ as the best initial set of communities.
  - Establish P_α as defined above.
  - Let each $p \in P_\alpha$ initially designate a sole community, leading to a set of $|P_\alpha|$ communities $\mathbb{C}_\alpha = \{C_1, \ldots, C_{|P_\alpha|}\}$.
  - Assign every $v \notin \mathbb{C}_\alpha$ to its closest community until all nodes belong to a community.

An essential insight of this approach revolves around the idea of ensuring a meaningful spatial distribution of the influential nodes of the network, striking a balance between proximity to facilitate cohesive communities and sufficient distance to avoid interference. To achieve this, the notion of $\alpha-$far nodes is introduced to serve as a criterion to evaluate the distance between potential influential vertices. By leveraging this parameter, one can successfully identify the best set of influential nodes capable of bunching and leading their surrounding vertices.

Now, by putting everything together, we devise the following tractable procedure for optimally solving (IPs-MM):

- Start with the initial communities obtained with the method explained above and employ the following row generation technique:

TABLE I: Networks under-study

ID	Network	n	m
1	Zachary's karate club [18]	34	78
2	Mexican Politicians [19]	35	117
3	Dolphin network [20]	62	159
4	Les Miserables [20]	77	254
5	p53 protein [21]	104	226
6	Books about U.S. politics [22]	105	441
7	American college football [23]	115	613
8	Citation graph drawing [24]	311	640
9	USAir97 [25]	332	2126
10	C. Elegans [26]	453	2025
11	Erdos collaboration [27]	472	1314
12	Electronic circuit [28]	512	819

1) Consider (IPs-MM) without any constraints.
2) Use the CPLEX solver and apply the branch and bound technique to obtain an optimal solution x^* to (IPs-MM).
3) Verify whether all constraints of (IPs-MM) are satisfied by x^*. If not, add the violated ones to the model and go to (2).

IV. Computational results

Within this section, we conduct a comprehensive performance evaluation of our proposed methodology. To maintain fairness, we take into account exactly the set of 12 networks used in [14]. These networks, outlined in Table I, are among the recognized and commonly utilized real-world networks utilized in this context, and each of them possesses a corresponding ground truth, representing the optimal community structures. Hence, it becomes convenient to evaluate the effectiveness of a community detection algorithm by quantifying the similarities between the algorithmically derived communities and the ground truth. To facilitate this evaluation, we employ the widely acknowledged performance metric NMI (*Normalized Mutual Information*).

A. Normalized Mutual Information (NMI)

NMI, as described in [29], is a widely recognized and established metric for evaluating the similarity between clusters. However, it can effectively measure the agreement between the optimal communities and those discovered by an algorithm. Consider a network G with n nodes, where $\mathcal{C}(\mathcal{A}) = C_1, \dots, C_k$ represents the communities obtained by algorithm $\mathcal{A}$, and $\mathcal{C}' = C'_1, \dots, C'_{k'}$ denotes the ground truth communities. The NMI value corresponding to the algorithm $\mathcal{A}$ can be computed as follows

$$NMI(\mathcal{A}) = \frac{-2\sum_{x=1}^{|\mathcal{C}|}\sum_{y=1}^{|\mathcal{C}'|}\frac{|C_x \cap C'_y|}{n} log(\frac{n|C_x \cap C'_y|)}{|C_x||C'_y|})}{\sum_{x=1}^{|\mathcal{C}|}\frac{C_x}{n} log(\frac{C_x}{n}) + \sum_{y=1}^{|\mathcal{C}'|}\frac{C'_y}{n} log(\frac{C'_y}{n})} \quad (22)$$

When the detected communities perfectly align with the ground truth, NMI attains its maximum value of one. Conversely, if the two sets exhibit no similarity, the NMI score is zero. In general, a higher NMI value indicates a more accurate and effective discovery of community structures.

B. Experiments

In what follows, we present a thorough evaluation demonstrating our proposed method's superiority in achieving high-quality communities over several competing algorithms. All the experiments are conducted on a computer system with a processor *Intel(R) Core i9-12900KF @ 3.2GHz, 16.6 Core(s)* and *128 GB* of *RAM*. Algorithms are implemented with C++, and CPLEX optimizer 12.9 is used for solving linear programming.

Fig. 2 delivers the comparisons by evaluating communities that are discovered based on the following cases:

- Our method (the blue curve): Obtaining the communities by optimally solving (IPs-MM) with the proposed method consisting of a row and column generation procedure.
- Method proposed in [14] (the red diagram): Solving the linear relaxation model of (IP-MM) via a row and column generation technique and applying a local search manner rounding procedure for obtaining the communities.
- Max-Min Modularity, proposed in [13] (green diagram): Using a user-defined relation matrix and applying a hierarchical heuristic algorithm for maximizing the Max-Min Modularity.

It is evident to conclude the promising outperformance of our proposed community detection method. In particular, the considerable gap between the blue and red curves clearly shows the advancement of using an exact method rather than relying on just heuristic approaches. While the technique in [14] took into account optimally solving the liner relaxation version of the (IP-MM), their proposed local search-based rounding procedure for obtaining the solution to (IP-MM) causes a significant error. The lever provided by the simpler model (IPs-MM), which was proven to be equivalent to (IP-MM), enabled us to seek an optimal solution to the model and, therefore, discover high-quality communities considerably better than those in [14]. This dominance could be more pronounced when noticing that the communities obtained in [14] were superior to a wide range of other algorithms.

Herein, for the sake of more visualization, Fig. 3 displays the schematic representations of the *Erdos collaboration* and *C. elegans* networks, along with the communities identified by the proposed algorithm.

Furthermore, to strengthen the assessment of our algorithm, we also decided to examine its execution time, for which we came to follow twofold perspectives: first, to determine how solving (IPs-MM) instead of (IP-MM) could enhance time complexity, and second, to compare the execution time of our model with that of the integer programming model proposed in [14] for the Max-Min Modularity Maximization problem. Fig. 4 illustrates the results of these comparisons.

Our proposed row and column generation technique, coupled with the intelligently determined initial solution, resulted in a significant speed improvement when optimally solving (IPs-MM), surpassing the performance of solving (IP-MM).

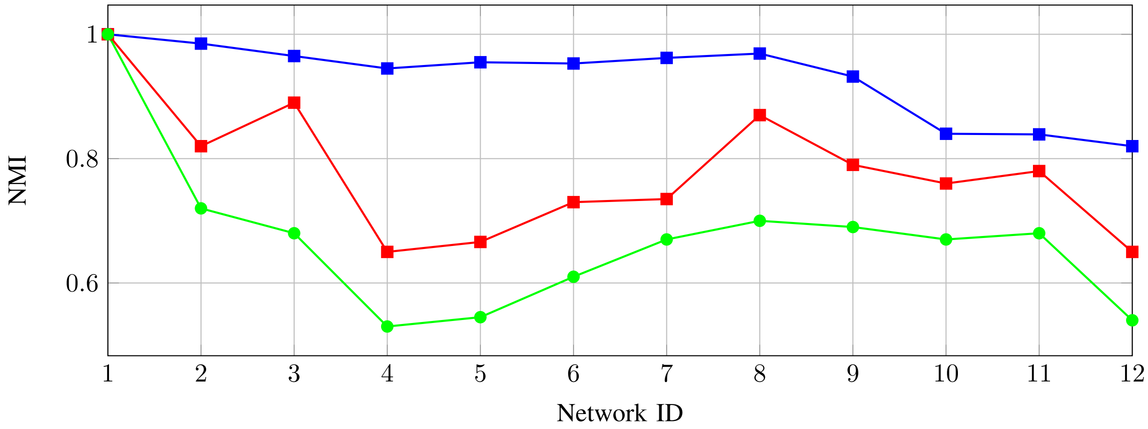
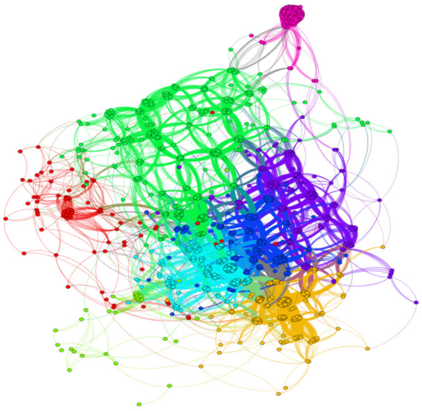
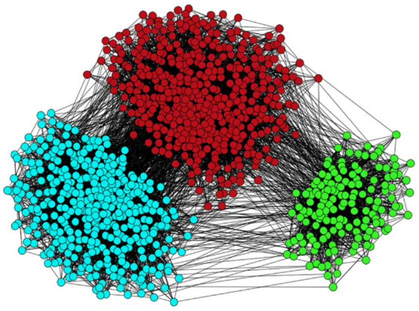


Fig. 2: Comparison between NMI values achieved by (i) blue curve: our method, (ii) red curve: method in [14], and (iii) green curve: the conventional Max-Min modularity.



(a) Erdos collaboration network



(b) C. Elegans network

Fig. 3: Two networks from Table I and their detected communities using the proposed algorithm.

This highlights the substantial impact of our simplified model and solution approach. Furthermore, our method demonstrated faster execution compared to solving the equivalent integer formulation of the sub-problem for the Max-Min Modularity Maximization problem proposed in [14], which further validates the efficiency of our simplification. As could be naturally expected due to the NP-hardness nature of the problem, our model performs slower than when solving the LP relaxation version of the model in [14] plus using the rounding algorithm. Nevertheless, the inaccurate results obtained in [14] reveal its untrustworthy against this work's proposed model.

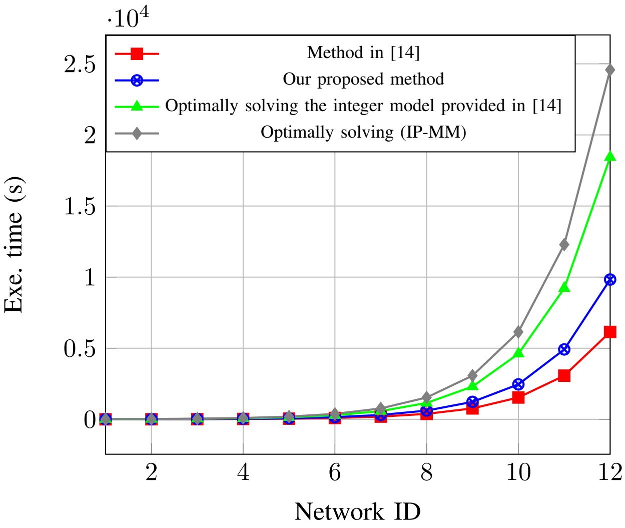


Fig. 4: The time elapsed (in terms of seconds) for solving different methods.

We complete this section by highlighting that even though the method presented in [14] can yield promising community structures in large-scale networks, for situations where accuracy is paramount, exact methods become significantly crucial. In such cases, the proposed method in this work can provide substantial assistance.

V. Conclusion

In this study, we addressed the Max-Min Modularity Maximization problem, a widely recognized metric for community evaluation. To enhance the problem's solution efficiency, we proposed an integer programming model that exhibits a reduced number of variables and constraints while preserving the same set of optimal solutions as the original model. By incorporating a row and column generation technique guided by an intelligently determined initial feasible solution, we were able to achieve optimal solutions in a remarkably efficient manner. The resulting solution provided us with a set of communities that exhibit notable similarities with the optimal community structures, indicating the effectiveness of our approach. This not only improved the overall quality of the obtained communities but also demonstrated the advantages of our model in terms of computational time.

References

[1] L. Jiang, L. Shi, L. Liu, J. Yao, and M. A. Yousuf, "User interest community detection on social media using collaborative filtering," *Wireless Networks*, pp. 1–7, 2019.

[2] M. Wang, C. Wang, J. X. Yu, and J. Zhang, "Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 998–1009, 2015.

[3] Y. Atay, I. Koc, I. Babaoglu, and H. Kodaz, "Community detection from biological and social networks: A comparative analysis of metaheuristic algorithms," *Applied Soft Computing*, vol. 50, pp. 194–211, 2017.

[4] D. Krioukov, M. Kitsak, R. S. Sinkovits, D. Rideout, D. Meyer, and M. Boguñá, "Network cosmology," *Scientific reports*, vol. 2, p. 793, 2012.

[5] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, "A model for scale-free networks: application to twitter," *Entropy*, vol. 17, no. 8, pp. 5848–5867, 2015.

[6] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, 2007, pp. 1–8.

[7] N. Tremblay and P. Borgnat, "Graph wavelets for multiscale community mining," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5227–5239, 2014.

[8] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[9] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Computing Surveys (CSUR)*, vol. 50, no. 4, pp. 1–37, 2017.

[10] A. Ferdowsi, M. Dehghan Chenary, and A. Khanteymoori, "Tscda: a dynamic two-stage community discovery approach," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 46, 2022.

[11] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the national academy of sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[12] A. Ferdowsi, "An integer programming approach reinforced by a message-passing procedure for detecting dense attributed subgraphs," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 569–576.

[13] J. Chen, O. R. Zaïane, and R. Goebel, "Detecting communities in social networks using max-min modularity," in *Proceedings of the 2009 SIAM international conference on data mining*. SIAM, 2009, pp. 978–989.

[14] A. Ferdowsi and A. Khanteymoori, "Discovering communities in networks: A linear programming approach using max-min modularity," in *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2021, pp. 329–335.

[15] T. N. Dinh and M. T. Thai, "Finding community structure with performance guarantees in complex networks," *arXiv preprint arXiv:1108.4034*, 2011.

[16] A. Miyauchi and Y. Miyamoto, "Computing an upper bound of modularity," *The European Physical Journal B*, vol. 86, no. 7, p. 302, 2013.

[17] M. E. Newman, "Analysis of weighted networks," *Physical review E*, vol. 70, no. 5, p. 056131, 2004.

[18] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[19] J. Gil-Mendieta and S. Schmidt, "The political network in mexico," *Social Networks*, vol. 18, no. 4, pp. 355–381, 1996.

[20] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[21] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical review E*, vol. 68, no. 6, p. 065103, 2003.

[22] A. Mahajan and M. Kaur, "Various approaches of community detection in complex networks: a glance," *International Journal of Information Technology and Computer Science (IJITCS)*, vol. 8, no. 35, 2016.

[23] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[24] N. Meghanathan, "A greedy algorithm for neighborhood overlap-based community detection," *Algorithms*, vol. 9, no. 1, p. 8, 2016.

[25] V. Batagelj and A. Mrvar, "Pajek datasets (2006)," 2009.

[26] A. Cangelosi and D. Parisi, "A neural network model of caenorhabditis elegans: the circuit of touch sensitivity," *Neural processing letters*, vol. 6, no. 3, pp. 91–98, 1997.

[27] A. Mrvar and V. Batagelj, *Pajek: Programs for Analysis and Visualization of Very Large Networks: Reference Manual: List of Commands with Short Explanation Version 5.10*. A. Mrvar, 2020.

[28] S. Chand and S. Mehta, "Community detection using nature inspired algorithm," in *Hybrid Intelligence for Social Networks*. Springer, 2017, pp. 47–76.

[29] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 119–130
DOI: 10.15439/2023F4407
ISSN 2300-5963 ACSIS, Vol. 35

Diffusion Limits for Shortest Remaining Processing Time Queues with Multiple Customer Types

Robert Gieroba
0000-0001-6419-3209
Maria Curie-Skłodowska University
in Lublin
Pl. Marii Curie-Skłodowskiej 1, 20-031 Lublin, Poland
Email: robert.gieroba@mail.umcs.pl

Łukasz Kruk
0000-0002-3073-959X
Maria Curie-Skłodowska University
in Lublin
Pl. Marii Curie-Skłodowskiej 1, 20-031 Lublin, Poland
Email: lukasz.kruk@mail.umcs.pl

***Abstract*—We consider a single-server queueing system with multiple customer types having bounded processing times in which users are scheduled according to the Shortest Remaining Processing Time (SRPT) discipline, with First In First Out (FIFO) as the tie-breaker. We assume that the processing times of jobs arriving in the system are bounded. We use probabilistic methods to find, under typical heavy traffic assumptions, a suitable approximation of the workload and queue length processes after a long time has passed and show that these processes are divided among the customer classes according to specific proportions, depending on their arrival rates and distributions of initial service times. Our results are confirmed by simulations.**

***Index terms*—Queueing systems, shortest remaining processing time, heavy traffic, diffusion approximations, multiple customer classes.**

I. Introduction

THE SHORTEST Remaining Processing Time (SRPT) service protocol assigns preemptive priority to the task with the smallest residual service time. It started gaining interest after Schrage had proved in [24] that it minimized the number of jobs in a single-server system. SRPT is also well known in queueing theory for minimizing the mean response time (Schrage and Miller [18]). Since then, its properties have been widely studied. Schreiber provides a summary of early work on SRPT in [25]. More recent research of this protocol includes investigating fairness (e.g., Wierman and Harchol-Balter [27]) or tail behavior (Núñez-Queija [19], Nuyens and Zwart [20]). In [12], Gromoll, Kruk and Puha prove, under typical heavy traffic assumptions, a diffusion limit theorem for a measure-valued state descriptor. Another approach, approximating SRPT by the Earliest Deadline First (EDF), leads to the same result (with a slight loss of generality) in [16]. The follow-up research on this topic includes obtaining diffusion limits under nonstandard spatial scaling by Puha [23] and limits for queues with heavy tailed service time distributions by Banerjee, Budhiraja and Puha in [4]. Moreover, Atar, Biswas, Kaspi and Ramanan presented in [3] a unified framework for analyzing single server queueing systems under various service protocols, including SRPT.

Some authors discussed the possibility of implementing the SRPT policy in practice. The main factor impeding it is its unfairness, which could potentially lead to a few tasks having much greater response times (so-called "starving"). However, it has been noted in many works that large jobs are only negligibly penalized at most. For example, Agrawal, Bansal, Harchol-Balter and Schroeder in [1] and [2] propose a method of improving the performance of Web servers by implementing SRPT-based scheduling. Another study by Harchol-Balter and Schroeder ([14]) presents the possibility of a great improvement of the performance of a Web server by changing the traditional fair scheduling policy to SRPT. There are also more recent papers concerning this topic. For example, [6] describes a way of improving the default Linux scheduler by using the existing CFS (Completely Fair Scheduler) and FIFO schedulers to approximate SRPT.

Recent studies suggest that the SRPT protocol performs well in the case of multiserver systems with a single queue. Grosof, Scully and Harchol-Balter proved in [13] that the mean response in the M/G/k queue under the SRPT discipline is asymptotically optimal in the heavy-traffic limit. Dong and Ibrahim ([8]) considered the multiserver M/G/k+G queue with impatient customers with the SRPT protocol and showed that in this setting, SRPT asymptotically maximizes the system throughput among all scheduling disciplines. However, it was also shown that the SRPT protocol can behave suboptimally in multiclass queueing networks to the extent of rendering the queueing system unstable ([7]).

Another related field of research involves studying resource sharing networks under the SRPT service discipline. In [10], [11], a notion of minimality, related to maximizing the corresponding cumulative transmission time with respect to jobs with residual service time not greater than a given threshold, was introduced and it was shown that SRPT is minimal in this sense. Moreover, in [11], another optimality criterion, local edge minimality, was proposed and it was proved that it characterizes a certain subclass of SRPT disciplines, named strong SRPT protocols.

This paper focuses on a single-server queueing system with multiple customer classes. Previous research on this topic includes the work of Peterson ([21]) concerning heavy-traffic limit theorems for queueing networks with multiple customers classes divided into two types: high-priority ones having a preemptive priority over low-priority ones, with customers within each of these types served according to the FIFO policy. In [17], Kruk and Sokołowska establish a fluid limit theorem for a single-server queueing model with K classes of customers, served according to the SRPT protocol, with FIFO used as the tie-breaker.

In this paper, we extend the analysis of [12] and prove a diffusion limit theorem for a multidimensional measure-valued state descriptor $\mathcal{Z}(t)$ defined in Section II, under usual heavy traffic assumptions. More precisely, we first describe a stochastic model for a single-server queuing system with multiple customer classes. We focus on the case of bounded service times of jobs arriving in the system. In order to obtain a diffusion limit, we consider a sequence of such models and apply diffusion scaling as in (1) to follow. We make typical heavy traffic assumptions, as detailed in Section III. The main results of this paper are Theorems 1 and 2. They enable us to easily obtain results for the corresponding workload and queue length processes. This method gives us a way to approximately predict the proportions between the workloads (and queue lengths) of the customer classes in the long run.

The paper is organized as follows. In the second section, we present the mathematical model of the queueing system outlined above and introduce stochastic processes describing its state. In the third section, we introduce a sequence of such models and describe the necessary assumptions. In Sections IV and V, we state and prove the main theorems of this paper. In Section VI, we provide a brief overview of computer simulations illustrating our results.

A. Notation

Let $\mathbb{N}$ denote the set of positive integers, let $\mathbb{R}$ denote the set of real numbers and let $\mathbb{R}_+ = [0, +\infty)$. For $a, b \in \mathbb{R}$, we write $a \vee b$ ($a \wedge b$) for the maximum (minimum) of a and b and $\lfloor a \rfloor$ for the largest integer not greater than a. By convention, a sum over the empty set of indices equals zero. The sets (a, b), $[a, b)$ and $(a, b]$ are empty for $a, b \in [0, \infty]$ with $a \geq b$. The Borel σ-field on $\mathbb{R}_+$ will be denoted by $\mathcal{B}(\mathbb{R}_+)$. For $B \in \mathcal{B}(\mathbb{R}_+)$, we denote the indicator of the set B by $\mathbb{I}_B$. We also define the function $\chi(x) = x,\ x \in \mathbb{R}_+$. For a function $g : \mathbb{R}_+ \to \mathbb{R}$ and $T > 0$, let $\|g\|_T = \sup\{|g(t)| : 0 \leq t \leq T\}$ and $\|g\|_\infty = \sup\{|g(t)| : t \geq 0\}$.

For a vector $a = (a_1, ..., a_K)$, with either real or measure-valued elements, by a_Σ we denote $\sum_{i=1}^K a_i$, unless stated otherwise, where it is a weighted sum. The same notation is used for vector-valued processes.

Let $\mathbf{M}$ denote the set of finite, nonnegative measures on $\mathcal{B}(\mathbb{R}_+)$. When $\mu \in \mathbf{M}$ and $a, b \in \mathbb{R}_+ \cup \{+\infty\}$, we will simply write $\mu(a, b)$, $\mu[a, b)$, $\mu(a, b]$ instead of $\mu((a, b))$, $\mu([a, b))$, $\mu((a, b])$, respectively. Moreover, we will write $\mu(x)$ instead of $\mu(\{x\})$ to denote the measure of a single-element set $\{x\}$. For $\xi \in \mathbf{M}$ and a Borel measurable function $g : \mathbb{R}_+ \to \mathbb{R}$ that is integrable with respect to ξ, define $\langle g, \xi \rangle = \int_{\mathbb{R}_+} g(x)\xi(dx)$.

The set $\mathbf{M}$ is endowed with the weak topology, that is, for $\xi_n, \xi \in \mathbf{M}$, we have $\xi_n \xrightarrow{w} \xi$ if and only if $\langle g, \xi_n \rangle \to \langle g, \xi \rangle$ as $n \to \infty$ for all bounded, continuous real functions g on $\mathbb{R}_+$. With this topology, $\mathbf{M}$ is a Polish space ([22]). We denote the zero measure in $\mathbf{M}$ by $\mathbf{0}$ and the measure in $\mathbf{M}$ that puts one unit of mass at a point $x \in \mathbb{R}_+$ by δ_x. For $x \in \mathbb{R}_+$, the measure δ_x^+ is δ_x if $x > 0$ and $\mathbf{0}$ otherwise. Let $\mathbf{M}_0$ denote the set of those elements of $\mathbf{M}$ that do not charge the origin and have a finite first moment.

We use " $\overset{d}{=}$ " for equality in distribution, " $\xrightarrow{fd}$ " to denote the convergence of finite-dimensional distributions of stochastic processes and " $\Rightarrow$ " to denote convergence in distribution of random elements of a metric space. All stochastic processes used in this paper are assumed to have paths that are right continuous with finite left limits (r.c.l.l.). For a Polish space $\mathcal{S}$, we denote by $\mathbf{D}([0, \infty), \mathcal{S})$ the space of r.c.l.l. functions from $[0, \infty)$ into $\mathcal{S}$, endowed with the Skorohod J_1 metric d ([9]). If $\mathcal{S} = \mathbb{R}$, we write $\mathbf{D}[0, \infty)$ instead of $\mathbf{D}([0, \infty), \mathbb{R})$. For $x \in \mathbf{D}([0, \infty), \mathbb{R}^n)$ and $t > 0$, define $x(t-) = \lim_{s \to t^-} x(s)$.

II. Stochastic model for an SRPT queue

The queueing model considered here consists of one server and K job types. Let $\mathcal{K} = \{1, ..., K\}$. The stochastic model involves a random initial condition $(\mathcal{Z}(0), S^x, x > 0)$, describing the system at time zero, together with a measure-valued state descriptor $\mathcal{Z} = (\mathcal{Z}_k, k \in \mathcal{K})$ specifying the time evolution of the system.

A. Initial condition

The initial condition for each class $k \in \mathcal{K}$ consists of the number $Z_k(0)$ of class k jobs in the queue at time zero, the initial service time for each class k job, and the functions describing the order in which the initial jobs with the same initial service time complete service.

Assume that $Z_k(0)$ is a nonnegative integer-valued and finite almost surely random variable for each $k \in \mathcal{K}$. Initial service times for each class k are given by the sequence $\{\tilde{v}_k^j\}_{j \in \mathbb{N}}$ of strictly positive, finite random variables. The initial job with service time $\tilde{v}_k^j, j \leq Z_k(0)$, is called job j for class k. The state of the system will be described by a counting measure with unit masses at the service times of the jobs present in the system. More formally, for $k \in \mathcal{K}$ we define the initial random measure $\mathcal{Z}_k(0) \in \mathbf{M}$ by

$$\mathcal{Z}_k(0) = \sum_{j=1}^{Z_k(0)} \delta_{\tilde{v}_k^j}.$$

Let $\mathcal{Z}(0) = (\mathcal{Z}_1(0), ..., \mathcal{Z}_K(0))$.

For every $x > 0$ the number of initial jobs with initial service time x that are served to completion once t units of time have been devoted to their service is $\lfloor t/x \rfloor \wedge \sum_{k \in \mathcal{K}} \mathcal{Z}_k(0)(x)$. Here we introduce processes $S_k^x,\ k \in \mathcal{K}$, that dictate how much service is allocated across classes. In particular, let $S_k^x(t)$ be the number of initial class k jobs with initial service time x that have left the system by the time that the server has devoted t units of time solely to serving jobs with this initial service time. We assume that the random functions $S_k^x(t)$ satisfy the following consistency conditions:

1) $S_k^x(0) = 0$,

2) $S^x_k(t)$ is nondecreasing and

$$S^x_k(t) = \mathcal{Z}_k(0)(x),\ t \geq x\mathcal{Z}_\Sigma(0)(x),\ k \in \mathcal{K},$$

3) we have

$$\sum_{k\in\mathcal{K}} S^x_k(t) = \lfloor t/x \rfloor \wedge \mathcal{Z}_\Sigma(0)(x).$$

The system $(\mathcal{Z}(0), S^x, x > 0)$, where $S^x = (S^x_1, ..., S^x_K)$, will be called the initial condition.

B. Stochastic primitives

Let E_k be the exogenous arrival process for class $k \in \mathcal{K}$. For $t \geq 0$, $E_k(t)$ is the number of class k arrivals to the system in the time interval $(0, t]$. For each k it is a (possibly delayed) renewal process with rate $\alpha_k > 0$ such that the interarrival times have standard deviation $a_k \geq 0$. Let $E(t) = (E_1(t), ..., E_K(t))$, $\alpha = (\alpha_1, ..., \alpha_K)$ and $a = (a_1, ..., a_K)$. In particular, α_Σ is the total arrival rate. For $t \geq 0$ and $k \in \mathcal{K}$ let $A_k(t) = Z_k(0) + E_k(t)$ and $A(t) = (A_1(t), ..., A_K(t))$. Then job j of class k arrives at time $T^j_k = \inf\{t \geq 0 : A_k(t) \geq j\}$. For $k \in \mathcal{K}$ and $j \in \mathbb{N}$ a random variable v^j_k represents the service time of the $(Z_k(0) + j)$th job of class k. We assume that the random variables $\{v^j_k\}_{j\in\mathbb{N}}$ are strictly positive and form an independent and identically distributed sequence with common distribution ν_k on $\mathbb{R}_+$ for each $k \in \mathcal{K}$. For $k \in \mathcal{K}$ let the sequences $\{v^j_k\}_{j\in\mathbb{N}}$ be mutually independent. Assuming that the mean $\langle\chi, \nu_k\rangle > 0$ and the standard deviation $b_k = \sqrt{\langle\chi^2, \nu_k\rangle - \langle\chi, \nu_k\rangle^2} \geq 0$ we define $\beta_k = \langle\chi, \nu_k\rangle^{-1}$ for each class. We put $\nu = (\nu_1, ..., \nu_K)$. Define $p_k = \alpha_k/\alpha_\Sigma,\ k \in \mathcal{K}$. Then $\nu_\Sigma = \sum_{k\in\mathcal{K}} p_k\nu_k$ is a mixture of service time distributions. It may be thought of as the distribution of the initial service time of a randomly chosen customer. Let $\rho = \alpha_\Sigma\langle\chi, \nu_\Sigma\rangle$ be the traffic intensity.

It will be convenient to combine the stochastic primitives for job classes into measure-valued load processes

$$\mathcal{V}_k(t) = \sum_{j=1}^{E_k(t)} \delta_{v^j_k},\ t \geq 0,\ k \in \mathcal{K},$$

and $\mathcal{V}(t) = (\mathcal{V}_1(t), ..., \mathcal{V}_K(t))$. Then for each $k \in \mathcal{K}$, $\mathcal{V}_k \in \mathbf{D}([0,\infty), \mathbf{M})$, since $E_k \in \mathbf{D}([0,\infty), \mathbb{R}_+)$.

C. Basic performance processes, service protocol

For $j \in \mathbb{N},\ k \in \mathcal{K}$ and $t \geq T^j_k$ let $w^j_k(t)$ denote the residual service time of job j in class k at time t. Thus, w^j_k decreases at rate one when the job j of class k is in service and is constant otherwise. In particular, w^j_k is identically equal to zero after the departure of the job j from the system.

Customers are served using the SRPT service protocol, which gives preemptive priority to the job with the shortest residual service time. In case of a tie, FIFO is used as a tie-breaking rule. When both the service times and the arrival times of two or more jobs are the same, we break the tie in an arbitrary manner.

The state of the system at time t will be described by a K-dimensional vector of counting measures with unit masses at the residual service times of the jobs of each class still present in the system. More formally, for $t \geq 0$ and $k \in \mathcal{K}$, the state descriptor of class k is defined as follows:

$$\mathcal{Z}_k(t) = \sum_{j=1}^{A_k(t)} \delta^+_{w^j_k(t)}.$$

Put $\mathcal{Z}(t) = (\mathcal{Z}_1(t), ..., \mathcal{Z}_K(t))$. For $t \geq 0$ and $k \in \mathcal{K}$, we define the workload of class k by $W_k(t) = \langle\chi, \mathcal{Z}_k(t)\rangle$ and $W(t) = (W_1(t), ..., W_K(t))$. Let $Q_k(t) = \langle 1, \mathcal{Z}_k(t)\rangle$ denote the number of customers of class $k \in \mathcal{K}$ present in the system at time t and define $Q(t) = (Q_1(t), ..., Q_K(t))$.

III. Diffusion limit

We define a sequence of systems to which the diffusion scaling is applied. Let $\mathcal{R}$ be a sequence of positive real numbers increasing to infinity. Consider an $\mathcal{R}$-indexed sequence of stochastic models, each defined as in Section II. For each $r \in \mathcal{R}$ there is an initial condition $(\mathcal{Z}^r(0), S^{r,x}, x > 0)$, stochastic primitives E^r_k and $\{v^{r,j}_k\}_{j\in\mathbb{N}}$ with parameters $\alpha^r_k, a^r_k, \nu^r_k, \beta^r_k$, b^r_k, p^r_k and ρ^r, for each class $k \in \mathcal{K}$. We also have arrival processes A^r with arrival times $\{T^{r,j}_k\}_{j\in\mathbb{N}},\ k \in \mathcal{K}$ a state descriptor $\mathcal{Z}^r$ and processes W^r, Q^r. The stochastic elements of each model are defined on a probability space $(\Omega^r, \mathcal{F}^r, \mathbf{P}^r)$ with expectation operator $\mathbb{E}^r$.

A diffusion scaling is applied to each model in the $\mathcal{R}$-indexed sequence as follows. For each $r \in \mathcal{R}$ and $t \geq 0$, let

$$\begin{aligned}
\hat{E}^r(t) &= \frac{1}{r}(E^r(r^2t) - r^2t\alpha^r),\\
\hat{\mathcal{Z}}^r(t) &= \frac{1}{r}\mathcal{Z}^r(r^2t),\\
\hat{W}^r(t) &= \frac{1}{r}W^r(r^2t),\\
\hat{Q}^r(t) &= \frac{1}{r}Q^r(r^2t),\\
\hat{\mathcal{V}}^r(t) &= \frac{1}{r}\left(\mathcal{V}^r(r^2t) - r^2t\alpha^r\nu^r\right).
\end{aligned} \tag{1}$$

A fluid scaling is applied to functions $S^{r,x}$. For each $r \in \mathcal{R}$, $x > 0$ and $t \geq 0$, let

$$\bar{S}^{r,x}(t) = \frac{1}{r}S^{r,x}(rt).$$

Let $\alpha = (\alpha_1, ..., \alpha_K) \in (0, +\infty)^K$, $a = (a_1, ..., a_K) \in (0, +\infty)^K$ and define $\alpha(t) = \alpha t,\ t \geq 0$. Let $\nu = (\nu_1, ..., \nu_K)$ be a vector of probability measures on $\mathbb{R}_+$ such that for each $k \in \mathcal{K}$

$$\nu_k(0) = 0, \qquad \langle\chi, \nu_k\rangle = \frac{1}{\alpha_k}, \qquad 0 < \langle\chi^2, \nu_k\rangle < \infty.$$

We make the following asymptotic assumptions for the sequence of stochastic primitives. Assume that as $r \to \infty$,

$$\alpha^r \to \alpha, \qquad a^r \to a, \qquad \hat{E}^r \Rightarrow E^*, \tag{2}$$

where E^* is a K-dimensional Brownian motion starting from zero with drift zero and covariance matrix $\Sigma = [\sigma_{ij}]$ such that

$\sigma_{kk} = a_k^2 \alpha_k^3,\ k \in \{1, ..., K\}$. In particular, if the coordinate processes of E^* are independent, then $\sigma_{ij} = 0,\ i \neq j$. Put

$$b_k = \sqrt{\langle \chi^2, \nu_k \rangle - \langle \chi, \nu_k \rangle^2},\ k \in \mathcal{K},$$

and $b = (b_1, ..., b_K)$. In addition, assume the heavy traffic condition that for some $\gamma \in \mathbb{R}$

$$\lim_{r \to \infty} r(1 - \rho^r) = \gamma. \tag{3}$$

For the sequence of service time distributions, we assume that $\nu^r = \nu$, i.e., ν^r does not depend on r. Then $\beta^r = \beta$, $p_k^r \to p_k := \frac{\alpha_k}{\alpha_\Sigma}$ as $r \to \infty$, where α is given by (2), $\rho^r \to 1$ as $r \to \infty$ and $b^r = b$. Define

$$x^* = \sup\{x \in \mathbb{R}_+ : \alpha_\Sigma \langle \chi \mathbb{I}_{[0,x]}, \nu_\Sigma \rangle < 1\}.$$

We assume that $x^* < \infty$.

For the sequence of diffusion scaled initial conditions $\{(\mathcal{Z}^r(0), S^{r,x}, x > 0)\}_{r>0}$ we assume that as $r \to \infty$

$$\hat{W}_\Sigma^r(0) \Rightarrow W_0^*, \tag{4}$$

where W_0^* is some random variable. It is well known ([15]) that from (2), (3), (4), the fact that service time distributions ν^r do not depend on r and that SRPT is a work conserving discipline, it follows that as $r \to \infty$

$$\hat{W}_\Sigma^r \Rightarrow W_\Sigma^*, \tag{5}$$

where W_Σ^* is a reflected Brownian motion with initial value $W_\Sigma^*(0) \overset{d}{=} W_0^*$ with drift $-\gamma$ and variance $(a_\Sigma^2 + b_\Sigma^2)\alpha_\Sigma$ per unit time. It can be shown ([12]) that if $x^* < \infty$ then $\hat{Q}_\Sigma^r \Rightarrow Q_\Sigma^* := \frac{W_\Sigma^*}{x^*}$.

Before we proceed, we state a standard result, as presented in [12].

Proposition 1. *For each* $r \in \mathcal{R}$, *let* $\{x_k^r\}_{k=1}^\infty$ *be an independent and identically distributed sequence of nonnegative random variables on a probability space* $(\Omega^r, \mathcal{F}^r, \mathbf{P}^r)$ *with finite mean* μ^r *and standard deviation* σ^r, *independent of a (possibly delayed) rate* $\alpha^r > 0$ *renewal process* B^r *such that the standard deviation of the interarrival times equals* $a^r \geq 0$. *Assume that* $\hat{B}^r \Rightarrow B^*$ *as* $r \to \infty$, *where*

$$\hat{B}^r(t) = \frac{1}{r}\left(B^r(r^2 t) - r^2 t \alpha^r\right),\ t \geq 0,$$

and B^* *is a one-dimensional Brownian motion starting from zero with drift zero and variance* $a^2\alpha^3$ *per unit time. Suppose that for some finite nonnegative constants* μ, σ, *and positive* α, a *we have that* $\mu^r \to \mu$, $\sigma^r \to \sigma$, $\alpha^r \to \alpha$ *and* $a^r \to a$ *as* $r \to \infty$. *Further assume that for each* $\varepsilon > 0$,

$$\lim_{r \to \infty} \mathbb{E}^r\left((x_1^r - \mu^r)^2 \mathbb{I}_{[|x_1^r - \mu^r| > r\varepsilon]}\right) = 0.$$

For $r \in \mathcal{R},\ n \in \mathbb{N}$ *and* $t \geq 0$, *let*

$$X^r(n) = \sum_{k=1}^n x_k^r \text{ and } \hat{X}^r(t) = \frac{X^r(\lfloor r^2 t \rfloor) - \lfloor r^2 t \rfloor \mu^r}{r}.$$

Then, as $r \to \infty$, $(\hat{B}^r, \hat{X}^r) \Rightarrow (B^*, X^*)$, *where* X^* *is a Brownian motion starting from zero with drift zero and variance* σ^2 *per unit time, independent of* B^*. *Furthermore, as* $r \to \infty$,

$$\hat{Y}^r(\cdot) \Rightarrow X^*(\alpha(\cdot)) + \mu B^*(\cdot),$$

where for each $r \in \mathcal{R}$ *and* $t \geq 0$,

$$\hat{Y}^r(t) = \frac{X^r(B^r(r^2 t)) - r^2 t \alpha^r \mu^r}{r},$$

and $\alpha(t) = \alpha t$.

IV. The case of $\nu_\Sigma(x^*) > 0$

The results in this section require all the assumptions made in Section 3. To simplify the notation, we will write S_k^r and $\bar{S}_k^r$ instead of S_k^{r,x^*} and $\bar{S}_k^{r,x^*}$ and similarly S^r and $\bar{S}^r$ instead of S^{r,x^*} and $\bar{S}^{r,x^*}$.

Theorem 1. *Let* $x^* < \infty,\ \nu_\Sigma(x^*) > 0$. *Assume that as* $r \to \infty$

$$\left(\hat{\mathcal{Z}}^r(0), \bar{S}^r, \hat{W}_\Sigma^r(0)\right) \Rightarrow (\mathcal{Z}^*(0), D, W_0^*) \tag{6}$$

in $\mathbf{M}^K \times (\mathbf{D}[0,\infty))^K \times \mathbb{R}$, *where*

$$\mathcal{Z}^*(0) = \left(\frac{p_k \nu_k(x^*)}{\nu_\Sigma(x^*)} \frac{W_0^*}{x^*} \delta_{x^*}, k = 1, ..., K\right) \tag{7}$$

and

$$D(t) = \left(\frac{p_k \nu_k(x^*)}{\nu_\Sigma(x^*)} \frac{t \wedge W_0^*}{x^*}, k = 1, ..., K\right),\ t \geq 0. \tag{8}$$

Then

$$\left(\hat{\mathcal{Z}}_k^r, k = 1, ..., K\right) \Rightarrow \left(p_k \frac{\nu_k(x^*)}{\nu_\Sigma(x^*)} \frac{W_\Sigma^*}{x^*} \delta_{x^*}, k = 1, ..., K\right). \tag{9}$$

A. Proof of Theorem 1

The general idea of the proof of Theorem 1 is to first show that in the diffusion limits all jobs with service times less than x^* are prioritized so that their corresponding workload and queue length processes vanish. This implies that the processes $\hat{\mathcal{Z}}_\Sigma^r$ converge weakly to a multiple of δ_{x^*}. Then we use functional limit theorems and the fact that FIFO is the tie-breaking rule to prove that in the diffusion limit the queue lengths and the workloads of job classes are divided according to the proportions in the statement of the theorem. In the proof, we follow the ideas of Peterson [21], with customers having the residual service times not less than x^* regarded as the low priority (L) ones with necessary modifications. The most important difference is that in [21] the priority of a job cannot change. In our setting however, the priority of a job can change from low to high (H). It happens every time a job with service time x^* receives some service and hence its residual service time is decreased. Consequently, our problem requires a somewhat more careful approach, described below in more detail.

1) Additional notation: For $k \in \mathcal{K}$, $r \in \mathcal{R}$ and $t \geq 0$ denote

$$V_k^r(t) = \langle \chi, \mathcal{V}_k^r(t) \rangle, \quad \hat{V}_k^r(t) = \langle \chi, \hat{\mathcal{V}}_k^r(t) \rangle,$$

$$\begin{aligned}
V_{k,H}^r(t) &= \langle \chi \mathbb{I}_{[0,x^*)}, \mathcal{V}_k^r(t) \rangle, \\
Q_{k,H}^r(t) &= \langle \mathbb{I}_{[0,x^*)}, \mathcal{Z}_k^r(t) \rangle, \\
W_{k,H}^r(t) &= \langle \chi \mathbb{I}_{[0,x^*)}, \mathcal{Z}_k^r(t) \rangle, \\
\hat{V}_{k,H}^r(t) &= \langle \chi \mathbb{I}_{[0,x^*)}, \hat{\mathcal{V}}_k^r(t) \rangle, \\
\hat{Q}_{k,H}^r(t) &= \langle \mathbb{I}_{[0,x^*)}, \hat{\mathcal{Z}}_k^r(t) \rangle, \\
\hat{W}_{k,H}^r(t) &= \langle \chi \mathbb{I}_{[0,x^*)}, \hat{\mathcal{Z}}_k^r(t) \rangle
\end{aligned}$$

and

$$\begin{aligned}
V_{k,L}^r(t) &= V_k^r(t) - V_{k,H}^r(t), \quad W_{k,L}^r(t) = W_k^r(t) - W_{k,H}^r(t), \\
\hat{V}_{k,L}^r(t) &= \hat{V}_k^r(t) - \hat{V}_{k,H}^r(t), \quad \hat{W}_{k,L}^r(t) = \hat{W}_k^r(t) - \hat{W}_{k,H}^r(t).
\end{aligned}$$

In general, subscript "H" stands for high priority jobs, i.e. jobs with residual processing times strictly lower than x^* and subscript "L" indicates low priority jobs, i.e. those with residual service times greater than or equal to x^*.

Recall that in our notation

$$\begin{aligned}
E_\Sigma^r(t) &= \sum_{k \in \mathcal{K}} E_k^r(t), \quad \hat{E}_\Sigma^r(t) = \sum_{k \in \mathcal{K}} \hat{E}_k^r(t), \\
E_\Sigma^*(t) &= \sum_{k \in \mathcal{K}} E_k^*(t), \quad W_\Sigma^r(t) = \sum_{k \in \mathcal{K}} W_k^r(t),
\end{aligned}$$

and similarly,

$$\begin{aligned}
Q_{\Sigma,H}^r(t) &= \langle \mathbb{I}_{[0,x^*)}, \mathcal{Z}_\Sigma^r(t) \rangle, \ \hat{Q}_{\Sigma,H}^r(t) = \langle \mathbb{I}_{[0,x^*)}, \hat{\mathcal{Z}}_\Sigma^r(t) \rangle, \\
W_{\Sigma,H}^r(t) &= \langle \chi \mathbb{I}_{[0,x^*)}, \mathcal{Z}_\Sigma^r(t) \rangle, \ \hat{W}_{\Sigma,H}^r(t) = \langle \chi \mathbb{I}_{[0,x^*)}, \hat{\mathcal{Z}}_\Sigma^r(t) \rangle, \\
W_{\Sigma,L}^r(t) &= W_\Sigma^r(t) - W_{\Sigma,H}^r(t), \ \hat{W}_{\Sigma,L}^r(t) = \hat{W}_\Sigma^r(t) - \hat{W}_{\Sigma,H}^r(t).
\end{aligned}$$

Let $L^r(t)$ denote the number of units of service dedicated to jobs with initial service times equal to x^* by time $t \geq 0$ in the r-th system. Observe for future reference that $S_k^r(L^r(t))$ is the number of fully served initial class k jobs with initial service time x^* by time t in the r-th system.

2) Concentration of the mass at x^*: We will first show that

$$\hat{Q}_{\Sigma,H}^r \Rightarrow 0, \ r \to \infty. \tag{10}$$

Observe that (10) implies

$$\hat{W}_{\Sigma,H}^r \Rightarrow 0, \ r \to \infty \tag{11}$$

since

$$\hat{W}_{\Sigma,H}^r \leq x^* \hat{Q}_{\Sigma,H}^r. \tag{12}$$

For $r \in \mathcal{R}$ and $t \geq 0$, let

$$\tau^r(t) = \sup\{s \in [0,t] : \hat{Q}_{\Sigma,H}^r(s) = 0\},$$

which equals zero by definition if

$$\{s \in [0,t] : \hat{Q}_{\Sigma,H}^r(s) = 0\} = \varnothing.$$

Then for $r \in \mathcal{R}$ and $t \geq 0$

$$\begin{aligned}
\hat{Q}_{\Sigma,H}^r(t) &\leq \hat{Q}_{\Sigma,H}^r(\tau^r(t)) + \frac{1}{r}\left(E_\Sigma^r(r^2 t) - E_\Sigma^r(r^2\tau^r(t)) + 1\right) \\
&= \hat{Q}_{\Sigma,H}^r(\tau^r(t)) + \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)) + \frac{1}{r} + r(t - \tau^r(t))\alpha_\Sigma^r.
\end{aligned}$$

Indeed, the inequality above follows from the fact that clearly $\tau^r(t) \leq t$, so we can bound the (diffusion scaled) number of high priority customers in the system at time t from above by an analogous number of customers at time $\tau^r(t)$ increased by the number of external arrivals in this time interval. The addition of 1 on the right-hand side of the inequality is needed because of a possibility that there is no job with service time less than x^* at time $r^2\tau^r(t)$ in the system and a job with service time x^* is chosen for processing at this time, which immediately increases the number of jobs with residual processing times less than x^* by 1.

First we find an upper bound on $\hat{Q}_{\Sigma,H}^r(\tau^r(t))$. Fix $r \in \mathcal{R}$ and $t \geq 0$. If $\tau^r(t) = 0$, then $\hat{Q}_{\Sigma,H}^r(\tau^r(t)) = \hat{Q}_{\Sigma,H}^r(0)$. Otherwise, $\tau^r(t) > 0$. If $\hat{Q}_{\Sigma,H}^r(\tau^r(t)) = 0$, then any nonnegative upper bound suffices, so we can also assume that $\hat{Q}_{\Sigma,H}^r(\tau^r(t)) > 0$. Then $\hat{Q}_{\Sigma,H}^r(\tau^r(t)-) = 0$ and $\hat{Q}_{\Sigma,H}^r(\tau^r(t)) > 0$. Therefore in the rth system at time $\tau^r(t)$ the exogeneous arrival process has a jump. This means that $\hat{Q}_{\Sigma,H}^r(\tau^r(t)) \leq \hat{E}_\Sigma^r(\tau^r(t)) - \hat{E}_\Sigma^r(\tau^r(t)-)$. Combining the bounds for $\tau^r(t) = 0$ or $\tau^r(t) > 0$ gives

$$\hat{Q}_{\Sigma,H}^r(\tau^r(t)) \leq \hat{Q}_{\Sigma,H}^r(0) + \hat{E}_\Sigma^r(\tau^r(t)) - \hat{E}_\Sigma^r(\tau^r(t)-),$$

where by convention $\hat{E}_\Sigma^r(0-) = \hat{E}_\Sigma^r(0) = 0$. Hence

$$\hat{Q}_{\Sigma,H}^r(t) \leq \hat{Q}_{\Sigma,H}^r(0) + \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)-) + \frac{1}{r} + r(t - \tau^r(t))\alpha_\Sigma^r. \tag{13}$$

For $r \in \mathcal{R}$ and $t \geq 0$ let $\theta^r(t) = t - \tau^r(t)$ and $\tilde{\theta}^r(t) = \theta^r(t) + \frac{1}{r^2}$. For now, suppose that

$$r\theta^r \Rightarrow 0. \tag{14}$$

Then, it follows from (14) that as $r \to \infty$,

$$\theta^r \Rightarrow 0 \quad \text{and} \quad \tilde{\theta}^r \Rightarrow 0. \tag{15}$$

Fix $r \in \mathcal{R}$ and $t \geq 0$. By (1), we have

$$\hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)-) = \hat{E}_\Sigma^r(t) - \frac{1}{r} E_\Sigma^r(r^2\tau^r(t)-) + r\tau^r(t)\alpha_\Sigma^r$$

which, together with the fact that the process E_Σ^r is nondecreasing, implies

$$\begin{aligned}
\hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)) &\leq \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)-) \\
&\leq \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r\left(\left(\tau^r(t) - \frac{1}{r^2}\right)^+\right) + \frac{\alpha_\Sigma^r}{r}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(t - \theta^r(t)) &\leq \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r(\tau^r(t)-) \\
&\leq \hat{E}_\Sigma^r(t) - \hat{E}_\Sigma^r\left(\left(t - \tilde{\theta}^r(t)\right)^+\right) + \frac{\alpha_\Sigma^r}{r}.
\end{aligned}$$

By (2), (15) and the fact that E^*_Σ is continuous almost surely, we obtain (see [5], Section 17) that for $t \geq 0$ as $r \to \infty$

$$\hat{E}^r_\Sigma(t) - \hat{E}^r_\Sigma(t - \theta^r(t)) \Rightarrow 0$$

and

$$\hat{E}^r_\Sigma(t) - \hat{E}^r_\Sigma\left((t - \tilde{\theta}^r(t))^+\right) + \frac{\alpha^r_\Sigma}{r} \Rightarrow 0.$$

Hence, as $r \to \infty$

$$\hat{E}^r_\Sigma(t) - \hat{E}^r_\Sigma(\tau^r(t)-) \Rightarrow 0. \tag{16}$$

Since the space $\mathbf{M}^K \times (\mathbf{D}[0,\infty))^K \times \mathbb{R}$ is separable, we can apply the Skorohod representation theorem ([5], Theorem 6.7) and assume that all the random elements in (6)-(8) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbf{P})$, on which as $r \to \infty$

$$\left(\hat{\mathcal{Z}}^r(0), \bar{S}^r, \hat{W}^r_\Sigma(0)\right)(\omega) \to (\mathcal{Z}^*(0), D, W^*_0)(\omega) \tag{17}$$

in $\mathbf{M}^K \times (\mathbf{D}[0,\infty))^K \times \mathbb{R}$ for almost every $\omega \in \Omega$. Fix such an ω. In this paragraph, all the random elements under consideration are evaluated at this ω. Observe that from the consistency conditions for functions S^x_k listed in Section II-A it follows that, for a given r, $\bar{S}^r_k(t) = \hat{\mathcal{Z}}_k(0)(x^*)$ for $t \geq x^*\hat{\mathcal{Z}}^r_\Sigma(0)(x^*)$. By (17), there exists a finite constant C such that $\hat{\mathcal{Z}}^r_\Sigma(0)(x^*) \leq C$ for all $r \in \mathcal{R}$. This means that all the functions $\bar{S}^r_k$ as well as the functions D_k are constant on $[Cx^*, \infty)$. In this case, convergence in the Skorohod topology to a continuous limit implies convergence in the uniform topology as well ([5], Section 12). Hence, by (17) as $r \to \infty$

$$|\hat{\mathcal{Z}}^r_\Sigma(0)(x^*) - \mathcal{Z}^*_\Sigma(0)(x^*)| = \left| \left\| \sum_{k\in\mathcal{K}} \bar{S}^r_k \right\|_\infty - \left\| \sum_{k\in\mathcal{K}} D_k \right\|_\infty \right| \tag{18}$$

$$\leq \left\| \sum_{k\in\mathcal{K}} \bar{S}^r_k - \sum_{k\in\mathcal{K}} D_k \right\|_\infty \to 0.$$

This, together with (6), (7), (17), gives us

$$\hat{Q}^r_{\Sigma,H}(0) = \langle \mathbb{I}_{[0,x^*)}, \hat{\mathcal{Z}}^r_\Sigma(0)\rangle \Rightarrow \langle \mathbb{I}_{[0,x^*)}, \mathcal{Z}^*_\Sigma(0)\rangle = 0, \ r \to \infty, \tag{19}$$

which in turn, together with (2), (13), (14), (16) implies (10).

Therefore it remains to prove (14). For each $r \in \mathcal{R}$ and $t \geq 0$, we examine the behavior of $W^r_{\Sigma,H}$ on time intervals of the form $(r^2\tau^r(t), r^2t]$ to derive an expression that relates $\hat{W}^r_{\Sigma,H}(t)$ and $\theta^r(t)$. In particular, since for each $r \in \mathcal{R}$ and $t \geq 0$, $Q^r_{\Sigma,H}(s) \neq 0$ for all $s \in (r^2\tau^r(t), r^2t]$ and the service discipline is SRPT, it follows that for each $r \in \mathcal{R}$ and $t \geq 0$,

$$\begin{aligned} W^r_{\Sigma,H}(r^2t) \leq\ & W^r_{\Sigma,H}(r^2\tau^r(t)) + V^r_{\Sigma,H}(r^2t) - V^r_{\Sigma,H}(r^2\tau^r(t)) \\ & + x^* - r^2(t - \tau^r(t)). \end{aligned}$$

Again, the addition of x^* on the right-hand side of the inequality is needed because of a possibility that there are only jobs with service time greater than or equal to x^* at time $r^2\tau^r(t)$ in the queue.

By applying diffusion scaling and rearranging, we obtain for $r \in \mathcal{R}$ and $t \geq 0$

$$\begin{aligned} \hat{W}^r_{\Sigma,H}(t) \leq\ & \hat{W}^r_{\Sigma,H}(\tau^r(t)) + \hat{V}^r_{\Sigma,H}(t) - \hat{V}^r_{\Sigma,H}(\tau^r(t)) + \frac{x^*}{r} \\ & + (\alpha^r_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle - 1)r\theta^r(t). \end{aligned}$$

Using the same line of reasoning that gave rise to (13), for $r \in \mathcal{R}$ and $t \geq 0$,

$$\begin{aligned} \hat{W}^r_{\Sigma,H}(t) \leq\ & \hat{W}^r_{\Sigma,H}(0) + \hat{V}^r_{\Sigma,H}(t) - \hat{V}^r_{\Sigma,H}(\tau^r(t)-) + \frac{x^*}{r} \\ & + (\alpha^r_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle - 1)r\theta^r(t). \end{aligned}$$

Since $\hat{W}^r_{\Sigma,H}(t) \geq 0$ for all $r \in \mathcal{R}$ and $t \geq 0$, it follows that for such r and t

$$\begin{aligned} (1 - \alpha^r_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle)r\theta^r(t) \leq\ & \hat{W}^r_{\Sigma,H}(0) + \hat{V}^r_{\Sigma,H}(t) \\ & - \hat{V}^r_{\Sigma,H}(\tau^r(t)-) + \frac{x^*}{r}. \end{aligned} \tag{20}$$

By (2) and the theorem assumption, we have that

$$\lim_{r\to\infty}\left(1 - \alpha^r_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle\right) = 1 - \alpha_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle > 0. \tag{21}$$

Hence, for r sufficiently large, $\left(1 - \alpha^r_\Sigma\langle\chi\mathbb{I}_{[0,x^*)}, \nu_\Sigma\rangle\right)\theta^r \geq 0$ for all $t \geq 0$. Using Proposition 1, one can prove (see [12]) that

$$\hat{V}^r_{\Sigma,H} \Rightarrow V^*_{\Sigma,H}, \ r \to \infty, \tag{22}$$

where $V^*_{\Sigma,H}$ is a Brownian motion starting from zero with zero drift and finite variance per unit time. Then (17) and (20)-(22) together imply that $\theta^r \Rightarrow 0$, $r \to \infty$. By the same line of reasoning that gave rise to (16), we have that for $t \geq 0$

$$\hat{V}^r_{\Sigma,H}(t) - \hat{V}^r_{\Sigma,H}(\tau^r(t)-) \Rightarrow 0.$$

By using this, (6), (7), (12) and (19)-(21) we obtain (14).

3) Proportional breakdown: For $r \in \mathcal{R}$ let $\sigma^r(t)$ be the time of arrival to the rth system of the job with service time x^* which most recently completed service before time t (if none has yet completed we define it as 0). Let $\bar{\sigma}^r(t) = \frac{1}{r^2}\sigma^r(r^2t)$, $t \geq 0$. We will now prove that $\bar{\sigma}^r \Rightarrow e$, where $e(t) = t$ for $t \geq 0$. For $t \geq 0$ we have

$$\begin{aligned} W^r_{\Sigma,L}(t) \geq\ & V^r_{\Sigma,L}(t) - V^r_{\Sigma,L}(\sigma^r(t)) - u^r(t) \\ & + x^*\left(\mathcal{Z}^r_\Sigma(0)(x^*) - S^r_\Sigma(L^r(t))\right), \end{aligned} \tag{23}$$

where $u^r(t)$ denotes the partial service (if any) that has been performed in the time interval $[\sigma^r(t), t)$ on the next job with service time x^*. Recall that $S^r_\Sigma(L^r(t))$ is the number of fully served initial jobs with initial service time x^* by time t in the rth system, so the last term is the workload of initial low priority jobs still present in the system at time t. The fact that (23) holds is a consequence of the FIFO discipline among the jobs with the same residual service time. Notice that $u^r(t) \leq x^*$. On the other hand, we also have that for $t \geq 0$

$$\begin{aligned} W^r_{\Sigma,L}(t) \leq\ & V^r_{\Sigma,L}(t) - V^r_{\Sigma,L}(\sigma^r(t)-) - u^r(t) \\ & + x^*\left(\mathcal{Z}^r_\Sigma(0)(x^*) - S^r_\Sigma(L^r(t))\right) + \langle\chi\mathbb{I}_{(x^*,\infty)}, \mathcal{Z}^r_\Sigma(0)\rangle, \end{aligned} \tag{24}$$

where $V^r_{\Sigma,L}(0-) = V^r_{\Sigma,L}(0) = 0$ by convention. The last term in (24) takes into account possible initial jobs with processing times greater than x^*. Under the diffusion scaling $\hat{W}^r_{\Sigma,L}(t) = \frac{1}{r}W^r_{\Sigma,L}(r^2t)$ we have $\hat{W}^r_{\Sigma,L} = \hat{W}^r_{\Sigma} - \hat{W}^r_{\Sigma,H}$ and by (5) and (11),

$$\hat{W}^r_{\Sigma,L} \Rightarrow W^*_{\Sigma}. \qquad (25)$$

Under the diffusion scaling, since $\mathcal{Z}^r_{\Sigma}(0)(x^*) - S^r_{\Sigma}(L^r(t)) \geq 0$, (23) yields for $t \geq 0$

$$\begin{aligned}\hat{W}^r_{\Sigma,L}(t) \geq \hat{V}^r_{\Sigma,L}(t) - \hat{V}^r_{\Sigma,L}(\bar{\sigma}^r(t)) - \frac{1}{r}u^r(r^2t)\\ + r\alpha^r_{\Sigma}\nu_{\Sigma}(x^*)(t - \bar{\sigma}^r(t)). \qquad (26)\end{aligned}$$

Since $\sigma^r(t) \leq t$, (26) gives us after rearranging

$$\begin{aligned}0 \leq t - \bar{\sigma}^r(t) \leq \frac{1}{r\alpha^r_{\Sigma}\nu_{\Sigma}(x^*)}\Big(\hat{W}^r_{\Sigma,L}(t)\\ -\hat{V}^r_{\Sigma,L}(t) + \hat{V}^r_{\Sigma,L}(\bar{\sigma}^r(t)) + \frac{1}{r}u^r(r^2t)\Big). \qquad (27)\end{aligned}$$

Let $T > 0$. Since convergence in the Skorohod topology to a continuous limit implies convergence in the uniform topology, we have $\|\hat{W}^r_{\Sigma,L}\|_T \Rightarrow \|\hat{W}^*_{\Sigma}\|_T$. For $t \in [0,T]$, we can bound $\hat{V}^r_{\Sigma,L}(\bar{\sigma}^r(t))$ by $\|\hat{V}^r_{\Sigma,L}\|_T$, so (27) after taking norms gives

$$\|t - \bar{\sigma}^r(t)\|_T \leq \frac{1}{r\alpha^r_{\Sigma}\nu_{\Sigma}(x^*)}\left(\|\hat{W}^r_{\Sigma,L}\|_T + 2\|\hat{V}^r_{\Sigma,L}\|_T + \frac{x^*}{r}\right).$$

Define the continuous functions $h_r : \mathbb{R} \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$ by $h_r(x) = \frac{x}{r}$ and $h(x) = 0, x \in \mathbb{R}$. Then, by Theorem 2.7 of [5], $h_r(\|\hat{W}^r_{\Sigma,L}\|_T + 2\|\hat{V}^r_{\Sigma,L}\|_T + \frac{x^*}{r}) \Rightarrow h(\|\hat{W}^*_{\Sigma}\|_T + 2\|\hat{V}^*_{\Sigma,L}\|_T) = 0$. Thus the right hand side above converges in probability to zero, which in turn implies that $\bar{\sigma}^r \Rightarrow e$ by Theorem 3.1 of [5].

For $t \geq 0$, let

$$I^r(t) := x^*\left(\hat{\mathcal{Z}}^r_{\Sigma}(0)(x^*) - \bar{S}^r_{\Sigma}(\bar{L}^r(rt))\right) + r\alpha^r_{\Sigma}\nu_{\Sigma}(x^*)(t - \bar{\sigma}^r(t)),$$

where $\bar{L}^r(t) = \frac{1}{r}L^r(rt)$. From (23)-(24) we have that

$$I^r(t) \leq \hat{W}^r_{\Sigma,L}(t) - \hat{V}^r_{\Sigma,L}(t) + \hat{V}^r_{\Sigma,L}(\bar{\sigma}^r(t)) + \frac{1}{r}u^r(r^2t), \qquad (28)$$

$$\begin{aligned}I^r(t) \geq \hat{W}^r_{\Sigma,L}(t) - \hat{V}^r_{\Sigma,L}(t) + \hat{V}^r_{\Sigma,L}(\bar{\sigma}^r(t)-) + \frac{1}{r}u^r(r^2t)\\ - \langle \chi\mathbb{I}_{(x^*,\infty)}, \hat{\mathcal{Z}}^r_{\Sigma}(0)\rangle. \qquad (29)\end{aligned}$$

Notice that

$$\langle \chi\mathbb{I}_{(x^*,\infty)}, \hat{\mathcal{Z}}^r_{\Sigma}(0)\rangle \Rightarrow 0, \; r \to \infty. \qquad (30)$$

Indeed,

$$\hat{W}^r_{\Sigma}(0) = \langle \chi\mathbb{I}_{[0,x^*]}, \hat{\mathcal{Z}}^r_{\Sigma}(0)\rangle + \langle \chi\mathbb{I}_{(x^*,\infty)}, \hat{\mathcal{Z}}^r_{\Sigma}(0)\rangle.$$

By (7) and (18), $\langle \chi\mathbb{I}_{[0,x^*]}, \hat{\mathcal{Z}}^r_{\Sigma}(0)\rangle \Rightarrow W^*_0$, so, taking (4) into account, we see that (30) must hold.

Using Proposition 1, one can prove (see [12]) that

$$\hat{V}^r_{\Sigma,L} \Rightarrow V^*_{\Sigma,L}, \; r \to \infty, \qquad (31)$$

where $V^*_{\Sigma,L}$ is a Brownian motion starting from zero with zero drift and finite variance per unit time. From (28)-(29), using (25), (30)-(31), the fact that $\bar{\sigma}^r \Rightarrow e$ and the Random Time Change Theorem ([5], Theorem 14.4) we have that

$$I^r \Rightarrow W^*_{\Sigma}. \qquad (32)$$

We can now obtain the desired breakdown for the workload processes. For $t \geq 0, k \in \mathcal{K}$ we have the following inequalities, analogous to (23)-(24):

$$\begin{aligned}W^r_{k,L}(t) \leq V^r_{k,L}(t) - V^r_{k,L}(\sigma^r(t)-) - u^r_k(t)\\ + x^*\left(\mathcal{Z}^r_k(0)(x^*) - S^r_k(L^r(t))\right) + \langle \chi\mathbb{I}_{(x^*,\infty)}, \mathcal{Z}^r_k(0)\rangle, \qquad (33)\end{aligned}$$

$$\begin{aligned}W^r_{k,L}(t) \geq V^r_{k,L}(t) - V^r_{k,L}(\sigma^r(t)) - u^r_k(t)\\ + x^*\left(\mathcal{Z}^r_k(0)(x^*) - S^r_k(L^r(t))\right), \qquad (34)\end{aligned}$$

where $u^r_k(t)$ denotes the partial service (if any) that has been performed in the time interval $[\sigma^r(t), t)$ on the next job of class k with service time x^*. Notice that $u^r_k(t) \leq x^*$ for all $k \in \mathcal{K}$.

Under diffusion scaling, (33) yields

$$\begin{aligned}\hat{W}^r_{k,L}(t) \leq \hat{V}^r_{k,L}(t) - \hat{V}^r_{k,L}(\bar{\sigma}^r(t)-) - \frac{1}{r}u^r_k(r^2t)\\ + x^*\left(\hat{\mathcal{Z}}^r_k(0)(x^*) - \bar{S}^r_k(\bar{L}^r(rt))\right) \qquad (35)\\ + r\alpha^r_k\nu_k(x^*)(t - \bar{\sigma}^r(t)) + \langle \chi\mathbb{I}_{(x^*,\infty)}, \hat{\mathcal{Z}}^r_k(0)\rangle\end{aligned}$$

and (34) yields

$$\begin{aligned}\hat{W}^r_{k,L}(t) \geq \hat{V}^r_{k,L}(t) - \hat{V}^r_{k,L}(\bar{\sigma}^r(t)) - \frac{1}{r}u^r_k(r^2t)\\ + x^*\left(\hat{\mathcal{Z}}^r_k(0)(x^*) - \bar{S}^r_k(\bar{L}^r(rt))\right) \qquad (36)\\ + r\alpha^r_k\nu_k(x^*)(t - \bar{\sigma}^r(t))\end{aligned}$$

for $t \geq 0, \; k \in \mathcal{K}$. Since $\mathcal{Z}^r_k \leq \mathcal{Z}^r_{\Sigma}$ and χ is nonnegative, from (30) we have that as $r \to \infty$

$$\langle \chi\mathbb{I}_{(x^*,\infty)}, \hat{\mathcal{Z}}^r_k(0)\rangle \Rightarrow 0. \qquad (37)$$

Suppose that for each $k \in \mathcal{K}$, $r \in \mathcal{R}$

$$\begin{aligned}I^r_k(t) := x^*\left(\hat{\mathcal{Z}}^r_k(0)(x^*) - \bar{S}^r_k(\bar{L}^r(rt))\right) + r\alpha^r_k\nu_k(x^*)(t - \bar{\sigma}^r(t))\\ \Rightarrow \frac{\alpha_k\nu_k(x^*)}{\alpha_{\Sigma}\nu_{\Sigma}(x^*)}W^*_{\Sigma}(t). \qquad (38)\end{aligned}$$

Using this, (35)-(37), the fact that $\bar{\sigma}^r \Rightarrow e$ and the Random Time Change Theorem, we can write that

$$\hat{W}^r_{k,L} \Rightarrow \frac{\alpha_k\nu_k(x^*)}{\alpha_{\Sigma}\nu_{\Sigma}(x^*)}W^*_{\Sigma} = p_k\frac{\nu_k(x^*)}{\nu_{\Sigma}(x^*)}W^*_{\Sigma}. \qquad (39)$$

From Section IV-A2 and (30) it follows that $\hat{\mathcal{Z}}^r_{\Sigma} \Rightarrow \frac{W^*_{\Sigma}}{x^*}\delta_{x^*}$. Taking this into account gives us (9). Therefore it remains to prove (38).

Recall from (2) that $\alpha^r \to \alpha$ as $r \to \infty$ and that by the Skorohod representation theorem we may assume that all the random elements in (6)-(8) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbf{P})$ such that (17) holds for almost every $\omega \in \Omega$. In what follows, all the random elements are evaluated at such an ω. Using an analogous line of reasoning as the one that led us to (18) we obtain that, as $r \to \infty$,

$$|\hat{\mathcal{Z}}_k^r(0)(x^*) - \mathcal{Z}_k^*(0)(x^*)| \to 0 \tag{40}$$

and, similarly,

$$\sup_{t \geq 0} |\bar{S}_k^r(t) - D_k(t)| \to 0. \tag{41}$$

Observe that

$$I_k^r - \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} I^r = x^* \left(\hat{\mathcal{Z}}_k^r(0)(x^*) - \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \hat{\mathcal{Z}}_\Sigma^r(0)(x^*) + \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \bar{S}_\Sigma^r(\bar{L}^r(rt)) - \bar{S}_k^r(\bar{L}^r(rt)) \right).$$

By (7),

$$\mathcal{Z}_k^*(0)(x^*) = \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \frac{W_0^*}{x^*} = \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \mathcal{Z}_\Sigma^*(0)(x^*),$$

so by (40),

$$\left| \hat{\mathcal{Z}}_k^r(0)(x^*) - \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \hat{\mathcal{Z}}_\Sigma^r(0)(x^*) \right| \to 0$$

as $r \to \infty$. Similarly, by (8) and (41)

$$\sup_{t \geq 0} \left| \bar{S}_k^r(\bar{L}^r(rt)) - \frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} \bar{S}_\Sigma^r(\bar{L}^r(rt)) \right| \to 0$$

as $r \to \infty$, which implies that the limits as $r \to \infty$ of I_k^r and $\frac{\alpha_k \nu_k(x^*)}{\alpha_\Sigma \nu_\Sigma(x^*)} I^r$ coincide and, by (32), proves (38).

V. The case of $\nu_\Sigma(x^*) = 0$

In this case we take a seemingly different approach. We start from modeling an exogeneous arrival process E_Σ common for all jobs. When a job arrives at the system, we randomly assign it to a class $1, ..., K$ with probabilities $p_1, ..., p_K$ correspondingly[1]. We also assume that there are no customers in the system at time 0. This is described more formally below.

Let $(\Omega, \mathcal{F}, \mathbf{P})$ be the probability space on which the stochastic elements of the model are defined. Let E_Σ be the exogenous arrival process, i.e. for $t \geq 0$, $E_\Sigma(t)$ is the number of arrivals to the system in the time interval $(0, t]$. It is a delayed renewal process with rate $\alpha_\Sigma > 0$ such that the interarrival times have standard deviation $a_\Sigma \geq 0$. Then job j arrives at time $T^j = \inf\{t \geq 0 : E_\Sigma(t) \geq j\}$.

Let $\{\varphi_i\}_{i \in \mathbb{N}}$ be i.i.d. random vectors, independent of E_Σ such that for each i $\varphi_i = (\varphi_{i,1}, ..., \varphi_{i,K})$, where $\varphi_{i,k} = 1$ if job i belongs to class k and $\varphi_{i,k} = 0$ otherwise. We assume that $\mathbf{P}(\varphi_i = e_k) = p_k$, where e_k is the kth unit vector in $\mathbb{R}^K$ and $p_k \in (0,1)$ for each $k \in \mathcal{K}$, $\sum_{k \in \mathcal{K}} p_k = 1$. Put $p = (p_1, ..., p_K)$. Let $\Phi(n) = (\Phi_1(n), ..., \Phi_K(n)) = \sum_{i=1}^n \varphi_i$. In words, $\Phi_k(n)$ is the number of class k jobs among the first n jobs which arrived in the system. We can now define $E_k(t) = \Phi_k(E_\Sigma(t))$, $k \in \mathcal{K}$, $t \geq 0$ and $E(t) = (E_1(t), ..., E_K(t))$. Put $\alpha_k = p_k \alpha_\Sigma$, $k \in \mathcal{K}$ and $\alpha = (\alpha_1, ..., \alpha_K)$. We define the processes $\mathcal{Z}$, W, Q and $\mathcal{V}$ analogously as in Section II. We assume that $\mathcal{Z}_k(0) = \mathbf{0}$ for each $k \in \mathcal{K}$.

Let v_i represent the initial service time of the ith job in the system. We assume that $(\varphi_i, v_i)_{i \geq 1}$ are i.i.d. random vectors independent of E and that ν_k is the conditional distribution of v_i under the condition of $\varphi_i = e_k$, $k \in \mathcal{K}$. We put $\nu_\Sigma = \sum_{k \in \mathcal{K}} p_k \nu_k$. For $j \in \mathbb{N}$, $k \in \mathcal{K}$ and $t \geq T^j$ let $w^j(t)$ denote the residual service time of job j at time t. As before, the SRPT protocol is used.

We apply diffusion scaling for a sequence of systems similarly as in Section III. Let $\mathcal{R}$ be a sequence of positive numbers increasing to infinity. Consider an $\mathcal{R}$-indexed sequence of stochastic models presented in the previous paragraph. For each $r \in \mathcal{R}$, there are stochastic primitives E_Σ^r, $\{\varphi_i^r\}_{i \in \mathbb{N}}$, Φ_k^r, E^r, $\{v_i^r\}_{i \in \mathbb{N}}$ with parameters $\alpha^r, a_\Sigma^r, p_k^r, \nu_k^r$. The stochastic elements of each model are defined on a probability space $(\Omega^r, \mathcal{F}^r, \mathbf{P}^r)$ with expectation operator $\mathbb{E}^r$ and variance operator Var^r. We also have arrival times $\{T^{r,j}\}_{j \in \mathbb{N}}$, a state descriptor $\mathcal{Z}^r$ and processes $W^r, Q^r, \mathcal{V}^r$.

A diffusion scaling is applied to each model in the $\mathcal{R}$-indexed sequence as in (1). Furthermore, for each $r \in \mathcal{R}$ and $t \geq 0$, let

$$\hat{\Phi}^r(t) = \frac{1}{r}(\Phi^r(\lfloor r^2 t \rfloor) - \lfloor r^2 t p^r \rfloor).$$

Let $\alpha_\Sigma, a_\Sigma \in (0, +\infty)$ and define $\alpha_\Sigma(t) = \alpha_\Sigma t$, $t \geq 0$, $p = (p_1, ..., p_k) \in (0,1)^K$, $\sum_{k \in \mathcal{K}} p_k = 1$. We make the following asymptotic assumptions for the sequence of stochastic primitives. Assume that as $r \to \infty$,

$$\alpha_\Sigma^r \to \alpha_\Sigma, \qquad a_\Sigma^r \to a_\Sigma, \qquad p^r \to p, \qquad \hat{E}_\Sigma^r \Rightarrow E_\Sigma^*, \tag{42}$$

where E_Σ^* is a Brownian motion starting from zero with drift zero and variance $a_\Sigma^2 \alpha_\Sigma^3$ per unit time. Moreover, we assume that $\nu^r = \nu$, i.e., ν^r does not depend on r.

We will now determine the limits of processes $\hat{\Phi}^r$ and $\hat{E}^r$ as $r \to \infty$. We first apply Proposition 1 to the processes $\hat{\Phi}^r = \left(\hat{\Phi}_k^r, r = 1, ..., K\right)$. Fix $k \in \mathcal{K}$. For each $r \in \mathbb{R}$, $\{\varphi_{i,k}^r\}_{i \in \mathbb{N}}$ are independent Bernoulli distributed random variables such that $\mathbf{P}^r(\varphi_{i,k}^r = 1) = p_k^r$. Hence $\mathbb{E}^r \varphi_{i,k}^r = p_k^r$ and $\mathrm{Var}^r\ \varphi_{i,k}^r = p_k^r(1 - p_k^r)$. Using the fact that $p_k^r \to p_k$ and Proposition 1, we obtain

$$\frac{1}{r} \left(\sum_{i=1}^{\lfloor r^2 t \rfloor} \varphi_{i,k} - \lfloor r^2 t \rfloor p_k \right) \Rightarrow \Phi_k^*, \ r \to \infty, \tag{43}$$

where Φ_k^* is a Brownian motion starting from zero with drift zero and variance $p_k(1 - p_k)$ per unit time. By using the Cramer-Wold device, multidimensional central limit theorem and Prohorov theorem, we can generalize this to joint convergence $\hat{\Phi}^r \Rightarrow \Phi^*$, where Φ^* is a K-dimensional Brownian motion starting from 0 with drift 0 and covariance matrix $C = [c_{ij}]_{i,j}$ such that $c_{ii} = p_i(1 - p_i)$, $c_{i,j} = -p_i p_j$ for $i \neq j$ (cf. [26], proof of Theorem 4.3.5). Moreover, this convergence is joint with (42) and Φ^* is independent of E_Σ^*.

[1] As we will see, this is a special case of the approach considered in Sections 1-4, with the exogeneous arrival process E having dependent coordinates.

Now we use Proposition 1 again to obtain the limit of the process $\hat{E}^r = \left(\hat{E}^r_k, r = 1, ..., K\right)$. Fix $k \in \mathcal{K}$. Observe that

$$\hat{E}^r_k(t) = \frac{\Phi^r_k\left(E^r_\Sigma(r^2t)\right) - r^2t\alpha^r_\Sigma p^r_k}{r}.$$

Therefore, by Proposition 1, $\hat{E}^r_k \Rightarrow E^*_k$, where $E^*_k(t) = \Phi^*_k(\alpha_\Sigma t) + p_k E^*_\Sigma(t), t \geq 0$. Note that E^*_k is a Brownian motion starting from 0 with drift 0 and variance per unit time $\alpha_\Sigma p_k(1-p_k)+p_k^2\alpha_\Sigma^3 a_\Sigma^2$. Therefore, by an analogous argument as in the previous paragraph, we obtain joint convergence $\hat{E}^r \Rightarrow E^*$, where $E^*(t) = \Phi^*(\alpha_\Sigma t) + pE^*_\Sigma(t), t \geq 0$. Note that E^* is a K-dimensional Brownian motion starting from 0 with drift 0 and covariance matrix $D = [d_{ij}]_{i,j}$ such that $d_{ii} = p_i^2 a_\Sigma^2\alpha_\Sigma^3 + \alpha_\Sigma^2 p_i(1-p_i)$, $d_{ij} = \mathrm{Cov}(\Phi^*_i(\alpha_\Sigma) + p_i E^*_\Sigma(1), \Phi^*_j(\alpha_\Sigma) + p_j E^*_\Sigma(1)) = p_i p_j a_\Sigma^2 \alpha_\Sigma^3 - \alpha_\Sigma^2 p_i p_j$, $i \neq j$.

Before we proceed, we present a general property of the SRPT protocol, which will be used in the following proofs.

Lemma 1. *Consider a single-server queueing system with customers served according to the SRPT protocol. Let $t \geq 0$ and let i, j be two jobs present in the system at time t with initial processing times v_i, v_j and residual processing times $w_i(t), w_j(t)$ respectively. Then the intervals $(w_i(t), v_i)$ and $(w_j(t), v_j)$ are disjoint.*

Proof. Suppose that $(w_i(t), v_i) \cap (w_j(t), v_j) \neq \varnothing$. First, consider the case when neither of the intervals is a subset of the other. Without loss of generality we may assume that $w_i(t) \leq w_j(t) < v_i \leq v_j$, thus $(w_i(t), v_i) \cap (w_j(t), v_j) = (w_j(t), v_i)$. This means that the job j, even though its initial processing time was not less than v_i, was partially served so that its residual processing time is lower than the initial processing time of the job i. Since the system uses the SRPT protocol, this is possible only when $v_i = v_j$ or when i arrived after j. In the first case, the start of the processing immediately breaks the tie so i cannot be partially processed before j gets fully serviced, thus $w_i(t) = v_i$ and therefore this case is impossible. In the second case, let t_1 be the time of the arrival of the job i. If $w_j(t_1) > v_i$, then for any $t \geq t_1$ the job j could only be chosen for processing after i had been fully serviced, which means that this case is impossible. If $w_j(t_1) \leq v_i$, then for any $t \geq t_1$ the job i could only be chosen for processing after j is fully serviced, which means that this case is impossible as well.

Finally, consider the case when one of the intervals is a subset of the other. Without loss of generality we can assume that $w_j(t) \leq w_i(t) < v_i \leq v_j$. Arguing as above, we can obtain that this is also impossible. This leads to a contradiction, therefore the intervals $(w_i(t), v_i)$ and $(w_j(t), v_j)$ are disjoint. □

We are now ready to formulate the main theorem in the case under consideration.

Theorem 2. *Let $x^* < \infty$, $\nu_\Sigma(x^*) = 0$. Assume that for every $k \in \mathcal{K}$ the limit*

$$\gamma_k = \lim_{x \uparrow x^*} f_k(x), \tag{44}$$

where f_k is the Radon–Nikodym derivative of the measure ν_k with respect to ν_Σ, exists. Under the assumptions of this section, we have that, as $r \to \infty$,

$$\left(\hat{\mathcal{Z}}^r_k, k = 1, ..., K\right) \overset{fd}{\to} \left(p_k\gamma_k \frac{W^*_\Sigma}{x^*}\delta_{x^*}, k = 1, ..., K\right). \tag{45}$$

Proof (somewhat heuritstic). We first show that

$$\hat{\mathcal{Z}}^r_\Sigma \Rightarrow \frac{W^*_\Sigma}{x^*}\delta_{x^*},\ r \to \infty. \tag{46}$$

and that, for any $x \in (0, x^*)$

$$\left\langle \mathbb{I}_{[0,x)}, \hat{\mathcal{Z}}^r_\Sigma(t) \right\rangle \Rightarrow 0, \quad \left\langle \chi\mathbb{I}_{[0,x)}, \hat{\mathcal{Z}}^r_\Sigma(t) \right\rangle \Rightarrow 0,\ r \to \infty. \tag{47}$$

This is done similarly as in the first part of the proof of Theorem 1.

Fix $t > 0$ and $\varepsilon > 0$. Let $\{i^r_j\}_j$ be the sequence of jobs present in the system r and having residual processing time in $(x^* - \varepsilon, x^*]$ at time t, i.e. such that $w^r_{i_j}(t) > x^* - \varepsilon$. In what follows, we will simply write i_j instead of i^r_j when the system in question can be inferred from the context. Let $n^r(t)$ be the number of such jobs (of all classes) in the rth system at time t. Notice that

$$\sum_{j=1}^{n^r(t)} v^r_{i_j} \geq W^r_\Sigma(t) - \left\langle \chi\mathbb{I}_{[0,x^*-\varepsilon]}, \mathcal{Z}^r_\Sigma(t) \right\rangle. \tag{48}$$

We claim that

$$\sum_{j=1}^{n^r(t)} v^r_{i_j} \leq W^r_\Sigma(t) - \left\langle \chi\mathbb{I}_{[0,x^*-\varepsilon]}, \mathcal{Z}^r_\Sigma(t) \right\rangle + \varepsilon. \tag{49}$$

Indeed, by Lemma 1, the intervals of the form $(w^r_{i_j}(t), v^r_{i_j})$ are pairwise disjoint. All of these intervals are contained in $(x^* - \varepsilon, x^*]$ by the definition of the sequence $\{i^r_j\}_j$, therefore the sum of their lengths cannot exceed ε.

By (5) and (47)-(49) we obtain that as $r \to \infty$[2]

$$\frac{1}{r}\sum_{j=1}^{n^r(r^2t)} v^r_{i_j} \Rightarrow W^*_\Sigma(t). \tag{50}$$

Observe that for $r \in \mathcal{R}$ and almost every (with respect to ν_Σ) $x \in (x^* - \varepsilon, x^*)$

$$\begin{aligned} &\mathbf{P}^r(\varphi^r_i = e_k | v^r_i = x) \\ &= \frac{\mathbf{P}^r(\varphi^r_i = e_k)}{\mathbf{P}^r(v^r_i = x)} \cdot \mathbf{P}^r(v^r_i = x | \varphi^r_i = e_k) \\ &= p^r_k f_k(x). \end{aligned} \tag{51}$$

Consider the sequence $\{v^r_{i_j}\varphi^r_{i_j,k}\}_j$. Obviously $\varphi^r_{i_j,k} = 1$ if the job i^r_j belongs to the class k and $\varphi^r_{i_j,k} = 0$ otherwise. Similarly as before we can show that the limits of $\hat{W}^r_k(t)$ and $\frac{1}{r}\sum_{j=1}^{n^r(r^2t)} v^r_{i_j}\varphi^r_{i_j,k}$ coincide. By (47) and the fact that $Q^r_\Sigma(t) = n^r(t) + \left\langle \mathbb{I}_{[0,x^*-\varepsilon]}, \mathcal{Z}^r_\Sigma(t) \right\rangle$ we obtain that as $r \to \infty$

$$\hat{n}^r(t) := \frac{1}{r}n^r(r^2t) \Rightarrow Q^*_\Sigma(t) = \frac{W^*_\Sigma(t)}{x^*}. \tag{52}$$

[2] In the context of diffusion scaling, we define the sequence $\{i_j\}_j$ for time r^2t instead of t.

If $n^r(r^2t)$ is of the order less than r, then by (47)-(48) and the bound $v_i \leq x^*$ for all i, $\hat{W}^r_\Sigma(t)$ is asymptotically negligible. On the other hand, if $n^r(r^2t)$ is large, using the law of large numbers, we get,

$$\frac{1}{n^r(r^2t)}\sum_{j=1}^{n^r(r^2t)} v^r_{i_j}\varphi^r_{i_j,k} \approx \frac{1}{n^r(r^2t)}\sum_{j=1}^{n^r(r^2t)} \mathbb{E}^r\left(v^r_{i_j}\varphi^r_{i_j,k}\right) \tag{53}$$

and by (51), (44),

$$\begin{aligned}\mathbb{E}^r\left(v^r_{i_j}\varphi^r_{i_j,k}\right) &= \int_{x^*-\varepsilon}^{x^*} u\mathbf{P}^r(\varphi^r_{i_j} = e_k | v^r_i = u)dF^r_j(u) \quad (54)\\ &= p^r_k\int_{x^*-\varepsilon}^{x^*} uf_k(u)dF^r_j(u)\\ &= p^r_k(\gamma_k + o(1))(x^* + O(\varepsilon)),\end{aligned}$$

where F^r_j is the distribution function of $v^r_{i_j}$ and $O(\varepsilon) \in [-\varepsilon, 0]$. Therefore, from (52)-(54) and (42) it follows that, as $r \to \infty$, any weak limiting distribution of a subsequence of the sequence

$$\frac{1}{r}\sum_{j=1}^{n^r(r^2t)} v^r_{i_j}\varphi^r_{i_j,k} = \frac{\hat{n}^r(r^2t)}{n^r(r^2t)}\sum_{j=1}^{n^r(r^2t)} v^r_{i_j}\varphi^r_{i_j,k}$$

is stochastically bounded from below by the distribution of

$$p_k\frac{W^*_\Sigma}{x^*}(\gamma_k(x^*-\varepsilon) + o(1))$$

and stochastically bounded from above by the distribution of

$$p_kW^*_\Sigma(\gamma_k + o(1)).$$

By letting $\varepsilon \downarrow 0$ and taking (46), (50) into account we obtain the desired breakdown and convergence of one-dimensional distributions. It is easy to extend this result to convergence of finite-dimensional distributions.

□

VI. Simulations

In this section we present the results of our computer simulations. We simulated the system described in Section II with two user classes. The times between arrivals of jobs of each class are exponentially distributed with parameters α_1, α_2 correspondingly.

We first consider the case when $\nu_\Sigma(x^*) > 0$. We assume that $\alpha_1 = 0.25$, $\alpha_2 = 1.25$. Hence $\alpha_\Sigma = 1.5$. The initial service times of the first class take the values $0.5, 1, 1.5$ with equal probabilities and the initial service times of the second class take the values $0.2, 0.3, 0.4, 0.6, 1.5$ with equal probabilities. Then we have $x^* = 1.5$, $\rho = 1$, $p_1 = 1/6$, $p_2 = 5/6$, $\nu_\Sigma(1.5) = 2/9$ and $p_1\nu_1(1.5)/\nu_\Sigma(1.5) = 1/4$. We assume that there are no customers in the system at time 0. The results of the simulation in this case are shown in Fig. 1.

Let us add an initial condition consisting of 25 jobs in each of two classes, with the same service time distributions as jobs arriving in the system in the corresponding class. Notice that their workloads are not even approximately distributed according to the asymptotic proportions stated in Theorem 1. The results are shown in Figure 2.

Now, let us change the initial condition. It still consists of 25 jobs in each class, but their processing times are distributed uniformly on $[0, 3]$, which is even further from the assumptions on the initial condition in Theorem 1. The results are shown in Fig. 3.

Now let us consider the case when $\nu_\Sigma(x^*) = 0$. Assume that $\alpha_1 = 1$, $\alpha_2 = 0.6$. Then $\alpha_\Sigma = 1.6$. The initial service times are uniformly distributed in the interval $[0, 1]$ and $[2/3, 1]$ correspondingly. This gives us $x^* = 1$, $\rho = 1$, $p_1 = 0.625$, $p_2 = 0.375$, $\gamma_1 = 4/7$ and $p_1\gamma_1 = 5/14 \approx 0.357$. We assume that there are no customers in the system at time 0. The results are shown in Fig. 4.

We add an initial condition consisting of 25 jobs in each of two classes, with the same service time distributions as jobs arriving in the system in the corresponding class. The results are shown in Fig. 5.

Let us summarize the results. In Fig. 1 we can observe in the left chart that the proportion of workload of class 1 to the total workload in the system stabilizes at $p_1\nu_1\left(\frac{3}{2}\right)/\nu_\Sigma\left(\frac{3}{2}\right) = 1/4$ after a long time has passed, which confirms that Theorem 1 holds true. Moreover, we can notice that the blue graph in the right chart illustrating the predicted workload of class 1 obtained by applying Theorem 1 “lies close” to the red graph presenting the actual workload of class 1. The prediction at a given time is more accurate, if there are more tasks in the system at this time.

Adding an initial condition in the simulation in Fig. 2 does not noticeably change the situation, even though at time 0 a large amount of workload is not distributed between the two classes according to the proportions required by the assumptions in Theorem 1. We can also observe less instability in the left chart, since with such a number of initial tasks the queue lengths remains at higher levels, where the proportions are more stable. However, in Fig. 3 we can see that an initial condition consisting of tasks greatly different from those arriving in the system results in a notably slower convergence. While the decreasing trend is still visible in the left chart, it is very small in magnitude and in the simulated time horizon the system did not manage to stabilize. Therefore, applying Theorem 1 to predict the proportion of class 1 workload to the total workload in the system gives us an inaccurate approximation and the assumptions for the initial conditions cannot be omitted.

In Fig. 4-5 we can see that if $\nu(x^*) = 0$, we also obtain similar results as above and can make analogous observations. This indicates that Theorem 2 (and even its generalized form, without assuming a zero initial condition) holds true.

VII. Conclusion

In this paper, we have proved a diffusion limit theorem for the measure-valued state descriptor for a single-server queuing system with multiple job classes and bounded processing times of arriving jobs. In particular, we have shown that, under suitable assumptions, the workload and the queue length in the

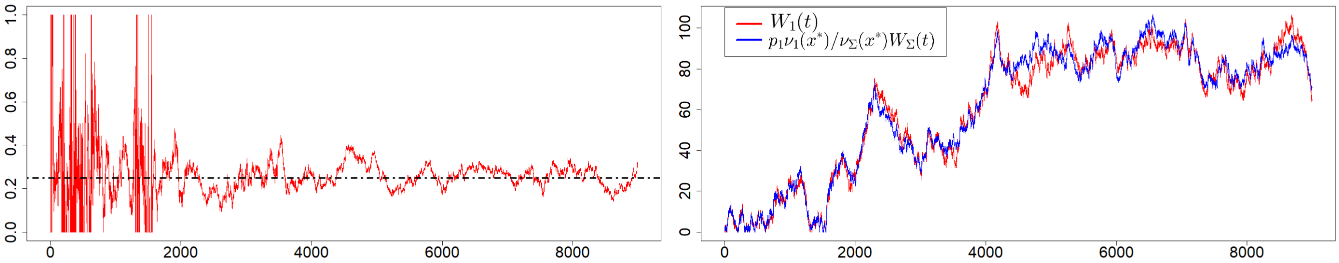


Fig. 1. Simulation in the case of $x^* < \infty, \nu_\Sigma(x^*) > 0$. The left chart illustrates the proportion of workload of class 1 to the total workload in the system as a function of time. The right chart presents the predicted workload of class 1 obtained as a result of applying Theorem 1 (blue) and the actual workload of class 1 (red).

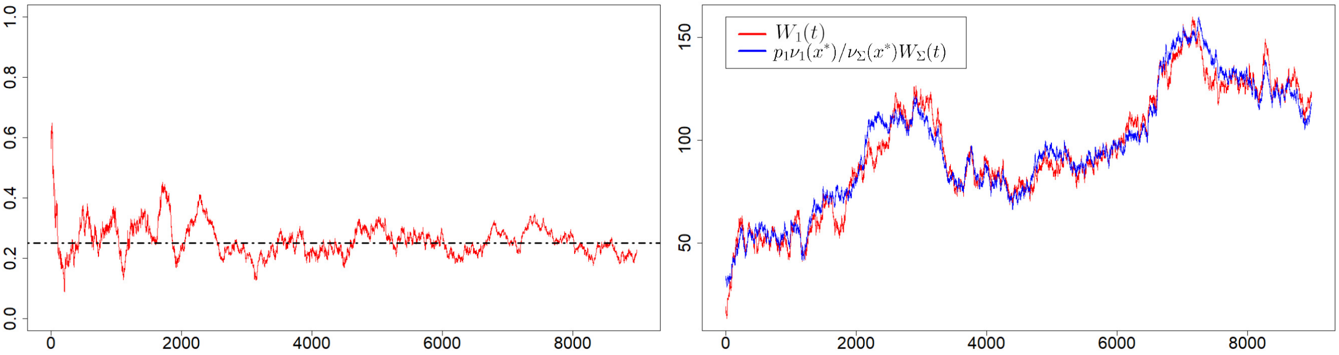


Fig. 2. Simulation in the case of $x^* < \infty, \nu_\Sigma(x^*) > 0$ with a non-zero initial condition. The interpretations of charts is the same as in Fig. 1.

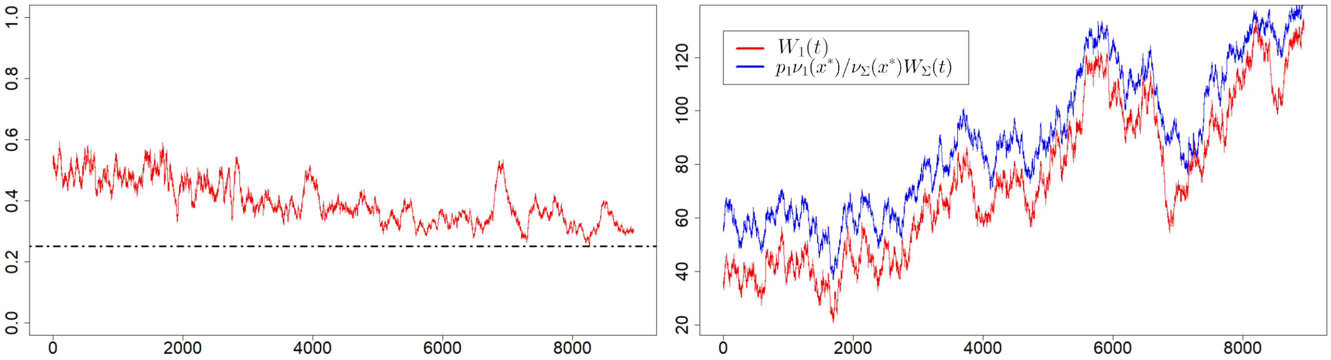


Fig. 3. Simulation in the case of $x^* < \infty, \nu_\Sigma(x^*) > 0$ with a different initial condition.

diffusion limit are divided between these classes according to specific proportions. This result can be applied in practice – it can be used to approximate proportions between workloads of different classes in the long run. The simulations presented in the last section indicate that it should be possible to further generalize Theorem 2 by removing the zero initial condition requirement.

References

[1] M. Agrawal, N. Barsal, M. Harchol-Balter, B. Schroeder, Implementation of SRPT scheduling in Web servers (2001). https://www.cs.cmu.edu/~harchol/Papers/srpt.impl.tech.rept.pdf

[2] M. Agrawal, N. Barsal, M. Harchol-Balter, B. Schroeder, Size-based scheduling to improve Web performance. ACM Transactions on Computer Systems. 21. (2002), https://doi.org/10.1145/762483.762486

[3] R. Atar, A. Biswas, H. Kaspi, K. Ramanan, A Skorokhod map on measure-valued paths with applications to priority queues. Annals of Applied Probability 28:418-481 (2018), https://doi.org/10.1214/17-AAP1309

[4] S. Banerjee, A, Budhiraja, A. L. Puha , Heavy traffic scaling limits for shortest remaining processing time queues with heavy tailed processing time distributions, Annals of Applied Probability **32**(4), 2587-2651 (2022), https://dx.doi.org/10.1214/21-AAP1741

[5] P. Billingsley, Convergence of Probability Measures (2nd Edition), John Wiley and Sons, Inc., New York, 1999.

[6] S. Cheng, Y. Cheng, Y. Fu, L. Liu, H. Wang, SFS: Smart OS scheduling for serverless functions, SC '22: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1-16 (2022), https://10.1109/SC41404.2022.00047

[7] T. Chojecki, Ł. Kruk, Instability of SRPT, SERPT and SJF queueing networks. Queueing Systems: Theory and Applications 101:57-92 (2022), https://doi.org/10.1007/s11134-021-09733-8

[8] J. Dong, R. Ibrahim, On the SRPT scheduling discipline in many-server

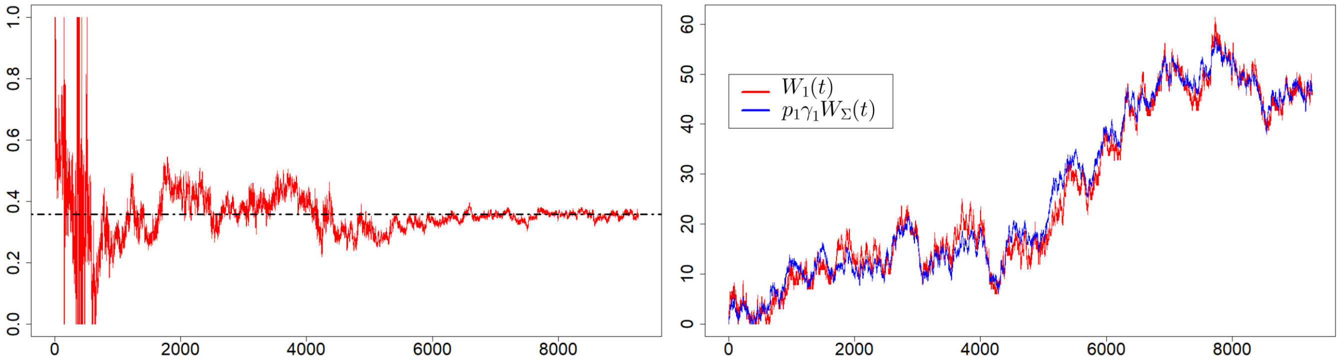


Fig. 4. Simulation in the case of $\nu_\Sigma(x^*) = 0$. As before, the left chart shows the proportion of workload of class 1 to the total workload in the system and the right chart presents the predicted workload of class 1 obtained as a result of applying Theorem 2 (blue) and the actual workload of class 1 (red).

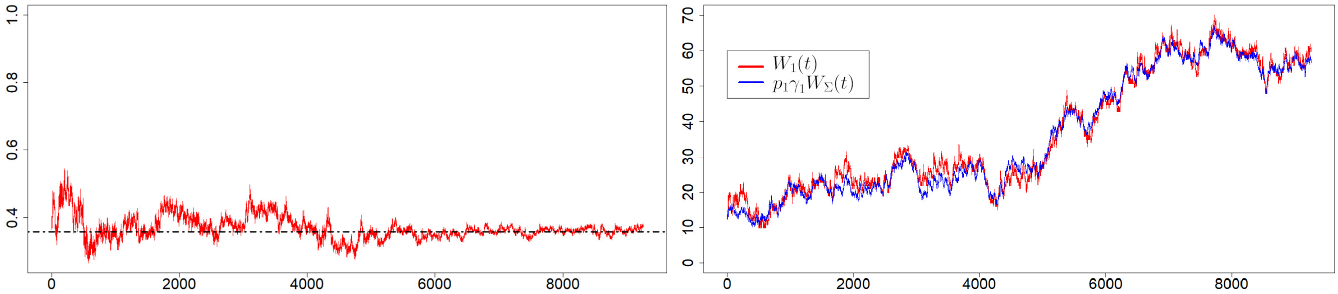


Fig. 5. Simulation in the case of $x^* < \infty, \nu_\Sigma(x^*) = 0$ with a non-zero initial condition.

queues with impatient customers. Management Science 67(12):7708-7718 (2021), https://doi.org/10.1287/mnsc.2021.4110

[9] S. N. Ethier, T. G. Kurtz. Markov Processes: Characterization and Convergence. John Wiley and Sons, Inc., New York, 1986.

[10] R. Gieroba, Ł. Kruk, Minimality of SRPT networks with resource sharing. WSEAS Transactions on Mathematics 20:74-83 (2021), https://doi.org/10.37394/23206.2021.20.8

[11] R. Gieroba, Ł. Kruk, Local edge minimality of SRPT networks with shared resources, Mathematical Methods of Operations Research, 96:459-492 (2022), https://doi.org/10.1007/s00186-022-00801-0

[12] H. C. Gromoll, Ł. Kruk, A. L. Puha, Diffusion limits for shortest remaining processing time queues, Stochastic Systems 1, 1-16, 2011, https://doi.org/10.1214/10-SSY016.

[13] I. Grosof, Z. Scully, M. Harchol-Balter, SRPT for multiserver systems. Performance Evaluation 127-128:154-175 (2018), https://doi.org/10.1145/3308897.3308902

[14] M. Harchol-Balter, B. Schroeder, Web servers under overload: How scheduling can help. ACM Transactions on Internet Technology 6 (2003), https://doi.org/10.1145/1125274.1125276

[15] D.L. Iglehart, W. Whitt, Multiple channel queues in heavy traffic I, Advances in Applied Probability 2, 150-177, https://doi.org/10.2307/3518347.

[16] Ł. Kruk, Diffusion Limits for SRPT and LRPT Queues via EDF Approximations, QTNA 2019, Lecture Notes in Computer Science, vol. 11688, pp. 263-275. Springer, Cham 2019, https://doi.org/10.1007/978-3-030-27181-7_16

[17] Ł. Kruk, E. Sokołowska, Fluid limits for multiple-input shortest remaining processing time queues, Mathematics of Operation Research 41, 1055-1092, 2016, https://doi.org/10.1287/moor.2015.0768.

[18] L. W. Miller, L. E. Schrage, The queue M/G/1 with the shortest remaining processing time discipline, Operations Research **14**, 670-684 (1966), https://doi.org/10.1287/opre.14.4.670

[19] R. Núñez-Queija: Queues with equally heavy sojourn time and service requirement distributions, Annals of Operations Research **113**, 101-117 (2002), https://doi.org/10.1023/A:1020905810996

[20] M. Nuyens, B. Zwart: A large deviations analysis of the GI/GI/1 SRPT queue, Queueing Systems **54**, 85-97 (2006), https://doi.org/10.1007/s11134-006-8767-1

[21] W. P. Peterson, A heavy traffic limit theorem for networks of Queues with multiple customer types, Mathematics of Operations Research 16, 90-118, 1991, https://doi.org/10.1287/moor.16.1.90

[22] Yu. V. Prohorov, Convergence of random processes and limit theorems in probability theory, Theory of Probability and its Applications 1, 157-214, 1956, https://doi.org/10.1137/1101016.

[23] A. L. Puha, Diffusion limits for shortest remaining processing time queues under nonstandard spatial scaling, Annals of Applied Probability **25**(6), 3381–3404 (2015), https://dx.doi.org/10.1214/14-AAP1076

[24] L. E. Schrage, A proof of the optimality of the shortest remaining processing time discipline, Operations Research 16, 687-690 (1968), https://doi.org/10.1287/opre.26.1.197

[25] F. Schreiber, Properties and applications of the optimal queueing strategy SRPT: a survey, *Archiv für Elektronik und Übertragungstechnik*, 47:372-378, 1993,

[26] W. Whitt, Stochastic-Process Limits: An Introduction to Stochastic-Process Limits and Their Application to Queues, Springer-Verlag, New York, 2002.

[27] A. Wierman and M. Harchol-Balter. Classifying scheduling policies with respect to unfairness in an M/G/1, Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 238-249, 2003, https://doi.org/10.1145/885651.781057

DOI: 10.15439/2023F6015
ISSN 2300-5963 ACSIS, Vol. 35

Genetic Algorithm for Planning and Scheduling Problem – StarCraft II Build Order case study

Konrad Gmyrek, Michał Antkiewicz, Paweł B. Myszkowski
Wrocław University of Science and Technology
Faculty of Information and Communication Technology
ul.I.Łukasiewicza 5, 50-371 Wrocław, Poland
Email: {konrad.gmyrek, michal.antkiewicz, pawel.myszkowski}@pwr.edu.pl

***Abstract*—The Planning and Scheduling (PS) problem plays a vital role in several domains, such as economics, military, management, finance, and games, where finding the optimal plan and schedule to achieve specific goals is essential. In this article, we present a Genetic Algorithm for the Planning and Scheduling (GAPS) problem in the StarCraft II Build Order Optimization problem (SC2 BO) context – as it signifies that modern strategy games present a more challenging environment than classical planning problems. We evaluate the performance of GAPS and compare it with state-of-the-art methods. Experimental results provide valuable insight into the effectiveness of GA in the context of the PS Problem under various configurations, notably in the context of Lamarckianism and the Baldwin Effect. Ultimately, this research enhances the understanding of GA application for the PS problem, offering notable insights regarding GA performance and potential for future work.**

I. Introduction

A *PLANNING* problem involves creating a series of strategic steps to achieve a specific goal. It often encapsulates order and conditions under which various actions must be performed while simultaneously factoring uncertainty, resources, and goals of a given problem instance. *Scheduling* is a decision-making process that manages the allocation of resources over time. Its primary objective is to optimize specific performance metrics, typically total completion time, often called *makespan*. Therefore, planning and scheduling problems could be described as determining the optimal sequence of actions to achieve specific goals in the most time and resource-efficient manner. Despite a long history of research into these problems, the need for solutions persists due to the consistent emergence of new challenges and their increasing complexity. Planning and scheduling apply to diverse real-world applications, including manufacturing, pathfinding, logistics, management, space exploration, telecommunications, economics and games (i.e. economic or computer ones).

This article aims to solve the planning and scheduling problem in the context of the economic development aspect in the StarCraft II (SC2) strategy game, in this area known as *Build Order Planing*.

In the following sections, we will provide the state-of-the-art analysis (Sec.II), define the exact problem (Sec.III), and present the proposed approach (Sec.III) and experimental results (Sec.IV) that describe the application of Genetic Algorithm (GA) to *Build Order Planing*. The last sections of the paper consist of conclusions (Sec.V) and future works (Sec.VI).

II. Related Work

A. Build Order-Planning

As described in the [5], Build Order Planning or Build Order (BO) optimization is a class of Automated Planning problems that arise in a video game genre called real-time-strategy (RTS). It is a process of finding the sequence (order) of actions to be made by the player to achieve a specific goal in the shortest makespan. This goal depends on the exact scenario and larger strategy, but it can be associated with creating a certain number of military units or gathering the precise amount of resources. Individual RTS games differ, but the basics are mostly similar. The player performs a series of actions like collecting resources, developing a base, and training military units, all in order to gain an advantage over the opponent and finally defeat him. Mentioned actions can be divided into two groups: strategical level actions (*macro*) and tactical level actions (*micro*). In this paper, we discuss a sequence of the macro actions only - called Build Order.

In the context of the SC2 game, simple build-order solutions, such as a quick attack with a small number of military units, may be countered easily by building defensive structures whose overall cost is lower compared to the cost of the attack. Therefore, engagement benefits the defending side. On the other hand, a defensive stance is disadvantageous against economic development, as it will overtake the defending player in the long term. Finally, development focus can be easily countered by an aggressive strategy (known as 'early rush'), which closes the cycle. Based on this stone-paper-scissors-like dynamic, both players can change their focus during a game to gain an advantage against the opponent. Selecting goals and switching them in-game time is a separate subject that can be described by a high-level strategy Artificial Intelligence (AI) system as described in [1]. We can imagine it as a sort of manager that dictates the overall strategy goal, however, the problem of determining the plan for achieving this goal in the most efficient manner is the subject of the presented research.

RTS games are interesting application domains for AI researchers because of the huge state spaces and concurrent actions. The most common instance of BO application in games is StarCraft (published by Blizzard Entertainment), a popular RTS game with over 10 million copies sold. StarCraft has received over 50 industry awards, including over 20 "Game of the Year" awards. It is considered the so-called e-sports game, as each player has the same chances at the beginning, does not contain random elements, and the result depends solely on the player's skill. As StarCraft (and its sequel - StartCraft II) is the most commonly used case, multiple other RTS games are known for the importance of early game build order. Such games as Age of Empires or Company of Heroes are worth mentioning. For the all mentioned real-time games, an important factor is the speed of action execution and fast reaction time. However, this is not an essential aspect of defining the BO problem. Turn-based games also make a good example.

BO-solving methods can be used to support players or design demanding opponents for games. It can also help with game balancing and exploit detection. For instance, when there are several civilizations (or races) to choose from, they should compete on a similar level. The existence of a specific action sequence that makes a civilization significantly stronger than the others is considered a product defect. The ability to simulate and detect it at an early stage of development is a significant improvement compared to the process of lengthy manual tests common in the game-development industry.

Authors of the [6] suggest that historically studied economic games such as Prisoner's Dilemma (where tit-for-tat derives) or Cournot Production games are far more streamlined in comparison to modern RTS games. In a broader look at this application, simulations of systems implemented in games refer to economic mechanisms known from the real world. An exemplary generalization can be interpreted as the optimization of the company's development strategy or even the investment strategy, where subsequent decisions depend on the results of previously taken steps. To support this assumption, Cobb-Douglas (CD) model presented in [14], has been applied by Webel et al. in [6] to the modern RTS game StarCraft: Brood War, where worker units gather resources, build infrastructure, and eventually lead to the construction of combat units. The winner of the game is the last one with a standing structure.

As suggested in [1], BO is an instance of a temporal planning problem. Temporal planning recognizes that actions take a certain amount of time to execute and acknowledges that multiple actions can occur concurrently under specific circumstances. For example, certain actions may have specific temporal requirements, such as waiting for a resource to become available or ensuring that one action finishes before another can start. By treating BO as a temporal planning problem, the planning process becomes more realistic and aligned with real-world scenarios than in the classical planning approach, which is streamlined, instantaneous, and does not consider actions interacting with each other.

To further emphasize treating planning in RTS games as a formal problem, Buro et al. in [1] present a study on the optimization of build order strategies in real-time strategy (RTS) games, highlighting the potential of using the Planning Domain Definition Language (PDDL) for modeling these problems. PDDL presented in [8] is a language utilized in automated planning competitions to standardize the description of planning domains and problems, enabling diverse planners to compete against each other. The authors of [1] also discuss the issue of concurrent execution and propose efficient mechanisms for ordering actions within the build order domain of RTS games. However, it is worth noting that even the most recent version of PDDL does not support object creation or deletion, which is essential in RTS games where object creation plays a vital role. While it is possible to simulate them implicitly within the language, it signifies that modern strategy games present a more challenging environment than classical planning problems.

Wei et al. in [3] were the first to address the BO problem as Planning and Scheduling problem with Producer/Consumer constraints. This attempt to mathematically model and approximate the BO problem is further supported by Blackford et al. article [5] about build order optimization in a multi-objective approach.

B. Methods

Several approaches to solving planning and scheduling problems exist in the literature, such as Ant Colony Optimization [13], Graph Search [5, 2], Stochastic Search [11], Evolutionary Algorithm [4, 2] and Machine Learning [9, 3].

Churchill et al. in [5] introduced a depth-first branch-and-bound algorithm (BnB) to address initial build orders in StarCraft. The authors implemented a tree search, with the root representing the initial game state, and conducted a depth-first exploration to identify an optimal solution. This solution aims to meet the goal within the shortest possible makespan. While it is possible to validate found build orders in the game environment, this approach is time- and resource-intensive. To address this challenge, the authors suggested developing a StarCraft economic simulator to measure the value of build orders effectively.

El-Nabarawy et al. in [11] implemented a Monte Carlo Tree Search (MCTS) based on a StarCraft mini-game called BuildMarines the goal is to find build orders that can provide players with a larger amount of Marines military units in a fixed amount of time. Both methods have significantly lower action space, and although they may be effectively applied to smaller problems, they differ from the efficiency and speed of evolutionary algorithms and machine learning techniques. The comparison of these methods has been described in the experiments section to support this claim further.

In the context of Evolutionary Algorithms (EA), Justesen et al. in [2] has successfully implemented the evolutionary-based method of Continual Online Evolutionary Planning (COEP) for build order optimization. The goal was to create a game agent that can, in-game time, find build orders that counter opponents' movements, also utilizing the BO simulator for evaluation. The fitness function is based on a heuristic that can describe how desirable founded BO is. Heuristic values short-term rewards higher than long-term rewards, which are very important in game planning agents because long-term build orders can provide the agents with optimal strategy and powerful armies, which are easier to counter by opponents.

Blackford et al. implement a multi-objective evolutionary algorithm (MOEA) detailed in their work [4] for the Total Annihilation RTS game. The multi-objective approach is advantageous because solutions evolve to satisfy different goals, and the algorithm does not find just one solution but a set of different solutions, each potentially providing different strategic advantages.

Another significant implementation of a machine learning-based game agent was achieved in the form of AlphaStar [9] for StarCraft II. This sophisticated AI has the capability to generate valid build orders dynamically during gameplay.

Liu et al. in their work [10] used a reinforcement learning technique to create a low-cost StarCraft II agent and suggested that on a larger scale, this technique could create a better agent using fewer resources. Although machine learning-based methods show immense potential in solving BO problems, they require considerable resources and time to develop a functioning model and, once created based on a problem, cannot be easily applied to another similar problem. Conversely, Genetic Algorithms (GA) only necessitate game data and a build order evaluation method to produce valid build orders. It makes GA a more universally applicable solution for these types of problems. GA can also be deployed within build orders for game balancing, which is a complex problem.

For the above reasons, we implement the Genetic Algorithm in our research. As for reference methods, we propose those mentioned earlier, specifically the ones proposed by Justesen et al. as cited in [2], the COEP method configuration (COEPc), and the method proposed by Blackford et al. in [4], MOEA method configuration (MOEAc). In addition, we will present an application of the Genetic Algorithm in comparison to approaches based on Branch and bound (BnB) and Monte Carlo Tree Search (MCTS) methods, as cited in [5, 11].

III. Problem Definition and proposed Method

A. Planning and Scheduling Problem Definition

We generalize the SC2 BO problem using Planning and Scheduling (PS) problem with Producer/Consumer constraints based on [4] work.

The planning aspect of the problem involves determining which actions need to be performed to achieve a certain goal G. The scheduling aspect, on the other hand, describes the timing of action execution with consideration to available resources R and minimizing the overall completion time of all actions $a \in A$, called makespan. Solution $s \in SP$ where SP is a solution space could be described by a vector of actions a as follows:

$$s = [a_0, a_1, ..., a_{n-1}] \tag{1}$$

where n is the solution size. The solution s changes the initial game state S^a into desired state of S^x (see Eq.2).

$$S^a \xrightarrow{a_0} S^b \xrightarrow{a_1} ... \xrightarrow{a_{n-1}} S^x \tag{2}$$

Every considered action a_x has a set of certain prerequisite conditions of execution. These prerequisites are described by resources $r \in R$. Wei et al. in [3] for the PS problem, distinguish two categories: Consumable, Renewable; and four types of resources: Consume, Produce, Borrow, and Require.

- **Consumable**
  - Consume - action can consume a specific amount of resources, for example, the cost of creating a wall (structure) in SC2 is: 100 gold, and 50 wood, so the action of making a wall would consume 100 gold and 50 wood.
  - Produce - action usually after completion provides a game environment with some kind of good (produces it). This good can be a soldier, building, research, or any other environmental element that changes the game state, but generally, we can define it as some kind of resource.
- **Renewable**
  - Borrow - action can borrow certain resources during execution time, for example in SC2 game, the action of creating a Zealot army unit needs to use Gateway building, Gateway can produce only one unit at a time, so it is borrowed by action to create Zealot action.
  - Require - a type of resource that multiple actions at the same time might require. For example, action mine coal borrows one miner and requires the mine to execute. However, in mine multiple miners can work so the resource is shared and now borrowed.

Note that given resources can belong to two or more types. For example, the SC2 Gateway building can be borrowed and produced. The list of resources associated with an action can be defined as a_R. For instance, in the action of creating a military unit a, a_R consists of consuming 100 gold and borrowing barracks for the production of the unit as follows: $a_R = \{100\ gold, barracks, null, military\, unit\}$

B. Constraints

To consider the build order a *feasible* solution, it is imperative that the set of constraints C is satisfied, which also limits SP.

Cumulative Producer/Consumer constraint refers to Consumable resources. It requires that when two actions are planned to occur simultaneously, the combined resources they need should not exceed the available quantity of those resources. For example, creating two Zealots (units) at the same time requires 200 crystals, 2 Gateways, and 4 supply consumption. Constraint can be expressed as:

$$\forall_{a \in s}\ a_R^t \leq T^t \tag{3}$$

where s is solution, a_R^t are resources required by the a action in time t and T is the total amount of resources available in time t.

Disjunctive constraint refers to borrowed resources, it states that for a pair of actions competing for execution at the same time. The constraint is satisfied when these actions do not overlap, and thereby the integrity of the system process is maintained. Constraint can be expressed as:

$$\forall_{(i,j)\in s^2, i\neq j}\ (i_{end} \leq j_{start}) \vee (j_{end} \leq i_{start}) \tag{4}$$

where pair $(i,j) \in s^2$ is every pair of two different actions that can be extracted from s. As a_{start} and a_{end}, we describe the start and end time of executing action a, respectively.

Exist constraint refers to Required resources. It states that certain resources must exist in order for action to execute. An example of an existing constraint in the context of SC2 could be Stalker army unit which we can not build until we build Cybernetics Core building. Constraint can be expressed as:

$$\forall_{a\in s}\forall_{r\in \text{exist}(a_R)}\ a_{start} \leq t \leq a_{end} \wedge r \in T^t \tag{5}$$

where $exist(a_R)$ is a subset of action a resource set a_R that contains only resources which must exist in order to execute action a. Resource r is not consumed.

According to the above definition, the problem could be defined as:
Given: $< G, R, C, A, S^a >$,
interpreted as follows. To find: Solution s that fulfills goal G in the shortest possible makespan, starting from state S^a, using resources R, considering only actions present in A, satisfying all constraints C.

C. SC2 BO problem as PS problem

SC2 is a strategic game that allows players to embody one of three distinct races, each differing in units, structures, and strategies. For our research purposes, we have selected the Protoss race, an advanced, alien-like race within the game. We denote the initial state S^a as a typical SC2 start state for the Protoss race, which includes the following resources: 50 crystals, 1 Nexus, 12 Probes, and a supply capacity of 15. Probes function as worker units that accumulate crystals at the Nexus base. These Probes have the capability to construct buildings, such as the Pylon, which not only increases the supply capacity but also permits the construction of more specialized buildings within its energy field. One such building is the Gateway, which facilitates the creation of basic military units, like the Zealot. In this context, we categorize actions like creating a Probe, constructing a Pylon or Gateway, or building a Zealot as macro actions, which is typical for SC2 BO. While we take into account all possible macro actions of Protoss race in our research, we will not delve into each one individually due to their sheer number.
To present SC2 BO as the PS Problem, the set of metrics $m \in M$ should be defined as follows:

- m^0 - total completion time,
- m^1 - total build cost of workers,
- m^2 - total build cost of the army,
- m^3 - total research cost,
- m^4 - total cost of defensive buildings.

The goal G within the SC2 BO context is represented as an *objective vector* $O = [m^1, m^2, m^3, m^4]$ comprising non-negative integer value. Assigning values on the vector positions defines exact scenarios. Each value within this vector can range from 0 to the maximum possible value of each metric m. However, m^0 is the total completion time of all actions in solution s extracted after BO simulation, which will be discussed later.

Based on each metric defined in vector O, we can divide the action space A into four distinct subsets, each containing actions that contribute solely to one specific objective metric value. This arrangement allows the given method to utilize every action in the game during the optimization process.

To address the scheduling aspect of the issue, we adopt a model based on the rule that each action is executed as swiftly as possible, under the condition of all prerequisites are met. For instance, if the task involves constructing two Zealots, this task will be executed concurrently if a sufficient amount of resources is available. It is the responsibility of the method to order actions in such a way that the arrangement is optimal in the context of the addressed rule. This model is adopted by a SC2 game simulator that conducts BO simulations. To acknowledge the need for the delay before executing certain actions, we propose *None* action, which will force a delay for the next action execution by a fixed amount of seconds.

The given definition allows the construction of solution $s \in SP$ (alternatively referred to as BO), where SP is the solution space, described by an ordered list of integers, each of which maps to a specific BO action in SC2 (an example presented in Figure 1).

In this paper, we engage with the complete action space of the Protoss race, which comprises 59 actions plus one additional *None* action. Each race in the game encapsulates actions space A of around 60 actions. In the case of the Protoss race, which we choose as the base for our research, the initial game state permits the execution of four possible actions. Notably, certain actions can expand the list of potential actions, making the explicit determination of the solution space a considerable challenge. Nevertheless, it's possible to estimate the entire set of potential genotypes by utilizing the length of the build order. This concept can be formally encapsulated in the following equation:

$$x = a^n \tag{6}$$

Here, n represents the size of the build order, a stands for the total size of the action space, and x signifies the complete number of genotypes – a similar representation is presented in [12]. For a comprehensive problem definition, particularly within the context of a BO problem, we need to define the size of the build order solution n. We will address size n in Sec.IV-B.

Finally, we define the evaluation function E, which value is to be minimized by the optimization method, as a function of end state S^x and objective vector O provided by simulator

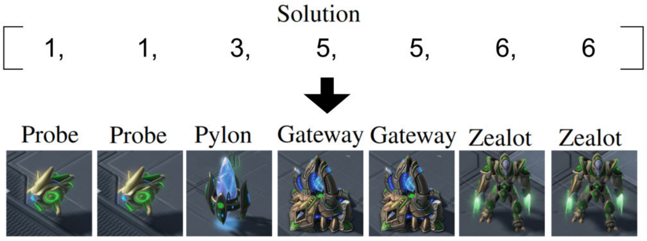


Fig. 1. The example solution – genotype (i.e. solution) presented as list of integer is mapped to list of SC2 actions

from all actions a in a given solution s. If the solution does not fulfill the objective described by the objective vector, we add a penalty for every metric m objective that is not fulfilled. E can be presented as the formula:

$$E(S^x, O) = S^{x^{m^0}} + \sum_{m \in M, m \neq m^0} max(0, O^m - S^{x^m}) \quad (7)$$

where $m \in M$ is metric, O is objective vector and S^{x^m} is end game state metric m acquired by simulation. The goal value of metric m, which is part of objective vector O can be formulated as O^m. As $S^{x^{m^0}}$ we understand the game end state time metric m^0 value. The $max(a, b)$ function represents the maximum of a and b. For example, for a given objective vector:
$O = \{m^1 = 0, m^2 = 200, m^3 = 0, m^4 = 0\}$
and end state S^x:
$\{m^0 = 120, m^1 = 800, m^2 = 100, m^3 = 100, m^4 = 0\}$
is evaluated as follows:

$$\begin{aligned} E(S^x, O) &= 120 + max(0, 0 - 800) + max(0, 200 - 100) \\ &\quad + max(0, 0 - 100) + max(0, 0 - 0) \\ &= 120 + 0 + 100 + 0 + 0 \\ &= 220 \end{aligned}$$

The mechanism for extracting metric values from S^x is built upon the simulation detailed in the following section.

D. Build Order Simulator

The BO simulator is a program inspired by previous state-of-the-art implementations. It takes input parameters such as the input game state S^a, action space A, build order s, and provides the Genetic Algorithm (GA) with the output game state S^x. The output is described by solutions s in-game execution time and end state metrics S^{x^M} values. The simulator focuses only on the BO aspect of the SC2 game and does not simulate enemy players.

The program simulates every second of the game in a loop until every action in the build order is completed or timeout is reached. During the simulation, we execute actions, often concurrently. Once an action is executed, we add the provided resource to the game state. We extract metrics from the provided game state S^n for the E function when the simulation ends. The simulator loops through BO and tries to execute the current action in each time frame. If it cannot, it just skips to another time frame, so during the evolution process, we have to ensure that every build order in the population is valid.

The simulation ends with a timeout if the build order is invalid, meaning it is impossible to execute based on game rules. However, in the proposed implementation, timeout never occurs, and the simulator validates correct solutions.

Build order completion is not a deterministic process, but it can be approximated by one. For instance, the time to build a Pylon can vary based on the place we want to build it. Nevertheless, this time is often similar, so it could be approximated by a fixed amount of time. For every action that needs to significantly change the location of execution, such as creating a new Nexus base, additional time is needed for travel.

E. Genetic Algorithm for Planning and Scheduling (GAPS)

We attempt to solve Planning and Scheduling (PS) problem on the example of the StarCraft II Build Order Optimization (SC2 BO) problem using a Genetic Algorithm (see **Algorithm 1**). GA is advantageous for exploring expansive solution spaces and strategy games encapsulating complex, deterministic environments, providing an ideal ground for conducting research in this field. Therefore we name our method Genetic Algorithm for Planning and Scheduling (GAPS). We propose a straightforward solution generation method that creates valid length solutions n based on an action tree. In this tree, state S^a is the root. From there, one action is selected randomly from the list of possible options, which transitions the system to the next state. This process is repeated, updating the list of possible actions based on the actual state until the size of the created solution equals n. After initialization, each created solution is evaluated and sorted based on fitness. To evaluate build order, we employ function E as previously described. We search for optimal configuration of the rest of the operators like selection, crossover, mutation, and repair in experiments (Sec.IV).

In the context of GA, Lamarckianism gives that an individual can modify their genotype in response to their environment and subsequently pass this change on to their offspring. In contrast, the Baldwin Effect posits that individuals can make non-genetic (phenotypic) changes over their lifetime in response to their environment. Those that successfully adapt in this way are more likely to survive and reproduce. Over time, genetic changes supporting these phenotypic adaptations could become more prevalent in the population, leading to the genetic assimilation of learned traits. Applying Lamarckianism to a fixed-length Build Order (BO) is challenging, requiring solutions to be repaired without altering its length. We provide the pseudocode of our base repair function, which eliminates actions that cannot be executed (see **Algorithm 2**). We then incorporate the Lamarckian approach by filling in missing actions with a 'None' action and overriding the genotype.

We implement the Baldwin Effect by repairing and evaluating a copy of the genotype, leaving the original genotype unchanged in the population.

Algorithm 1 GAPS

```
Require: $gameState$, $actionSpace$, $solution$, $populationSize$, $generations$, $solutionSize$, $crossoverRate$, $mutationProb$
Initialize $population \leftarrow generatePopulation(...)$
for $i$ = 0 to $generations - 1$ do
    $sort(population)$
    $best = population[0]$
    Initialize $children$ as an empty list
    $children.append(best)$
    for $i$ = 0 to $populationSize - 1$ do
        $parent1$, $parent2$
        $selection(population, parent1, parent2)$;
        if $underProbabilityThreshold(crossoverRate)$ then
            $child \leftarrow crossover(parent1, parent2)$
        end if
        if $underProbabilityThreshold(mutationProb)$ then
            $child \leftarrow mutate(child)$
        end if
        $repairedSolution \leftarrow repair(child)$;
        $fitness \leftarrow evaluate(repairedSolution)$;
        $children.append(child)$
    end for
    $populaton \leftarrow children$
end for
return $population[0]$
```

Algorithm 2 Repair Algorithm

```
Require: $gameState$, $actionSpace$, $solution$
Initialize $resultBuildOrder$ as an empty list
for $i$ = 0 to $solution.buildOrderSize - 1$ do
    $action \leftarrow solution.buildOrder[i]$
    if $isActionPossible(gameState, actionSpace, action)$ then
        $resultBuildOrder.append(action)$
    end if
end for
return $resultBuildOrder$
```

IV. Experiments

In this section, all research experiments are presented to answer four research questions:

- **RQ1** - How do different genetic operators (crossover, mutation) influence the evolution process? ($a30case$)
- **RQ2** - How does GA configuration (budget, selection, crossover rate, mutation probability) influence the evolution process? ($allcases$)
- **RQ3** - How effective is GA using Lamarckianism in comparison to Baldwin Effect? ($allcases$)
- **RQ4** - How effective selected GA configuration based on previous experiments is in comparison to some state-of-the-art setups in the context of all test cases? ($allcases$)

The following sections present details and results of developed experiments to answer the above research questions.

A. Test cases

We prepared three objective scenarios for enriching experiments: $aggressive$, $balanced$, and $development$. Each scenario is divided into three problem sizes: 30, 60, and 150, where problem size describes how long action sequences we want to examine. For 9 cases, we propose objective vectors $O = [m^1, m^2, m^3, m^4]$. In the $aggressive$ scenario, the goal is to gain as much army value as soon as possible, the development scenario aims to create a strong economy, and the balanced scenario is a mix of economy, army, research, and defense; we propose values of each O based on game experience. This approach allows us to examine the algorithm's behavior in response to varying types of problems in the context of different levels of complexity.

Based on our knowledge of the game and previous manual experiments, we propose setting budgets for all examined methods corresponding to three problem sizes. Additionally, we use a short code to name each case; for instance, an aggressive problem of size 30 is labeled $a30$. The case $a30$ will serve as the base case. For the $a30$ case, we define $O = [0, 2000, 0, 0]$, meaning that, given a maximum of 30 actions, a selected method needs to identify a build order that yields an army value of 2000 in the shortest possible makespan (see Tab.I on p.7).

For testing procedures in tuning/general experiments, as default *a30* scenario is used. For methods comparison, all test cases are used.

B. Setup

All experiments were done using the computer with AMD EPYC 7H12 64-Core Processor, Ubuntu operating system, and C++20. Because of the non-determinism nature of GA, for every configuration, we repeat the experiment 10 or 30 times, depending on the experiment.

For each case scenario, we propose to examine three different solution sizes described by $n = sum(O) * 1.5\%$. For example, in the case of $a30$ we examine population sizes 60, 90, and 150. We can define the number of generations as $number\, of\, generations$ = $budget/population\, size$. Therefore we have specific problem sizes for each scenario with defined budgets. For each problem size, we have three population size/generations proportions constrained by budget. For example, in the aggressive scenario for problem size 30, we have three population size/generations proportions: 60/750, 90/500, and 150/300.

We conduct crossover/mutation experiments incorporating three standard crossovers: one-point crossover, two-point crossover, and uniform crossover. We also examine three different mutations: random, bit flip, and mixed mutation presented in COEPc. For our configuration experiment, we validate three distinct population sizes/generations proportions within a specified budget across all nine scenarios, using

TABLE I
TEST CASES – SCENARIOS

Genotype size	**Budget**	**Aggressive**	**Balanced**	**Development**
30	45000	$a30\ O = [0, 2000, 0, 0]$	$b30\ O = [1100, 1000, 250, 150]$	$d30\ O = [1500, 200, 0, 0]$
60	150000	$a60\ O = [0, 4000, 0, 0]$	$b60\ O = [2000, 1000, 400, 300]$	$d60\ O = [2200, 400, 0, 0]$
150	750000	$a150\ O = [0, 10000, 0, 0]$	$b150\ O = [3300, 1000, 250, 150]$	$d150\ O = [5500, 0, 0, 0]$

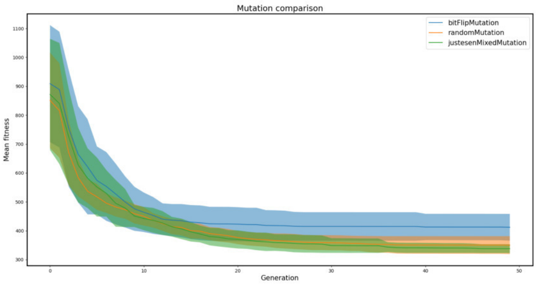


Fig. 2. Mutation comparison for $a30$

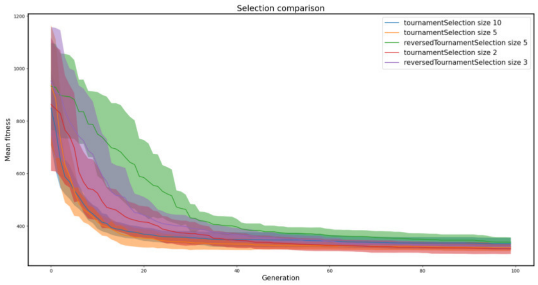


Fig. 3. Selection results for $a30$

TABLE II
MUTATION COMPARISON $a30$

Mutation type	mean	std_dev
Bit Flip	409.9	46.8
Mixed	314.3	19.2
Random	329.2	20.7

TABLE III
SELECTION COMPARISON TYPE $a30$ REPEATS

Tournament size	mean	std_dev
reversed 5	315.1	14.51
reversed 3	298.4	14.31
2	305.5	17.69
5	309.0	17.25
10	315.1	14.51

various tournament selection sizes, crossover rates, and mutation probabilities. It is important to note that mutation probability refers to the chance of applying a mutation that alters precisely one gene in the genotype, an approach inspired by state-of-the-art implementations. In addition to the standard configuration experiment, we investigate a further variant: a reverse tournament of size n that selects a parent randomly based on the $n-1$ best candidates and thus considerably lowers selection pressure.

Once we establish the GAPS configuration (GAPSc), we explore it in the context of Lamarckianism and the Baldwin Effect.

Finally, we compare GAPSc with BnB and MCTS as well as COEPc and MOEAc for a state-of-the-art comparison. State-of-the-art GA-based methods configurations:

- COEPc - random selection (best 25%), two-point crossover (crossover rate 100%), mixed mutation (probability 50%), Baldwin Effect-based repair,
- MOEAc - tournament selection (size 2), one-point crossover (crossover rate 90%), bit flip mutation (probability 100%), Baldwin Effect-based repair.

C. How does crossover/mutation influence GAPS effectiveness? – **RQ1**

The experiments with genetic operators (i.e. crossover and mutation) showed that although chosen crossover does not have much impact on the evolution, mutation can be quite vital (see Tab.II and Fig.2). The best mutation proved to be presented in COEP mixed mutation, which encapsulated four different mutations where each can be applied to created offspring. It helps enrich population exploitation of solution space, improving evolution quality.

D. How to configure the GAPS? – **RQ2**

In the initial stage of experiments, the configuration experiment proved the insignificance of validated population sizes.

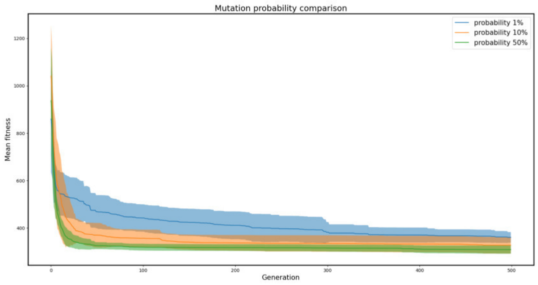


Fig. 4. Mutation probability comparison ($a30$)

TABLE IV
MUTATION PROBABILITY COMPARISON FOR $a30$, BEST FITNESS FOR 10 REPEATS

Mutation probability	mean	std_dev
1%	360.5	22.3
10%	327.5	35.1
50%	309.0	17.2

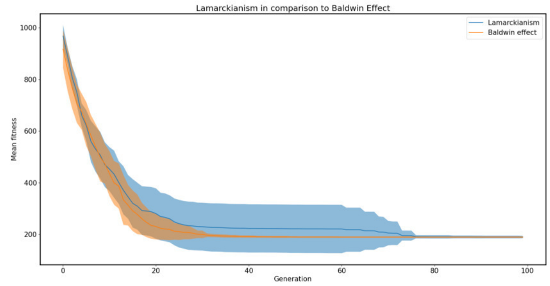


Fig. 5. Lamarckianism in comparison to Baldwin Effect for $d60$

TABLE VI
COMPARISON RESULTS OF BNB, MCTS AND GAPS FOR $a30$, BEST FITNESSES FROM 30 REPEATS

Method	mean	std_dev
GAPS	318.7	32.8
BnB	1933	*0.0*
MCTS	1827.3	43.1

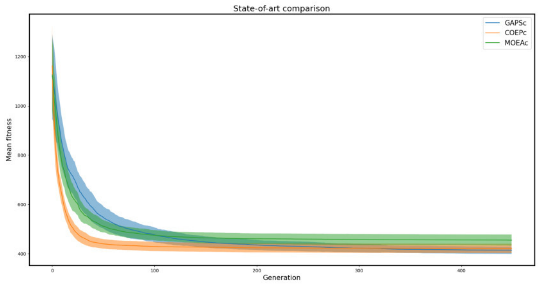


Fig. 6. Comparison results for GA for $b150$

It could be suggested that we search too few configurations to show a significant difference. However, this gives us valuable information about the evolution process.
The selection/crossover rate/mutation probability experiment provided us with the best configuration for the next experiments, which proved to be reversed tournament selection with size 3, crossover rate = 40%, and mutation probability = 50% – see Tab.III with Fig.3. and Tab.IV with Fig.4. Based on experimental results, we can define the best-found configuration of GAPS (GAPSc): random correct solutions generation, tournament selection (size 5), one-point crossover (crossover rate 30%), mixed mutation (probability 50%), and Baldwin Effect-based repair.

E. Lamarckianism in comparison to Baldwin Effect – **RQ3**

Results of GAPS in the context of Lamarckianism and the Baldwin Effect varies based on the scenario. At times, one outperforms the other. However, generally speaking, the Baldwin Effect tends to be the more stable choice (see Tab.V and Fig. 5).

F. State-of-the-art setups comparison – **RQ4**

Experimental results showed that BnB and MCTS methods could not find any solutions for any scenarios under a given budget because they cannot search for solutions of defined sizes (ex. 30). Therefore to extract some solutions, we compare them by giving the task of finding a solution with the most military units in a given budget and then comparing these solutions to the solution found by GA in $a30$ case (see Tab.VI). The experiment shows that under a budget of 45000, associated with a problem of size 30, BnB can find an optimal solution for acquiring 200 army value. MCTS, on the other hand, can find a suboptimal solution that can provide 300 army value. In comparison, for the same budget, GAPS is able to find a solution that achieves the given goal of 2000 army value on average in 318.7 in-game seconds.

TABLE V
LAMARCKIANISM IN COMPARISON TO BALDWIN EFFECT FOR $d60$, BEST FITNESSES FROM 30 REPEATS

Repair type	mean	std_dev
Lamarckianism	187.8	2.7
Baldwin Effect	188.9	1.9

The results from the GA-based experiments reveal interesting insights (see Table VII and Figure 6). These suggest that our GAPS configuration (GAPSc) often discovers superior solutions in smaller, $aggressive$ cases. Conversely, COEPc tends to generate better solutions in larger, more $balanced$ cases. The outcomes for $development$ cases are quite similar.

The explanation for this might be rooted in the specifications of the distinct scenario. $Aggressive$ cases typically require highly optimized solutions with a specific order of action execution. In contrast, $development$ cases permit a wider range of solutions, all yielding similar results.

G. Comparison to known build order

We conduct a final experiment in which we take a well-rated Stalker rush build order from a website that stores the best Sc2 build orders. We simulate this order, extract metrics, and then input those metrics as an objective vector for our implementation to check if GAPS can find a similar build order. The experiment reveals that while the website build order takes 221 in-game seconds, our method only requires 205 seconds to produce a build order described by the same objective vector. Although using the same actions, the solution provided by GAPS completely reorders them in a manner

TABLE VII
COMPARISON RESULTS FOR GA FOR ALL SCENARIOS

	GAPSc		**COEPc [2]**		**MOEAc [4]**	
Case	mean	std_dev	mean	std_dev	mean	std_dev
a30	**302.17**	11.34	306.83	20.27	356.63	37.29
b30	258.2	10.15	**251.26**	10.78	320.033	32.22
d30	**187.23**	2.06	**187.83**	2.29	207.03	39.16
a60	**409.3**	16.21	420.23	19.55	452.467	22.57
b60	318.5	14.5	**309.36**	7.99	373.36	27.17
d60	**246.57**	3.55	**245.2**	4.99	262.2	10.71
a150	731.0	9.1	**707.7**	20.25	785.9	20.96
b150	561.27	18.62	**534.56**	15.37	644.3	39.93
d150	**1422.13**	1.2	**1422.87**	1.83	**1422.83**	1.84

that allows a significant number of actions to be executed concurrently.

V. CONCLUSION

This study offers several key conclusions regarding applying and optimizing GAPS for planning and scheduling problems in the context of the game StarCraft II.

Mutation operators significantly impact the evolutionary process. The mixed mutation strategy described in [2] provided the best results, enhancing the population's ability to exploit the solution space and thereby improving the quality of evolution.

Search for optimal GAPS configuration, including population size, number of generations, selection, crossover rate, and mutation probability, reveals that population size and number of generations did not significantly impact the experiment results. The best-founded configuration proved to be: reversed tournament selection of size 3, one-point crossover with crossover probability of 40% as well as mixed mutation [2] with mutation probability of 50%.

The results were scenario-dependent when comparing Lamarckianism and the Baldwin Effect in GA. The Baldwin Effect emerged as a more stable choice across different problem sizes and scenarios. This stability is primarily due to the Baldwin Effect's ability to balance exploration and exploitation in the solution space, reducing the risk of premature convergence to suboptimal solutions.

In contrast to other state-of-the-art configurations, GAPSc demonstrated proficiency in smaller, aggressive cases where precision and optimization were vital. Conversely, COEPc prevailed in larger, balanced cases where flexibility and a range of solutions were advantageous. Therefore, careful consideration of case characteristics is crucial when deciding the most appropriate method to apply. In doing so, we can utilize the strengths of each method, ensuring optimal outcomes.

The final experiment underscored the potential of GA in optimizing game strategies. In comparison to a highly-rated Stalker rush build order from a renowned strategy website, the GA generated a similar build order in fewer in-game seconds, showcasing its potential to create efficient and competitive game strategies. The time difference (16sec.) might cause a significant difference at the very beginning of the high-ranked match.

In conclusion, the findings of this study underscore the effectiveness and potential of Genetic Algorithms in tackling complex planning and scheduling problems, similar to StarCraft II Build Order Optimization. Furthermore, they provide a foundation for future exploration and optimization in this domain.

VI. FUTURE WORKS

Several promising future research directions exist for enhancing the Genetic Algorithm applied to BO.

Firstly, the development of more sophisticated objective metrics could be beneficial. Such metrics would encapsulate the tactical nuances of each unit in the game, providing a richer representation of the problem domain within the evolutionary process. This enhancement could lead to solutions that better reflect the complex dynamics of the game.

A natural extension of this is the application of a multi-objective Genetic Algorithm. It would enable the simultaneous optimization of several objective functions, providing a more comprehensive optimization that considers the nature of planning and scheduling problems.

Using surrogate models, also known as meta-models, could help accelerate the evaluation process within the GA. These models, built based on existing evaluation data, can provide fast, approximate evaluations, significantly speeding up the evolutionary process.

Exploring different approaches could also increase the effectiveness. One promising method that aligns well with the nature of Genetic Algorithms in the context of presented results is Extreme Optimization (EO). This technique focuses on the most complex components of a solution and attempts to improve them. Thus, it could be particularly effective in complex environments like those encountered in StarCraft II, where optimizing numerous suboptimal solutions is necessary. Combining GA with EO could lead to a more efficient search process. While GA excels at exploring a broad solution space, adding EO allows it to focus more specifically on the areas that need the most improvement, leading to more exploitation of the solutions space and enhancing effectiveness.

Integrating GA with a high-level agent could validate its performance and potentially create a powerful StarCraft II bot. This agent could use the GA-generated solutions as part of its decision-making process, while the feedback from the agent's performance could further inform and refine the evolutionary process.

GAPS can be extended to numerous domains, such as economics, and military simulations, due to its inherent flexibility and adaptability. In the economic field, GAPS can be applied to optimize portfolio management, improve supply chain efficiency, and refine resource allocation plans in large-scale projects, effectively scheduling investments over time to maximize returns while minimizing risk. Similarly, in military simulations, GAPS can optimize strategies for defense and offense, logistics, troop deployment, and disaster response by assessing multiple scenarios accordingly.

The potential directions for enhancing the presented GA are vast and multidimensional, ranging from more nuanced objective functions to hybrid techniques and high-level agent integration. These could all contribute to a more robust, efficient, and capable GA for tackling the complex problem of game planning and scheduling.

Finally, more advanced simulations or games can be examined, to verify how the increased number of resource management options affects the results. It could include a higher number of resources, a resources trade/exchange system with a simple demand/supply mechanism, and soft constraints based on the simulation-specific rules. Those extensions can bring the environment significantly closer to real-world scenarios. Another interesting, yet challenging aspect to simulate is risk management. In both game and economic environments multiple players compete with each other, thus long-term and error-prone plans can be replaced by less efficient but safer strategies. It is related to the previously mentioned multi-objective optimization as the risk exposure might be one of the minimized objectives.

References

[1] Kovarsky, A., and Buro, M. (2006) "A first look at build-order optimization in real-time strategy games." Proceedings of the GameOn Conference. 2006.

[2] Justesen, Niels and Risi, Sebastian. (2017). Continual online evolutionary planning for in-game build order adaptation in StarCraft. 187-194. 10.1145/3071178.3071210.

[3] Wei, LZ and LW Sun. (2009) “Build Order Optimisation For Real-time Strategy Game”, http://www.nus.edu.sg/nurop/2009/SoC/nurop-LimZhanWei.pdf

[4] Blackford, J., and Lamont, G. (2014) "The real-time strategy game multi-objective build order problem." Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Vol. 10. No. 1. 2014.

[5] Churchill, D., and Buro M. (2011) "Build order optimization in starcraft." Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Vol. 7. No. 1.

[6] Weber, Bryan S. (2018) "Standard economic models in nonstandard settings–starcraft: Brood war." 2018 IEEE Conference on Computational Intelligence and Games (CIG). IEEE,.

[7] Buro, M., and Kovarsky, A. (2007) "Concurrent action execution with shared fluents.", AAAI Conf. 2007: 950-955.

[8] Fox, M., and Derek Long. "PDDL2. 1: An extension to PDDL for expressing temporal planning domains." Journal of artificial intelligence research 20 (2003): 61-124.

[9] Vinyals, O., Babuschkin, I., Czarnecki, W.M. et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575, 350–354 (2019). https://doi.org/10.1038/s41586-019-1724-z

[10] Liu, Ruo-Ze and Pang, Zhen-Jia and Meng, Zhou-Yu and Wang, Wenhai and Yu, Yang and Lu, Tong. (2022) "On Efficient Reinforcement Learning for Full-length Game of StarCraft II", Journal of Artificial Intelligence Research 75, 2022, pp.213-260.

[11] El-Nabarawy, Islam and Arroyo, K. and Wunsch, D. (2020). "StarCraft II Build Order Optimization using Deep Reinforcement Learning and Monte-Carlo Tree Search", https://arxiv.org/pdf/2006.10525.pdf.

[12] M. Kuchem, M. Preuss and G. R. (2013) "Multi-objective assessment of pre-optimized build orders exemplified for StarCraft 2" 2013 IEEE Conference on Computational Intelligence in Games (CIG), 2013, pp. 1-8, doi: 10.1109/CIG.2013.6633626.

[13] C.W. Leung, T.N. Wong, K.L. Mak, R.Y.K. Fung, (2010) Integrated process planning and scheduling by an agent-based ant colony optimization, Computers and Industrial Engineering, Vol. 59 (1), pp.166-180.

[14] Cobb, Charles W., and Paul H. Douglas. "A theory of production." (1928).

DOI: 10.15439/2023F4617
ISSN 2300-5963 ACSIS, Vol. 35

Discovering relationships between data in an enterprise information system using log analysis

Łukasz Korzeniowski
0000-0001-8458-9825
Nordea Bank Abp SA
Satamaradankatu 5, FI-00020,
Helsinki, Finland
Email: lukasz.korzeniowski@protonmail.com

Krzysztof Goczyła
0000-0003-3009-8988
Gdańsk University of Technology,
Faculty of Electronics,
Telecommunication and Informatics
Narutowicza 11/12, Gdańsk, Poland
Email: kris@eti.pg.edu.pl

***Abstract*—Enterprise systems are inherently complex and maintaining their full, up-to-date overview poses a serious challenge to the enterprise architects' teams. This problem encourages the search for automated means of discovering knowledge about such systems. An important aspect of this knowledge is understanding the data that are processed by applications and their relationships. In our previous work, we used application logs of an enterprise system to derive knowledge about the interactions taking place between applications. In this paper, we further explore logs to discover correspondence between data processed by different applications. Our contribution is the following: we propose a method for discovering relationships between data using log analysis, we validate our method against a benchmark system AcmeAir and we validate our method against a real-life system running at Nordea Bank.**

I. Introduction

LARGE enterprises, especially those with a long history of operation, face the challenge of modernizing their processes and adapting them to the changing environment. One of the key, and often the most challenging, aspects of such adaptation is the modernization of the enterprise's IT infrastructure. This is usually a complex endeavor, that includes exiting legacy systems, reorganizing the architecture, and harmonizing the data usage across the enterprise system. Each of these activities requires the enterprise architects team to have a good understanding of the existing IT infrastructure constituting the enterprise system, including the processes, applications participating in them, and the data being processed.

In [1] we proposed a method for discovering the knowledge about the interactions between applications in an enterprise system based on the analysis of application logs. We chose this type of analysis as the basis for our method due to some interesting properties of logs. Firstly, logging is a common practice present in IT from its very beginning, meaning that both legacy and modern applications are expected to create some sort of log (a trace of the actions executed by an application). Secondly, logs contain rich information, which blends the working application's technical and business aspects. Lastly, log entries tend to be kept up-to-date with the executed application code. All of these properties make application logs a perfect candidate for deriving the actual knowledge about various aspects of the enterprise system in an automated way.

In this paper, we further explore the potential of application log analysis in terms of supplying enterprise architects with valuable information about the enterprise system. This time, we focus on the data processed by applications. We try to find correspondence between data processed by different applications, which can be treated as good candidates for reconstructing relationships between data models used by different applications of the system. This information can be valuable in many aspects – it allows for the "detection" of pieces of information used by various applications, it allows for explaining information from legacy applications by finding its correspondence to information in modern (well-documented) applications, and it allows for attaching some business meaning to information by utilizing the business part of log entries. Our work fits in the domain model extraction area of the landscape of automated log analysis proposed by the authors of [2]. Other research in this area includes ontology discovery with process mining [3], search query categorization into a predefined taxonomy using search log analysis [4], or learning expert knowledge on applying security rules based on security log [5]. Our contribution to the body of knowledge is three-fold:

- we propose a method for automated discovery of relationships between data processed by applications, using textual analysis of the log content,
- we validate our method on the benchmark system AcmeAir [6],
- we validate our method on a real-life system running at Nordea Bank.

The rest of the paper is organized as follows. In Section II, we present a formal statement of the problem. Section III describes the proposed method of log analysis. In section IV, we introduce the two datasets used for the evaluation of our method and we define the evaluation criteria. In Section V, we compare our method with alternative approaches and describe the related work. We present the conclusions and plans for future work in Section VI.

This work was supported by the Gdańsk University of Technology and Nordea Bank

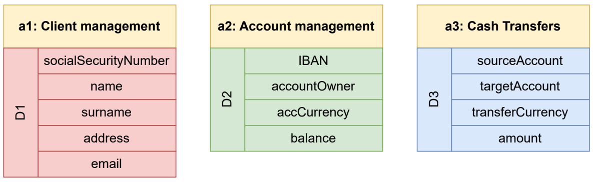


Fig. 1. An example of a fragment of an enterprise system S from the banking domain, related to the execution of cash transfers. The yellow color denotes different applications constituting the system. Red, green, and blue colors denote the data attributes processed by each of the applications.

II. Problem Statement

Let enterprise system S consists of a set of applications $A = \{a_1, \ldots, a_n\}$. Let each application $a_i \in A$ process some data represented by a set of data attributes $D^i = \{d_1^i, \ldots, d_{k^i}^i\}$. For each data attribute $d_k^i \in D^i$, we denote the set of potential values that the attribute can take by $V(i, k)$.

A fragment of an example enterprise system with applications and respective data attributes is presented in Fig. 1. The presented fragment is related to the processing of cash transfers in a bank and consists of three applications - managing clients, managing accounts, and cash transfer execution.

For any two data attributes d_k^i, d_l^j, we define their level of similarity using the Jaccard index

$$J(d_k^i, d_l^j) = \frac{|V(i,k) \cap V(j,l)|}{|V(i,k) \cup V(j,l)|} \tag{1}$$

We define a data relationship graph $G(S) = (U, E)$ as an undirected graph, where E is the set of edges consisting of all pairs of related data attributes d_k^i, d_l^j, and U is the respective set of vertices. An example of such a graph is presented in Fig. 2.

Let $L(a_i) = (l_1^i, \ldots, l_{k^i}^i)$ denote the log of application $a_i \in A$, represented by a tuple of log entries.

We define the problem of discovering data relationships as follows. Having a set of application logs $L = \{L(a_1), \ldots, L(a_n)\}$, find approximate graph $G^{'} = (U^{'}, E^{'})$.

III. Proposed Method

A. Method Overview

We propose a method that treats a log as a text. Such an approach has several benefits. It does not require any arbitrary assumptions to be made about the log content, which results in a broader scope of the method's usability. It also does not require additional preprocessing of the log, apart from unifying the log format across applications. Our method does not require any knowledge of the underlying data attributes for each application – they are discovered from the log automatically. Fig. 3 presents an overview of the steps that our method consists of.

Our method is parameterized by the following hyperparameters:

- N – the maximum length of n-gram embedding of a token,
- ies – inner embedding similarity threshold,

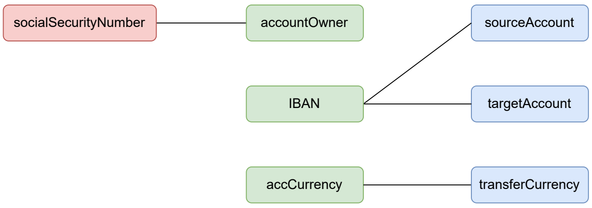


Fig. 2. A data relationship graph for the example system presented in Fig. 1. Edges represent relationships between data attributes from different applications. The coloring of nodes matches the coloring in Fig. 1.

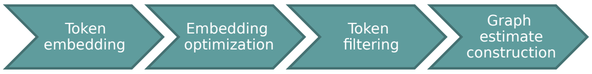


Fig. 3. Overview of data relationship discovery method.

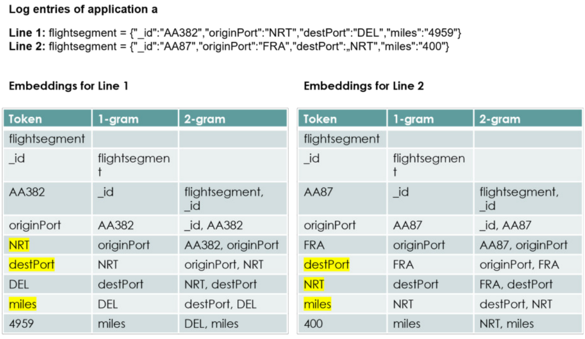
- oes – outer embedding similarity threshold,
- mt – minimum number of tokens sharing the same embedding that is required for the embedding to be considered,
- ml – minimum length for a token to be considered valid.

The following subsections describe each of the steps in detail.

B. Token Embedding

We process the log of each application $a \in A$ separately. For each log entry, we extract a list of tokens using the regular expression $[@.A-Za-z0-9\_]+$. For each token in the log entry and each $k \in \{1, \ldots, N\}$, we create its embedding as a k-gram of words [7] preceding the given token in the log line. Such an approach means that the same token can have multiple embeddings for a given value of k, depending on the context (consisting of k preceding tokens). An example of such embeddings is presented in Fig. 4. It can be noticed that "NRT", "destPort" and "miles" tokens have different embeddings for both log lines due to different neighbor tokens in each of the lines.

For each value of k and for each unique embedding e, we maintain the set of all tokens sharing this embedding within the $L(a)$, which we denote as $Emb^a(e, k)$. We denote the set of all k-gram embeddings within a as Emb_k^a, and the set of



Log entries of application a

Line 1: flightsegment = {"_id":"AA382","originPort":"NRT","destPort":"DEL","miles":"4959"}
Line 2: flightsegment = {"_id":"AA87","originPort":"FRA","destPort":„NRT","miles":"400"}

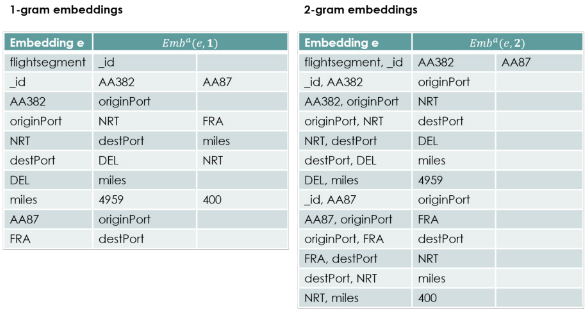
Embeddings for Line 1

Token	1-gram	2-gram
flightsegment		
_id	flightsegmen t	
AA382	_id	flightsegment, _id
originPort	AA382	_id, AA382
NRT	originPort	AA382, originPort
destPort	NRT	originPort, NRT
DEL	destPort	NRT, destPort
miles	DEL	destPort, DEL
4959	miles	DEL, miles

Embeddings for Line 2

Token	1-gram	2-gram
flightsegment		
_id	flightsegmen t	
AA87	_id	flightsegment, _id
originPort	AA87	_id, AA87
FRA	originPort	AA87, originPort
destPort	FRA	originPort, FRA
NRT	destPort	FRA, destPort
miles	NRT	destPort, NRT
400	miles	NRT, miles

Fig. 4. 1- and 2-gram embeddings for sample log entries. Yellow color denotes tokens with multiple embeddings.



1-gram embeddings

Embedding e	$Emb^a(e,1)$	
flightsegment	_id	
_id	AA382	AA87
AA382	originPort	
originPort	NRT	FRA
NRT	destPort	miles
destPort	DEL	NRT
DEL	miles	
miles	4959	400
AA87	originPort	
FRA	destPort	

2-gram embeddings

Embedding e	$Emb^a(e,2)$	
flightsegment, _id	AA382	AA87
_id, AA382	originPort	
AA382, originPort	NRT	
originPort, NRT	destPort	
NRT, destPort	DEL	
destPort, DEL	miles	
DEL, miles	4959	
_id, AA87	originPort	
AA87, originPort	FRA	
originPort, FRA	destPort	
FRA, destPort	NRT	
destPort, NRT	miles	
NRT, miles	400	

Fig. 5. Tokens sharing the same 1- and 2-gram embedding for sample log entries from Fig. 2.

all embeddings within $L(a)$ as $Emb^a = \bigcup_{k \in \{1,\ldots,N\}} Emb^a_k$. Fig. 5 shows groups of tokens sharing the same embedding.

For given k-gram embedding e consisting of tokens $(t_k, t_{k-1}, \ldots, t_1)$ and $l < k$, we define an l-cut operation on e, denoted as $e|l$, as follows: $e|l = (t_l, t_{l-1}, \ldots, t_1)$.

C. Embedding Optimization

The result of the previous step is ambiguous – the same groups of tokens are represented with multiple embeddings, for different values of $k \in \{1, \ldots, N\}$. To remove this ambiguity, we search for the highest value of k such that k-gram embedding of the groups of tokens represents them better than $k+1$-gram embedding. The longer the embedding, the more precisely it describes the represented tokens.

We start with the set of initial embeddings E that consists of all 1-gram embeddings. For each 2-gram embedding e, we then take the set of all tokens that it represents $Emb^a(e, 2)$. We compare this set with the set of all tokens represented as 1-cut of e, using the Jaccard index. If the index value is above the ies threshold, we substitute the $e|1$ embedding in E with e. We repeat this procedure for longer k-grams until $k = N$, each time comparing the k-gram embedding with its k-1-cut. In the end, the set E contains all the embeddings for log $L(a)$ after optimization, which we denote as $OEmb^a$. Fig. 6 shows the two consecutive steps of the optimization process.

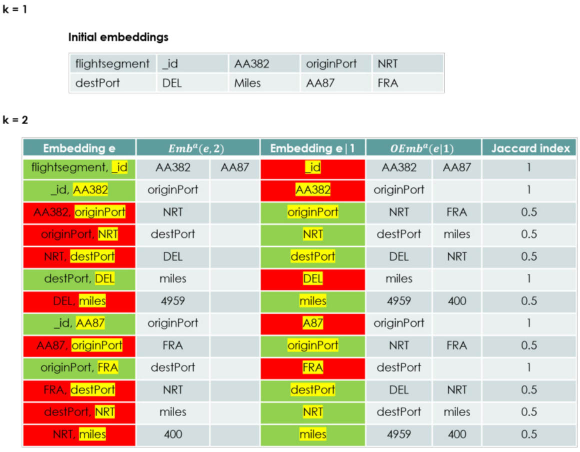
$OEmb^a(e)$ denotes the set of tokens represented by embedding e within log $L(a)$. The above algorithm ensures that there is only one embedding (one value of k) that represents a given set of tokens. Fig. 7 presents the example outcome of the optimization process.

D. Token Filtering

We further optimize the set of embeddings. For each application log $L(a)$ and each embedding $e \in OEmb^a$:

- we discard e if $|OEmb^a(e)| < mt$,
- we define the filtered set of tokens, such that $\forall_{e \in OEmb} FEmb^a(e) = \{t \in OEmb^a(e) | length(t) \geq ml\}$.

$FEmb^a(e)$ denotes the final set of tokens represented by the embedding e within application log $L(a)$, while $FEmb^a$ represents the set of all final embeddings for application a,



k = 1

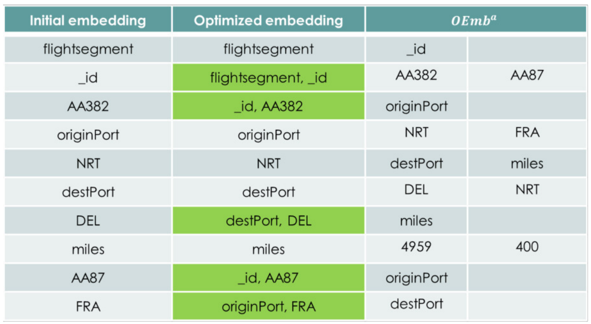
Initial embeddings

flightsegment	_id	AA382	originPort	NRT
destPort	DEL	Miles	AA87	FRA

k = 2

Embedding e	$Emb^a(e,2)$		Embedding e | 1	$OEmb^a(e\|1)$		Jaccard index
🟩 flightsegment, _id	AA382	AA87	🟥 _id	AA382	AA87	1
🟩 _id, AA382	originPort		🟥 AA382	originPort		1
🟥 AA382, originPort	NRT		🟩 originPort	NRT	FRA	0.5
🟥 originPort, NRT	destPort		🟩 NRT	destPort	miles	0.5
🟥 NRT, destPort	DEL		🟩 destPort	DEL	NRT	0.5
🟩 destPort, DEL	miles		🟥 DEL	miles		1
🟥 DEL, miles	4959		🟩 miles	4959	400	0.5
🟩 _id, AA87	originPort		🟥 A87	originPort		1
🟥 AA87, originPort	FRA		🟩 originPort	NRT	FRA	0.5
🟩 originPort, FRA	destPort		🟥 FRA	destPort		1
🟥 FRA, destPort	NRT		🟩 destPort	DEL	NRT	0.5
🟥 destPort, NRT	miles		🟩 NRT	destPort	miles	0.5
🟥 NRT, miles	400		🟩 miles	4959	400	0.5

Fig. 6. Steps of the example optimization process for embeddings presented in Fig. 5 and ies threshold of 0.9. The yellow color denotes the $k-1$ cut of the embedding. Each row in the table presents a step in the process. Green cells denote the embeddings that have been accepted and red cells denote the embeddings that have been rejected.



Initial embedding	Optimized embedding	$OEmb^a$	
flightsegment	flightsegment	_id	
_id	🟩 flightsegment, _id	AA382	AA87
AA382	🟩 _id, AA382	originPort	
originPort	originPort	NRT	FRA
NRT	NRT	destPort	miles
destPort	destPort	DEL	NRT
DEL	🟩 destPort, DEL	miles	
miles	miles	4959	400
AA87	🟩 _id, AA87	originPort	
FRA	🟩 originPort, FRA	destPort	

Fig. 7. The outcome of the example process presented in Fig. 6 Green cells denote the embeddings that have been extended as part of the optimization.

and $FEmb = \bigcup_{a \in A} FEmb^a$ is a set of all embeddings in the system.

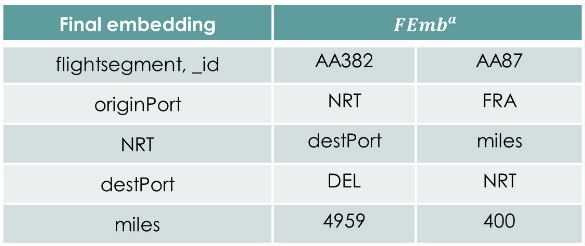
The first optimization removes embeddings that represent only a few tokens. Such embeddings are interpreted as representations of static parts of the log entry, which are of less interest in terms of data relationship discovery. The second optimization removes short tokens, which decreases the chance of discovering accidental relationships.

E. Graph Estimate Construction

We calculate a distance matrix between all $FEmb$ embeddings using the Jaccard index as the distance measure:

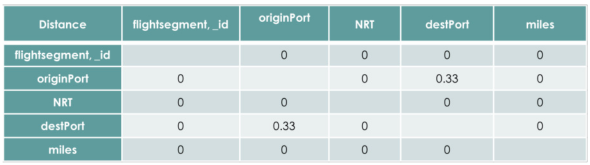
$$\forall_{e_1, e_2 \in FEmb, e_1 \neq e_2} dist(e_1, e_2) = \frac{|FEmb(e_1) \cap FEmb(e_2)|}{|FEmb(e_1) \cup FEmb(e_2)|} \tag{2}$$

We filter pairs of embeddings based on their distance, using oes as the threshold, above which the embedding relationship is considered strong enough and should be retained. The retained embedding pairs form the edges of our approximate graph $G^{'}$ and respective embeddings become the vertices of the graph. Fig. 9 presents an example distance matrix



Final embedding	$FEmb^a$	
flightsegment, _id	AA382	AA87
originPort	NRT	FRA
NRT	destPort	miles
destPort	DEL	NRT
miles	4959	400

Fig. 8. The final set of embeddings for sample log entries presented in Fig. 4.



Distance	flightsegment, _id	originPort	NRT	destPort	miles
flightsegment, _id		0	0	0	0
originPort	0		0	0.33	0
NRT	0	0		0	0
destPort	0	0.33	0		0
miles	0	0	0	0	

Fig. 9. Distance matrix for final embeddings from Fig. 8, based on the Jaccard index.

for the final embeddings presented in Fig. 8. The respective graph estimate is shown in Fig. 10, which shows a detected relationship between *originPort* and *destPort* data attributes.

IV. Method Evaluation

A. Dataset Overview

We evaluate our method using two datasets:

- AcmeAir – a dataset allowing us to assess the accuracy of our method on a benchmark system, that we have full knowledge about,
- NDEASET2 – a dataset from a real-life enterprise system running at Nordea Bank.

Both datasets are described in detail in subsequent sections.

B. Benchmark Dataset

AcmeAir [6] is an open-source implementation of a fictitious flight reservation system that is commonly used as a benchmark system [8]. It consists of a web application and several restful web services that contribute entries to a common log file. AcmeAir provides very limited documentation that allows deriving only the data model (Fig. 11).

The AcmeAir documentation does not provide a component diagram but limits itself only to listing backend services with their responsibilities:

- BookingService - creating new bookings and canceling existing bookings for a given customer,
- CustomerService - creating and updating customer information, managing the customer's session,
- FlightService - allowing users for searching for flights by airports and/or departure dates and searching for flight segments.

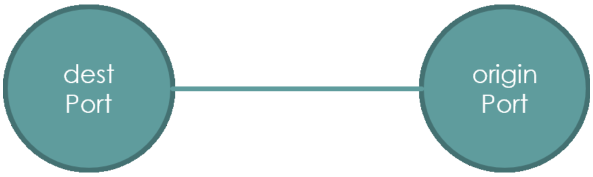


Fig. 10. Graph estimate for the set of final embeddings and oes threshold of 0.25.

However, based on the analysis of the system's source code, we have reconstructed missing elements of the component model, which are presented in Fig. 12.

All AcmeAir components put their log entries into a common log file in a standardized manner. Each log entry produced by AcmeAir consists of the following elements (see Fig. 13 for a log sample):

- Timestamp – date and time of creating the log entry,
- Logging level – either DEBUG or INFO,
- Source – the name of the component that created the entry,
- Content – logged message.

In our analysis, we use logs produced by a fork of AcmeAir [9] which extends the logging to cover HTTP requests/responses issued within AcmeAir. We use the AcmeAir driver – a built-in simulator of user interactions with the system – to generate a log file used in further analysis. The workload is generated in a multi-threaded manner. We use five threads to simulate concurrent actions taken by users of the web application. It is important to note that although the log is generated in a multithreaded way, log entries do not contain any information allowing one to identify a thread within which specific entries are logged. We use a log fragment of 1000000 lines (file size: 314MB). We have found that larger log fragments do not introduce more information as the patterns appearing in the log keep repeating.

As part of log preprocessing, we performed the following transformations to the log:

- we transform the content into a CSV format to be aligned with our real-life dataset. Each log entry is described by *timestamp*, *source*, and *message*.
- For the log entries where HTTP request/responses are logged, we derive the source from the HTTP endpoint. Such an approach better reflects the actual relationship between the data and the service that processes them.

Both the original and the preprocessed logs are available in [9].

C. Real-life Dataset

We used a log dataset NDEASET2 from a real-life system deployed at Nordea Bank, which we refer to as NDEASYS. As compared to the NDEASET1 dataset described in our previous work [1], NDEASET2 covers a full working week of the NDEASYS. The total size of the dataset is larger by an order of magnitude (95GB). It covers one more application and one additional process (daily reporting). We followed the same rules for removing bias as described in [1] to ensure proper diversity of the dataset, which include:

- team diversity – applications built by teams of different sizes, experiences, locations,

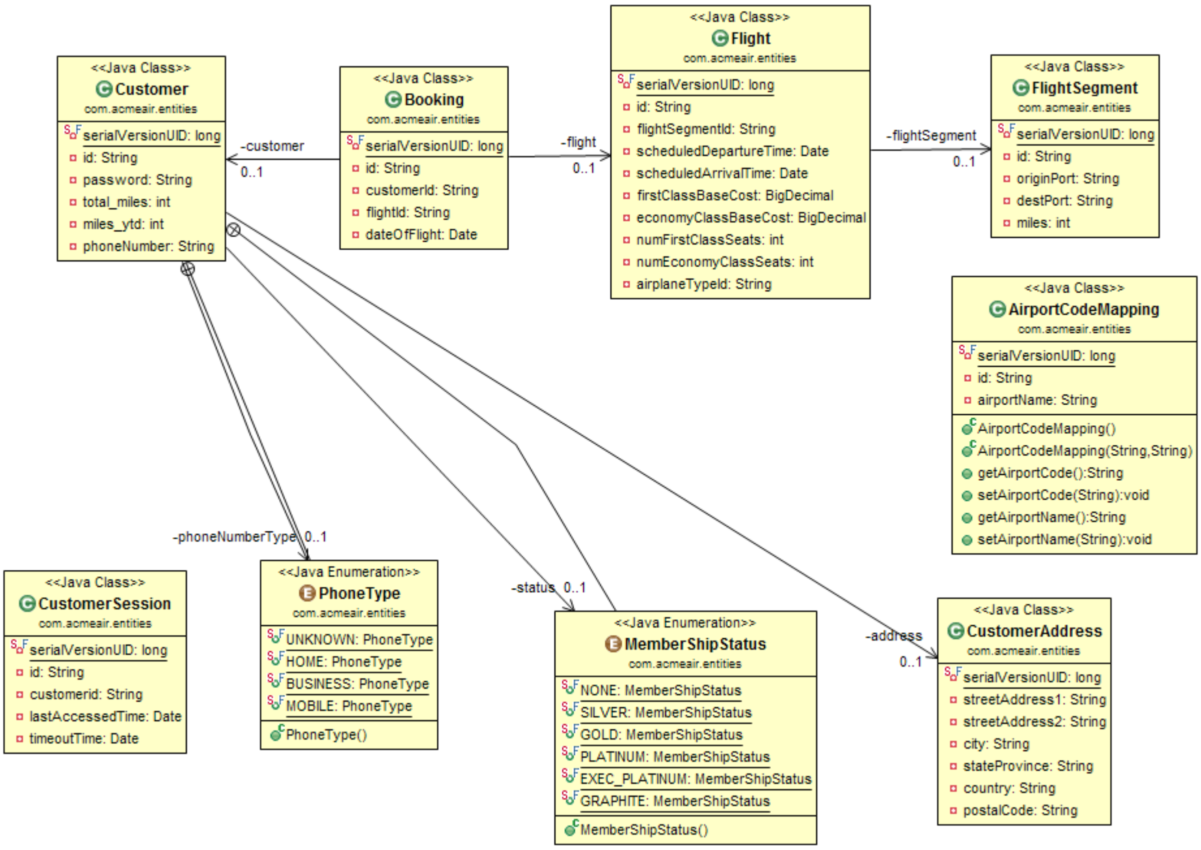


Fig. 11. Class diagram of AcmeAir system available in its documentation.

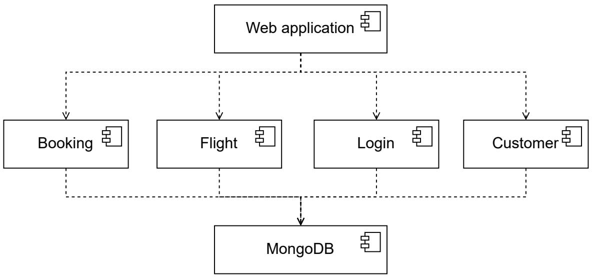
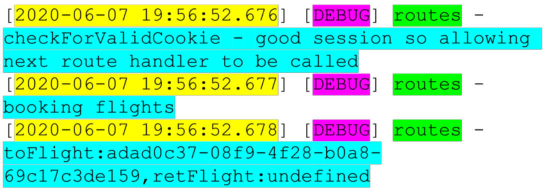


Fig. 12. Component model of AcmeAir.



```
[2020-06-07 19:56:52.676] [DEBUG] routes -
checkForValidCookie - good session so allowing
next route handler to be called
[2020-06-07 19:56:52.677] [DEBUG] routes -
booking flights
[2020-06-07 19:56:52.678] [DEBUG] routes -
toFlight:adad0c37-08f9-4f28-b0a8-
69c17c3de159,retFlight:undefined
```

Fig. 13. An AcmeAir log sample. Yellow color denotes the timestamp, magenta – the logging level, green - the source of the log entry, and cyan – the message.

- application diversity – we included dedicated business applications, purely technical components, and shared service platforms (e.g., storage or communication services),
- time diversity – applications built in different periods,
- integration diversity – applications communicating using different interfaces and exchange formats.

The bias of the dataset related to logs being collected within the same company is mitigated by the fact that Nordea does not enforce any strict rules for log creation and log content for non-regulatory logging.

The fragment of the architecture of the NDEASYS that contributes to the creation of logs within NDEASET2 is shown in Fig. 14. Applications B and D are both providing data of the same type to A, but using different formats. Application A enriches the received data with data from applications C, E, and F. The final result is aggregated as part of the daily reporting and the aggregated data are provided to application G.

Table I describes the characteristics of the NDEASYS2 dataset. As part of the log preprocessing, we unify all of the logs in the dataset to a common CSV format with timestamp, source, and message columns, which match the format of the AcmeAir logs that we use as a benchmark. Fig. 15 presents an anonymized example of log entries in NDEASET2.

TABLE I
CHARACTERISTICS OF THE NDEASYS2 DATASET

App	Log size [MB]	Application diversity	Team diversity		Time diversity		Integration diversity	
		Type	Size	No. locations	Dev. period	Dev. duration [months]	Integration style	Format
A	26000	dedicated	3	2	2020-2022	24	Messaging, RPI	Swift (ISO15022, ISO 20022), JSON
B	165	technical	1	2	2020	1	Messaging	Swift (ISO15022)
C	7000	shared service	2	2	2016-2020	48	RPI	JSON
D	18	technical	1	2	2020	6	Messaging	Swift (ISO15022)
E	64000	shared service	3	2	2016-2022	72	RPI	JSON
F	153	shared service	3	2	2016-2022	72	RPI	JSON
G	80	dedicated	1	2	2020	1	Messaging	Swift (ISO15022

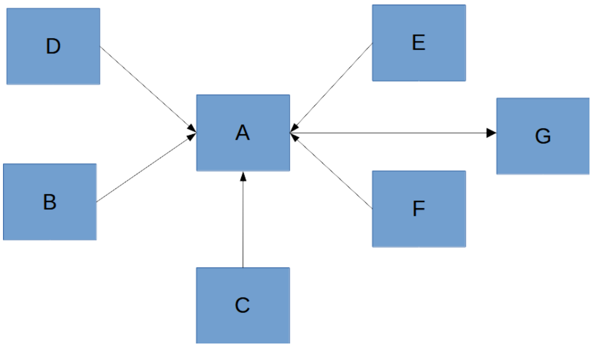


Fig. 14. The architecture of the system used for evaluation. Lines denote pairs of interacting applications, and arrows denote the direction of the data flow.

```
1647330650034,E,"
logger=c.n.t.i.r.q.QueryCallStatsObserver,
operation=QUERY_LATEST_IN_GROUP, clientId=X,
clientLibrary=null, clientVersion=null,
hostName=a01.com, correlationId=969a81d3-8ad8-4a5f-
84e8- 0868bfd65ddb, action=query_start, , domain=Y/Z,
requestCondition={""extracted.id"":
""0122318714085000""},  condition={""extracted.id"":
""0122318714085000""},
groupByFields=[Id], sortFields=[timestamp], limit=0,
payload=true"
```

Fig. 15. Example of one log entry from NDEASET2. Yellow color denotes the timestamp, green –the source application, and cyan –the message.

D. Evaluation Criteria

For each dataset, we use evaluation criteria suitable to the level of knowledge about the system that the dataset is based on. For AcmeAir, we can assume full knowledge about the system and its data model due to its simplicity and open-source nature. Based on the data model presented in Fig. 9, we construct the reference data relationships graph shown in Fig. 16, which serves as our ground truth. Arrows are introduced for clarity only. They denote the direction of the relationship and are not taken into consideration during method evaluation. Vertical lanes represent different services depicted in Fig. 10. Nodes within a lane represent data attributes managed by a given service. Each node is described by an attribute and entity it belongs to (e.g. *Customer* is an entity and *id* – is its attribute). Edges of the graph represent relationships between data attributes. Only the attributes that are related to one another are presented in the graph.

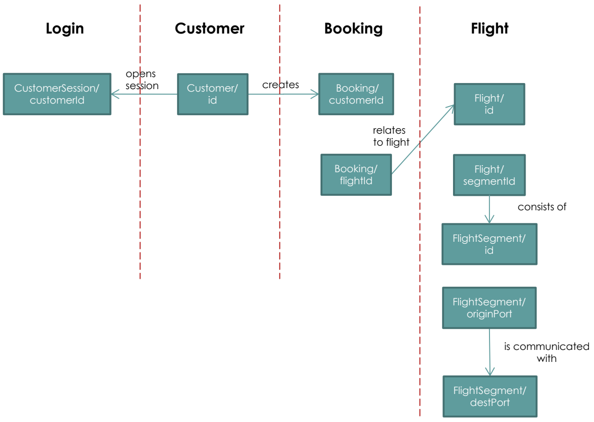


Fig. 16. Graph of data relationships in AcmeAir system, serving as our ground truth.

We use the F1 score of the set of graph edges to determine the accuracy of our method for the AcmeAir dataset.

For the NDEASET2 dataset, we are unable to construct a reference graph. However, our knowledge of the NDEASYS system allows us to assess, whether or not the discovered relationship is correct. Therefore, for this dataset, we use only the precision of the discovered set of edges as the measure of our method's accuracy.

E. Benchmark Results

To be able to compare the outcome of our method with the ground truth presented in Fig. 16, we need to perform two types of mapping:

- mapping of discovered data attributes to the data attributes from Fig. 11 (details of mapping are presented

TABLE II
MAPPING OF DATA ATTRIBUTES BETWEEN THE ACMEAIR LOG AND THE REFERENCE DATA RELATIONSHIP GRAPH

Log		**Reference data relationship graph**	
Service	**Discovered** attribute	**Entity**	**Attribute**
Booking	returnbookingid	Booking	id
	departbookingid	Booking	id
	number	Booking	id
	_id	Booking	id
	customerid	Booking	customerId
	userid	Booking	customerId
	user	Booking	customerId
	byuser	Booking	customerId
	flightid	Booking	flightId
	retflightid	Booking	flightId
	retflight	Booking	flightId
	toflight	Booking	flightId
	toflightid	Booking	flightId
	retflightsegid	Flight	segmentId
	toflightsegid	Flight	segmentId
Flights	scheduledarrival time	Flight	scheduledArrival Time
	scheduleddeparture time	Flight	scheduledDeparture Time
	_id	Flight	id
	flightsegmentid	Flight	flightSegmentId
	toairport	FlightSegment	destPort
	fromairport	FlightSegment	originPort
	originport	FlightSegment	originPort
	destport	FlightSegment	destPort
	miles	FlightSegment	miles
Customer	_id	Customer	id
	user	Customer	id
	byid	Customer	id
Login	login	Customer	id
Web	user	Customer	id
	miles	FlightSegment	miles
	_id	FlightSegment	id
	destport	FlightSegment	destPort
	originport	FlightSegment	originPort

in Table II),

- mapping of the denormalized data model present in the logs to a normalized model present in Fig. 11.

The latter mapping is necessary because the information in the logs of individual services represents some part of the view of the overall schema of the application. Additionally, the information in logs overlaps between the logs of different services.

It is expected that logs contain more information than only related to data schemas, thus the resulting data relationship graph will be richer (having more nodes and edges) than the reference graph from Fig. 16. To provide a fair assessment of our method, we first present the results on a subgraph limited to the set of nodes from Fig. 16. Then we discuss separately the rest of the graph.

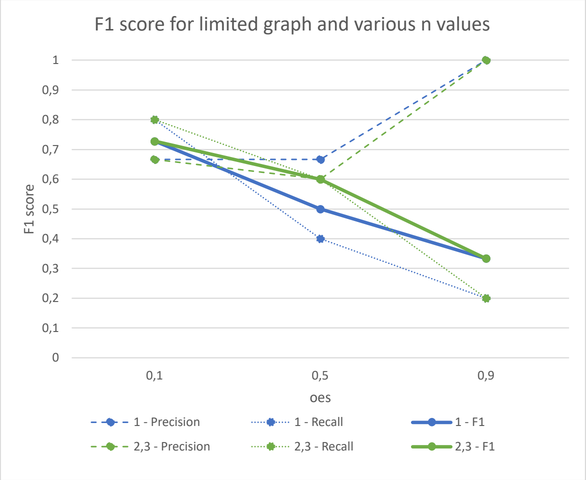


Fig. 17. The results of our method were measured by the F1 score. Colors denote the outcome for different values of n. Dashed lines represent precision, dotted – recall, and solid – F1 measure. The horizontal axis represents different values of the oes meta-parameter.

Fig. 17 Shows the results of our method for the limited graph under different values of meta-parameters. It can be observed that for $n = 2$ and $n = 3$, the results are the same, and the n meta-parameter, in general, does not have a key influence on the overall score. A much more significant parameter is oes, which determines the perception of similarity between data attributes. The best F1 score of 0,72 was reached for the oes value of 0.1. We interpret such a low value as our method needing an argument to exclude a relationship (low level of similarity). This follows the intuition that, even though two attributes are closely corresponding to one another, they do not have to share the same values across the whole log. It is more probable that the set of shared values would be rather small because of a different set of services being triggered depending on the processes executed in the system.

An example of a graph generated by our method is shown in Fig. 18. The thick edges denote data relationships that match the reference data relationship graph from Fig. 16. We can see that the generated graph is much richer in information and our method gives a couple of interesting insights into the underlying data schemas. Firstly, it shows the potential of detecting unknown relationships based on data. An example is a relationship between *CustomerSession/customerId* and *Booking/customerId* which was detected based on matching values that both attributes take. An interesting part of the graph is a complete subgraph under Web service, whose nodes represent airport names. The fact that these attributes were extracted and are connected means that these values must frequently occur next to one another in logs. It is the case since when a flight is booked, origin and destination airports are logged in a single log entry. The fact that these attributes form a complete subgraph is the effect of the low cardinality of the set of airports. Combined with the automated

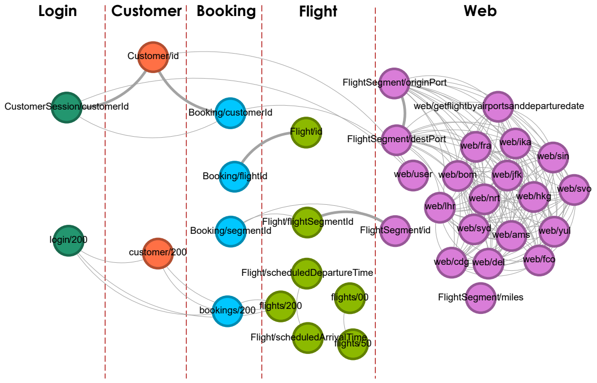


Fig. 18. A data relationship graph was generated for n=3, ies=0.9, and oes=0.1. Vertical lanes denote services depicted in Fig. 12. Nodes within a lane represent data attributes discovered in the log of a given service. Thick edges represent data relationships that match the reference relationship graph in Fig. 16.

TABLE III
CHARACTERISTICS OF THE NDEASYS2 DATASET

n	Discovered data attribute (n-gram embedding)
2	booking, byuser
2	flightsegment, _id
3	customer, by, user
3	flights, by, user

generation of the AcmeAir log sample, this resulted in all the origin/destination airport combinations being exhausted. It shows that our method is susceptible to the identification of false relationships for data of low cardinality. This is normally addressed in our method by choosing only long enough tokens (ml meta-parameter of our method). For the AcmeAir dataset, however, the minimum token length was set to 3 to utilize the airport data as a means of detecting other relationships.

In Fig. 18 it can be observed that the graph contains nodes representing data attributes that are not present in the original model in Fig. 11, e.g. *web/user* or *web/getflightsbyairportanddeparturedate*. These attributes come from the analysis of log entries that do not directly log the whole data entity but are representants of typical observation-point logging used by developers [10]. Such attributes are valuable since they provide an additional semantic context to interpret other attributes. In the case of *web/user*, we can interpret that a customer is a user of the system.

The graph presented in Fig. 18 is created after performing the attribute mapping described at the beginning of the section. The full set of results achieved by our method for various meta-parameters is available in [9] in the GEXF format. However, it is interesting to analyze raw data attributes. Table III presents a sample extract of raw attributes represented by n-grams for $n \in \{2, 3\}$. The set of tokens constituting the n-gram can also be viewed as a semantic context, which can be used for the refinement of the generated graph in the future.

Our method has also identified some false positives – numeric nodes that represent an accidental correlation of data that should be treated as noise. Filtering out this noise is one of the challenges for our future work.

F. Real-life System Results

Our method has been executed against a log of the real-life system running at Nordea Bank - NDEASYS. We used the following meta-parameters to get the best results:

- $N = 4$,
- $ies = 0.9$,
- $oes = 0.1$,
- $mt = 10$,
- $ml = 5$.

Since, due to the size of the domain, it is hard to obtain the full ground truth, we focused on identifying falsely discovered data relationships. The edges of the resulting graph were validated using our domain knowledge. Our method achieved a precision of 0.98 on NDEASET2.

We performed an additional validation of the achieved results based on the analysis of data flows within NDEASYS. Based on Fig. 14, it is expected that data relationships are found between applications B, G, A, and F, which represent the main data flow. The discovered data relationships are aligned with the diagram in Fig. 14. For every edge of the architecture graph, there is at least one discovered data attribute relationship between the two applications.

Also, other properties of our method revealed on the AcmeAir dataset were confirmed:

- larger values of N tend to introduce an additional semantic context to the data relationship graph,
- observation-point logging entries introduce a domain nomenclature that puts other discovered attributes into a concrete context.

While promising, these results still need to be validated especially in terms of the recall and F1 measures. The size of the data and the big number of discovered relationships do not allow us to present an anonymized visualization of the retrieved graph, analogous to Fig. 18.

G. Threats to Validity

Internal threats to validity include the construction of the NDEASET2 dataset and the level of detail of logs. We tried to mitigate both threats by choosing applications that provided a decent level of diversity. We find team diversity especially important, as it ensures log entries are created by developers from different teams, potentially following different standards. Also, the application diversity (different types of applications included in the dataset) is meant to increase the overall representativeness of the dataset. As for the quality of logs, we will cover its influence on our method's results in our future work.

The biggest external threat to validity is the dataset from Nordea Bank, which might not be representative of the whole bank and enterprise systems in general. This threat might come from the specific dataset that was chosen for research but also from potential formal or informal rules regarding logging practices enforced by Nordea. We tried to mitigate this threat by applying our method to a benchmark system, which is however much simpler than an average enterprise application, both business and technical-wise. In the future, we will try to find other datasets to further mitigate these threats. It is worth noting that although publicly available datasets are very beneficial for the validation of our method and the general reproducibility of research, they will never be close to the size and complexity of real-life enterprise systems. Therefore, apart from selecting open-source datasets, we will seek further datasets inside Nordea Bank to retain real-world validation for our research.

The threat to construct validity is an insufficient measure of the quality of our method for the NDEASET2 dataset (we are using only precision, not using recall, and not being able to construct the reference graph). Construction of the reference graph for NDEASYS is one of the goals of our future work.

Conclusion validity is threatened by data attributes with low cardinality of values (e.g. currency symbols in the banking industry or airport codes in the flight booking domain). Such data attributes can be falsely considered as related just because of a high chance of them getting the same values. Another threat to conclusion validity is the low level of detail of logs can result in relationships between data attributes not being detected.

V. Related Work

Discovering relationships between data is a topic studied by schema matching discipline [11]. This domain is very broad and uses several techniques, including matching strings, matching words in certain languages, matching graphs, or using ontologies representing knowledge in certain fields. The goal of traditional schema matching is to match elements between two schemas given as input. The authors of [12] and [13] study a more general case where the number of schemas to be matched is greater than two. Finding similarities in attribute values (called duplicates) is the basis of the method proposed in [14]. The authors perform schema matching based on a small number of matching values in two schemas. [15] proposes a method for matching knowledge graphs. The authors split the process into schema-level and instance-level matching. In the first phase, matching is performed using only schema information, while in the second phase, the result is further refined by matching the values that schema attributes take. All of the approaches in the schema matching domain assume full knowledge of the individual schemas being matched. What is in the area of interest is only finding the correspondence between the attributes of the schemas. This is a significant difference compared to our method, which needs to discover both the schemas and their corresponding attributes.

Semantic data type discovery is a field of research that focuses on assigning types that have well-defined meanings to schema attributes. As compared to regular type detection (e.g. whether an attribute is a string, int, or boolean), semantic types hold much more information (e.g. postal code, surname, country) and as such could be used to match attributes of different schemas. [16] introduces *Sherlock*, a supervised-learning approach to semantic type detection. It uses the VisNet dataset to train a classifier that detects one of the 78 semantic types defined in the *T2Dv2Gold Standard* dataset. The authors of [17] propose the *SATO* algorithm, which extends *Sherlock* by incorporating the concept of context. Data attributes are matched not only based on the values they take but also based on the neighbor attributes in the same schema. Such an approach allows to properly classify semantic types for attributes with a low number of samples. Both [17] and [18] rely on model training for a given dataset which contrasts with the unsupervised approach we take to derive data relationships. [18] describes *RaF-STD*, an unsupervised learning approach to semantic type detection. The authors exploit triples of schema attributes that share common values and iteratively introduce higher-level virtual attributes representing the notion of similarity. This method does not require any prior knowledge of the source schemas but requires the existence of such schemas, which we do not assume in our method.

Automated log analysis is a field of research that focuses on the extraction of data from logs in an automated fashion. The authors of [2] split this field based on the type of knowledge being extracted. According to this classification, domain model extraction is a field that is somewhat relevant to data attribute discovery. [19] presents a method for discovering an ontology based on event logs and process mining techniques. The method is validated using a dataset of stack overflow posts and proved to generate a valid ontology in a computer science domain. The use of an event log requires intensive log preprocessing, which is not the case with our method, which operates on raw application logs, with very little preprocessing to unify log format across applications.

Log template generation is a field of research that aims in finding patterns in lines of log files as part of a log analysis task. Typically such patterns split a log line into static and variable parts, which could be used to discover data attributes in our method. [20] and [21] are two common methods for discovering log patterns. According to [1], both of these methods do not cope well with log lines of variable length (e.g. XML document being logged), which can be a typical case of logging entry/exit data. The authors of [10] classify such log entries under the observation-point logging category and show it is one of the most common types of log entries. [1] proposes an *SLT* method for log template generation, which is well-suited for handling log entries of variable length but the result is coarse-grained – it does not provide any information on the position of the variable parts within a log line. Our method for data relationship discovery uses the context of neighbor tokens to detect data attributes, so the position of the variable tokens within a log entry is essential.

Word embedding is a sub-field of natural language processing that aims in representing words in some text corpus as multi-dimensional vectors. This corresponds to the initial phases of our method, where we perform the embedding of tokens of a log line. [22] and [23] are the two most popular methods for word embedding. For both, there is a large set of pre-trained models over text corpora in various languages. However, their usability to log analysis is limited. Firstly, they require training of the model on a particular text corpus, as the model is specific to the logs being analyzed. Secondly, the set of words in log corpora is infinite due to information like unique identifiers being commonly logged.

The method for discovering interactions between applications described in [1] shares similar ideas to the method described in this paper. It looks for rare tokens in various applications' logs and uses the findings to justify the hypothesis of the existence of a relationship between applications. The main difference between [1] and this paper is that [1] is focused on detecting the relationships between applications, while this paper focuses on detecting relationships between data attributes. For this purpose, we are looking not only at the rare but also the more frequent tokens. Discovery of a relationship of more frequent tokens (e.g. currency codes) in two applications cannot be treated as a good justification for the existence of a dependency (data exchange) between the applications. Therefore the method presented in this paper cannot be considered a generalization of the method described in [1]. These methods are rather complementary and the identification of relationships in data can be used to refine the set of interactions discovered by [1].

VI. Conclusions and Future Work

In this paper, we proposed an unsupervised method for the automated discovery of data relationships in an enterprise system. We validated our method on a synthetic and real-world dataset. In both cases, it proved to be a useful tool for obtaining an overview of the data processed by applications within an enterprise system. For the real-world dataset, the method has successfully detected correspondences between data attributes in different applications, which can have multiple uses - failure diagnosis, schema discovery, or schema mapping, to name a few.

In the future, we will extend the method to take advantage of additional information – locality of log entries, data attributes described by n-grams with higher values of n, and semantics hidden in the observation-point logging entries. We will further analyze the retrieved relationship graphs for the NDEASET2 dataset and work on establishing the ground truth for NDEASYS to validate the recall and F1 measures. We will also combine the proposed method for data relationship discovery with the method of interaction discovery [1] to form a comprehensive approach for discovering knowledge about an enterprise system based on automated log analysis.

Acknowledgment

This paper was written in cooperation with the Nordea Bank which provided the log dataset and an overview of the systems that were subject to this study.

References

[1] L. Korzeniowski and K. Goczyla, "Discovering interactions between applications with log analysis," 2022. doi: 10.15439/2022F172 p. 861 – 869.

[2] L. Korzeniowski and K. Goczyla, "Landscape of automated log analysis: A systematic literature review and mapping study," *IEEE Access*, vol. 10, pp. 21 892–21 913, 2022. doi: 10.1109/ACCESS.2022.3152549

[3] D. Barua, N. T. Rumpa, S. Hossen, and M. M. Ali, "Ontology based log analysis of web servers using process mining techniques," 2019, Conference paper. doi: 10.1109/ICECE.2018.8636791 p. 341 – 344.

[4] S.-L. Chuang and L.-F. Chien, "Enriching web taxonomies through subject categorization of query terms from search engine logs," *Decision Support Systems*, vol. 35, no. 1, pp. 113–127, 2003. doi: https://doi.org/10.1016/S0167-9236(02)00099-4 Web Retrieval and Mining.

[5] S. Khan and S. Parkinson, "Eliciting and utilising knowledge for security event log analysis: An association rule mining and automated planning approach," *Expert Systems with Applications*, vol. 113, p. 116 – 127, 2018. doi: 10.1016/j.eswa.2018.07.006

[6] Acmeair, "A java implementation of the acme air sample application." last accessed: 2023-07-24. [Online]. Available: https://github.com/acmeair/acmeair

[7] C. D. Manning, H. Schütze, and G. Weikurn, "Foundations of statistical natural language processing," *SIGMOD Record*, vol. 31, no. 3, p. 37 – 38, 2002. doi: 10.1145/601858.601867

[8] C. M. Aderaldo, N. C. Mendonça, C. Pahl, and P. Jamshidi, "Benchmark requirements for microservices architecture research," 2017. doi: 10.1109/ECASE.2017.4 p. 8 – 13.

[9] Acmeair, "A nodejs implementation of the acme air sample application with extended logging." last accessed: 2023-07-24, commitId: 59e8545c1e5264107e60706a360e0c8133aa8f9e. [Online]. Available: https://github.com/lkorzeni11/acmeair-nodejs

[10] Q. Fu, J. Zhu, W. Hu, J.-G. Lou, R. Ding, Q. Lin, D. Zhang, and T. Xie, "Where do developers log? an empirical study on logging practices in industry," 2014, Conference paper. doi: 10.1145/2591062.2591175 p. 24 – 33.

[11] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3730 LNCS, p. 146 – 171, 2005. doi: 10.1007/11603412_5

[12] E. Rahm and E. Peukert, "Large-scale schema matching," in *Encyclopedia of Big Data Technologies*, 1st ed., S. Sakr and A. Zomaya, Eds. Springer Publishing Company, Incorporated, 2019. ISBN 331977526X

[13] E. Rahm and E. Peukert, "Holistic schema matching," in *Encyclopedia of Big Data Technologies*, 1st ed., S. Sakr and A. Zomaya, Eds. Springer Publishing Company, Incorporated, 2019. ISBN 331977526X

[14] A. Bilke and F. Naumann, "Schema matching using duplicates," 2005. doi: 10.1109/ICDE.2005.126 p. 69 – 80.

[15] X. Xue and H. Zhu, "Matching knowledge graphs with compact niching evolutionary algorithm," *Expert Systems with Applications*, vol. 203, 2022. doi: 10.1016/j.eswa.2022.117371

[16] M. Hulsebos, A. Satyanarayan, K. Hu, T. Kraska, M. Bakker, Demiralp, E. Zgraggen, and C. Hidalgo, "Sherlock: A deep learning approach to semantic data type detection," 2019, Conference paper. doi: 10.1145/3292500.3330993 p. 1500 – 1508.

[17] D. Zhang, Y. Suhara, J. Li, M. Hulsebos, Demiralp, and W. C. Tan, "Sato: Contextual semantic type detection in tables," *Proceedings of the VLDB Endowment*, vol. 13, no. 11, p. 1835 – 1848, 2020. doi: 10.14778/3407790.3407793

[18] F. Piai, P. Atzeni, P. Merialdo, and D. Srivastava, "Fine-grained semantic type discovery for heterogeneous sources using clustering," *VLDB Journal*, vol. 32, no. 2, p. 305 – 324, 2023. doi: 10.1007/s00778-022-00743-3

[19] D. Barua, N. T. Rumpa, S. Hossen, and M. M. Ali, "Ontology based log analysis of web servers using process mining techniques," 2019, Conference paper. doi: 10.1109/ICECE.2018.8636791. ISBN 978-153867482-6 p. 341 – 344.

[20] R. Vaarandi and M. Pihelgas, "Logcluster - a data clustering and pattern mining algorithm for event logs," 2015, Conference paper. doi: 10.1109/CNSM.2015.7367331 p. 1 – 7.

[21] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," 2017, Conference paper. doi: 10.1109/ICWS.2017.13 p. 33 – 40.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, Conference paper.

[23] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," 2014, Conference paper. doi: 10.3115/v1/d14-1162 p. 1532 – 1543.

DOI: 10.15439/2023F7831
ISSN 2300-5963 ACSIS, Vol. 35

The Use of Digital Technologies in German Business Consultancies

Christian Leyh
0000-0003-0535-0336
Technical University of Central Hesse – University of Applied Sciences
THM Business School
Wiesenstr. 14,
35390 Gießen, Germany
Email: Christian.Leyh@w.thm.de

Marcel Lange
Technical University of Central Hesse – University of Applied Sciences
THM Business School
Wiesenstr. 14,
35390 Gießen, Germany
Email: Marcel.Lange@w.thm.de

Alisa Lorenz
0000-0002-8547-1391
Technical University of Central Hesse – University of Applied Sciences
THM Business School
Wiesenstr. 14,
35390 Gießen, Germany
Email: Alisa.Lorenz@w.thm.de

***Abstract*—The increased digitalization of the economy and society has triggered drastic changes in companies and confronted them with enormous challenges. The consulting industry is also strongly affected by this digital transformation. In an exploratory study, we examined digitalization's impact on management consultancies in Germany and their use of digital technologies. In our article, we provide insights into which digital technologies are considered to be important for the different phases of consulting and how digitalization affects the work of management consultants. From the perspective of such consultants, we discuss the technologies currently in use and offer an outlook on their future importance based on an online survey. Our results reveal that many business consultancies already rely heavily on digitalization and use a variety of digital technologies in all phases of the consulting process. Although business consultancies currently use well-established technologies, they remain aware of the growing importance of modern technologies for the future of consulting services.**

***Index Terms*—Digitalization, Digital technology, Consulting, Business consultancy**

I. Introduction

TODAY more than ever, society as a whole is undergoing a rapidly evolving digital transformation. Government institutions, households, enterprises, and their interactions are all changing due to the increased prevalence and rapid growth of digital technologies. Especially for enterprises, it has never been more important to be able to rely on a deep understanding of information technologies (IT) in general and of digital innovation in particular. The persistently high level of dynamism in everyday business today shows that constant changes and adaptations in response, including ones due to digitalization, will be the rule, not the exception, in the future economy. Worldwide digital networking, the automation of individual or even all business processes, and the restructuring of existing business models are just a few of the wide-ranging effects of digitalization. Indeed, the consequences of digitalization are omnipresent, as is the question of whether such changes should be viewed as positive or negative [1]–[4]. In either case, nearly all companies will have to pursue increased digital transformation, at least to some extent, in order to remain competitive in the (global) market [5].

Digitalization is partly of unprecedented importance owing to the COVID-19 pandemic, which delivered an unparalleled shock of uncertainty across all state borders and industries. Nationwide store closures, contact restrictions, and mandatory home offices forced companies to use contactless distribution channels and to facilitate remote work. Those developments drove a push toward digitalization as numerous processes in companies had to be digitalized and as companies themselves had to prove their resilience [6], [7].

One industry especially challenged by uncertainty during the COVID-19 pandemic was the consulting industry. As reported by the German Association of Management Consultancies—that is, the BDU (https://www.bdu.de/en)—the revenue growth of business consultancies collapsed in 2020 for the first time since 2010. The pandemic also dramatically altered the working methods of consultants. Guided by the motto of "New Work," the BDU reported massive contact restrictions and significant changes in workplace and working time models [8]. To be sure, business consultancies had to radically reorient themselves to meet the challenges of the COVID-19 pandemic.

At the same time, the pandemic, sometimes hailed as an accelerator of digital transformation, clarified that companies should and indeed were seizing the moment as a launchpad for not only digital transformation but also structural change. However, even before the pandemic, consulting companies faced the same challenges associated with digitalization that other companies faced as well. New competitors, new demands from customers seeking to professionalize their own digitalization [9], new requirements imposed by digitalization in providing consulting services, and the need for new skills and know-how on the part of consulting companies all confronted the classic people-oriented business of consulting with the need for changes in service provision, just as in other industries. In that light, "business as usual" was not a valid business strategy for many consulting companies even before the COVID-19 pandemic and became especially impractical due to the pandemic. Instead, the consulting industry has had to increasingly implement digital technologies in the various phases of the consulting process and, in turn, deal with emerging opportunities and innovations. In that context, the question thus arises as to what extent business consultancies are already using digital technologies. Therefore, to gain insight into how relevant digitalization already is and how much more relevant it will become in the consulting industry, we sought to survey the current and future significance of digitalization. We also wanted to investigate the general perception of digitalization among consultants in a bid to provide a basic picture of their opinions regarding digitalization. Beyond that, to gain comprehensive insight into business consultancies, we aimed to examine the current and future significance of digital technologies and trends for consultants.

To those ends, we developed a study using an online questionnaire to evaluate the status quo of the use of digital technologies in business consultancies in Germany, which we chose to examine due to our cultural background. The research question for our study was:

To what extent do business consultancies in Germany use digital technologies?

In response to that question, in this article we present selected results of our study. Following this section addressing our motivation for the study, we provide a short theoretical background before describing our study's design and our method of data collection. Afterward, we present and discuss selected results in light of our research question. The article closes with a summary of the main results and an outlook for future research in the field.

II. Theoretical Background: Business Consultancies

Business consultancies can be characterized in light of their consulting focus. In our study, we classified consultancies with reference to the BDU's classification, which divides the market for consulting into four classic fields [8]:

- Strategy consulting
- IT consulting
- Organization and process consulting
- Human resources consulting

To begin, strategy consulting is considered to be the most demanding field of consulting. Not only does it occur exclusively within the top management of companies, but the topics also concern the core of all corporate activities—that is, the corporate strategy [10]. The goal of a consultant in strategy consulting is to help the client to define long-term goals and develop a course of action to achieve the corporate strategy. Achieving that goal involves analyzing the current business situation, identifying opportunities and challenges, and developing a tailored strategy [11].

By comparison, IT consulting addresses the widest variety of consulting topics of all four classic fields of consulting. The topics range from the creation of business-critical individual software and the implementation of standard software and web-based applications to system integration and the optimization of IT architectures and infrastructure [11].

Next, organization and process consulting builds on the concepts of strategy consulting. By contrast, however, consultants work at the operational level, and contact between the client company and the consultancy usually occurs not within top management but mostly in middle and lower management [11]. Organization and process consulting deals with the optimization of organizational structures and processes within a company. Its goal is to improve the efficiency, effectiveness, and agility of the company by reviewing and, if necessary, adapting its business processes [12].

Last, human resources consulting focuses on both the managers and employees of a company. Among other activities, it involves the promotion of professional and social skills, usually facilitated in training courses [12].

No matter the field, a key factor of success for business consultancies is the consulting approach that they adopt. At base, successful consulting requires an understanding of the consulting process. In the literature, the consulting process is described in various procedure models, which differ less in their content than in the number of phases conceived as being part of the process. Barchewitz and Armbrüster [13] have described the consulting process in a three-phase model involving planning, realization, and control. Bodenstein and Herget [14], by contrast, have presented a four-phase process model involving conception, contract design, implementation, and conclusion. In our study, we followed the procedure model developed by Seifert [15], which comprises six phases:

- Acquisition
- Project preparation
- Problem analysis
- Problem-solving
- Implementation
- Post-processing

First, acquisition forms the basis of the consulting process, because in that phase a business consultancy seeks to win an order from a client [15]. A general exchange of information also occurs, after which the business consultancy submits a bid for the project order. Once the consultancy has received the order, a consultancy contract is negotiated between the parties [11].

Second, in project preparation, the project team is defined, the team's members are given access to all relevant systems, and further organizational arrangements are made [15].

Third, problem analysis focuses on gathering, deepening, and evaluating information. During that phase, the current situation is analyzed, and a formulation to meet the project's objective is finalized [11].

Fourth, problem-solving is the core phase of a consulting project [11]. Therein, a strategy for realizing a solution to the problem is presented. To that purpose, different alternative solutions are designed, evaluated, and presented to the client, who subsequently selects one of them to pursue [15].

Fifth, during implementation, the selected solution is implemented. The process is carefully planned to ensure successful implementation, and, afterward, the results are reviewed, and, if necessary, then the solution is optimized [14].

Sixth and last, post-processing considers both the client and the consultancy. On the client's side, the phase involves the conclusion of the project, including the achievement of the project's objectives. On the consultancy's side, it entails the preparation of documentation, assessments, and results for reuse [15].

III. Data Collection

Our research question was designed to afford access to initial insights into how consulting firms view and use digital technologies. To gain such insights, we adopted an exploratory approach in our study, which we conceive as being a starting point for more in-depth research in the future. For that reason, we make no claim regarding the representativeness of participants in the study.

Questionnaire Design

Overall, our questionnaire included 20 questions, divided into seven blocks of questions:

- General information about the participants
- Importance of digitalization
- Importance of digital technologies
- Degree of digitalization
- Importance of the business model
- Use of digital technologies
- Perception of digitalization and future trends

General information about the participants: In the first block of questions, participants were asked four fact-focused questions as a means to later categorize them in data analysis. Question 1 inquired into the number of employees in the participants' companies, the responses to which were used to classify the companies into micro, small, medium, and large companies. Question 2 asked participants about the area of consulting in which they were most active. Last, Questions 3 and 4 addressed the participants' professional experience by inquiring into the number of years spent in the profession and the number of client and consulting projects undertaken.

Importance of digitalization: The second block of questions, containing Questions 5–8, addressed the current and future importance of digitalization, along with its importance during the COVID-19 pandemic. To that end, participants were asked to indicate digitalization's importance for themselves as consultants in Question 5 and for their company in Question 7. In between, Question 6 asked for an assessment of digitalization's expected importance in the next five years from the participant's perspective. Last, Question 8 inquired into how digitalization's importance has changed from the company's perspective since the pandemic.

Importance of digital technologies: In the third block of questions, Questions 9 and 12 sought to determine the importance of digital technologies and trends in business consulting. To that purpose, a list of 14 digital technologies was created with reference to the literature. To ensure consistency in understanding, potentially unfamiliar technologies were briefly explained. Meanwhile, Questions 10 and 11 asked participants about the importance of those technologies during the COVID-19 pandemic. Those questions allowed us to determine both the current state of digital technologies in business consultancies and the most significant technologies for the consultants during the COVID-19 pandemic.

Degree of digitalization: In the fourth block of questions, Question 13 asked participants to select one of four statements that best describes the current level of digitalization in their respective companies.

Importance of the business model: The fifth block of questions consisted of Question 14, which asked about the COVID-19 pandemic's impact on the company's business model.

Use of digital technologies: To gain more granular insight, the first question of the sixth block of questions, Question 15, asked the consultants to rate their current use of digital technologies during the different phases of the consulting process. Afterward, Question 16 asked them to rate their expected use of digital technologies in the next five years, and Question 17 asked them to select the technologies that they use in each phase of the consulting process.

Perception of digitalization and future trends: The intention of the seventh and final block of questions was to determine how the participants perceived digitalization at present and in the future. To that end, Questions 18 and 19 asked participants to evaluate specific opportunities by responding to different statements. The questions were intended to capture the participants' opinions on digital technologies. Last, Question 20 inquired into the participant's personal attitude toward digitalization.

Data Collection

As a result of several pretests with various researchers of the Technical University of Central Hesse and different practitioners, the questionnaire was improved. The general aim of the pretests was to assess the questionnaire's instructions as well as the individual questions for comprehensibility and errors.

Next, mostly using email, we invited consultants to participate in our study. To that end, we contacted all business consultancies that were members of the BDU at the time of data collection, and their responses were our primary source for contact information. The emails were sent between January 15 and February 15, 2023. We also shared the link to the online questionnaire on business platforms such as LinkedIn (www.linkedin.com) and XING (www.xing.com) and with personal contacts in our business networks.

When the survey period ended, the online questionnaire had been completed 291 times. Of those 291 questionnaires, 187 had been completed in full. Before data analysis, those 187 questionnaires were checked for plausibility, with special attention to whether any pattern in the answers might suggest that the participant had only clicked through the questionnaire at random. As a result, we had to exclude only one data set, meaning that 186 data sets were analyzed for the results presented in the following section.

IV. Selected Results

Participants' General Characteristics

To be able to differentiate responses along the lines of company size, the business consultancies were grouped according to the number of employees. Table I provides an overview of the respective company sizes.

Table I.
Participant structure by number of employees (n=186)

Number of employees	Absolute frequency	Relative frequency
1–10	47	25.3%
11–49	30	16.1%
50–249	19	10.2%
>250	90	48.4%

Table I shows that 47 consultants from micro-enterprises and 30 from small enterprises participated in the survey. The smallest group of participants, totaling 19, was represented by medium-sized companies, whereas the largest proportion of participants, totaling 90, came from large companies.

The distribution of participants across the different fields of consulting (see Section 2) was highly heterogeneous. Because Question 2 allowed multiple answers, the 186 participants gave a total of 245 answers. The most represented field was organization and process consulting, with 78 responses, followed by IT consulting with 70, strategy consulting with 44, and human resources consulting with 36. Added to that, 17 participants selected the answer option “Other.”

Concerning the experience of the participants in terms of their years spent working as consultants, Table II shows that 105 participants had up to 10 years of work experience and that 81 had at least 10 years of work experience.

Table II.
Participants by years of work experience (n=186)

Years of work experience	Absolute frequency	Relative frequency
<1	5	2.7%
1–5	66	35.5%
6–10	34	18.3%
11–15	20	10.8%
16–20	20	10.8%
21–25	18	9.7%
26–30	13	7.0%
31–35	4	2.2%
36–40	6	3.2%
> 40	0	0%

The participants' professional experience with consulting projects was also queried. Whereas only 11 consultants had previously worked on 1–3 projects, 43 had been involved in 4–9 projects, 37 in 10–19 projects, 20 in 20–29 projects, and 17 in 30–39 projects. In the largest group, 58 participants had been involved in more than 40 projects.

Digitalization: General Aspects

The participants were also asked to assess the current role of digitalization in their day-to-day work. For a detailed look at their responses, Figure 1 shows how participants with up to 10 years of professional experience responded versus participants with more than 10 years of professional experience. On the one hand, participants with up to 10 years of professional experience attributed “medium significance” and “high significance” to digitalization in their day-to-day work in nearly equal measure, at rates of 44.8% and 48.6% respectively. By contrast, only 6.7% participants selected “low significance.” On the other hand, 63.0% of participants with more than 10 years of professional experience characterized digitalization as having “high significance” in their daily work, whereas 30.9% of them selected “medium significance” and another 6.2% selected “low significance.” Remarkably, none of the participants selected “no significance” to answer the question. It is therefore clear that digitalization was perceived as playing a greater role in the day-to-day work of consultants with more than 10 years of professional experience than for ones with up to 10 years of such experience.

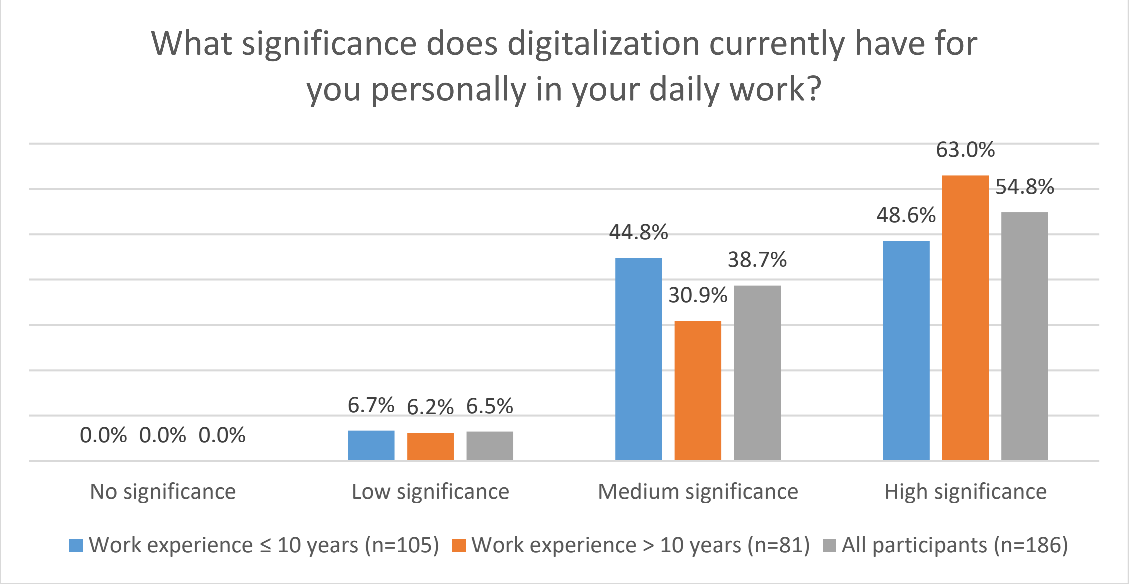


Fig 1. Significance of digitalization for consultants according to work experience (n=186; relative frequency)

Regarding digitalization in general, the participants were additionally asked to assess the level of digitalization in their consultancies by choosing one of the following levels:

- **Level 1:** We predominantly rely on consulting processes in which our consultants work together with the customer on-site. Technologies such as chat, video-conferencing, and other digital collaboration tools are rarely used in projects.
- **Level 2:** We carry out projects in which our consultants and customers work together at separate locations. However, most of our projects are based on on-site, face-to-face interaction.
- **Level 3:** Digital technologies are an integral part of our business model. We specifically manage the personal deployment of consultants on-site and no longer include it in every project.
- **Level 4:** Our business model is based predominantly on digital technologies. Consultants work on-site with clients only in particularly critical phases and in regard to particularly complex problems.

Given those four statements, only 22 of the 186 participants selected Level 4 to characterize digitalization at their companies. By contrast, 83 selected Level 3, 68 selected Level 2, and, least frequently, 13 selected Level 1.

To present the level of digitalization in greater detail, Figure 2 depicts the level of digitalization of the business consultancies by company size. As shown, 18.9% of large companies were characterized as having Level 4 digitalization, followed by 10.5% of medium-sized companies. Micro-enterprises accounted for the largest share of Level 1 digitalization, at 12.8%, while small companies had the second-largest share, at 10.0%. Those results clearly show that larger companies seem to have a higher level of digitalization than smaller companies.

Turning to the perception of digitalization, we asked participants whether they perceived digitalization primarily as a threat to or as an opportunity for their companies. Figure 3 provides a breakdown of their answers based on company size. The top bar of the graph shows the overall results, which indicates that 113 participants perceived digitalization in their companies “clearly as an opportunity” and 61 as “more like an opportunity.” The remaining 12 participants perceived digitalization in their companies as both an opportunity and a threat (i.e., “opportunity/threat”). Notably, none of the participants selected the answer options “more like a threat” or “clearly as a threat.” The other four bars in the graph show the evaluation by company size. Of the 90 participants from large companies, 56 perceived digitalization at their companies “clearly as an opportunity,” 29 as “more like an opportunity,” and 5 as “parts/parts.” The picture sharpens for medium-sized companies; of those 19 consultants, 17 perceived digitalization in their companies “clearly as an opportunity” and 2 as “more like an opportunity.” Thus, medium-sized companies had the highest proportion of participants who selected “clearly as an opportunity.” The 30 participants from small companies also only selected two answer options; 19 selected “clearly as an opportunity,” while 11 selected “more like an opportunity.” By contrast, of the 47 participants in micro-enterprises, 21 chose “clearly as an opportunity,” 19 chose “more like an opportunity,” and 7 chose “parts/parts.”

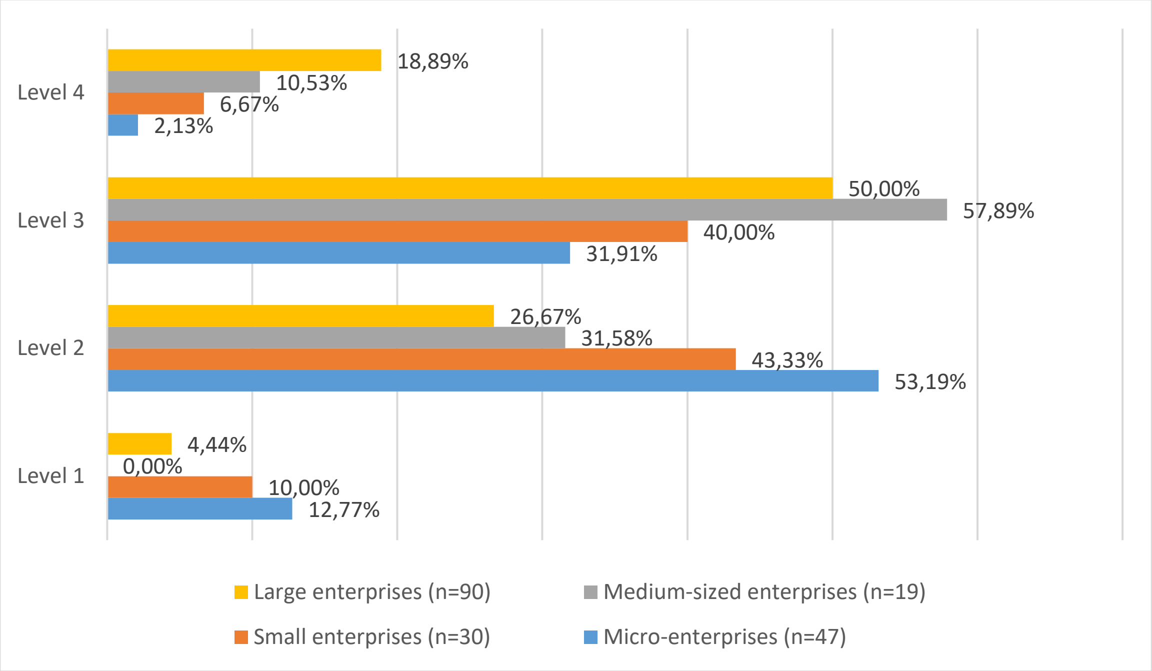


Fig 2. Level of digitalization by company size (relative frequency)

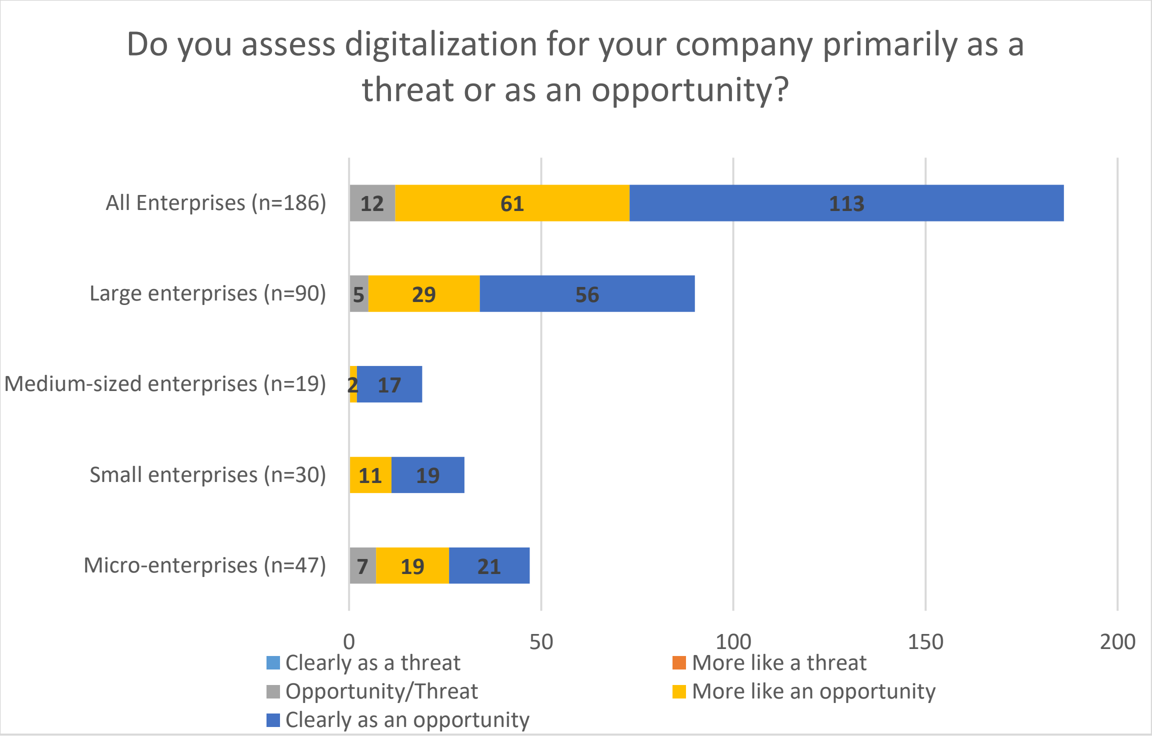


Fig 3. Perception of digitalization (n=186; absolute frequency)

Use of Digital Technologies

This section presents the results of our analysis of the data from questions concerning the use of digital technologies in business consultancies.

To begin, focusing on the current and future significance of digital technologies in business consultancies, participants were asked to assess the current significance of 14 specific technologies. They were next asked to assess the importance of those technologies for their consultancies in the next five years. To evaluate those data, the verbalized answers were coded and recorded as arithmetic mean values. The following coding was chosen:

- 1 = *no importance*
- 2 = *low importance*
- 3 = *medium importance*
- 4 = *great importance*

Table III provides an overview of the results. Because not every participant assessed every technology, the table also provides the number of participants who assessed the particular technology.

TABLE III.
SIGNIFICANCE OF DIGITAL TECHNOLOGIES (N=186; MULTIPLE ANSWERS POSSIBLE)

Digital technology	Current significance (arithmetic mean)	Future significance (arithmetic mean)
Knowledge management systems (n=180)	2.96	3.38
Virtual marketplace for consultants and customers (n=180)	2.96	2.99
Social media (n=180)	2.18	3.29
Self-service consulting (n=175)	2.05	2.98
Open community and expert platforms (n=171)	2.32	3.01
Mobile computing (n=171)	3.54	3.81
Artificial intelligence (n=182)	2.66	3.62
Document management systems (n=181)	2.99	3.34
Data/process mining (n=171)	2.52	3.47
Crowdsourced consulting (n=170)	2.05	2.96
Cloud computing (n=180)	3.29	3.69
Chats (n=183)	3.22	3.31
Big data analytics (n=176)	2.52	3.55
Audio/video-conferencing (n=186)	3.72	3.75

As Table III shows, audio/video-conferencing was viewed as being the most important digital technology, with a mean value of 3.72 out of 4.00. In second place was mobile computing, with a mean of 3.54, followed by cloud computing, with a mean of 3.29. The importance of the mean values becomes particularly clear when looking at the technologies in the lower ranks. Social media came in third lowest place, with a mean value of 2.18, followed by crowdsourced consulting and self-service consulting as lowest in ranking, each with a mean of 2.05.

The difference between the arithmetic means of “Current significance” and “Future significance” indicates which digital technologies may become the focus of consulting firms in the next five years. The third-largest difference was 0.96 for artificial intelligence technology, closely followed by social media, with a difference of 1.01. The largest difference, 1.03, was with big data analytics.

Next, participants were asked to indicate the current and anticipated future use of digital technologies in their consulting process. Again, the arithmetic mean was used for evaluation. To that end, the verbalized answers were coded as follows:

- 1 = *no use*
- 2 = *very little use*
- 3 = *low use*
- 4 = *medium use*
- 5 = *high use*
- 6 = *very high use*

Figure 4 shows the participants’ evaluation of the current and anticipated future use of the technologies in the consulting process undertaken by their respective business consultancies. The figure readily clarifies that the anticipated future use of digital technologies in all phases of the consulting process was rated higher than the current use.

To gain a comprehensive view of the current use of digital technologies in the consulting process, participants had the opportunity to assign the 14 listed technologies to the individual phases of the process and could select multiple response options. Table IV provides an overview of the results. When the numbers of the various technologies per phase were totaled, digital technologies emerged as being used most frequently in problem analysis, followed by problem-solving and project preparation. Implementation ranked fourth, followed by acquisition. Last, post-processing was reported to involve the fewest digital technologies. Altogether, the results suggest that the diversity of digital technologies used is most often greatest in the middle phases of the consulting process.

Regarding the use of digital technologies in the different fields of consulting, participants in IT consulting selected different digital technologies most frequently in all six phases of the consulting process, closely followed by participants in organization and process consulting. Participants in strategy consulting reported using nearly half as many different digital technologies as in IT consulting or organization and process consulting. Meanwhile, participants in human resources consulting reported using the fewest different digital technologies.

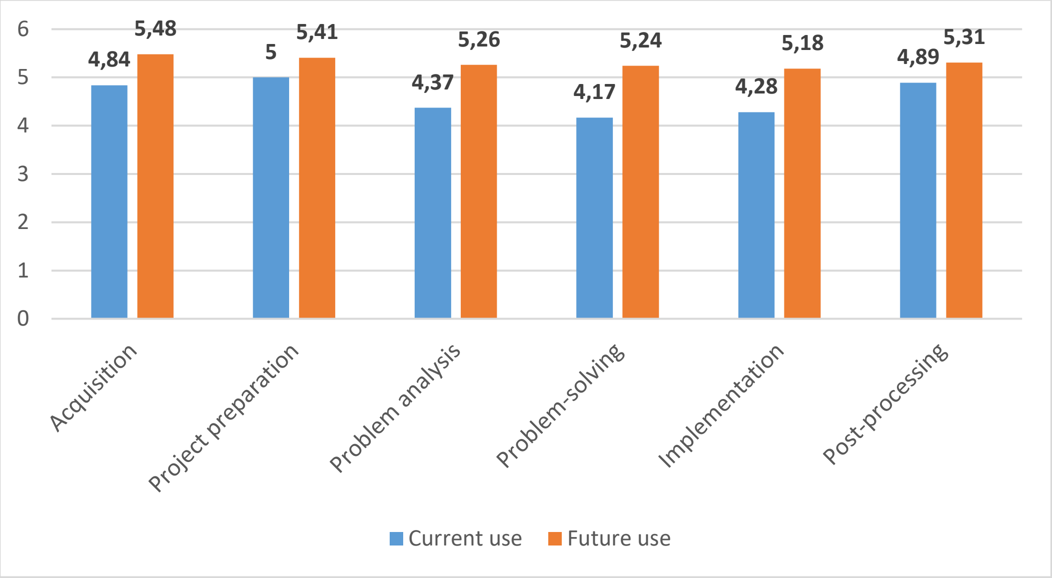


Fig 4. Use of digital technologies (n=182; arithmetic mean)

TABLE IV.
DIGITAL TECHNOLOGIES PER PHASE OF THE CONSULTING PROCESS (N=186; ABSOLUTE FREQUENCY, MULTIPLE ANSWERS POSSIBLE)

	Acquisition	**Project preparation**	**Problem analysis**	**Problem-solving**	**Implementation**	**Post-processing**
Knowledge management systems	66	133	121	125	95	100
Virtual marketplace for consultants and customers	92	36	35	39	27	18
Social media	144	24	21	23	18	20
Self-service consulting	11	29	92	37	23	13
Open community and expert platforms	68	37	41	62	29	16
Mobile computing	123	139	143	140	138	131
Artificial intelligence	19	28	77	78	56	18
Document management systems	92	124	115	117	113	122
Data/process mining	16	32	102	79	35	17
Crowdsourced consulting	17	31	42	62	30	29
Cloud computing	72	118	120	118	116	100
Chats	90	136	126	124	116	112
Big data analytics	20	26	111	67	30	15
Audio/video-conferencing	106	166	143	135	128	143

Last, Table V provides an overview of the participants' opinions on five statements regarding the use of digital technologies. For each statement, the arithmetic mean was again calculated, and the verbalized scale was coded as follows:

- 1 = *strongly disagree*
- 2 = *somewhat disagree*
- 3 = *parts/parts*
- 4 = *somewhat agree*
- 5 = *strongly agree*

Table V shows that the statement "By using digital technologies, the work–life balance in the consulting industry is improved" was only partly agreed with, with a mean value of 3.71. By contrast, the statement "By using digital technologies, there is an increase in the efficiency of consulting" had the highest level of agreement of the five statements, with an mean of 4.3. The lowest level of agreement, with a mean of 2.82, was achieved by the statement "By using digital technologies, the quality of the result delivered to the customer is improved." The statement, "By using digital technologies, new customers and markets can be addressed," was agreed to by significantly more participants, with a mean value of 4.25. The rating of the remaining statement, "By using digital technologies, a differentiation from competitors is made possible," had a mean value of 3.82. Based on the five mean scores, it can be concluded that the participants were more likely to agree than disagree with the statements.

TABLE V.
OPINIONS ON THE USE OF DIGITAL TECHNOLOGIES (N=186; ABSOLUTE FREQUENCY AND ARITHMETIC MEAN)

By using digital technologies...						
Statement	**Strongly agree**	**Somewhat agree**	**Parts/parts**	**Somewhat disagree**	**Strongly disagree**	**Arithmetic mean**
...a differentiation from competitors is made possible. (n=181)	64	56	31	24	6	3.82
...new customers and markets can be addressed. (n=182)	89	60	24	8	1	4.25
...the quality of the result delivered to the customer is improved. (n=186)	48	72	54	10	2	2.82
...there is an increase in the efficiency of consulting. (n=186)	85	77	19	4	1	4.3
...the work–life balance in the consulting industry is improved. (n=184)	49	65	42	23	5	3.71

V.Conclusion, Limitations, and Directions for Future Research

The results of our analysis suggest that consultants currently consider digitalization as to be of medium to high importance in their business consultancies. Taking into account their work experience, digitalization seems slightly more important for more experienced consultants than for less experienced ones. In terms of the four classic fields of consulting that we considered in our study, digitalization currently seems to be most important in strategy consulting and human resources consulting. No matter the field—indeed, overall—digitalization is not perceived as being exclusively a threat (vs. an opportunity). In fact, for 60.1% of consultants, digitalization is clearly perceived as an opportunity and for 32.8% is perceived as being at least somewhat of an opportunity. Therefore, digitalization is seen in an almost entirely positive light by consultants. On top of that, consultants perceive an opportunity to increase efficiency in the consulting process by using digital technologies and believe that the technologies will allow new markets and customers to be reached.

From the consultants' perspective, traditional technologies such as audio- and video-conferencing, mobile computing, and cloud computing are currently the most important for their business consultancies. By contrast, analytical tools are used only sporadically but increasingly more often in larger companies. Beyond that, technologies such as self-service consulting, virtual marketplaces for customers and consultants, and crowdsourced consulting are rarely used. According to the participants, established technologies will continue to play the most important role in their business consultancies in the next five years. Nevertheless, they also expect the use of analytical tools and social media to increase in importance. From their perspective, digital technologies in general will play an important part in developing future business consultancies, and their use stands to have a major impact on the efficient delivery of good consulting services in the future.

Within the concrete phases of the consulting process, the fewest technologies are used during acquisition and post-processing. By contrast, technologies are increasingly used from the phrases of preparation to implementation. From the consultants' perspective, the use of technologies in all phases of the consulting process will increase in the future; even so, they expect priority use after acquisition and before post-processing. Within the individual phases, established technologies are preferred and used the most frequently. Social media, by comparison, is primarily used in acquisition, and analytical tools are used especially during problem analysis. However, a clear change in the technologies used within the phases was not observed despite the existence of social media and virtual marketplaces.

In sum, it can be stated that business consultancies clearly see the benefits of digitalization and of using digital technologies. Nevertheless, they continue to rely on more established technologies. In response to that tendency, future research needs to produce a more detailed, diversified view of the use of different digital technologies. That need is especially evident considering the potential impact of widely discussed generative AI tools such as ChatGPT. On that count, qualitative studies should be conducted in individual fields of consulting and with more specific consideration of company size, especially the size of the companies using the consulting service, to further pinpoint the importance of digital technologies for the consulting process in general and for its respective phases. Future research should also analyze the use of digital technologies with reference to the different types of consulting projects, including logistics projects, IT/digitalization projects, and human resources projects, in order to identify and highlight differences. Last, we recommend investigating barriers to and challenges in using digital technologies in business consultancies and how they can be minimized.

As most empirical studies, ours was limited in multiple ways. Due to our approach, our results possess limited statistical generalizability. However, the method applied allowed us to identify important details and obtain initial insights into the experience of business consultants, which was the chief focus of our study. Another limitation was that the participants' origins were limited to German business consultancies. Because German-specific trends could have influenced the results, the results reflect the situation in one country only.

References

[1] C. Leyh, T. Schäffer, K. Bley, and S. Forstenhäusler, "Assessing the IT and Software Landscapes of Industry 4.0-Enterprises: The Maturity Model SIMMI 4.0," in *Information Technology for Management: New Ideas and Real Solutions*, E. Ziemba, Ed., Cham: Springer International Publishing, 2017, pp. 103–119. doi: 10.1007/978-3-319-53076-5_6

[2] M. Pagani, "Digital Business Strategy and Value Creation: Framing the Dynamic Cycle of Control Points," *MIS Quarterly*, vol. 37, no. 2, pp. 617–632, 2013, doi: 10.25300/MISQ/2013/37.2.13

[3] S. Mathrani, A. Mathrani, and D. Viehland, "Using enterprise systems to realize digital business strategies," *Journal of Enterprise Information Management*, vol. 26, no. 4, pp. 363–386, 2013, doi: 10.1108/JEIM-01-2012-0003

[4] C. Leyh, K. Bley, and M. Ott, "Chancen und Risiken der Digitalisierung – Befragungen ausgewählter KMU," in *Arbeit 4.0 – Digitalisierung, IT und Arbeit*, J. Hofmann, Ed., Wiesbaden: Springer Fachmedien, 2018, pp. 29–51. doi: 10.1007/978-3-658-21359-6_3

[5] K. Bley, C. Leyh, and T. Schäffer, "Digitization of German Enterprises in the Production Sector – Do They Know How 'Digitized' They Are?" in *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS 2016)*, 2016.

[6] C. Leyh, K. Köppel, S. Neuschl, and M. Pentrack, "Critical Success Factors for Digitalization Projects," in *Proceedings of the 16th Conference on Computer Science and Intelligence Systems*, 2021, pp. 427–436. doi: 10.15439/2021F122

[7] K. Luban, R. Hänggi, and G. Bernard, "Einleitung," in *Erfolgreiche Unternehmensführung durch Resilienzmanagement*, K. Luban and R. Hänggi, Eds., Berlin, Heidelberg: Springer, 2022, pp. 1–14. doi: 10.1007/978-3-662-64023-4_1

[8] BDU e.V., "Facts & Figures zum Beratungsmarkt 2021," BUD - German Association of Management Consultancies, Bonn, 2021. [Online]. Available: https://www.bdu.de/media/355573/facts-figures-vorjahr.pdf

[9] J. M. Kawohl, R. Waubke, and F. Höselbarth, "Digitale Transformation von Unternehmensberatungen – wie Consulting sich verändern wird," 2017. [Online]. Available: http://www.peoplebrand.de/img/studie_transformation-von-unternehmensberatungen.pdf

[10] A.-W. Scheer and A. Köppen, Eds., *Consulting*. Heidelberg: Springer, 2000. doi: 10.1007/978-3-642-98079-4

[11] D. Lippold, *Grundlagen der Unternehmensberatung*. Wiesbaden: Springer Fachmedien, 2016. doi: 10.1007/978-3-658-12882-1

[12] R. Bodenstein, "Unternehmensberatung – Typologie, Felder und Rollen," in *Exzellenz in der Unternehmensberatung*, R. Bodenstein, I. A. Ennsfellner, and J. Herget, Eds., Wiesbaden: Springer Fachmedien, 2022, pp. 19–39. doi: 10.1007/978-3-658-34589-1_2

[13] C. Barchewitz and T. Armbrüster, *Unternehmensberatung - Marktmechanismen, Marketing, Auftragsakquisition*. Wiesbaden: Deutscher Universitätsverlag, 2004. doi: 10.1007/978-3-322-81779-2

[14] R. Bodenstein and J. Herget, *Consulting Governance: Strukturen, Prozesse und Regeln für erfolgreiche Beratungsprojekte*. Heidelberg: Springer, 2022. doi: 10.1007/978-3-662-65299-2

[15] H. Seifert, "Virtualisierung von Beratungsleistungen: Grundlagen der digitalen Transformation in der Unternehmensberatung," Technische Universität Ilmenau, Ilmenau, 2017.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 161–169
DOI: 10.15439/2023F6002
ISSN 2300-5963 ACSIS, Vol. 35

A Framework for Assessing the Sustainability Impact of Intelligent Transport Systems in the Smart City Context

Alisa Lorenz
0000-0002-8547-1391
Technische Hochschule Mittelhessen – THM Business School
Wiesenstr.14,
35390 Giessen, Germany,
University of Cologne, Albertus-Magnus-Platz, 50923 Cologne, Germany
Email: alisa.lorenz@w.thm.de

Nils Madeja
0000-0001-9558-2004
Technische Hochschule Mittelhessen - THM Business School
Wiesenstr.14,
35390 Giessen
Email: nils.madeja@w.thm.de

Christian Leyh
0000-0003-0535-0336
Technische Hochschule Mittelhessen - THM Business School
Wiesenstr.14,
35390 Giessen
Email: christian.leyh@w.thm.de

***Abstract*—In their transition to smart cities, an increasing number of cities are pursuing strategies to improve efficiency of transport. One strategy is to achieve smart mobility, for which cities implement intelligent transport systems (ITS). Simultaneously, municipalities recognize their responsibility for creating a sustainable environment for citizens in the face of challenges like overpopulation, land shortage, and climate change. Interestingly, many ITS initiatives mainly focus on technical outcomes and overlook their impact on sustainability despite its key benefit for smart mobility. To fill this gap, we develop a framework for assessing the sustainability impact of ITS initiatives in this paper. We analyze the Sustainable Development Goals (SDGs) defined by the United Nations and relate them to various concepts of ITS to derive our framework. Thereby, our work bridges two fundamental perspectives for further research and supports decision-makers in choosing ITS initiatives that contribute to both smart mobility and sustainability.**

***Index Terms*—smart city, smart mobility, smart traffic management, intelligent transport systems, sustainability**

I. Introduction

SMART mobility can be viewed as a comprehensive and collective term encompassing various data-driven concepts for maneuvering individuals, groups of people or objects in one or multiple geographies and influences our present and future [1]. Considering the complex challenges of the current decade, such as overpopulation, demographic change, globalization, space shortage, and dense traffic, cities aim to stay attractive to their citizens and provide a livable environment [2], [3]. With over 50% of the global population living in urban areas, their citizens can especially profit from the opportunities of smart traffic management and the management of high traffic volume in congested environments [4]. Simultaneously, the environmental and social challenges driven by climate change raise the need for municipalities to take responsibility and counteract the negative influences on their citizens. Consequently, more cities aim to use the advances of digitalization to create value for smart and sustainable mobility of citizens [5]–[8]. Some researchers even point out that cities cannot become smart without being sustainable, making sustainability an important factor in smart city projects [9].

Intelligent transport systems (ITS) provide a set of technical applications and aim to provide innovative services for different modes of transport and traffic management [1], [10], [11]. They empower citizens to make better decisions regarding their mobility and enable safer and better-coordinated transport networks. Since road traffic is responsible for about 65% of the CO2 emissions in cities and is likely to increase in the future, ITS promise to mitigate the negative effects of traffic on the environment [12]. However, while sustainability is a key factor in smart mobility initiatives, many ITS projects are still mainly focused on technical criteria and measures and do not seem to analyze their impact on sustainability [13]–[15].

Therefore, we target the intersection of sustainability and ITS in this paper. We analyze the Sustainable Development Goals (SDGs) defined by the United Nations (UN) to determine which goals, targets, and indicators have implications for the development of ITS in the context of smart mobility towards a sustainable smart city. Specifically, we explore different perspectives on ITS and relate them to the SDGs to answer the following question:

> Which Sustainable Development Goals, targets, and indicators are relevant for assessing the sustainability of intelligent transport systems?

To answer this question, we review the relevant literature and combine it with international agreements and resolutions to derive a framework for measuring the effectiveness of ITS strategies on sustainability. Therefore, the paper is structured as follows. First, we define the term intelligent transport systems and set it into the context of smart cities and sustainability. We then develop our framework based on the literature and describe the implications for further research and practice.

With our research, we contribute to the research fields of sustainable smart cities and mobility while also providing practical implications for sustainable ITS. With our findings, we want to inspire municipal decision-makers and technical leaders to consider sustainability factors to build data-driven solutions that leave a positive impact on society and nature. In fact, we are currently facing this specific challenge in a project for data-driven traffic management funded by the German Federal Ministry for Digital and Transport. Hence, we want to share our approach to support other cities considering or planning ITS projects.

II. Related Literature

A. Smart Cities and Smart Mobility

With the trend of urbanization and population growth, cities become increasingly populated while space and resources are limited. Urban areas will face challenges in meeting the needs of their growing number of citizens in many sectors, such as housing, transportation, energy systems, education, and healthcare, which leads to the need for sustainable development [16]. The concept of smart cities has been evolving in the last decade and aims to enhance quality of life in urban areas by using the opportunities of information and communication technologies (ICT), hardware, algorithms, and data to create a positive impact on life in cities [7].

Smart cities are characterized by the six areas *smart economy, smart people, smart governance, smart environment, smart living,* and *smart mobility,* which are all interlinked [17]. Smart mobility is an especially relevant building block of smart cities. The improvement of mobility with technical advances can save resources, increase efficiency, and provide accessibility [4]. More specifically, smart mobility is defined as "a set of coordinated actions addressed at improving the efficiency, the effectiveness and the environmental sustainability in cities," which is characterized by transport and the use of information and communication technology [6]. It further consists of local accessibility, (inter-)national accessibility, availability of ICT infrastructure, and sustainable, innovative and safe transport systems [17]. Smart mobility has direct implications for fulfilling the Sustainable Development Goals defined by the United Nations and will contribute to the future of city planning and logistics [18]. However, research still shows gaps regarding the consideration of potential sustainability factors that directly affect citizens, e.g., air quality [19]. The field of smart mobility therefore leaves high potential for future research for more sustainable cities.

B. Smart Traffic Management and Intelligent Transport Systems

A variety of terms is employed to denote the technical applications, data-driven services, and conceptual advances for data-driven traffic management; the most frequently used terms are intelligent transport systems, smart traffic management, transport/travel demand management, and smart mobility management. Though not completely congruent, these terms exhibit a high semantic overlap and are used interchangeably. In this paper, we consistently use the term intelligent transport systems (ITS) since it has been researched for more than two decades now [20] and is used by the United Nations and European Parliament [11].

ITS are defined as all technical solutions and construction concepts related to traffic [1]. The United Nations Economic Commission for Europe (UNECE) further describes them as "a set of procedures, systems and devices that enable (a) improvements in the mobility of people and transportation of passengers and goods, through the collection, communication, processing and distribution of information and (b) the acquisition of feedback on experience and a quantification of the results gathered" [21]. The European Parliament defines ITS in a slightly more generalized manner as communication systems to provide services related to different modes of transport and traffic management which supports a safer, more coordinated, and smarter use of transport networks for users [11]. All definitions, however, have the target of technology-based and data-driven traffic management in common that aims to improve mobility. ITS further consist of various tools based on information and communication technology and support the concept of smart mobility [10]. Examples for specific applications are traffic light control systems or analytical tools that influence transport management.

In addition to various definitions, several perspectives on ITS focus on different means and needs. In Fig. 1, we summarize four of the most prominent perspectives and definitions and in the following describe them in more detail for a broad understanding of the concept.

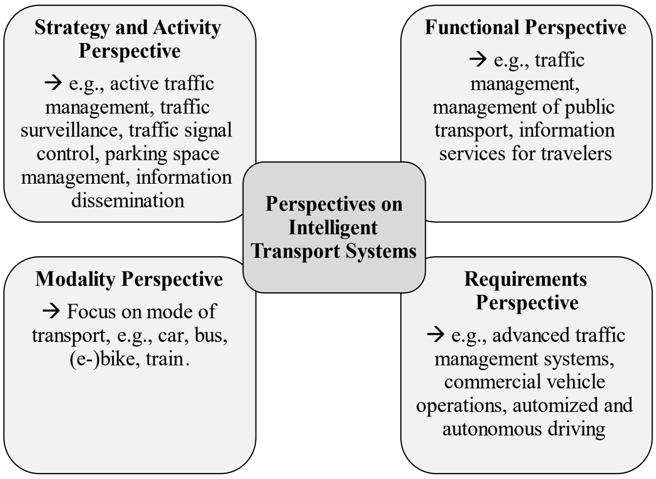

Strategy and Activity Perspective
→ e.g., active traffic management, traffic surveillance, traffic signal control, parking space management, information dissemination
Functional Perspective
→ e.g., traffic management, management of public transport, information services for travelers
Perspectives on Intelligent Transport Systems
Modality Perspective
→ Focus on mode of transport, e.g., car, bus, (e-)bike, train.
Requirements Perspective
→ e.g., advanced traffic management systems, commercial vehicle operations, automized and autonomous driving

Fig. 1. Perspectives on intelligent transport systems, own illustration derived from [1], [10], [21], [22], [23]

The strategy and activity perspective on ITS [22], [23] is defined by the U.S. Department of Transportation and provides the broadest and most granular perspective. It is focused on ITS strategies and details them into activities. Depending on the publication, 16 to 26 related activities are defined. The strategies and sample activities are:

- Traffic management and operations (e.g., traffic surveillance, traffic signal control, speed and intersection warning systems, bicycle and pedestrian crossing enhancements),
- Road weather management operations (e.g., road weather information systems, winter roadway operations),
- Maintenance and construction management (e.g., coordination activities for construction management, work zone management),
- Incident and energy management (e.g., emergency management, emergency vehicle routing), and
- Public transportation management (e.g., electronic fare collection and integration, multimodal travel connections, transit surveillance).

The functional perspective [10] on ITS describes functions of ITS like management or information provision, consisting of:

- Traffic management,
- Management of public transport,
- Management of cargo transport and fleet of vehicles,
- Traffic safety management and monitoring systems for violation of regulations,
- Management of road incidents and emergency services,
- Information services for travelers and electronic payment services, and
- Electronic systems for collecting tolls for road use.

Some of these functions also overlap with the activities from the strategy and activity perspective, which shows that there is no clear separation of the different perspectives. According to this definition, ITS operation is particularly focused on information collection from different systems, processing of this information, and the provision of related recommendations.

The requirements perspective [1] focuses on requirements profiles and specific technical systems. Again, there are overlaps to both the strategic and functional perspectives. The related systems are:

- Advanced Traffic Management (ATMS),
- Advanced Traveler Information (ATIS),
- Advanced Vehicle Control (AVCS),
- Commercial Vehicle Operations (CVO),
- Advanced Public Transportation (APTS),
- Rural Transportation (ARTS),
- Automized and Autonomous Driving,
- Intelligent Traffic Data (Smart Traffic), and
- Vehicle Networks (Connected Vehicles).

These systems have a direct impact on activities such as emergency management, information management, or innovation management.

While the previous perspectives provide a more generic view, ITS can also be categorized according to the modes of transport they address. That leads to the modality perspective [1], comprising:

- (Motor) car traffic,
- Public transport (bus, train, city train, subway),
- (e-)Bike,
- Motorcycle,
- Plane, or
- Vessel.

The modal split is especially important in relation to sustainable solutions. However, in contrast to the previous ones, this perspective does not present activities or strategies. In summary, every perspective provides a slightly different view on ITS while they have a focus on more efficient, safe, and sustainable intelligent traffic management and the consideration of the modal split in common.

C. Sustainable Development Goals

In 2015 the United Nations formally acknowledged the need for transformative change towards sustainability and defined 17 goals for sustainable development (SDG). The resulting resolution defines sustainability as "meeting the needs of the present without compromising the ability of future generations to meet their own needs" by considering environmental concerns, social aspects, and economic development [16]. Therefore, sustainability encourages growth and technological progress by focusing on people's needs while also making sure that choices in the present do not exhaust resources needed in the future.

All 17 goals are interdependent, and they are defined by 169 targets overall. The progress for achieving these targets can be tracked by 231 unique indicators (ibid). The goals are set to be achieved by 2030 and have universal relevance, as they have been aligned between all 191 UN member nations and thus represent a collective understanding of sustainability.

Many of these goals build an important foundation for the progress towards smart mobility. The application of traffic-related technology and smart services in cities reflects the original idea of smart mobility. However, smart mobility also calls for a balance of technology with the needs of citizens that are reflected by the sustainability factors [24]. Recent research shows that there is high potential for analyzing the contribution of smart cities to achieve sustainable development [25] and some researchers even go as far as to point out that cities cannot become smart without being sustainable, making sustainability an important factor in smart city projects [26]. Further, researchers have covered ICT adoption for sustainable development in the industry context, highlighting the link between ICT and sustainability [27].

The presented literature shows that sustainability is important for smart mobility in smart cities. Therefore, we aim to provide a framework of the important SDGs that relate to ITS in order to be able to assess related projects and support decision-making in smart mobility initiatives.

III. Methods

In the previous chapter, we showed the need for ITS that are not only smart but also sustainable. However, to our knowledge, there is no framework that would help researchers and decision-makers assess whether and how smart traffic management measures contribute to sustainability. Therefore, we dedicate this research to analyzing the various perspectives on ITS and bringing them into the context of the SDGs. To answer our research question, we combine the Sustainable Development Goals, their targets, and related indicators in a framework that shows the sustainability factors ITS can influence. Based on the relevant literature on sustainability, smart mobility, and ITS, we conduct conceptual development in our study. Besides scientific literature, we also include international agreements and recommendations into the development for several reasons: First, UN resolutions can be seen as universally relevant because 191 states from the world community have committed to their achievement. Recommendations by the UNECE are similarly relevant and, while not legally binding, provide a more detailed view than the resolutions. Second, we aim to ground our framework on existing work while developing a new concept through the combination of several perspectives. A literature exploration can provide different views and serve as a foundation for developing a unified understanding. Third, the combination of scientific literature with international agreements allows for both rigorous and relevant contributions. The detailed process of the framework development is described in the following chapter.

IV. Framework Development

A. Overview

In the following, we describe the process of our framework development in detail and explain how we combined the SDGs with perspectives from ITS, filtered and refined them across different stages, and finally brought them together for a central view. Fig. 2 summarizes this process and shows how both strands are first considered individually and then merged into the final framework.

B. Determination of Relevant Sustainable Development Goals

To determine the relevant SDGs for our framework and their relation to ITS, we searched for key terms in the resolutions A/RES/70/1 as the original 2030 Agenda for Sustainable Development and A/RES/71/313, which additionally contains the later-adopted indicators to the goals [16], [28]. We started out with all 17 goals, 169 targets, and 231 indicators (248 indicators including duplicates) and filtered them according to the terms in Table 1. We determined the terms according to the perspectives of ITS described in the previous chapter and used collective terms, e.g., "transport," for all related terms. We further added terms that focus on cities since we used them as the application context of our study. Additionally, we added the pollution perspective as a result of traffic and as one goal of ITS. The detailed rationale behind the terms can also be found in Table 1.

The resulting set contained 8 goals, 16 targets, and 16 indicators. We then eliminated two further targets and five indicators. First, we ruled out target 14.1 because it is related to marine pollution, which falls outside the scope of our framework that focuses on traffic on land. We also excluded two indicators that contained the term "urban" as a description for the measurement process and therefore did not apply in terms of content (indicators 1.1.1., 4.5.1, 11.6.1). Finally, we also excluded goal 12.c and indicator 12.c.1 because they are re-

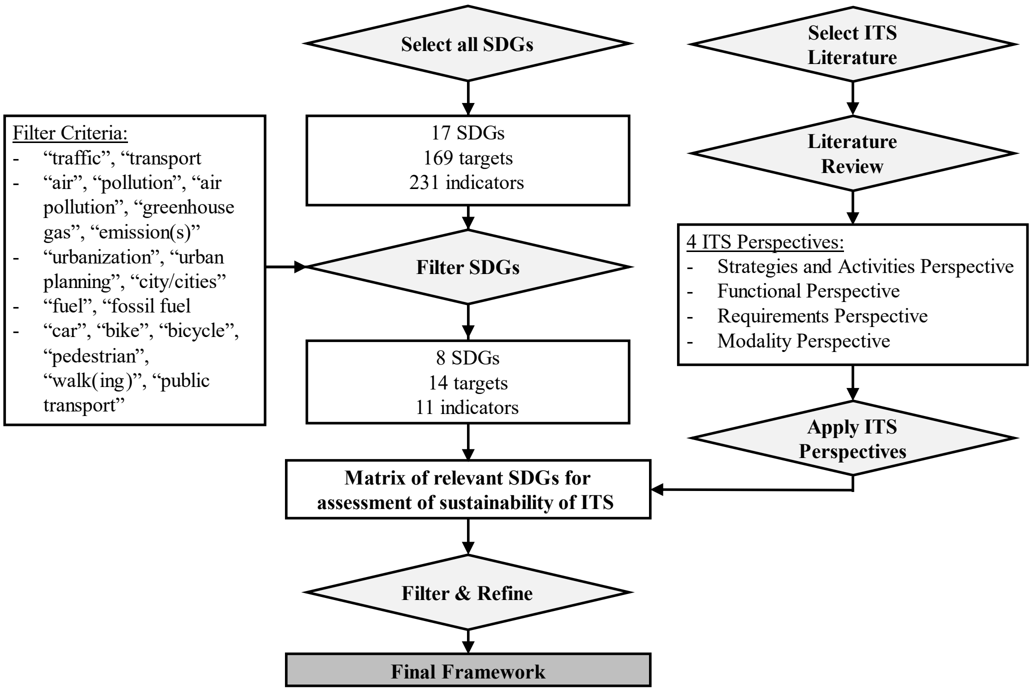

Select all SDGs
17 SDGs
169 targets
231 indicators
Filter Criteria:
- "traffic", "transport
- "air", "pollution", "air pollution", "greenhouse gas", "emission(s)"
- "urbanization", "urban planning", "city/cities"
- "fuel", "fossil fuel
- "car", "bike", "bicycle", "pedestrian", "walk(ing)", "public transport"
Filter SDGs
8 SDGs
14 targets
11 indicators
Select ITS Literature
Literature Review
4 ITS Perspectives:
- Strategies and Activities Perspective
- Functional Perspective
- Requirements Perspective
- Modality Perspective
Apply ITS Perspectives
Matrix of relevant SDGs for assessment of sustainability of ITS
Filter & Refine
Final Framework

Fig 2. Method of Framework Development, own illustration

TABLE I.
FILTER TERMS FOR SUSTAINABLE DEVELOPMENT GOALS, TARGETS, AND INDICATORS

Term	Rationale
"traffic", "transport"	Direct relation to the traffic component of ITS
"air", "pollution", "air pollution", "greenhouse gas", "emission(s)"	Direct relation to the consequences of (motorized) traffic and traffic density as well as the goal of ITS to mitigate the effects on the environment with data-driven systems and technology
"urbanization", "urban", "urban planning", "city", "cities"	Application context of a (smart) city
"fuel", "fossil fuel"	Resources for motorized traffic that is the dominant form of traffic in cities
"car", "bike", "bicycle", "public transport", "train", "pedestrian", "walk(ing)"	Relation to modes of transport on land that are part of ITS

lated to subsidies that are not decided on the level of municipalities and therefore not in the scope of ITS in the context of smart cities. Afterward, we added the superordinated targets or goals related to targets or indicators since they would not apply to the search terms. Hence, we did not add indicators to related targets when they did not apply to the criteria, leaving some indicators blank. From these constraints remained 8 goals, 14 targets, and 11 indicators that we applied to different perspectives on ITS and that can be found in the matrix in Table 2.

C. Determination of Intelligent Transport System Perspectives

After the preselection of relevant SDGs, we determined the relevant ITS dimensions. As presented in the literature review, there are multiple perspectives of ITS that overlap and align in some parts but still provide different views and concepts. In general, we opted for the definition by the UNECE which contains the aspects of improvements in the mobility of people, transportation of passengers and goods, the collection, communication, processing, and distribution of information as well as the acquisition of feedback on experience and quantification of the gathered results. Based on this definition, we aimed to equally consider all four perspectives on ITS that we derived from literature and presented in Fig. 1. We decided against choosing only one of the perspectives in order to include SDGs that are relevant but do not relate to all perspectives. By using several perspectives, we further aim for more transparency, a broader view, and stability in the evaluation to assess the SDGs for developing our final framework.

D. Combination of SDGs and ITS

After filtering the relevant SDGs and determining the ITS perspectives, we combined both dimensions in a matrix (Table 2). On the Y-axis (e.g., in the rows), we entered the Sustainable Development Goals together with the related targets and indicators. We chose a hierarchical view to represent goals, targets, and indicators in an accessible way. On the X-axis (e.g., in the columns), we added the four perspectives on ITS. We then analyzed how measures of each of the ITS perspectives could contribute to the sustainability targets and indicators. We used the resulting intersections to record the results of our analysis. Perspectives that directly contributed to a target or indicator were marked with "XX" in the related row. Perspectives with a more indirect relation were marked with only one "X" to indicate a lower relevance.

After analyzing each relation, we summarized the findings in a central framework (Table 3). We summed up the labels that indicate each relevance per row to determine whether a target or indicator is primarily or secondarily impacted by ITS. We considered targets and indicators that were marked as relevant ("XX") in relation to all ITS perspectives as primarily impacted by ITS and indicators that were either marked as less relevant ("X") or relevant for less than all four perspectives, as secondarily impacted.

E. Final Framework

From the analysis performed, we were able to determine six main sustainability indicators and eight targets that relate to a total of six Sustainable Development Goals and are primarily influenced by ITS. The remaining five indicators and five targets can be influenced by ITS but probably with lower intensity. Therefore, they were marked as indirectly influenced. The final framework is displayed in Table III.

While SDG 11, "Sustainable Cities and Communities," is the most represented goal in the framework with the highest number of related targets and indicators, the other goals are equally relevant. Our framework especially highlights goals that do not relate directly to traffic, like SDG 7, "Affordable and Clean Energy," or SDG 6, "Clean Water and Sanitation."

Our framework provides an overall view of the most important SDGs as a recommendation for researchers and practitioners to consider in the development process of intelligent transport systems. It further stresses that there is not only one perspective on sustainability and that ITS solutions could target sustainability in multiple areas. Some measures might also contribute to multiple SDGs at the same time, e.g., targeting indicator 9.4.1, "CO2 emission per unit of value added," might also have a positive impact on indicator 13.2.2, "Total greenhouse gas emissions per year," due to the general reduction of emissions.

While it might not be possible to consider every factor equally, it serves as a starting point to create awareness of sustainability targets and indicators in the context of ITS. The framework can be used by researchers, but especially but practitioners, to reflect upon projects and initiatives related to intelligent transport systems and sustainability. We recommend using it as a checklist to determine, whether at least one of the goals, targets or indicators is addressed with the planned initiative. The framework is best used in the planning phase of new ITS projects to determine possible sustainability goals, targets, and indicators that might play a role in the projects. Throughout initiatives, it can then help in making the general assessment of the contribution to sustainability of the solutions more transparent.

V. Limitations And Implications

While we aim for a broad and deep analysis in the creation of our framework, we would like to address some weaknesses in the approach and potential for future work. First, one challenge of a literature review is to include both relevant and novel literature as well as consider established "basic" literature. Despite our diligence in the selection process, we cannot claim to have a complete overview. Further, we recognize that other researchers might choose different publications. Additionally, the selection of the four perspectives on ITS might be influenced by subjective perception. As discussed in the literature review, there are many different terms for and perspectives on ITS. One reason might be the interdisciplinary character of ITS where different perspectives from traffic engineering, traffic planning, business administration, and information systems intersect. We see potential for future work in the attempt to find one definition of ITS that includes all perspectives in order to establish a common transdisciplinary understanding.

Second, the inclusion and exclusion criteria of the relevant Sustainable Development Goals were chosen with the application context in mind but are nevertheless subjective and offer room for discussion. Applying different search terms might lead to a different result set and might influence the final matrix. Further, there might be additional SDGs that do not have a direct relation to ITS but might still be considered when developing such systems. For example, SDG target 16.7 calls for ensuring responsive, inclusive, participatory, and representative decision-making at all levels. From a more social perspective, inclusive decision-making in the process of choosing and developing ITS measures, e.g., by including citizens, could contribute to this target as well. While we do not consider the development process of the applications or social factors in our analysis, our framework is easily adjustable and could include these factors in the future or be enhanced for projects with special emphasis on these dimensions.

Third, we recognize that the assessment of each SDG in relation to ITS might be subjective. We conducted the assessment to the best of our knowledge and based it on the descriptions of each SDG, target, and goal. However, other researchers might have rated the criteria differently. In future work, the rating could be enhanced with more expertise by including more researchers.

Fourth, our framework shows the criteria for assessing ITS from a qualitative perspective but does not provide quantitative measurement criteria. The UN does provide some implications for measurement in its definition of indicators. However, they are on a rather high level and need to be adjusted to and detailed for the specific context. Therefore, we suggest a follow-up study on developing a specific measurement for assessing ITS quantitatively.

VI. Conclusion

In this paper, we described the need for more sustainable actions in the mobility sector and pointed out that previous ITS projects and research seemed to lack the consideration of sustainability factors. We further argued that municipalities have a particular responsibility towards sustainability due to their position as decision-makers for their citizens. We developed a framework that considers both different ITS perspectives and sustainability factors and determined which Sustainable Development Goals, targets, and indicators are relevant for the assessment of the sustainability of intelligent transport systems and should be considered when developing ITS strategies.

To summarize, our framework supports decision-makers in municipalities with an approach to assess their selected or planned ITS initiatives regarding sustainability factors. This could help to make more conscious decisions towards a higher quality of living in cities and contribute to the economy, society, and nature at the same time. Hence, we call for a municipal traffic management that is not only smart but also sustainable to contribute to a livable future.

TABLE II.
ASSESSMENT MATRIX OF SDGS IN RELATION TO ITS

Sustainable Development Goals, targets and indicators			**ITS Perspectives**			
SDG	**SDG Target**	**SDG Indicator**	**STR**	**FNC**	**REQ**	**MOD**
Goal 3: Good Health and Well-Being	3.6 Halve the number of global deaths and injuries from road traffic accidents	3.6.1 Death rate due to road traffic injuries	XX	XX	XX	XX
	3.9 Substantially reduce number of deaths and illnesses from hazardous chemicals and pollution and contamination	3.9.1 Mortality rate attributed to household and ambient air pollution	XX	XX	XX	XX
Goal 6: Clean Water and Sanitation	6.3 Improve water quality by reducing pollution, eliminating dumping and minimizing release of hazardous chemicals and materials	-	XX	XX	XX	XX
Goal 7: Affordable and Clean Energy	7.1 Ensure universal access to affordable, reliable and modern energy services	7.1.2 Proportion of population with primary reliance on clean fuels and technology	XX	XX	XX	XX
Goal 9: Industry, Innovation and Infrastructure	9.4 Upgrade infrastructure and retrofit industries to make them sustainable, with increased resource-use efficiency and greater adoption of clean and environmentally sound technologies and industrial processes	9.4.1 CO2 emission per unit of value added	XX	XX	XX	XX
Goal 11: Sustainable Cities and Communities	11.2 Provide access to safe, affordable, accessible and sustainable transport systems for all, improving road safety, notably by expanding public transport	11.2.1 Proportion of population that has convenient access to public transport, by sex, age and persons with disabilities	XX	XX	XX	XX
	11.3 Enhance inclusive and sustainable urbanization and capacity for participatory, integrated and sustainable human settlement planning and management	11.3.2 Proportion of cities with a direct participation structure of civil society in urban planning and management		XX		
	11.6 Reduce the adverse per capita environmental impact of cities; special attention to air quality	11.6.2 Annual mean levels of fine particulate matter (e.g. PM2.5 and PM10) in cities (population weighted)	XX	XX	XX	XX
	11.7 Provide universal access to safe, inclusive, and accessible green and public spaces	11.7.1 Average share of the built-up area of cities that is open space for public use for all, by sex, age, and persons with disabilities				XX
	11.a Support positive economic, social and environmental links between urban, peri-urban, and rural areas by strengthening development planning	11.a.1 Number of countries that have national urban policies or regional development plans that respond to population dynamics and ensure balanced territorial development			XX	
	11.b Increase the number of cities/settlements adopting and implementing policies and plans towards inclusion, resource efficiency, mitigation, and adaptation to climate change, etc.	-	XX	XX	XX	XX
Goal 12: Responsible Consumption and Production	12.4 Environmentally sound management of chemicals and all wastes and reduce their release to air, water and soil	-	XX	XX	XX	XX
Goal 13: Climate Action	13.2 Integrate climate change measures into national policies, strategies and planning	13.2.2 Total greenhouse gas emissions per year	X	X	X	X
Goal 16: Peace, Justice and strong Institutions	16.1 Significantly reduce all forms of violence and related death rates everywhere	16.1.4 Proportion of population that feel safe walking alone around the area they live after dark	XX	X		X

Legend:

STR = Strategy & Activity Perspective

REQ = Requirements Perspective

FNC = Functional Perspective

MOD = Modality Perspective

TABLE III.
FINAL ASSESSMENT FRAMEWORK FOR THE SUSTAINABILITY IMPACT OF ITS

		Relevant Sustainable Development Goals, targets and indicators for ITS	
		SDG	**SDG Target/Indicator**
Impact of ITS on SD goals, targets and indicators	**Primary impact**	Goal 3: Good Health and Well-Being	3.6 Halve the number of global deaths and injuries from road traffic accidents 3.6.1 Death rate due to road traffic injuries
			3.9 Substantially reduce number of deaths and illnesses from hazardous chemicals and pollution and contamination 3.9.1 Mortality rate attributed to household and ambient air pollution
		Goal 6: Clean Water and Sanitation	6.3 Improve water quality by reducing pollution, eliminating dumping, and minimizing release of hazardous chemicals and materials
		Goal 7: Affordable and Clean Energy	7.1 Ensure universal access to affordable, reliable, and modern energy services 7.1.2 Proportion of population with primary reliance on clean fuels and technology
		Goal 9: Industry, Innovation and Infrastructure	9.4 Upgrade infrastructure and retrofit industries to make them sustainable, with increased resource-use efficiency and greater adoption of clean and environmentally sound technologies and industrial processes 9.4.1 CO2 emission per unit of value added
		Goal 11: Sustainable Cities and Communities	11.2 Provide access to safe, affordable, accessible, and sustainable transport systems for all, improving road safety, notably by expanding public transport 11.2.1 Proportion of population that has convenient access to public transport, by sex, age, and persons with disabilities
			11.6 Reduce the adverse per capita environmental impact of cities; special attention to air quality 11.6.2 Annual mean levels of fine particulate matter (e.g. PM2.5 and PM10) in cities (population weighted)
			11.b Increase the number of cities/settlements adopting and implementing policies and plans towards inclusion, resource efficiency, mitigation, and adaptation to climate change, etc.
		Goal 12: Responsible Consumption and Production	12.4 Environmentally sound management of chemicals and all wastes and reduce their release to air, water, and soil
	Secondary impact	Goal 11: Sustainable Cities and Communities	11.3 Enhance inclusive and sustainable urbanization and capacity for participatory, integrated, and sustainable human settlement planning and management 11.3.2 Proportion of cities with a direct participation structure of civil society in urban planning and management
			11.7 Provide universal access to safe, inclusive, and accessible green and public spaces 11.7.1 Average share of the built-up area of cities that is open space for public use for all, by sex, age, and persons with disabilities
			11.a Support positive economic, social and environmental links between urban, peri-urban, and rural areas by strengthening development planning 11.a1 Number of countries that have national urban policies or regional development plans that respond to population dynamics and ensure balanced territorial development
		Goal 13: Climate Action	13.2 Integrate climate change measures into national policies, strategies, and planning 13.2.2 Total greenhouse gas emissions per year
		Goal 16: Peace, Justice and strong Institutions	16.1 Significantly reduce all forms of violence and related death rates everywhere 16.1.4 Proportion of population that feel safe walking alone around the area they live after dark

References

[1] B. Flügge, Smart Mobility – Connecting Everyone. Wiesbaden: Springer Fachmedien Wiesbaden, 2017. doi: 10.1007/978-3-658-15622-0.

[2] T. Chen, J. Ramon Gil-Garcia, and M. Gasco-Hernandez, "Understanding social sustainability for smart cities: The importance of inclusion, equity, and citizen participation as both inputs and long-term outcomes," Journal of Smart Cities and Society, vol. 1, no. 2, pp. 135–148, 2022, doi: 10.3233/SCS-210123.

[3] V. Morabito, "Big Data and Analytics for Government Innovation," in Big Data and Analytics, Cham: Springer International Publishing, 2015, pp. 23–45. doi: 10.1007/978-3-319-10665-6_2.

[4] R. Faria, L. Brito, K. Baras, and J. Silva, "Smart mobility: A survey," in 2017 International Conference on Internet of Things for the Global Community (IoTGC), IEEE, 2017, pp. 1–8. doi: 10.1109/IoTGC.2017.8008972.

[5] G. M. Jonathan, "Digital Transformation in the Public Sector: Identifying Critical Success Factors," 2020, pp. 223–235. doi: 10.1007/978-3-030-44322-1_17.

[6] C. Benevolo, R. P. Dameri, and B. D'Auria, "Smart Mobility in Smart City," 2016, pp. 13–28. doi: 10.1007/978-3-319-23784-8_2.

[7] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," Journal of Internet Services and Applications, vol. 6, no. 1, 2015, doi: 10.1186/s13174-015-0041-5.

[8] F. K. S. Chan and H. K. Chan, "Recent research and challenges in sustainable urbanisation," Resour Conserv Recycl, vol. 184, 2022, doi: 10.1016/j.resconrec.2022.106346.

[9] T. Yigitcanlar, Md. Kamruzzaman, M. Foth, J. Sabatini-Marques, E. da Costa, and G. Ioppolo, "Can cities become smart without being sustainable? A systematic review of the literature," Sustain Cities Soc, vol. 45, pp. 348–365, 2019, doi: 10.1016/j.scs.2018.11.033.

[10] B. Kos, "Intelligent Transport Systems (ITS) in Smart City," 2019, pp. 115–126. doi: 10.1007/978-3-030-17743-0_10.

[11] European Parliament, "Directive 2010/40/EU of the European Parliament and of the Council of 7 July 2010 on the framework for the deployment of Intelligent Transport Systems in the field of road transport and for interfaces with other modes of transport," 2010.

[12] L. Chapman, "Transport and climate change: a review," J Transp Geogr, vol. 15, no. 5, pp. 354–367, 2007, doi: 10.1016/j.jtrangeo.2006.11.008.

[13] Z. Li, R. al Hassan, M. Shahidehpour, S. Bahramirad, and A. Khodaei, "A Hierarchical Framework for Intelligent Traffic Management in Smart Cities," IEEE Trans Smart Grid, vol. 10, no. 1, pp. 691–701, 2019, doi: 10.1109/TSG.2017.2750542.

[14] A. Saikar, M. Parulekar, A. Badve, S. Thakkar, and A. Deshmukh, "TrafficIntel: Smart traffic management for smart cities," in 2017 International Conference on Emerging Trends and Innovation in ICT, ICEI 2017, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 46–50. doi: 10.1109/ETIICT.2017.7977008.

[15] T. Devi, K. Alice, and N. Deepa, "Traffic management in smart cities using support vector machine for predicting the accuracy during peak traffic conditions," Mater Today Proc, vol. 62, pp. 4980–4984, 2022, doi: 10.1016/j.matpr.2022.03.722.

[16] General Assembly of the United Nations, "Transforming our World: The 2030 Agenda for Sustainable Development. A/RES/70/1," 2015.

[17] R. Giffinger and G. Haindlmaier, "Smart cities ranking: an effective instrument for the positioning of the cities?," ACE: Architecture, City and Environment, vol. 4, no. 12, pp. 7–26, 2010, doi: 10.5821/ace.v4i12.2483.

[18] S. Paiva, M. Ahad, G. Tripathi, N. Feroz, and G. Casalino, "Enabling Technologies for Urban Smart Mobility: Recent Trends, Opportunities and Challenges," Sensors, vol. 21, no. 6, p. 2143, 2021, doi: 10.3390/s21062143.

[19] E. J. Tomaszewska and A. Florea, "Urban smart mobility in the scientific literature — bibliometric analysis," Engineering Management in Production and Services, vol. 10, no. 2, pp. 41–56, 2018, doi: 10.2478/emj-2018-0010.

[20] J. Andersen and S. Sutcliffe, "Intelligent Transport Systems (ITS) – An Overview," IFAC Proceedings Volumes, vol. 33, no. 18, pp. 99–106, 2000, doi: 10.1016/S1474-6670(17)37129-X.

[21] United Nations Economic Commission for Europe, "Intelligent Transport Systems (ITS) for sustainable mobility," 2012.

[22] M. Grant, P. Noyes, L. Oluyede, J. Bauer, and M. Edelman, "Developing and Sustaining a Transportation Systems Management & Operations Mission for Your Organization. A Primer for Program Planning.," Reston, Washington, Boulder, 2017.

[23] J. Clark, M. Neuner, S. Sethi, J. Bauer, L. Bedsole, and A. Cheema, "Transportation Systems Management and Operations in Action," Washington, D.C., 2017.

[24] D. Soeiro, "Smart Cities, Well-Being and Good Business: The 2030 Agenda and the Role of Knowledge in the Era of Industry 4.0," 2020, pp. 55–67. doi: 10.1007/978-3-030-40390-4_5.

[25] A. M. Toli and N. Murtagh, "The Concept of Sustainability in Smart City Definitions," Front Built Environ, vol. 6, 2020, doi: 10.3389/fbuil.2020.00077.

[26] T. Yigitcanlar, "Smart cities: an effective urban development and management model?," Australian Planner, vol. 52, no. 1, pp. 27–34, 2015, doi: 10.1080/07293682.2015.1019752.

[27] E. Ziemba, "Exploring Levels of ICT Adoption and Sustainable Development – The Case of Polish Enterprises," Sep. 2019, pp. 579–588. doi: 10.15439/2019F145.

[28] General Assembly of the United Nations, "Global indicator framework for the Sustainable Development Goals and targets of the 2023 Agenda for Sustainable Development. A/RES/71/313," 2017.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 171–181
DOI: 10.15439/2023F5263
ISSN 2300-5963 ACSIS, Vol. 35

Time-series Anomaly Detection and Classification with Long Short-Term Memory Network on Industrial Manufacturing Systems

Tijana Markovic [§]
School of Innovation,
Design and Engineering
Malardalen University
72123, Vasteras, Sweden
tijana.markovic@mdu.se

Alireza Dehlaghi-Ghadim [§]
Research Institute of Sweden (RISE)
School of Innovation, Design and
Engineering, Malardalen University
72164, Vasteras, Sweden
alireza.dehlaghi.ghadim@ri.se,mdu.se

Miguel Leon [§]
School of Innovation,
Design and Engineering
Malardalen University
72123, Vasteras, Sweden
miguel.leonortiz@mdu.se

Ali Balador
School of Innovation,
Design and Engineering
Malardalen University
72123, Vasteras, Sweden
ali.balador@mdu.se

Sasikumar Punnekkat
School of Innovation,
Design and Engineering
Malardalen University
72123, Vasteras, Sweden
sasikumar.punnekkat@mdu.se

***Abstract*—Modern manufacturing systems collect a huge amount of data which gives an opportunity to apply various Machine Learning (ML) techniques. The focus of this paper is on the detection of anomalous behavior in industrial manufacturing systems by considering the temporal nature of the manufacturing process. Long Short-Term Memory (LSTM) networks are applied on a publicly available dataset called Modular Ice-cream factory Dataset on Anomalies in Sensors (MIDAS), which is created using a simulation of a modular manufacturing system for ice cream production. Two different problems are addressed: anomaly detection and anomaly classification. LSTM performance is analysed in terms of accuracy, execution time, and memory consumption and compared with non-time-series ML algorithms including Logistic Regression, Decision Tree, Random Forest, and Multi-Layer Perceptron. The experiments demonstrate the importance of considering the temporal nature of the manufacturing process in detecting anomalous behavior and the superiority in accuracy of LSTM over non-time-series ML algorithms. Additionally, runtime adaptation of the predictions produced by LSTM is proposed to enhance its applicability in a real system.**

***Index Terms*—anomaly detection, anomaly classification, machine learning, deep learning, LSTM, sensor data, manufacturing systems**

I. Introduction

MODERN manufacturing systems have hundreds of sensors that record a huge amount of data, which provides an opportunity to use data science to improve the performance of manufacturing processes [1]. Valuable information and knowledge can be extracted from these data using different techniques, such as machine learning algorithms, statistical analysis methods, data visualization techniques, etc. [2]. One very important aspect that can be addressed is the detection of anomalous behavior in the manufacturing system. Abnormal process behavior is a major problem that can cause a decrease in the quality of a product or a complete process failure that results in the direct loss of a huge amount of money and raw material. The process may fail due to malfunctioning equipment, poor maintenance, external hacker attack, etc. All components and sensors in the system must be continuously monitored, and prompt actions should be provided if any deviations from normal behavior are identified.

This paper aims to detect anomalous behavior in industrial manufacturing systems by considering the temporal nature of the manufacturing process. This temporal nature can be presented by time-series data, which can be easily obtained from any manufacturing system. The time-series is a collection of observations made chronologically, characterized by large data size and high dimensionality [3]. In this paper, the time-series Machine Learning (ML) algorithm, more specifically, Long Short-Term Memory (LSTM) network, is applied on a synthetically generated dataset called Modular Ice-cream factory Dataset on Anomalies in Sensors (MIDAS) [4], which was created using a simulation of a modular manufacturing system for ice cream production. Two different problems are addressed: Anomaly Detection (AD) and Anomaly Classification (AC).

The main contributions of the paper can be summarized as follows:

- an analysis of the LSTM performance (in terms of accuracy, execution time, and memory consumption) in a new data set on manufacturing systems for AD and AC,

[§] Equal contribution

- a comparison in performance between LSTM performance and non-time-series ML algorithms,
- runtime adaptation of the predictions produced by LSTM to enhance its applicability in a real system.

The rest of the paper has been divided into the following sections. Section II provides a description of related research. A background on deep learning and more specifically on LSTM is provided in Section III. The experimental setup, and the results and discussion for the conducted experiments are given in Sections IV and V, respectively. The conclusion and the future work wrap up the paper in Section VI.

II. Related Work

Detecting anomalies in time-series data using ML techniques has recently piqued the interest of many researchers and has been the subject of several studies. Time-series data can be found in various application domains such as smart manufacturing, health monitoring, cyber security, and smart energy management [5]. The data source and its application can have a significant impact on the efficacy of different AD approaches. Moreover, to select an appropriate approach, data characteristics should be also considered, such as label availability and data volume. Several surveys exist that study techniques for AD with respect to different application domains [6], [7], [8]. The focus of the work presented in this paper is to detect anomalies in industrial systems through the analysis of data from multiple sensors.

Identifying patterns or deviations in industrial sensors can be done using various techniques such as statistical analysis, simple ML algorithms, and deep learning. Simple ML methods often utilize classical techniques to model the distribution of time-series data. The authors in [9] used One-Class Support Vector Machine (OC-SVM) and extended Kalman filter for real-time sensor AD in connected automated vehicle sensors. Their proposed method combines model-based signal filtering and ML technique to detect anomalous sensor readings and/or malicious cyber attacks. In [10], the authors used multivariate linear regression and Gaussian mixture models to detect anomalies in the reading sensor values from engine-based machines. They create a model from the correct behavior of the system based on sensor values such as fuel usage, engine load, and oil pressure to gauge and identify specific failures in the machine by comparing the received data with the normal model.

In recent years, deep learning models have shown promising results in time-series modeling [11]. The deep learning ability to automatically learn complex patterns from huge amounts of data makes it suitable for time-series AD. In the application of deep learning on time-series data, there are three clear approaches that can be seen in the literature.

The first approach is to use LSTM. In [12], an automated approach to detecting anomalies using a supervised LSTM and statistical analysis is introduced. This study uses an LSTM neural network to predict non-robust statistical properties and robust ones. This model identifies anomalies in time-series data by combining statistical analysis with a supervised LSTM neural network. In [13], a supervised LSTM-based model is introduced to detect abnormal data generated by mechanical equipment. The model extracts spatial features from the visual representation of the signal's frequency spectrum over time using a convolutional model and detects anomalies using a residual LSTM. In [14], authors introduced an LSTM-based real-time AD algorithm for time-series that can tolerate minor pattern changes. This algorithm automatically calculates the detection threshold based on changes in the data pattern. They employed two LSTM models, each with a distinct threshold. The first threshold is derived by taking into account all data points, whereas the second threshold is determined by taking into account those data points that are considered normal. Two LSTM models work in parallel, where one LSTM model finds single-point anomalies, while the other detects anomalous based on the long-term threshold.

The second approach combines an autoencoder with LSTM for unsupervised/semi-supervised problem-solving. In [15], an LSTM-based scheme is proposed for multi-sensor AD. It uses normal data to train an encoder-decoder model, which reconstructs normal data with minimal error. Anomalies are detected by measuring reconstruction error and likelihood. In [16], a model using autoencoders and residual error detects anomalies in sound sensor data for complex machines, aiding early maintenance planning. [14] introduces an LSTM-based real-time AD algorithm for time-series, adjusting the detection threshold based on pattern changes. Two LSTM models work together for single-point and long-term AD. In [17], a stacked autoencoder connects gated neural networks and LSTMs for short-term and long-term AD in discrete manufacturing. Authors in [18] characterized multi-sensor time series with a deep convolutional autoencoder model. They used a non-linear bidirectional LSTM and linear auto-regressive model for prediction. Multi-Scale Convolutional Recurrent Encoder-Decoder (MSCRED) framework proposed in [19], where they used convolutional recurrent encoder-decoder for detection and diagnosis in multivariate time series data. In this framework, the spatial information is encoded into signature matrices using a convolutional encoder, while the temporal information is modeled using an attention-based ConvLSTM. In [20], the authors proposed the Multivariate Time-series Anomaly Detection via Graph Attention Network (MTAD-GAT) framework, which treats each univariate time series as an individual feature and includes a feature-oriented graph attention layer and a time-oriented graph attention layer to detect the complex dependencies of multivariate time series. Furthermore, to find anomalies, it computes the inference score of forecasting-based model prediction reconstruction-based model predictions. Variational Autoencoder Generation Adversarial Networks (LSTM-based VAE-GAN) method proposed in [21] which uses Generative Adversarial Networks (GAN) for time series AD to monitor the equipment sates. This method trains the encoder, the generator and the discriminator with LSTM models and detects anomalies based on reconstruction error and discrimination results. Corizzo et al. [22] presented an AD model that leverages spatial awareness through a stacked autoencoder architecture. The proposed method utilizes spatial autocorrelation patterns generated by the autoencoder during the distinctive encoding phase to detect anomalies based on distance measures. The authors evaluated the effectiveness of their algorithm using geo-distributed data collected from renewable energy plants.

The final approach is to use a Convolutional Neural Network (CNN). In [23] a fault detection and diagnosis model using a CNN in semiconductor manufacturing is proposed. The authors applied windowing techniques to deal with variable-length multiple time-series data and trained CNNs to detect anomalies in the quality of electric wafers. This model is equipped with a mechanism to identify the contribution of each sensor on anomalies to provide traceable information for fault diagnosis. The work presented in [24] proposes a new deep learning architecture for supervised AD on multi-sensor data using a combination of CNN and Recurrent Neural Networks (RNN). This architecture is proposed to detect anomalies in elevators with different sensors in terms of type and count. They used individual CNN networks for each sensor as a first layer to extract features. This layer eliminates the need for preprocessing and provides architecture design flexibility for the next layers. The next layer of this architecture aggregates these features using LSTM or RNN networks using the windowing technique, where each window can model part of the time-series. The work presented in [25] proposed an algorithm based on CNN and Spectral Residual (SR) to improve the performance of AD on public datasets and Microsoft production data. They converted time series data with the SR model and then used visual saliency detection for AD in time series.

The main difference between this paper and all the above mentioned papers is that this paper focuses on the application of basic LSTM approach to detect and classify anomalies in analog sensors data from a manufacturing environment and compares that approach with different non-time-series ML algorithms.

III. Long Short-term Memory Recurrent Neural Network

A. Artificial Neural Network

Artificial Neural Network (ANN) [26], [11] is an ML algorithm inspired by the human brain. Artificial neurons are organized into layers (input layer, one or more hidden layers, and output layer), where the output of one layer is used as input to the next layer. An example of ANN architecture can be seen in Fig. 1. The layers are interconnected with connections that have parameters called weights. The output of each layer is calculated as follows

$$\overline{O}_{layer_i} = activation((W_{layer_i})^T \times \overline{O}_{layer_{i-1}} + \overline{B}_{layer_i}) \qquad (1)$$

where $O_{layer_{i-1}}$ is the output of the previous layer (if i is the first layer, then $O_{layer_{i-1}}$ is equal to the input values X), $(W_{layer_i})^T$ is the transpose of the weights that connect layer i and layer $(i-1)$ and B_{layer_i} are random weights that are not multiplied by any input. Finally, an activation function transforms the output into a different range. There are different activation functions as sigmoid, tanh, softmax, ReLU, LeakyReLU, etc. [26]. Three activation functions that are used in the paper are: sigmoid (Eq. (2)), tanh (Eq. (3)) and softmax (Eq. (4)). The sigmoid activation function transforms the data into $(0,1)$ interval, while tanh transforms it to $(-1,1)$. The softmax activation function is only used for the output layer and it does not normalize the output where the values are in the $(0,1)$ range and the summation of all values is equal to 1.

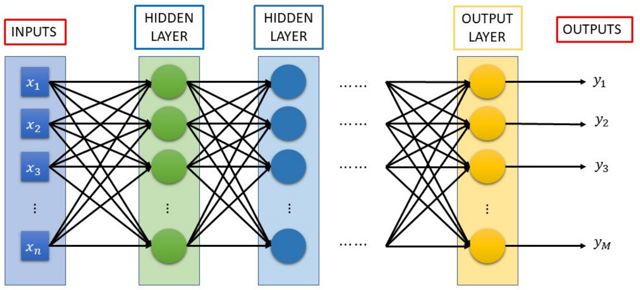


Fig. 1: Artificial Neural Network

$$sigmoid(x) = \sigma(x) = \frac{1}{1+e^{-x}} \qquad (2)$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \qquad (3)$$

$$softmax(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \qquad (4)$$

The goal of the ANN training process is to adjust its weights (black connections in Fig. 1) to fit the objectives. This is done using the backpropagation algorithm [27], [11]. Once the weights are trained, the ANN is ready to be used.

B. Recurrent Neural Network

Recurrent Neural Networks (RNN) is a type of Artificial Neural Network that takes into consideration time-series data. In order to compute the output at time t, the calculations made in $t-1$ are used. Fig. 2 shows an example of RNN, and how it uses the different time steps. To be more specific, the equations that compute the output for time step t ($\overline{Y}(t)$) are as follows

$$\overline{Y}(t) = activation((W_y)^T \times \overline{S}(t)) \qquad (5)$$

$$\overline{S}(t) = activation((W_x)^T \times \overline{X}(t) + (W_s)^T \times \overline{S}(t-1)) \qquad (6)$$

where W_y, W_s, W_x are weight matrices for the connection between the hidden layer and the output layer, the hidden layer connected to itself, and the connection between the input layer and the hidden layer, respectively. Additionally, $\overline{X}$ and $\overline{S}$ are vectors that represent the input and the output of the hidden layer, respectively.

Unfortunately, RNN has a problem, called grading vanishing problem, when training with backpropagation algorithm [26]. For this reason, a more advanced type of network is needed.

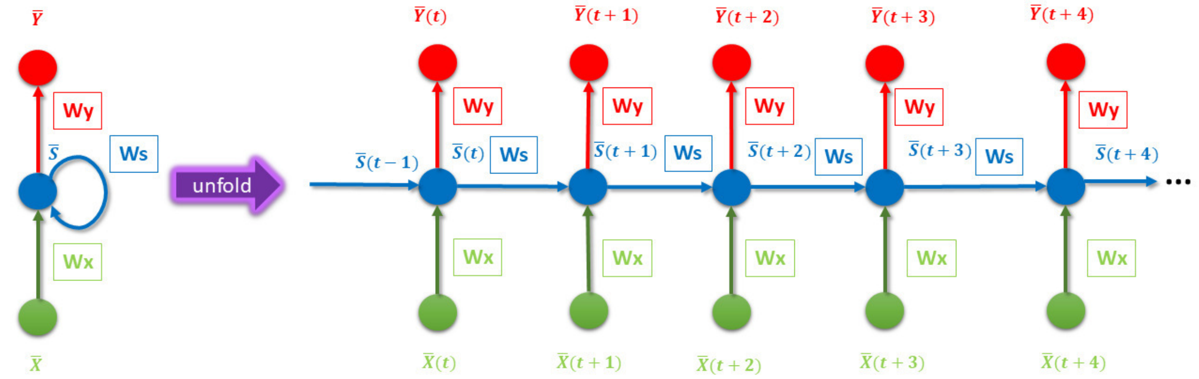


Fig. 2: Recurrent Neural Network in its fold and unfolded version.

C. Long-Short Term Memory Neural Network

Long-short Term Memory (LSTM) neurons are presented to solve the grading vanishing problem. Each of this units is composed of:

- Two memories:
  - **Long-term memory (*LTM*)** which passes the information that has been used for a long period of time to the current time step. Additionally, thanks to this memory, the gradient vanishing problem is solved.
  - **Short-term memory (*STM*)** which saves the information that has been used in the previous time step.
- Four different areas (gates):
  - **Forget gate** which finds information that can be discarded. It is calculated as follows:

$$\overline{F}(t) = \sigma((W_f)^T \times [\overline{X}(t), \overline{STM}(t-1)] + \overline{B}_f) \qquad (7)$$

    where W_f represents the weights of the forget gate, $\overline{X}$ the input information, $\overline{STM}$ the short-term memory, t the time step and $[\overline{a}, \overline{b}]$ the concatenation of the vector a with the vector b.
  - **Input gate** which finds information that is relevant $(\overline{R}(t))$ and needs to be modified into the LTM, and weights the values based on their importance $(\overline{I}(t))$ into the [-1,1] range. The calculations are as follows:

$$\overline{R}(t) = \sigma((W_r)^T \times [\overline{X}(t), \overline{STM}(t-1)] + \overline{B}_r) \qquad (8)$$

$$\overline{I}(t) = tanh((W_i)^T \times [\overline{X}(t), \overline{STM}(t-1)] + \overline{B}_i) \qquad (9)$$

    where W_r and W_i are the weights used to calculate the relevance and the importance, respectively. In the same way, $\overline{B_r}$ and B_i represent the biases used to calculate the relevance and the importance, respectively.
  - **Update gate** which modifies LTM by adding the information found in the input gate. Additionally, LTM needs to be modified by $\overline{F}$ calculated in Eq. (7). The LTM vector $(\overline{LTM})$ for the current time step t is calculated as follows:

$$\overline{LTM}(t) = \overline{LTM}(t-1) \times \overline{F}(t) + \overline{R}(t) \times \overline{I}(t) \qquad (10)$$

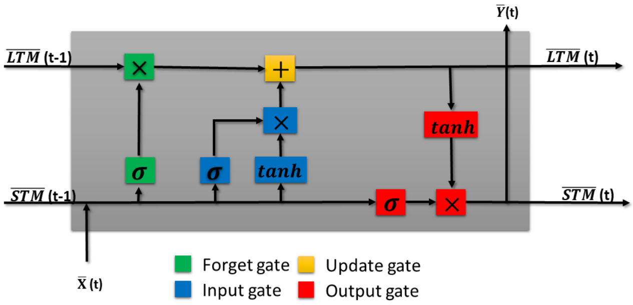


Fig. 3: Long-Short term memory unit. STM and LTM stand for Short-Term Memory and Long-Term Memory, respectively, where t indicates the time step.

  - **Output gate** which calculates the final output that depends on the input information, the output used in the previous time step and the LTM that is previously modified. It is calculated as follows:

$$\overline{Y}(t) = \overline{STM}(t) = \sigma((W_y)^T \times [\overline{X}(t), \overline{STM}(t-1)] + \overline{B}_y) \times tanh(\overline{LSTM}(t)) \qquad (11)$$

    where W_y is the weight matrix used in the use gate, while $\overline{B}_y$ is the bias used on the same gate.

The complex units explained above are used to replace the simple units in the hidden layer of RNN (blue circles in Fig. 2). An overview of the entire LSTM unit is given in Fig. 3.

IV. Experiments

A. Dataset Description

The experiments presented in this paper are conducted on a publicly available dataset called MIDAS[1], which is created using a modular manufacturing simulation environment, where an ice cream making process is simulated [4]. The simulated system is composed of 6 interconnected modules, and each module has various analog and digital sensors [28], [29]. To

[1] https://github.com/vujicictijana/MIDAS/

create MIDAS dataset, three different types of anomalies in analog sensors were injected during the simulation process, and each of them represents modifying the value in a different way: freeze value, step change and ramp change. From the moment of anomaly injection, the actual sensor value was replaced by a modified value until the end of the current run, which means that the simulation behavior was changed from that point since the controllers have access to the wrong values. In this way, different scenarios were simulated, from malfunctioning sensors to external intrusions. The anomalies were injected into the 8 different analog sensors, one sensor at a time.

One run of the system represents a full ice cream making process according to the given recipe, from mixing all the ingredients to producing the ice cream cones as the final product. All runs in the dataset are generated using the same recipe and they are completely independent of each other. The dataset contains a separate csv file for each of 1000 runs, where 258 runs represent normal behavior and 742 runs contains anomalies. There are three different types of anomalies: Freeze value (24.9% of total runs), Step change (24.% of total runs) and Ramp change (24.6% of total runs). The dataset is divided into 500 run for training, 100 for validation and 400 runs for testing data.

Each run has around 36000 instances and one instance represents system state at a certain time point, which results in 36,124,859 instances in the entire dataset. Each instance has 60 columns, including ordinal number of instance within one run, all sensor readings for all modules, and information about anomaly injection. If the instance contains an anomaly the anomaly type, sensor where the anomaly was injected, and actual sensor value are provided. The dataset is well balanced since the percentage of anomalous instances is 50.33%, and approximately one third of anomalous instances belongs to each type of anomaly.

B. Dataset preprocessing

During the dataset preprocessing phase, the following steps were conducted:

- Output variable was defined - The “Anomaly” column is used as the output variable. This column contains 4 different values (- for no anomaly, Step, Ramp, and Freeze) that were encoded to numbers from 0 to 3, in the given order.
- Input variables were defined - All sensor readings for all modules were used as input variables, except the ones that have the same value in the entire dataset, which resulted in 38 input variables.
- Input variables were normalized - All input variables are binary values or real numbers. Binary values remained unchanged and real numbers were normalized in the range [0, 1] using the Min-Max normalization technique.
- Numeric values were transformed to float32 - By default, the columns in the dataset have float64 as a data type. To prevent memory overflow, the data type of the normalized numbers was changed from float64 to float32, with a negligible effect on the results of the LSTM.
- Sliding window approach was applied - In order to generate time-series data for LSTM a sliding window approach was applied. The experiments were conducted with different window sizes and the windows were generated only within a run. Additionally, only the output in the last time step was consider as an output for AD/AC.

C. Experimental settings

Two different problems were addressed: Anomaly Detection (AD) and Anomaly Classification (AC) of abnormal time points on multivariate temporal data using supervised ML algorithms.

All experiments were performed in a MacBook Pro 2019 2.6GHz, Intel Core i7, 16 GB RAM. Python programming language and Tensorflow[2] [30] ML library were used to implement the experiments. The experiments were performed with 500-100-400 runs for training, validation and testing, respectively.

All experiments were conducted with all input variables that were selected during the preprocessing phase (full data), but also with the reduced number of input variables (reduced data). For the experiments with the reduced number of input variables only variables that contain sensor readings for sensors in which the anomalies were injected are considered (8 variables). During all experiments the time and memory consumption was recorded with a goal to analyse how demanding the resulting LSTM is. Each experiment was repeated 5 times and the average results are presented.

Four different experiments have been carried out:

1) *Study of hyper-parameters in LSTM for AD and AC* - The validation set was used to find the best combination of hyper-parameters in LSTM: number of hidden neurons and number of epochs. Values 1, 2, 5, 10 and 20 were tested for both parameters, for both AD and AC. Additionally, two different activation functions were tested for AD (Sigmoid and Softmax), while Softmax activation function was used for AC.
2) *Study of window sizes in LSTM for AD and AC* - The validation set was used to find the best window size with the best combination of hyper-parameters from the first experiment. The window sizes that were tested include 1, 2, 5, 10, and 20 for both AD and AC. The maximal number was selected to cover all time steps within half of a second.
3) *LSTM performance for AD and AC* - LSTM was trained with the best combination of the hyper-parameters from the first experiment and the best window size from the second experiment and tested with the testing data.
4) *Runtime adaptation of LSTM prediction* - Since the resulting LSTM will be applied in a simulator of a modular manufacturing system, it is needed to prevent false negative and false positive alarms as much as

[2] https://www.tensorflow.org/

possible. After it is certain that an anomaly has occurred, the alarm should be on until the run ends or until someone checks the systems. This means that all the remaining time points until the end of that run should be considered as anomalous. To achieve this, runtime adaption of current prediction (y_i) was implemented by checking the previous predictions and changing it according to Eq. 12. Different amount of past time points that are considered ($t \in \{10, 20, 30 \ldots 3000\}$) and different rates of points that have to be predicted as anomalies ($r \in \{1, 2, \ldots 9\}$) were tested on validation set. Before the moment when certainty is achieved, predictions will be considered false positive if less than 10% of predictions in the previous period are predicted as anomalies, so the final prediction will be normal behaviour. All adaptions are made considering only data from the specific run.

$$y_i = \begin{cases} 1 & \text{if } \exists_{k \leq i} \sum_{j=k-t-1}^{k} y_j = \frac{t}{r} \\ 0 & \text{otherwise, if } \sum_{j=i-t-1}^{i} y_j < \frac{t}{10} \\ y_i & \text{otherwise} \end{cases} \tag{12}$$

V. Results and Discussion

A. Study of hyper-parameters in LSTM for AD and AC

The performance of LSTM with different combinations of hyper-parameters on the validation set is presented in Fig. 4 for AD and in Fig. 5 for AC. It can be noticed that the accuracy of LSTM is higher as the number of epochs increases, for both AD and AC, which is expected because the longer training process enables LSTM to fit better to the data. The same happens if the number of hidden neurons is increased, which means that the LSTM can better generalize the model. However, increasing from 10 to 20, for both epochs and hidden neurons, does not bring a big improvement.

With respect to the activation function in the output layer for AD, it can seen how the use of softmax gives improvements in all cases except when there is one neuron in the hidden layer. This is normal since having two neurons in the output layer helps generalize better than having only a single unit.

Finally, if considering using all the input variables or only the variables where anomalies were injected, it can be noticed that using all available information gives benefits to the proposed model, which means that other variables in the process are also affected by the anomalies.

Taking into consideration all of the above statements it was decided to use softmax as an activation function for the output layer for AD, and 10 as the number of hidden neurons and the number of epochs for all problems. The selected combination did not have the absolute highest accuracy, however, the results are very similar with the best combination. This option is selected because it makes the model more efficient in terms of time and memory. Although the results using the full data are better, the experiments in the following sections are conducted using both options.

TABLE I: Accuracy of LSTM on validation set for different window sizes for AD and AC with all and reduced input variables

Problem Window	AD		AC	
	full	reduced	full	reduced
1	84.01	82.72	68.81	68.29
2	86.75	85.21	69.58	69.51
5	87.45	86.73	69.74	69.89
10	88.06	86.07	71.27	69.01
20	**88.51**	**86.92**	**71.52**	**70.55**

B. Study of window sizes in LSTM for AD and AC

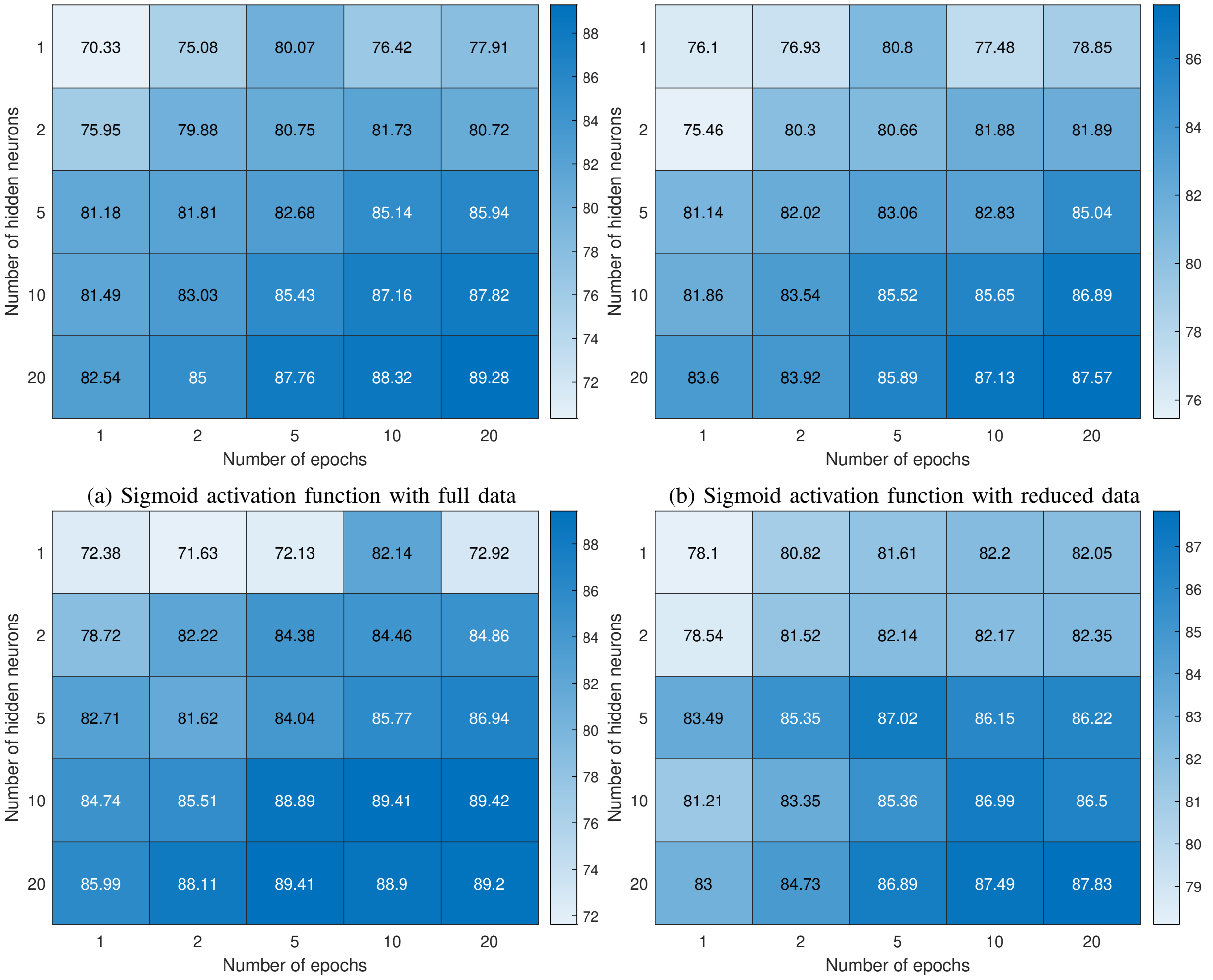
The best combination from the previous section is used to train LSTM with different window sizes and the results are presented in Table I. As expected, as the window size increases the accuracy of LSTM is higher, since the model considers more time steps back in time. It can be observed that the improvement is around 1 percentage point in almost all cases for full data in AD for window sizes 1, 2 and 5. The differences from window size 5 to 10 and 10 to 20 are around 0.5 percentage points. On the other hand, on full data for AC, the improvement is only big when comparing window size 1 to window size 2 (around 3 percentage points). Then, for the rest of the window sizes the accuracy remains stable. If the reduced data are considered, a similar behavior can be noticed for both AD and AC.

If the results between window size 1 (which considers only a single time step) and window size 20 (which considers half a second of the manufacturing process) are considered, the improvement can be noticed in all cases varying from 2.2 to 4.5 percentage points. This clearly shows the benefit of considering time-series when making a predictions.

C. LSTM performance for AD and AC

The window size that achieved the best performance on the validation set (Table I) was used to measure the LSTM performance on the test set. The accuracy of LSTM and the comparison with non-time-series ML algorithms is presented in Table II, for both AD and AC. The non-time-series ML algorithms that were considered include: Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), and Multi-Layer Perceptron (MLP). The parameters used for these algorithms can be found in paper [4].

It can be observed that LSTM has the highest accuracy, obtaining a better performance than the best non-time-series algorithm (MLP) by 2.7 and 0.12 percentage points for AD and AC, respectively. This again proves that considering the previous data can help detect anomalous behavior. Additionally, it can be noticed that LSTM that uses reduced data also has higher accuracy than all non-time-series algorithms for AD, while it is better than all of them and almost equal to MLP for AC. This means that historical information can bring more benefits than using data from all sensors without considering the temporal nature of the process.



(a) Sigmoid activation function with full data

(b) Sigmoid activation function with reduced data

(c) Softmax activation function with full data

(d) Softmax activation function with reduced data

Fig. 4: Hyper-parameters selection for LSTM network for AD - Heatmaps with accuracies of LSTM on the validation set

TABLE II: LSTM performance for AD and AC on testing set and comparison with non-time-series ML algorithms

Problem / Algorithm	AD	AC
LR	64.60	51.42
DT	69.97	56.53
RF	75.07	62.49
MLP	82.36	70.73
LSTM (full data)	**85.06**	**70.85**
LSTM (reduced data)	84.49	69.89

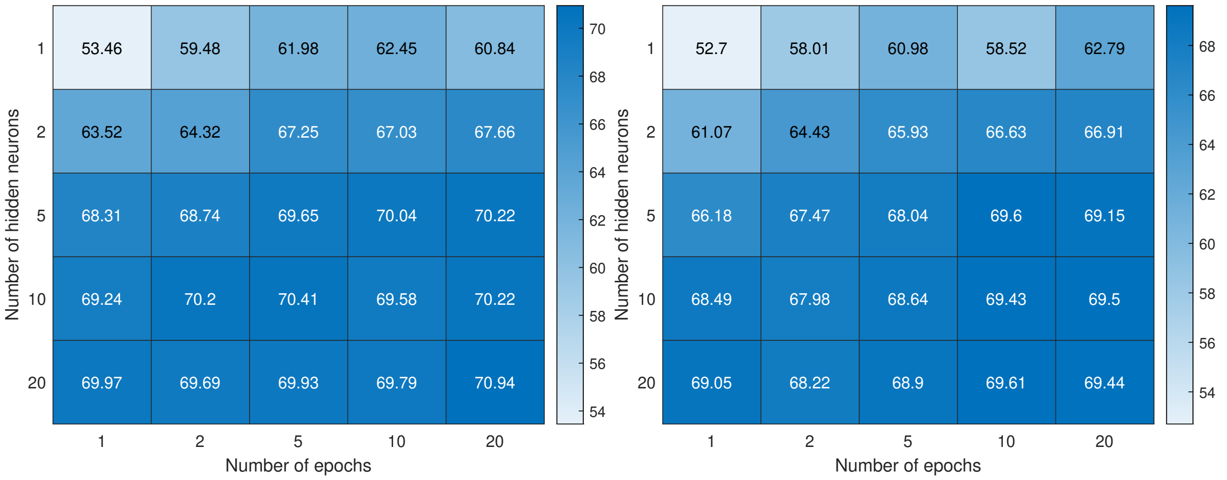
D. Runtime adaptation of LSTM prediction

The performance of LSTM on validation set after making a runtime adaption of the predictions is shown in Figure 6. It can be observed that the best results were achieved if all the previous time points ($r = 1$) in the selected period are predicted as anomalous. The best results were achieved when the amount of previous time points was 390. This means that the system needs around 8 seconds of continues positive predictions to be certain that an anomaly happened. Additionally, it can be noticed that after the selected number of time points (t) the accuracy is stabilized. Comparing all considered combinations, they have the same trend. However, with smaller number of required ones, the system needs more time to be certain about the anomaly, which results in lower overall accuracy.

After applying real-time adaption on the testing set, the overall accuracy of the model improved in average 1.8 percentage points, resulting in overall average accuracy of 86.9%. The actual performance of the runtime adaptation is showed in Fig. 7. It can be noticed that by adding the runtime adaption some of the wrong predictions were successfully fixed (some of the false positive and negative alarms were prevented).

E. Time and memory consumption

An important aspect to consider is the time that LSTM requires to create the model, as well as the time that is needed to apply the model to detect/classify the anomalies in a real-time environment. Firstly, a comparison between model training and testing time depending on the window size and



(a) Softmax activation function with full data

(b) Softmax activation function with reduced data

Fig. 5: Hyper-parameters selection for LSTM network for AC - Heatmaps with accuracies of LSTM on the validation set

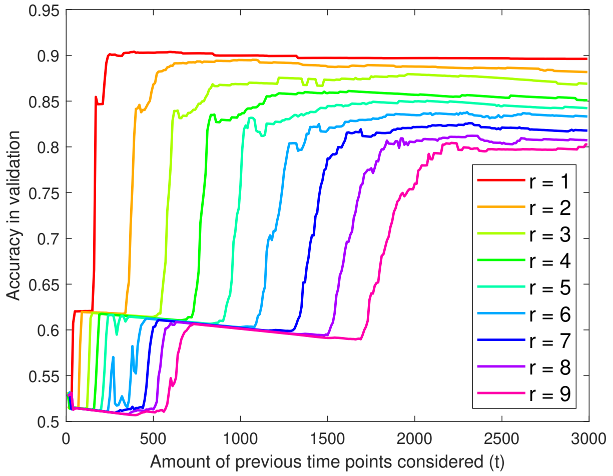


Fig. 6: Accuracy of adapted LSTM on validation set for different combinations of amount of previous time points considered (t) and the rate of anomalous predictions (r)

the amount of data (full data and reduced data) is given in Fig. 8. It can be observed how the time increases linearly for training (Fig. 8a), except when the full data with window size 20 is used. The main reason for this increase in time is that the RAM memory is not large enough to allocate the whole dataset, so the computer is forced to use the hard-drive to load the data. Secondly, if the testing time is considered (Fig. 8b), it can be noticed that all models with all window sizes would be able to run on a real-time system. The time between data readings within the system is equal to 25 ms, which is much higher than the slowest LSTM model that was evaluated (which requires less than 0.05 ms).

Additionally, the time used by LSTM and the non-time-series ML algorithm are compared with respect to training and testing time (Table III). Using LSTM requires more time than the non-time-series ML algorithms for both training and testing, which is expected since the LSTM uses more data points within every example. When focusing only on the testing time, it can seen that the LSTM is 84 times slower than non-time-series version of ANN (MLP) for AD, while it is 39 times slower for AC.

TABLE III: Average time of LSTM for training on entire training set and testing of one single case and comparison with non-time-series ML algorithms

Problem / Algorithm	AD		AC	
	Train (s)	**Test (ms)**	**Train (s)**	**Test (ms)**
LR	889.06	0.00032	4520.92	0.00027
DT	415.44	0.00048	447.42	0.00063
RF	1493.27	0.00432	2197.63	0.00576
MLP	11362.16	0.00052	17925.70	0.00110
LSTM (full data)	45367.63	0.04386	61860.21	0.04328
LSTM (reduced data)	3028.62	0.04475	4702.19	0.04509

Finally, the memory used by the training data, for both full and reduced data, is given in Fig. 9. It can be seen that the required memory increases linearly, however, with the full data from the window size 10, the dataset requires more memory than the amount of memory available in the RAM. This also proves that the limit of RAM is the reason why time increases exponentially after window size 10 (as shown in Fig. 8a).

VI. Conclusions and Future Work

In this paper, LSTM is applied to detect and classify anomalies in the manufacturing system. Experiments were conducted on the synthetically generated dataset called MIDAS that contains various anomalies in analog sensors data during ice cream making process and realistically represents a manufacturing process. The results showed that considering time-series nature of the data is beneficial for the accuracy of both detection and classification of anomalies. The LSTM had better performance than all non-time-series ML algorithms

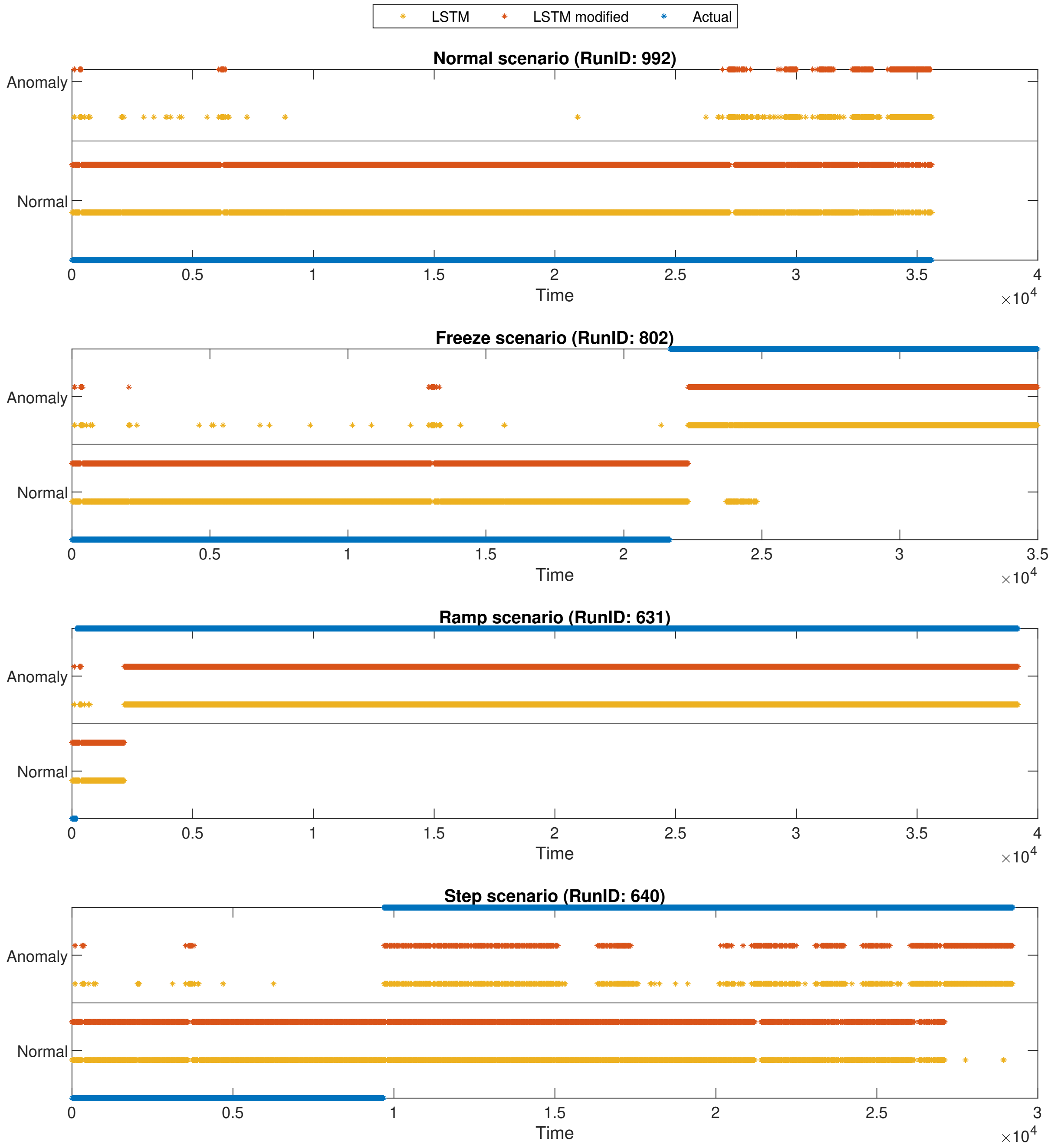


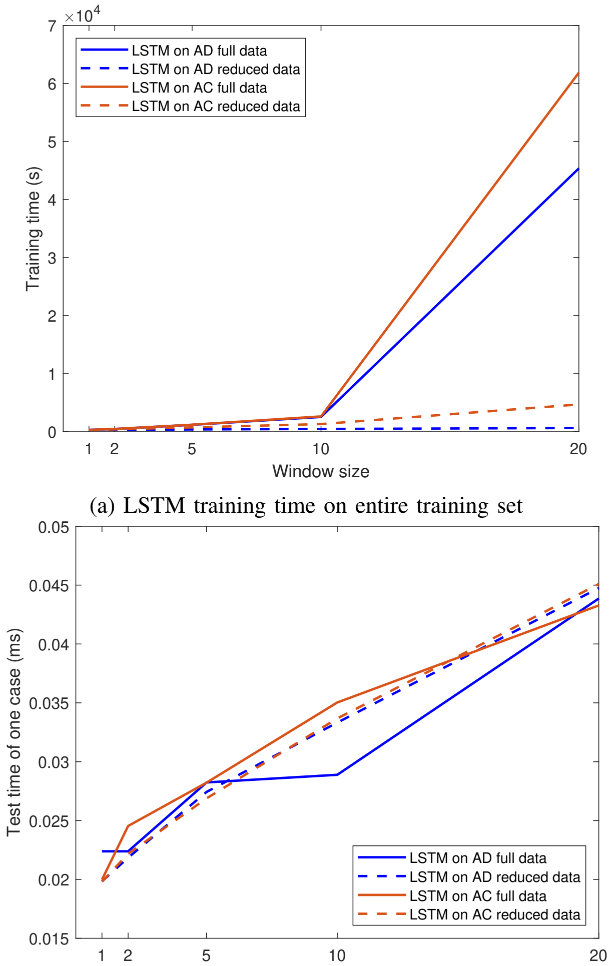
Fig. 7: Predictions of LSTM and LSTM after runtime adaptation (LSTM modified) for 4 full scenarios from the testing set for anomaly detection. Every time-point prediction is plotted.

it was compared with. On the other hand, it is more time consuming, but it can still run on a real-time system. Furthermore, the predictions generated by LSTM were adapted during runtime, with a goal to improve its performance and applicability in real systems.

As a future work, we plan to integrate the proposed model in the simulation environment that is used to generate MIDAS dataset. Additionally, we plan to develop a federated learning approach for LSTM that will be able to distribute the AI on different system levels.

ACKNOWLEDGMENT

This work has been partially supported by the H2020 ECSEL EU projects Intelligent Secure Trustable Things (InSecTT) and Distributed Artificial Intelligent System (DAIS). InSecTT (www.insectt.eu) has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 876038 and DAIS (https://dais-project.eu/) has received funding from



(a) LSTM training time on entire training set

(b) LSTM testing time for one case

Fig. 8: Time comparison of LSTM during training and testing with different window sizes.

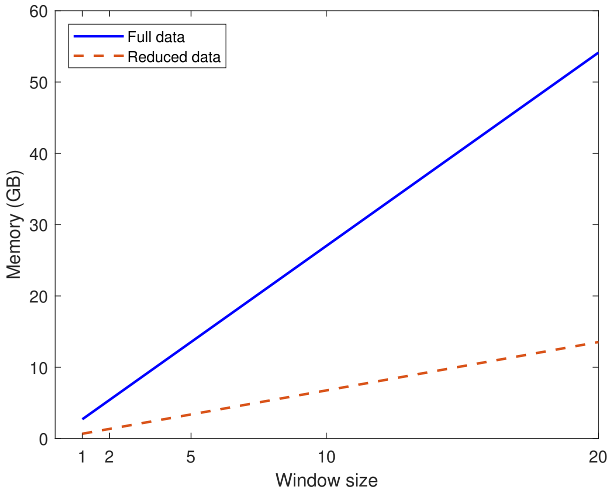


Fig. 9: Memory used by the training data

the ECSEL JU under grant agreement No 101007273. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Sweden, Spain, Italy, France, Portugal, Ireland, Finland, Slovenia, Poland, Netherlands, Turkey.

REFERENCES

[1] B. Esmaeilian, S. Behdad, and B. Wang, "The evolution and future of manufacturing: A review," *Journal of manufacturing systems*, vol. 39, pp. 79–100, 2016.

[2] A. K. Choudhary, J. A. Harding, and M. K. Tiwari, "Data mining in manufacturing: a review based on the kind of knowledge," *Journal of Intelligent Manufacturing*, vol. 20, pp. 501–521, 2009.

[3] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[4] T. Markovic, M. Leon, B. Leander, and S. Punnekkat, "A modular ice cream factory dataset on anomalies in sensors to support machine learning research in manufacturing systems," *IEEE Access*, vol. 11, pp. 29 744–29 758, 2023.

[5] K. Choi, J. Yi, C. Park, and S. Yoon, "Deep learning for anomaly detection in time-series data: review, analysis, and guidelines," *IEEE Access*, 2021.

[6] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using lstm networks," *Computers in Industry*, vol. 131, p. 103498, 2021.

[7] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for iot time-series data: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6481–6494, 2019.

[8] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: a comprehensive evaluation," *Proceedings of the VLDB Endowment*, vol. 15, no. 9, pp. 1779–1797, 2022.

[9] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE transactions on intelligent transportation systems*, vol. 22, no. 3, pp. 1411–1421, 2020.

[10] G. Shah and A. Tiwari, "Anomaly detection in iiot: A case study using machine learning," in *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*, 2018, pp. 295–300.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] W. Jia, R. M. Shukla, and S. Sengupta, "Anomaly detection using supervised learning and multiple statistical methods," in *2019 18th IEEE International Conference On Machine Learning and Applications (ICMLA)*. IEEE, 2019, pp. 1291–1297.

[13] G. Hong and D. Suh, "Supervised-learning-based intelligent fault diagnosis for mechanical equipment," *IEEE Access*, vol. 9, pp. 116 147–116 162, 2021.

[14] M.-C. Lee, J.-C. Lin, and E. G. Gan, "Rere: A lightweight real-time ready-to-go anomaly detection approach for time series," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020, pp. 322–327.

[15] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[16] D. Y. Oh and I. D. Yun, "Residual error based anomaly detection using auto-encoder in smd machine sound," *Sensors*, vol. 18, no. 5: 1308, 2018.

[17] B. Lindemann, N. Jazdi, and M. Weyrich, "Anomaly detection and prediction in discrete manufacturing based on cooperative lstm networks," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1003–1010.

[18] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[19] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 1409–1416.

[20] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 841–850.

[21] Z. Niu, K. Yu, and X. Wu, "Lstm-based vae-gan for time-series anomaly detection," *Sensors*, vol. 20, no. 13, pp. 3738–3750, 2020.

[22] R. Corizzo, M. Ceci, G. Pio, P. Mignone, and N. Japkowicz, "Spatially-aware autoencoders for detecting contextual anomalies in geo-distributed data," in *International conference on discovery science*. Springer, 2021, pp. 461–471.

[23] C.-Y. Hsu and W.-C. Liu, "Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semi-conductor manufacturing," *Journal of Intelligent Manufacturing*, vol. 32, pp. 823–836, 2021.

[24] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head cnn–rnn for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, 2019.

[25] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 3009–3017.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415–1442, 1990.

[28] B. Leander, T. Marković, A. Čaušević, T. Lindström, H. Hansson, and S. Punnekkat, "Simulation environment for modular automation systems," in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2022, pp. 1–6.

[29] B. Leander, T. Markovic, and M. Leon, "Enhanced simulation environment to support research in modular manufacturing systems," in *IECON 2023–49th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2023, pp. 1–6.

[30] M. A. et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

DOI: 10.15439/2023F4382
ISSN 2300-5963 ACSIS, Vol. 35

XOR-based decomposition and its application in memory-based and reversible logic synthesis

Tomasz Mazurkiewicz
0000-0001-7305-2379
Cyber Command
Warsaw, Poland
Email: kontakt@tomaszmazurkiewicz.pl

***Abstract*—In this research paper, we propose a novel approach to digital circuit design using XOR-based decomposition. The proposed technique utilizes XOR gates as a fundamental building block for decomposing complex Boolean functions into simpler forms, leading to more efficient and compact digital circuits. We demonstrate the effectiveness of our approach in two different contexts: memory-based logic synthesis and reversible logic synthesis. In particular, we demonstrate that the proposed technique can efficiently reduce the number of input variables, which is a crucial task when using memories in the design. Obtained results prove that the XOR-based approach can efficiently complement variable reduction and dimensionality reduction algorithms. Furthermore, we show its application in generating the XOR-AND-XOR form of a reversible function and demonstrate how to combine it with another technique, i.e., a functional decomposition for reversible logic synthesis.**

I. Introduction

LOGIC synthesis is a very important process that enables efficient design and implementation of complex digital circuits. Its importance continues to grow since the complexity and performance demands of digital circuits are still increasing. In recent years, a specific type of logic synthesis, i.e., memory-based synthesis has gained attention from researchers, leading to the development of numerous algorithms and techniques in this field, making it a promising approach for the design of modern electronic systems. Especially logic synthesis of incompletely specified Boolean functions was deeply analyzed.

This technique approaches the design of digital circuits from a different perspective than traditional logic synthesis methods. Instead of focusing solely on Boolean logic gates and their interconnections, memory-based logic synthesis incorporates memories, such as static random-access memory (SRAM) and read-only memory (ROM), as fundamental building blocks in the design process. Due to that, this approach can improve the performance, area, and power consumption of digital circuits.

Lately, a synthesis of specific functions, called index generation functions [6], [9], [13], gained significant interest due to the practical applications of their implementations in network hardware, e.g., in telecommunication and cybersecurity.

Due to the properties of index generation functions, typically fewer variables than initial N variables can be used to represent those functions. It is important, especially in memory-based logic synthesis [11], where the memory size strongly depends on the number of input variables.

In the literature, the application of linear (i.e., XOR-based) decomposition in index generation functions minimization was widely investigated. However, XOR-based logic synthesis is a relatively new approach to a digital circuit design that leverages the properties of the exclusive OR (XOR) gate as a fundamental building block. It is worth noticing that this approach can often exploit the symmetries of Boolean functions, leading to more efficient circuits.

This approach implements a function as a composition of linear and general functions. The layer of XOR gates implements the first one, while the second one is typically implemented using memory (RAM/ROM). A typical decomposition scheme is presented in Fig. 1. Variable reduction is an optional step that reduces the number of variables, i.e. it removes those variables that can be removed without loss of any information. The outputs of this algorithm become the inputs to a linear function algorithm. This algorithm finds P reduction equations that use XOR combinations of subsets of the input variables. In the end, the general function is implemented using $2^P Q$ memory bits, where Q denotes the number of output variables.

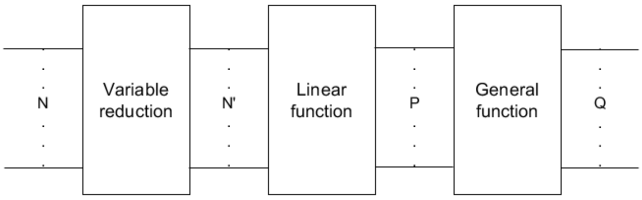


Fig. 1: The linear decomposition scheme.

Similar approaches, i.e. reduction of a number of variables (or dimensionality reduction) and implementation of a reduced general function, are used in other fields, e.g. in reversible logic synthesis [2] and data mining [3]. Especially the first field is a promising research area due to its potential to improve the energy efficiency of digital circuits. Work on the potential of the exposition of an XOR relationship in the logic synthesis of boolean functions has also been carried out. For example, Czajkowski and Brown [5] showed that it leads to significant resource savings for MCNC (Microelectronics Center of North Carolina) benchmark functions.

In this paper, we analyze how the XOR-based method used previously in index generation functions minimization [6] can be generalized and applied to any function represented using binary input vectors. We present the whole algorithm and show its usefulness using standard benchmark functions. We also present how the proposed approach can be used in the fields mentioned above. In particular, lately, the novel form called XORAX (XOR-AND-XOR) was proposed [2] to represent a function. This form can ease the reversible synthesis of some functions. In this paper, we show that the XOR-based decomposition can be used to perform the first step of XORAX form generation. We also prove that the proposed method can be used to improve the results obtained using a functional decomposition [10].

II. Preliminaries

A. Basic notation

Let N denote the number of input variables, and K denote the number of rows in a function truth table. Notice that $K = 2^N$ for completely specified Boolean functions.

To present an algorithm of XOR-based decomposition, we introduce a concept of discernibility set. It will be denoted as $C_{p,q}$, where p and q ($p, q \in \{1, 2, \ldots, K\}, p < q$) are indexes of vectors of $\{0, 1\}^N$, such that $F(p) \neq F(q)$, where $F(p)$ is the output value for row number p in a function truth table. We define discernibility set as follows:

$$C_{p,q} = \{x \in X : x(p) \neq x(q)\}, \tag{1}$$

where X denotes the set of input variables.

In particular, $C_{p,q}$ represents input variables where vectors number p and q differ. For example, in Table I the first and third vectors differ on variable x_2. Thus, $C_{1,3} = \{x_2\}$.

TABLE I: An example function.

idx	x_1	x_2	x_3	$F(X)$
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	0
8	1	1	1	0

Notice that for any index generation function, the condition $F(p) \neq F(q)$ is true for every possible pair of values p and q since the output values are unique consecutive integer values. This observation simplifies computations for such functions. However, it leads to higher memory consumption since many more $C_{p,q}$ sets might be generated.

The collection of all $C_{p,q}$ will be denoted as RC, i.e.

$$RC = \{C_{p,q} : p, q \in \{1, 2, \ldots, K\}, p < q\}. \tag{2}$$

Its complement, i.e., collection of all sets that are not present in the RC, will be denoted as $COM(RC)$. Additionally, the complement limited to r-element sets will be denoted as $COM(RC^r)$. The discernibility sets for all possible values of p and q can be represented using the discernibility matrix.

Example 2.1: Consider the example function ($N = 3, O = 1, K = 8$) presented in Table I. All calculated C_{pq} sets for this function (i.e., RC) are presented in Table II. Notice that pairs of p and q values such that $F(p) = F(q)$ were omitted in the calculation (e.g. $C_{1,2}$ is not present since the value for both the first and second rows equals zero). Based on the calculated sets, we get

$$RC^1 = \{\{x_1\}, \{x_2\}\},$$

$$RC^2 = \{\{x_1, x_3\}, \{x_2, x_3\}\},$$

$$RC^3 = \emptyset.$$

Therefore, we get the following complements:

$$COM(RC^1) = \{\{x_3\}\},$$

$$COM(RC^2) = \{\{x_1, x_2\}\},$$

$$COM(RC^3) = \{\{x_1, x_2, x_3\}\}.$$

TABLE II: $C_{p,q}$ sets for the example function

p, q	$C_{p,q}$	p, q	$C_{p,q}$
1,3	$\{x_2\}$	3,7	$\{x_1\}$
1,4	$\{x_2, x_3\}$	3,8	$\{x_1, x_3\}$
1,5	$\{x_1\}$	4,7	$\{x_1, x_3\}$
1,6	$\{x_1, x_3\}$	4,8	$\{x_1\}$
2,3	$\{x_2, x_3\}$	5,7	$\{x_2\}$
2,4	$\{x_2\}$	5,8	$\{x_2, x_3\}$
2,5	$\{x_1, x_3\}$	6,7	$\{x_2, x_3\}$
2,6	$\{x_1\}$	6,8	$\{x_2\}$

Since we check whether $F(p) \neq F(q)$, the proposed approach can be applied also to functions that do not return only 0 or 1. In particular, both multiple-output functions and functions with multiple-valued output can be decomposed using the same technique as long as input vectors are represented as binary vectors.

Example 2.2: Consider the function presented in Table III. It represents the following mapping: $F : \{0, 1\}^6 \to \{1, 2, 3\}$. Using the same approach as in Example 2.1, we get the RC presented in Table IV.

TABLE III: An example with multiple-valued output.

idx	x_1	x_2	x_3	x_4	x_5	x_6	$F(X)$
1	1	0	0	0	0	0	1
2	0	1	1	1	1	0	1
3	0	0	1	0	0	1	2
4	1	0	1	1	1	1	2
5	0	1	0	0	1	0	3
6	0	1	0	0	0	0	3

B. Memory-based logic synthesis

Memory-based logic synthesis plays an important role in embedded systems, which often rely heavily on memory for storing and processing data. For example, nonvolatile Read Only Memories (ROMs) are used to store some fixed data used in a design, or to implement a truth table or a state machine.

TABLE IV: $C_{p,q}$ sets for the example function

p,q	$C_{p,q}$	p,q	$C_{p,q}$
1,3	$\{x_1, x_3, x_6\}$	2,5	$\{x_3, x_4\}$
1,4	$\{x_3, x_4, x_5, x_6\}$	2,6	$\{x_3, x_4, x_5\}$
1,5	$\{x_1, x_2, x_5\}$	3,5	$\{x_2, x_3, x_5, x_6\}$
1,6	$\{x_1, x_2\}$	3,6	$\{x_2, x_3, x_6\}$
2,3	$\{x_2, x_4, x_5, x_6\}$	4,5	$\{x_1, x_2, x_3, x_4, x_6\}$
2,4	$\{x_1, x_2, x_6\}$	4,6	$\{x_1, x_2, x_3, x_4, x_5, x_6\}$

This technique helps to optimize the design of digital circuits that interface with memory.

In modern embedded systems, Field-Programmable Gate Arrays (FPGAs) are used very often due to the wide range of applications, including signal processing, video processing, networking, and data storage. The fundamental building blocks used to implement a digital circuit described typically using a hardware description language, are Look-Up Tables (LUTs). A LUT is a small, programmable memory block that can store a truth table, which implements a Boolean function. The number of inputs to a LUT can vary depending on the FPGA architecture, but in modern devices equals typically 4 to 6. Additionally, FPGAs very often contain memories.

The main goal of memory-based logic synthesis [11] is to minimize the memory usage of the design. The typical approach is to use functional decomposition to divide a function into smaller subfunctions that can be efficiently implemented using memories or LUTs. Additionally, a variable reduction [3] is often used to remove redundant information from a function. However, this technique fails for some functions [6], or the number of variables can be further minimized. For example, the XOR-based decomposition reduces the number of variables by using the additional layer of XOR gates.

C. Reversible logic synthesis

Reversible logic synthesis is a process of designing circuits that can run both forward and backwards, meaning that can perform both computation and its inverse. In order to do so, the circuit has to have the same number of input and output variables and the implemented function must be a bijection (i.e., the input values can be uniquely determined from the output values and vice versa). In this process, the main goal is to minimize the number of gates required to implement a function.

In order to unify the approach for the comparison of different reversible logic synthesis methods, several gate libraries are used. The smallest complete set of gates, proposed in [14] and called the NCT gate library, contains three gates: NOT, CNOT and Toffoli gates. Any reversible function can be implemented using a combination of those three gates.

The first gate is a one-bit gate that performs the logical negation operation. The second gate is a two-bit gate, meaning it has two input signals and two output signals, which implements the following mapping $(x, y) \rightarrow (x, x \oplus y)$, where $\oplus$ denotes the XOR function. The Toffoli gate is a three-bit gate with two control bits and a single target bit. It implements the following mapping $(x, y, z) \rightarrow (x, y, xy \oplus z)$.

The NCT library is commonly used in the design of reversible circuits due to its simplicity and efficiency. However, it can be extended to include some additional gates, such as SWAP and Fredkin gates (NCTSF library) or multiply-controlled gates (GT and GT&GF libraries). Reversible circuits are implemented as a cascade of those reversible logic gates.

There are several significant applications of reversible logic synthesis. Due to the asymptotic zero power dissipation achievable by reversible computation [1], it can be used in low-power computing, making them an attractive option for mobile devices, wearables, and Internet of Things (IoT) devices. Reversible circuits are also important in quantum computing since quantum algorithms require reversible logic gates for their implementation. Reversible logic synthesis might be also useful in optical computing and DNA computing.

III. XOR-based decomposition

The existence of the XOR-based decomposition can be verified using a simple test [6], [9] : $x_i \oplus x_j$ is a decomposition function of F iff $\{x_i, x_j\} \in COM(RC^2)$. Similarly, pair of functions $x_i \oplus x_j$ and $x_j \oplus x_k$ is a decomposition function of F iff $\{x_i, x_j, x_k\} \in COM(RC^3)$, and so on.

Therefore, to find a decomposition function, we need to generate RC^r and look for a decomposition using the increasing value of r. In the result, we get:

$$F(x_1, x_2, \ldots, x_N) = G(y_1, y_2, \ldots, y_{N-1}). \qquad (3)$$

The first $N - r + 1$ inputs in the G function correspond to those inputs of the F function that are not used in a decomposition function. The last $r - 1$ inputs are obtained using a decomposition function, e.g. y_{N-1} might equal to $x_i \oplus x_j$ if a decomposition function for $r = 2$ is found.

Example 3.1: Consider again the function F from our previous example (Example 2.1). Notice that both $COM(RC^2)$ and $COM(RC^3)$ are not empty. Therefore, two possible decompositions are:

1) $y_1 = x_3$ and $y_2 = x_1 \oplus x_2$,
2) $y_1 = x_1 \oplus x_2$ and $y_2 = x_2 \oplus x_3$.

In the first case, we get

$$F(x_1, x_2, x_3) = G(x_3, x_1 \oplus x_2),$$

while in the second case we get

$$F(x_1, x_2, x_3) = G(x_1 \oplus x_2, x_2 \oplus x_3).$$

In both cases, the number of variables was reduced by one. Truth tables for both cases are presented in Table V. Notice that the number of rows is also reduced (from 8 to 4). The first column shows the row numbers from the original truth table to which the newly generated value corresponds.

The same approach can be applied to the function from Example 2.2. Based on the RC we know that $x_1 \oplus x_3$ is a decomposition function, since $\{x_1, x_3\} \notin RC \Rightarrow \{x_1, x_3\} \in COM(RC^2)$. Therefore, we get:

$$y_1 = x_2,\ y_2 = x_4,\ y_3 = x_5,\ y_4 = x_6,\ y_5 = x_1 \oplus x_3.$$

TABLE V: The example function after decomposition.

(a) Function after decomposition number 1.

$idxs$	y_1	y_2	$G(Y)$
1, 7	0	0	0
3, 5	0	1	1
2, 8	1	0	0
4, 6	1	1	1

(b) Function after decomposition number 2.

$idxs$	y_1	y_2	$G(Y)$
1, 8	0	0	0
2, 7	0	1	0
4, 5	1	0	1
3, 6	1	1	1

Algorithm 1 Finding decomposition

```
$labels \leftarrow [\{i\} : i \in \{1, 2, \ldots, N\}]$
$dm, bins \leftarrow$ Algorithm_2()
while $True$ do
    $dec \leftarrow$ Algorithm_3($dm, bins$)
    if $dec$ is $None$ then
        break
    end if
    $dm \leftarrow$ Algorithm_4($dm, dec$)
    $labels \leftarrow$ Algorithm_5($labels, dec$)
end while
return $labels$
```

Algorithm 2 Generation of discernibility matrix

```
1:  $dm \leftarrow \emptyset$
2:  for $p \leftarrow 1$ to $K - 1$ do
3:      for $q \leftarrow p + 1$ to $K$ do
4:          if $F(p) \neq F(q)$ then
5:              $dm = dm \cup (v_p \oplus v_q)$
6:          end if
7:      end for
8:  end for
9:  $bins \leftarrow split\_to\_bins(dm)$
10: return $dm, bins$
```

Algorithm 3 Finding decomposition (single iteration)

```
1:  for $a \leftarrow 2$ to $N$ do
2:      if $|B_a| = \binom{n}{a}$ then
3:          continue
4:      else if $|B_a| = 0$ then
5:          return $\{x_i : i \in \{1, 2, \ldots, a\}\}$
6:      end if
7:      for $comb \in combinations(n, a)$ do
8:          if $\neg \exists v \in B_a : \forall c \in comb : v(c) = 1$ then
9:              return $\{x_i : i \in comb\}$
10:         end if
11:     end for
12: end for
13: return $None$
```

The complete algorithm is presented as Algorithm 1. Firstly, Algorithm 2 is used to generate a discernibility matrix. Array *labels* is used to represent the current form of a function. Based on that the Algorithm 3 is used to find a decomposition. If a decomposition is found, we need to modify the matrix (Algorithm 4) and labels (Algorithm 5, where Δ denotes the symmetric difference of two sets).

The proposed algorithm uses the discernibility matrix to find a possible representation of an input function using XOR gates. This matrix represents all generated $C_{p,q}$ sets for all possible pairs p and q ($p < q$). In Algorithm 2 a pseudocode for matrix generation is presented. Notice that in the 5th line, we check whether the value of the function differs between rows number p and q. If so, the value of the XOR operation of those rows is added to the matrix. An added row has ones on those positions where vectors number p and q differ, i.e. the $C_{p,q}$ set. In order to reduce the computational complexity of the whole algorithm, we analyze RC^r using the increasing value of r. Therefore, the rows from the matrix are divided into bins, based on their size, i.e. bin B_i represents RC^i. Furthermore, repeating values from each bin are removed.

The process of multilevel function minimization consists of iteration of the basic decomposition steps, presented as Algorithm 3. To speed up computation, we start each iteration by checking whether $|B_a| = \binom{n}{a}$ or $|B_a| = 0$. In the first case, $RC^a = \emptyset$. Thus, it is impossible to find a decomposition for that value of a. On the other hand, in the second case $C^a = \emptyset$. Thus, we can simply return $\{x_i : i \in \{1, 2, \ldots, a\}\}$. The found decomposition (lines 7-11) is represented as a vector. Indexes of bits set to 1 in this vector represent input variables that will be used to minimize an input function. For example, $comb = (11000)$ shows that XOR of the first and second input variables is a decomposition function of F, i.e.

$$F = G(x_3, x_4, x_5, x_1 \oplus x_2).$$

Notice that in this approach we return the first found function. Such an approach is called First-Fit [6]. Other approaches to selecting decomposition functions were proposed in the literature. However, the described one is the fastest and provides good results in terms of the solution quality (i.e., the number of variables). On the other hand, it generates slightly worse results for specific functions, e.g. *M-out-of-N* coders.

For example, if *comb* $= (11000)$, then we get the following content of *labels*: $[\{3\}, \{4\}, \{5\}, \{1, 2\}]$ that represents the function G mentioned above. Notice that found decomposition function (i.e., using variables x_1 and x_2) is used as the last input to the new representation, while the other input variables come before it. Therefore, the algorithm will more likely find a decomposition using input variables that have not been used in the previous iterations. In the result, the compound degree (i.e., the number of inputs to the XOR operation) of each variable y_i might be similar.

Since the number of $C_{p,q}$ sets strongly depends on the value of K, the proposed approach is very efficient especially if $K \ll 2^N$.

The described approach treats the value of a function as a single output. However, a multi-output logic function

$$F : B^N \rightarrow B^Q, \tag{4}$$

Algorithm 4 Modification of discernibility matrix

```
$jdx \leftarrow 1$
for $idx \leftarrow 1$ to $N$ do
    if $idx \notin dec$ then
        $new\_dm[:, jdx] = dm[:, idx]$
        $jdx \leftarrow jdx + 1$
    end if
end for
for $idx \leftarrow 1$ to $|dec| - 1$ do
    $col1 \leftarrow dm[:, dec[idx]]$
    $col2 \leftarrow dm[:, dec[idx + 1]]$
    $new\_dm[:, jdx] = col1 \oplus col2$
    $jdx \leftarrow jdx + 1$
end for
return $new\_dm$
```

Algorithm 5 Modification of labels

```
$new\_labels \leftarrow \emptyset$
for $idx \leftarrow 1$ to $|labels|$ do
    if $x_{idx} \notin dec$ then
        $new\_labels \leftarrow new\_labels \cup labels[idx]$
    end if
end for
for $idx \leftarrow 1$ to $|dec| - 1$ do
    $l1 \leftarrow labels[dec[idx]]$
    $l2 \leftarrow labels[dec[idx + 1]]$
    $new\_labels \leftarrow new\_labels \cup (l1 \Delta l2)$
end for
return $new\_labels$
```

Algorithm 6 Finding decomposition for each output

```
$res \leftarrow \emptyset$
for $i \leftarrow 1$ to $Q$ do
    $res = res \cup$ Algorithm_1$(X, Y[:, i])$
end for
return $res$
```

where $B = \{0, 1\}$ can be implemented using XOR-based decomposition to each output variable separately. Recall that we denote by Q the number of output variables. The proposed approach is presented as Algorithm 6. In that case, the final result is a composition of found decompositions, where each function returns a single bit value.

IV. Application in memory-based logic synthesis

In this paper, we applied the proposed approach to some well-known functions. The obtained results are presented in Table VI. The variable reduction algorithm was applied to all analyzed functions, and the total number of reducts and the size of the shortest one are both presented in the table. Each reduct was then used as an input function to our linear decomposition algorithm. For each function, we show how many decompositions with the specified value of output variables (P) were found. *xor5* function is presented here to prove that described method correctly finds decomposition with a single multi-input XOR gate.

In the table, the Δ_1 and Δ_2 columns display the reduction factor in memory usage. The reduction factor is calculated using two different scenarios. The first formula, which is calculated using the following equation:

$$\Delta_1 = 2^{N-P} \tag{5}$$

compares the memory usage of the implementation from an input function to the implementation using XOR-gate decomposition. The second formula, which is calculated using the following equation:

$$\Delta_2 = 2^{N'-P} \tag{6}$$

compares the memory usage obtained after variable reduction to the final implementation, where N' is the size of the shortest reduct. In both formulas, the smallest value of P obtained for a function is used. For example, $\Delta_1 = 2$ means that memory usage is halved compared to the direct implementation (i.e., without using a decomposition algorithm).

The proposed approach can not directly operate on *don't care* terms, since it relies on the XOR operation. Typically, such terms can be ignored, set to 0 or set to 1. For *add6* and *clpl* function, we set each *don't care* term to 0. For *9sym* function, we add rows to make it a complete function and analyzed all possible values of *don't care* terms.

As already mentioned, memory-based logic synthesis was often used in the implementation of index generation functions. a well-known example of such a function consists of ten 40-bit vectors [12] and a unique consecutive integer value from 1 to 10 is assigned for every vector. In this paper, we denote this function as *igf40*. Using the variable reduction algorithm [3], it is possible to find more than 2200 reducts with $N' = 4$, and more than 100k in total ($N' \in \{4, 5, 6, 7\}$). In this paper, we applied the described approach to all those functions after variable reduction. For example, in Table VIIa, we present a function after variable reduction, where $N' = 5$. In Table VIIb we present that function after linear decomposition, where $y_4 = x_4 \oplus x_{32}$ and $P = 4$. In the end, the number of variables was minimized to 4 for more than 99% of reducts.

The most striking observation to emerge from the results is that the proposed approach lead to significant memory usage minimization. It is because obtained values of P are smaller than both the number of input variables N and the sizes of the shortest reducts.

When Algorithm 1 fails for some functions, e.g. *rd53*, Algorithm 6 can be used. In that particular case, $N = 5$, $K = 32$, and $Q = 3$. Therefore, we can apply the proposed approach three times, each time focusing on a single output signal. This technique leads to decompositions with the following number of inputs: $P_1 = N = 5$, $P_2 = 4$, and $P_3 = 1$. In the end, the memory usage is minimized from $2^5 * 3 = 96$ bits to $2^5 + 2^4 + 2^1 = 50$ bits. Notice that the third function does not need to be implemented using memory, since the number of inputs after decomposition (i.e., P_3) equals 1.

TABLE VI: Experimental results

Database / function	N	K	no. of reducts	size of the shortest reduct	P	no. of decompositions	Δ_1	Δ_2
9sym	9	87 (512)	1	9	8	1	2^1	2^1
add6	12	432	1	12	10	1	2^2	2^2
br1	12	34	2	7	6	2	2^6	2^1
br2	12	35	3	8	6	3	2^6	2^2
clpl	11	20	9	8	4	5	2^7	2^4
					5	4		
igf40	40	10	100172	4	4	99552	2^{36}	0
					5	620		
house	17	232	4	8	6	2	2^9	2^2
					9	2		
kaz	22	31	5574	5	2	2	2^{20}	2^3
					3	43		
					4	677		
					5	4347		
					6	505		
xor5	5	16 (32)	1	5	1	1	2^4	2^4

TABLE VII: Decomposition of *igf40* function (single reduct).

(a) A function after variable reduction.

x_4	x_7	x_8	x_{26}	x_{32}	F
0	0	1	0	1	1
1	1	1	1	0	2
1	0	1	1	1	3
1	1	0	1	1	4
1	0	0	1	0	5
1	1	0	0	0	6
0	1	1	0	0	7
1	1	1	0	0	8
0	1	0	1	1	9
0	0	1	0	0	10

(b) A function after linear decomposition.

y_1	y_2	y_3	y_4	F'
0	1	0	1	1
1	1	1	1	2
0	1	1	0	3
1	0	1	0	4
0	0	1	1	5
1	0	0	1	6
1	1	0	0	7
1	1	0	1	8
1	0	1	1	9
0	1	0	0	10

TABLE VIII: Experimental results (the multi-output approach)

Database / function	N	M	MEM	Values of P	MEM'
adr4	8	5	1280	{8,7,6,4,2}	468
igf40 *{6,16,24,31}*	4	4	64	{4,3,4,3}	48
igf40 *{4,7,8,26,32}*	5	4	128	{1,3,3,3}	26
rd53	5	3	96	{5,4,1}	48
s27_split	7	4	512	{7,6,7,4}	336
z4	7	4	512	{7,5,4,2}	180

The size of the memory required to implement a function is calculated as

$$MEM = 2^N * Q. \tag{7}$$

Furthermore, the size of a memory required after the application of Algorithm 6 equals

$$MEM' = \sum_{i=1}^{Q} 2^{P_i}. \tag{8}$$

The experimental results obtained using the multioutput approach are presented in Table VIII. The original memory size and the memory size after decomposition, using equations (7) and (8) respectively, were both presented. The function *igf40 {6,16,24,31}* is a function *igf40* after variable reduction, where $P = Q = 4$ and the values in the brackets are indexes of input variables left after the reduction. Thus, the memory size can not be further minimized using that approach. However, the multioutput approach leads to lower memory consumption. We also present a second reduct, where $N' = 5 \neq Q$. Similar results were obtained for other analyzed functions, leading to a significant minimization of memory usage ($MEM' < MEM$).

V. Application in reversible logic synthesis

Recently, a novel three-level XOR-AND-XOR form was proposed [2] to represent autosymmetric functions. It extends a popular Exclusive Sum of Products (ESOP) form (i.e., XOR-AND form) by adding an additional XOR level to the representation. Using that form, a reversible circuit implementing a function can be easily constructed.

The first XOR level is used to compute reduction equations. In particular, it reduces the number of input variables to a f_k function using XOR gates. Therefore, this step is crucial in this technique and influences the Quantum Cost of the circuit at most. Notice that this step is analogous to the approach described earlier in this paper. It corresponds to the scheme presented in 1, where the linear function corresponds to the reduction equations and the general function corresponds to the restriction function f_k. Therefore, XOR-based decomposition can be used to find reduction equations. Recall from Section II-C that those equations can be implemented using several CNOT gates if we find a decomposition function with the value of r limited to two.

Secondly, a f_k function is implemented using any logic minimization tool. Notice that $k = N - P$. For example, an ABC [4] tool can be used to synthesize a function. *&esop* function derives ESOP from an AND-Inverter graph AIG, and *&exorcism* performs heuristic ESOP minimization [8]. This representation can be then used to find a reversible circuit that implements a f_k.

The final step is used to perform *uncomputation*, meaning that the original value is being restored on those lines that were affected by the reduction equation. It can be achieved by adding the CNOT gates used in the first level in reverse order.

Example 5.1: Consider the following function in ESOP form:

$$ESOP(F) = x_1x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_2x_6.$$

We apply the proposed approach to find reduction equations. In the first iteration of the algorithm, we get $x_3 \oplus x_4$ as a decomposition function (since all are smaller in lexicographic order two input functions are in RC^2). Therefore,

$$F(X) = G(x_1, x_2, x_5, x_6, x_3 \oplus x_4).$$

Applying the algorithm the second time, we get $x_5 \oplus x_6$. Since it leads to $COM(RC) = \emptyset$, the algorithm ends. In the end, we get the following reduction equations:

$$y_1 = x_1,\ y_2 = x_2,\ y_3 = x_3 \oplus x_4,\ y_4 = x_5 \oplus x_6.$$

In the result, we get the following ESOP representation of the restriction f_2 ($k = 6 - 4 = 2$):

$$ESOP(f_2) = y_1y_3 \oplus y_2y_4.$$

Thus, we get the following XORAX representation of F:

$$XORAX(F) = x_1(x_3 \oplus x_4) \oplus x_2(x_5 \oplus x_6).$$

It contains 6 literals and 2 products. Notice that the ESOP representation contains 8 literals and 4 products.

In Fig. 2 the obtained reversible circuit is presented. The first two CNOT gates are used to represent the reduction equations (i.e, $x_3 \oplus x_4$ and $x_5 \oplus x_6$). Next, two Toffoli gates represent the f_2 restriction based on its ESOP form. Finally, the last two CNOT gates recover the initial value of both x_4 and x_6 variables by applying XOR operations one more time. Four CNOT gates and two Toffoli gates are used in total. Therefore, the total Quantum Cost (QC) equals

$$QC = 4 * 1 + 2 * 5 = 14.$$

Notice that the straightforward implementation from an ESOP form requires four Toffoli gates, where each of them realizes the single product of two variables. Thus, its Quantum Cost equals 20.

The proposed approach can also be combined with a functional decomposition technique [10]. This method decomposes an input function into smaller irreversible functions that are connected, and the original function is preserved. The most important fact is that each function is smaller (in terms of the number of inputs), which makes it easier to synthesize. This technique was also used in memory-based logic synthesis to reduce memory consumption [7].

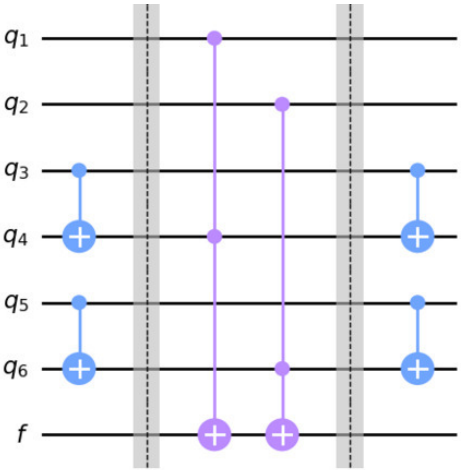


Fig. 2: Reversible circuit derived from a XORAX representation.

Each of the smaller functions is synthesized to get a reversible circuit and then combined into a single circuit. Because irreversible functions are implemented, a *garbage* output is introduced. In particular, the original value of the signal does not have to be restored on a line that was affected by reversible gates.

Example 5.2: Consider a function presented in Table IX. Using the ABC tool, we get a circuit, where Quantum Cost equals 120. Using functional decomposition it is possible, to divide this function into two functions (denoted G and H). The first function has three inputs (x_3, x_4, x_5) and one output (g). The second one has four inputs (x_1, x_2, x_6, g) and one output. The mapping between inputs and output for both functions can be found using a graph colouring method [7]. Truth tables for both functions are presented in Table X. Using the ABC tool, we get Quantum Cost 30 and 38 respectively, meaning 68 in total.

TABLE IX: An example function.

x_1	x_2	x_3	x_4	x_5	x_6	$F(X)$
0	0	0	0	0	0	1
0	1	0	0	1	0	1
0	1	0	1	1	0	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
1	0	1	1	0	0	0
0	1	0	0	0	0	0
1	0	1	0	0	0	1
0	1	0	1	0	0	0

Both functions can be decomposed using the XOR-based approach. For function G, we get $\{x_3, x_4, x_5\} \in COM(RC)$. Therefore,

$$y_1 = x_3 \oplus x_4,\ y_2 = x_4 \oplus x_5.$$

Linear function can be implemented using a single CNOT gate,

TABLE X: Functional decomposition of a function.

(a) Function G.

x_3	x_4	x_5	G
0	0	0	0
0	0	1	1
0	1	1	0
1	1	0	1
0	1	0	0
1	0	0	0

(b) Function H.

x_1	x_2	x_6	g	H
1	0	0	0	1
0	1	0	0	0
1	0	0	1	0
0	0	1	1	0
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1

while the general function synthesized using the ABC tool has $QC = 6$.

For function H, we get $\{x_2, g\} \in COM(RC)$. Therefore,

$$y'_1 = x_1,\; y'_2 = x_6,\; y'_3 = x_2 \oplus g.$$

In that case, the linear function can be implemented using two CNOT gates. The general function synthesized using the ABC tool has $QC = 5$. In total, Quantum Cost was minimized from 68 to 14.

Interestingly, this specific input function can be directly decomposed using the proposed approach. In that case, after four iterations of Algorithm 1, we get:

$$y_1 = x_4 \oplus x_6,\; y_2 = x_1 \oplus x_2 \oplus x_3 \oplus x_5.$$

Those equations can be implemented using four CNOT gates (i.e., one for y_1 and three for y_2), while the general function has $QC = 6$, leading to $QC = 10$ in total.

VI. Conclusion

In this paper, we showed how XOR-based decomposition, which has been previously used in index generation functions synthesis, can be generalized and efficiently applied in memory-based and reversible logic synthesis. We provided a complete algorithm and used well-known benchmark functions to prove that it can provide significant minimization of memory usage. The presented results highlight the potential of this technique for memory-based logic synthesis. Furthermore, we showed that our algorithm can be easily combined with other techniques proposed in the literature to achieve promising results in reversible logic synthesis. Based on our research we believe that XOR-based decomposition might become a valuable tool for other researchers and practitioners.

References

[1] Bennett C. H., "Logical Reversibility of Computation," in: *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525–532, 1973, https://doi.org/10.1147/rd.176.0525.

[2] Bernasconi A., Berti A., Ciriani V., Corso G. D. and Fulginiti I., "XOR-AND-XOR Logic Forms for Autosymmetric Functions and Applications to Quantum Computing," in: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, https://dx.doi.org/10.1109/TCAD.2022.3213214.

[3] Borowik G. and Łuba T., "Fast Algorithm of Attribute Reduction Based on the Complementation of Boolean Function," *Advanced Methods and Applications in Computational Intelligence*, 2014, pp. 25–41, https://dx.doi.org/10.1007/978-3-319-01436-4_2.

[4] Brayton, R., Mishchenko, A., "ABC: An Academic Industrial-Strength Verification Tool, " in: *Computer Aided Verification. CAV 2010. Lecture Notes in Computer Science, vol 6174*. Springer, Berlin, Heidelberg. https://dx.doi.org/10.1007/978-3-642-14295-6_5.

[5] Czajkowski T. S. and Brown S. D., "Functionally linear decomposition and synthesis of logic circuits for FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 12, pp. 2236–2249, https://dx.doi.org/10.1109/TCAD.2008.2006144.

[6] Mazurkiewicz T. and Łuba T., "Linear and Non-linear Decomposition of Index Generation Functions," *in 26th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, 2019, pp. 246–251, https://dx.doi.org/10.23919/MIXDES.2019.8787031.

[7] Mazurkiewicz T., "Non-disjoint functional decomposition of index generation functions," *IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, Miyazaki, Japan, 2020, pp. 137-142, https://dx.doi.org/10.1109/ISMVL49045.2020.00-16.

[8] Mishchenko A. and Perkowski M., "Fast Heuristic Minimization of Exclusive-Sums-of-Products," in: *5th International Reed-Muller Workshop*, 2001.

[9] Łuba T., Borowik G. and Jankowski C., "Gate-based decomposition of index generation functions", in: *Proc. SPIE. 10031, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2016. Vol. 10031. SPIE*, 2016. pp. 100314A–1–100314A–10, https://dx.doi.org/10.1117/12.2248754.

[10] Rawski M. and Szotkowski P., "Reversible logic synthesis of boolean functions using functional decomposition," *22nd International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)*, Torun, Poland, 2015, pp. 380-385, https://dx.doi.org/10.1109/MIXDES.2015.7208547.

[11] Sasao T., "Memory-Based Logic Synthesis," *Computer Science*, Springer, 2011, https://dx.doi.org/10.1007/978-1-4419-8104-2.

[12] Sasao T., "Index Generation Functions, Logic Synthesis for Pattern Matching", EPFL Workshop on Logic Synthesis & Verification, Lausanne, Switzerland, 2015.

[13] Sasao T., "Index Generation Functions," *Synthesis Lectures on Digital Circuits & Systems*, Springer Cham, 2020, https://dx.doi.org/10.1007/978-3-031-79911-2.

[14] Toffoli, T., "Reversible Computing," in: *Proceedings of the 7th Colloquium on Automata, Languages and Programming*, Springer-Verlag, pp. 632–644, http://doi.org/10.1007/3-540-10003-2_104.

DOI: 10.15439/2023F444
ISSN 2300-5963 ACSIS, Vol. 35

Balancing Privacy and Accuracy in Federated Learning for Speech Emotion Recognition

Samaneh Mohammadi[1,3], Mohammadreza Mohammadi[1,2], Sima Sinaei[1], Ali Balador[3], Ehsan Nowroozi[4], Francesco Flammini[3], and Mauro Conti[2]
[1] RISE Research Institutes of Sweden, Västerås, Sweden.
Email: {samaneh.mohammadi, mohammadreza.mohammadi, sima.sinaei}@ri.se
[2] University of Padua, Padua, Italy. Email: mauro.conti@unipd.it
[3] Mälardalen University, Västerås, Sweden. Email: {ali.balador, francesco.flammini}@mdu.se
[4] Queen's University Belfast, Centre of Secure Information Technologies, Belfast, Northern Ireland, United Kingdom. Email: e.nowroozi@qub.ac.uk

***Abstract*—*Context:* Speech Emotion Recognition (SER) is a valuable technology that identifies human emotions from spoken language, enabling the development of context-aware and personalized intelligent systems. To protect user privacy, Federated Learning (FL) has been introduced, enabling local training of models on user devices. However, FL raises concerns about the potential exposure of sensitive information from local model parameters, which is especially critical in applications like SER that involve personal voice data. Local Differential Privacy (LDP) has prevented privacy leaks in image and video data. However, it encounters notable accuracy degradation when applied to speech data, especially in the presence of high noise levels. In this paper, we propose an approach called LDP-FL with CSS, which combines LDP with a novel client selection strategy (CSS). By leveraging CSS, we aim to improve the representatives of updates and mitigate the adverse effects of noise on SER accuracy while ensuring client privacy through LDP. Furthermore, we conducted model inversion attacks to evaluate the robustness of LDP-FL in preserving privacy. These attacks involved an adversary attempting to reconstruct individuals' voice samples using the output labels provided by the SER model. The evaluation results reveal that LDP-FL with CSS achieved an accuracy of 65-70%, which is 4% lower than the initial SER model accuracy. Furthermore, LDP-FL demonstrated exceptional resilience against model inversion attacks, outperforming the non-LDP method by a factor of 10. Overall, our analysis emphasizes the importance of achieving a balance between privacy and accuracy in accordance with the requirements of the SER application.**

***Index Terms*—Federated Learning, Privacy-preserving Mechanism, Differential Privacy, Speech Emotion Recognition**

I. Introduction

SPEECH Emotion Recognition (SER) is a cutting-edge technology that detects and interprets emotions conveyed through spoken language [1]. Its potential impact spans multiple sectors, including customer service, mental health, education, and entertainment [2], [3]. However, the traditional centralized SER model, which involves gathering user speech data and training a single model on a central server, poses privacy risks. Analyzing speech data can expose sensitive information like biometric identity, personality traits, location, emotional state, age, gender, and overall health [4]. To address these ethical and privacy concerns, regulation like the General Data Protection Regulation (GDPR) [5] has been implemented to safeguard personal data. Privacy must be a top priority when developing and implementing SER applications across various domains.

Federated learning (FL) has emerged as a promising solution to privacy concerns in various fields and applications [6]. FL maintains local data on end devices and trains ML models on local client devices without transferring raw data to a central server. This preserves data privacy and ensures compliance with regulations such as GDPR. For SER applications, the initial processing of speech data and training perform on clients' devices, and only local model parameters are sent to the central server for model aggregation [7].

However, FL faces new privacy concerns when it comes to transmitting local model parameters between clients and servers [8]. This is a significant concern because the transmission parameters can be exploited by third parties, enabling them to launch attacks that reconstruct raw speech data or features, thereby revealing sensitive information [9]. To address this issue, additional privacy mechanisms have been proposed together with FL to safeguard such applications.

One promising mechanism that has emerged is differential privacy (DP), which offers a potential solution to protect individual data points in certain cases. DP can be implemented in FL through two forms: local differential privacy (LDP) applied on the client side and global differential privacy (GDP) used on the server side [10]. LDP, a well-established technique, involves the addition of carefully calibrated noise to each client's model parameters before transmitting them to the central server [11], [12], [13].

Integrating LDP in FL for SER applications offers several benefits. It ensures the privacy and confidentiality of individual speech data, protecting sensitive information from unauthorized access and potential attacks. By introducing noise to the model parameters, LDP prevents the reconstruction of raw speech data or features by malicious third parties, thereby preserving users' privacy. Despite the promising potential of LDP in FL for SER applications, there is a noticeable lack of research published in reputable conferences or journals specifically addressing the utilization of LDP in this context. This gap highlights the need for further investigation and exploration to fully understand the effectiveness and practical implications of integrating LDP into FL for SER.

However, when applied to SER applications, LDP does not offer acceptable accuracy due to the adverse effects of adding noise to voice data, which can distort the audio signal [14]. Furthermore, adding noise to SER model parameters can affect the model's utility by distorting or misaligning the parameters, leading to errors in the model's output [15]. This compromise in accuracy is especially detrimental to most SER applications, which rely on precise results for industrial use [16]. Therefore, when developing LDP mechanisms in FL for SER, it is necessary to find a concrete solution that effectively mitigates the impact of noise on SER accuracy while still maintaining robust privacy protections.

This paper proposes a method, referred to as LDP-FL with CSS, which integrates LDP with a novel client selection strategy (CSS) to enhance privacy while preserving the acceptable accuracy of SER in the FL system. LDP is utilized to protect clients' speech datasets, while CSS is employed to minimize the negative impact of noise scaling on the model updates, resulting in more representative updates and improved accuracy.

Moreover, our study focuses on adapting the model inversion attack, initially developed for facial recognition models [17], for the SER model through appropriate configuration adjustments. This attack attempts to reconstruct speech features by considering the adversary's knowledge of a particular client's emotion label and their local SER model. The primary goal is to evaluate the effectiveness of the LDP method in safeguarding against such attacks within the FL setup. Finally, we comprehensively evaluated the LDP-FL with CSS approach, specifically focusing on assessing its alignment with SER requirements and analyzing the trade-off between accuracy and privacy.

The novel contributions of this paper can be summarized as follows:

- We introduce a novel approach that combines local differential privacy in federated learning (LDP-FL) with a client selection strategy (CSS) to enhance privacy while mitigating the impact of noise on SER accuracy.
- We implement model inversion attacks to assess the robustness of LDP-FL and determine its effectiveness in preserving privacy. These attacks involve an adversary attempt to reconstruct individuals' voice samples based on the output labels provided by the SER model.
- We conduct a comprehensive evaluation of the LDP-FL with CSS approach on public SER datasets, considering important parameters such as privacy budget, noise scale, failure probability, and clipping threshold value. Our evaluation focuses on assessing how well our method meets SER requirements and analyzing the balance between accuracy and privacy.

The rest of the paper is structured as follows. Section II covers background and related work on privacy-preserving FL and SER. A reference system description is provided in Section III, including the SER non-functional requirements, threat model, the proposed method, and implementation of the model inversion attack. Section IV presents the experimental results obtained using the proposed approach. Lastly, Section V concludes the paper and provides insights for future developments.

II. Background and Related Works

In this section, we will provide an overview of the background and related work on LDP mechanisms in FL. We will then discuss the use of FL for SER applications and its related work.

A. Privacy-preserving Federated Learning

FL protects user privacy by decentralizing data from the central server to edge devices; however, sharing information with servers (e.g., model weights) can pose privacy threats [8]. Since FL requires central servers and clients to exchange model update parameters, attackers with white-box access obtain the model, its architecture, weight parameters, and any hyperparameters needed for predictions. When using black-box scenarios, the adversary can observe only the outputs of the model on arbitrary inputs [9].

LDP has become an increasingly popular technique for privacy-preserving in FL [10]. LDP can prevent individual devices' data from being leaked to the central server during the model training process [11], [12]. This technique involves adding artificial noise to each model's updated parameters before sharing it with the central server. Recent work proposed a framework called NbAFL that utilized LDP and demonstrated its capability to meet DP requirements under different protection levels by appropriately adapting various variances of artificial noise [12]. Another study proposed LDP-based stochastic gradient descent (SGD) that guarantees a given LDP level by providing a noise variance limit after multiple rounds of weight updates using a tight composition theorem [13].

B. Speech Emotion Recognition using Federated Learning

SER technology aims to recognize and understand human emotions through speech. SER systems analyze the audio signals from human speech and use ML algorithms to detect patterns and classify the emotional states conveyed by the speech [2]. Building SER models requires significant amounts of data, including sensitive personal information such as speech signals and emotions. However, centralized storage of this data presents privacy risks. To mitigate these risks, FL is a promising solution that allows models to be trained collaboratively on decentralized devices without the need to transfer raw data [7].

The paper [18] introduces an FL-based approach for building a private decentralized SER model. The proposed method utilizes data-efficient federated self-training to train SER models with minimal on-device labelled samples. However, the proposed method only relies on the FL framework as a privacy-preserving technique and does not consider any

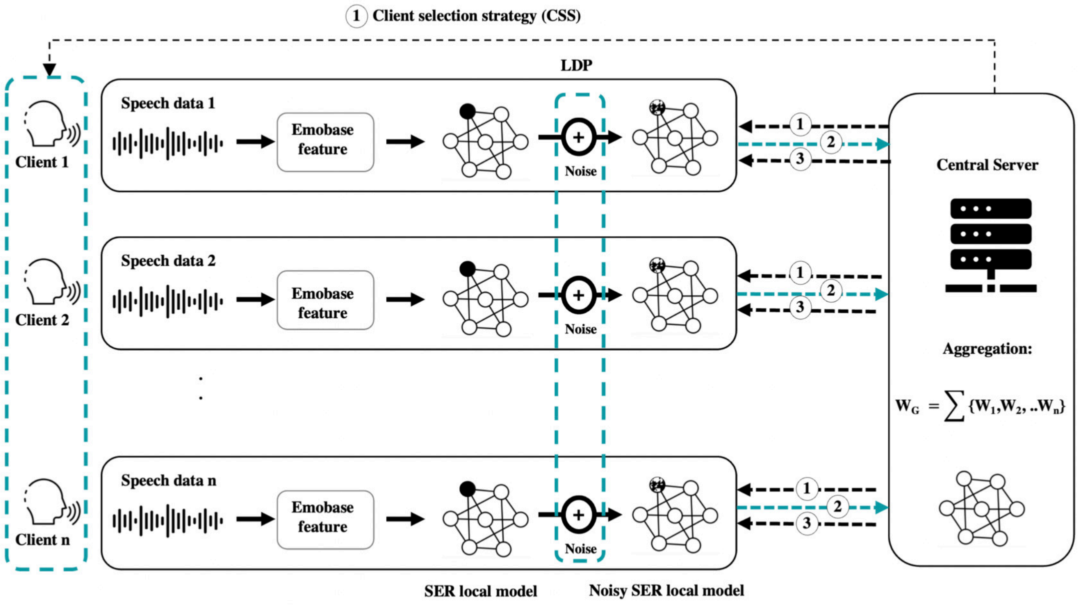


Fig. 1: An overall overview of LDP-FL with CSS for SER application.

threat models from clients or servers in FL, nor does it consider any other privacy-preserving techniques. Similarly, another work [19] proposes a federated adversarial learning framework to protect both data and deep neural networks in SER. The framework comprises an FL framework for data privacy and adversarial training during the training stage for model robustness. However, like the previous method, it only relies on the FL framework for privacy preservation and does not consider other privacy-preserving techniques in FL.

III. System Description

In this section, we will address the non-functional requirements of the SER application (III-A), discuss the associated threat model (III-B), present the proposed LDP-FL with CSS method (III-C) along with algorithms and details, and finally describe the model inversion attack for speech features using algorithms (III-D).

A. Non-functional Requirements of Speech Emotion Recognition Application

Non-functional requirements refer to the characteristics or qualities of a system that are related to its performance rather than its specific functionality. In the context of SER applications, important non-functional requirements include privacy and accuracy. Satisfying these requirements is critical to ensure user needs and expectations while complying with legal requirements. In this part, we provide a more detailed explanation and explain how we meet these requirements in the evaluation section IV.

1) *Privacy:*
   a) Personal speech data must be kept on local devices only [5].
   b) The central server or a potential eavesdropper must be not able to infer sensitive information from the local model parameters
2) *Accuracy:*
   a) The level of accuracy of SER applications must be kept high enough to identify the correct emotions from speech samples reliably. We can consider a baseline accuracy of a minimum 70% in detecting the four basic emotions - neutral, sad, happy, and angry [20].

It is important to highlight that those requirements can be highly interdependent. For instance, privacy-preserving approaches can have an impact on accuracy due to e.g. the distributed setup the usage of FL, applied noise of the LDP method, etc. Additionally, when implementing a SER in an FL setup, it has been demonstrated in reference [18] that there is a potential for a 0-5% accuracy drop.

B. Threat Model

In this paper, we assume the server follows the honest-but-curious (HBC) paradigm. Under this paradigm, the server is not malicious and adheres to the FL protocol, but may still possess a curiosity about the data or models of other clients [21]. While the individual client datasets are kept locally in FL, the intermediate parameter w_i needs to be shared with the server, which can potentially expose clients' private information, as evidenced by model inversion attacks. For instance, in [17], researchers demonstrated a model inversion attack capable of reconstructing images from a facial recognition system.

C. Proposed Method: LDP-FL with CSS

We present "LDP-FL With CSS," a novel approach that combines local differential privacy (LDP) and a client selection strategy (CSS) in federated learning to balance client privacy and the accuracy of the SER model. Our method addresses privacy requirement 1.a by processing and training clients' speech data locally on their devices within the FL setup. By leveraging LDP techniques, Gaussian noise is incorporated into the local updates before transmitting them to the central server. This implementation provides robust protection against the inference of sensitive information, thereby fulfilling the requirement of 1.b and mitigating potential risks in the threat model. To meet the accuracy requirements 2.a, we incorporate CSS, prioritizing clients with larger data pools and involving them in each training round. This strategy aims to mitigate the potential negative impact of LDP on accuracy while enhancing it to meet the desired accuracy levels.

Figure 1 outlines the proposed method, which consists of three main steps. Firstly, the server broadcasts the initialized global SER model and uses the CSS method to select clients for training. In the second step, the chosen clients analyze speech data, extracting Emobase features (explained in Sec. IV-A) and train their models. They also update their local model parameters using the global model. Privacy is ensured by applying the LDP method to each parameter before sharing them with the server. The clients then share the noisy model parameters with the server. In the third step, the server aggregates the received noisy local model parameters and returns the updated global model to the clients. We provide Algorithm 1 as a comprehensive outline of the LDP-FL with CSS approach. Subsequently, we will delve deeper into the concepts of LDP and CSS in the context of FL.

Algorithm 1: LDP-FL with CSS

```
   Input: Number of iterations: T, Number of selected
          clients: K, Local minibatch size: B, Initial
          global model: $w_0$, Learning rate: $\eta$, Clipping
          threshold: C, LDP parameters: $\epsilon$ and $\delta$
1  Initialization:
2  Initialize the global model parameters $w_0$
   for $t \leq T$ do
3      The server broadcasts current model $w_t$
4      K: Client Selection Strategy (CSS)
5      Clients-side:
       for $i \in 1, 2, ..., K$ do
6          for each batch $b \in B_i$ do
7              Compute gradient $g(b) \leftarrow \nabla_w L^i(w_t; b)$
8          Clip gradient $\overline{g}(b) \leftarrow g(b)/Max(1, \frac{\|g(b)\|}{C})$
9          Add Noise
           $\tilde{g_i} = \frac{1}{|B|}(\sum_{b \in B} \overline{g}(b) + N(0, \sigma^2 C^2 I)$
10         Share $\tilde{g_i}$ with server
11     Server-side:
12     Aggregate $\tilde{g} = \frac{1}{K}\sum_{i=1}^{K} \tilde{g_i}$
13     Global model update $w_{t+1} \leftarrow w_t - \eta.\tilde{g}$
```

1) Local Differential Privacy (LDP): LDP is defined under the setting where the user does not trust anyone (not even the central data collector) [11]. In this setting, users themselves apply a random perturbation to protect their privacy. Each user runs a random perturbation algorithm, denoted as M, on their data and shares the perturbed results with the aggregator or central server. In LDP, the privacy budget, denoted as ϵ, represents the amount of privacy protection desired, with a higher value of ϵ implying a lower level of privacy. While δ represents the probability that an LDP mechanism fails to provide the specified privacy guarantee. Here is a formal definition of LDP:

Definition 1 ((ϵ, δ)-LDP [22]): A randomized mechanism M satisfies (ϵ, δ)-LDP if and only if for any pairs of input values v and v' in the domain of M, and for any possible output $y \in$ S , it holds:

$$Pr[M(v) = y] \leq e^{\epsilon} Pr[M(v') = y] + \delta. \tag{1}$$

Theoretically, (ϵ, δ)-LDP means that a mechanism M achieves (ϵ, δ)-LDP with probability at least $1 - \delta$.

To implement the LDP mechanism in a FL setup, we followed the approach described in reference [23]. Specifically, we incorporated artificial Gaussian noise into the clients' model parameters. In order to ensure that the given noise distribution $Z \sim N(0, \sigma^2 C^2 I)$ preserves (ϵ, δ)-LDP, for any $\epsilon < cq^2 T$, $\delta > 0$, and T number of epoch, we choose noise scale $\sigma \geq c\frac{q\sqrt{Tlog(1/\delta)}}{\epsilon}$, where the constant c and sampling probability q. In this result, Z is the value of an additive noise for client gradient.

In Algorithm 1, during time slot t, each selected client $i \in k$ trains its local dataset by minimizing the loss function ∇L^i (lines 6-8). For each client, the gradient $g(b)$ is calculated for each $b \in B_i$. To limit the impact of each gradient $g(b)$, we apply clipping using the $\|L\|_2$ norm. Specifically, $g(b)$ is replaced by $g(b)/\max(1, \frac{\|g(b)\|_2}{C})$ where C is the clipping threshold (line 7). This clipping mechanism ensures that if the norm $\|g\|_2$ is less than or equal to C, the gradient g remains unchanged. However, if $\|g\|_2$ exceeds C, it is scaled down to have a norm of C, thereby controlling the contribution of large gradients.

After clipping the gradient, we compute the average of all gradients in set B and add a scaled Gaussian noise $Z \sim N(0, \sigma^2 C^2 I)$ to each client's gradient to achieve LDP in lines 9-10. The resulting noisy gradient $\tilde{g_i}$ is then shared with the server in line 11. On the server side, upon receiving the noisy gradients $\tilde{g_i}$ from the selected clients, the server performs the FedSGD algorithm by aggregating the gradients $\tilde{g} = \frac{1}{K}\sum_{i=1}^{K} \tilde{g_i}$. Subsequently, the global model is updated using $W_{t+1} \leftarrow W_t - \eta \cdot \tilde{g}$ and utilized for the next iteration in lines 13-14.

2) Clients Selection Strategy (CSS): To mitigate potential noise effects and uphold the initial accuracy of SER models in FL, we introduce a refined client selection strategy named

(CSS). Our proposed approach involves carefully selecting clients for FL training, employing two distinct criteria.

Algorithm 2: Clients Selection Strategy (CSS)

```
Input: Number of iterations: $T$, Clients list: $L$,
       Number of selected Clients: $K$
Output: List of selected clients
1 for $t \leq T$ do
2     Half of selection: $M = K/2$
3     $\mathcal{C}$ = sorted $L$ in descending order by sample size
4     Selected clients = $\mathcal{C}$ [:M]
5     Remaining clients = randomly select $\mathcal{C}$ [M:]
6 Return K = selected clients + remaining clients
```

Firstly, we select half of the clients from a larger pool of candidates based on their sample size. This criterion ensures that clients with larger local datasets are given preference. By incorporating larger local datasets, which are more likely to yield accurate and representative model updates, we aim to enhance the overall model accuracy. Secondly, to mitigate selection bias, the remaining half of the clients are randomly chosen. This random selection mechanism introduces an element of diversity and reduces the potential bias that could arise from selecting clients based solely on their sample size.

Algorithm 2 outlines the client selection strategy (CSS) method used in each training round of the overall Algorithm 1. Our method selects the top half of the clients based on their sample size, giving those clients with the largest sample sizes a higher probability of being chosen for each round of training (line 4). To reduce bias in client selection, we combine our proposed method with random selection for the remaining clients (line 5). We then combine the two sets of selected clients to obtain the final selection (line 6).

We ensure that there is no overlap between the two sets of selected clients to guarantee that each client is selected precisely once per training round. By employing the CSS approach, we strike a balance between leveraging large local datasets for training and maintaining diversity within the FL system. This methodology effectively minimizes noise effects and fosters the preservation of the initial model accuracy in SER models trained through FL.

D. Model Inversion Attack for Speech Emotion Recognition Models

A model inversion attack takes place when an adversary gains access to a model's output and potentially its parameters, aiming to infer sensitive training data. In our paper, we adjust the existing work conducted in the field of face recognition [17] and adapt it for speech emotion recognition by changing some configurations. In this scenario, we assume that the attacker possesses knowledge of a single emotion label, such as neutral, sad, happy, or angry, as well as the model used by the clients. The objective of the adversary is to reconstruct the speech data features associated with a specific client and the corresponding emotion label.

The target of model inversion attack in this case is the inversion of speech features, which represent high-level statistical characteristics of a client's speech. Each intensity value in the features corresponds to a floating-point value. In our attack scenarios, we assume that the attacker does not possess exact knowledge of the feature values they are trying to infer. We consider feature vectors with n components and four emotion label classes, and we model each emotion recognition classifier as a function $\tilde{f} : [0,1]^n \rightarrow [0,1]^4$. The output of the model is a probability vector, where each component represents the probability that the feature vector belongs to a specific emotion label. We use the notation $\tilde{f}_{\text{label}}(x)$ to refer to the ith component of the output corresponding to the emotion label. The Algorithm 3 provides a comprehensive outline of the model inversion attack specifically designed for speech emotion recognition models.

Algorithm 3: Model inversion attack for speech emotion recognition models

```
Input: Number of iteration: $T$, Best score: $\gamma$, Target
       model: $\tilde{f}$, Learning rate: $\eta$
Output: Related speech features to target label
$c = 1 - \tilde{f}_{label}(x)$
$x_0 \leftarrow 0$
for $t \leq T$ do
    $x_t \leftarrow \text{Process}(x_{t-1} - \eta \cdot \nabla c(x_{t-1}))$
    if $c(x_t) \geq \max(c(x_{t-1}), \ldots, c(x_{t-\beta}))$ then
        break
    if $c(x_t) \leq \gamma$ then
        break
return $[\arg\min_{x_t}(c(x_t)), \min_{x_t}(c(x_i))]$
```

The algorithm utilizes gradient descent to minimize a cost function involving the emotion recognition model $\tilde{f}$ for model inversion. Gradient descent iteratively updates a candidate solution by moving towards the negative gradient direction. The cost function, denoted as c, is defined based on $\tilde{f}$. The model inversion attacks employ gradient descent for a maximum of T iterations with a step size of η. After each iteration, the resulting feature vector is processed using a post-processing function called Process, which can apply various manipulations to the speech features, such as denoising and sharpening, depending on the specific attack. The descent terminates if the cost does not improve within β iterations or if the cost exceeds a threshold γ. In such cases, the best candidate is returned as a result.

IV. Experiment Results

In this section, we present an industrial use case and the simulation setting. We evaluate the impact of the LDP-FL method on SER accuracy, considering parameters like noise scale, failure probability, and clipping threshold. We analyze the effect of CSS on SER accuracy within the LDP-FL framework and investigate the robustness of LDP-FL against model inversion attacks. Finally, we discuss the crucial task of achieving the optimal balance between privacy levels and accuracy.

A. Usecase Description and Simulation Setting

DAIS[1] (Distributed Artificial Intelligent System) [24] is a pan-European project that aims to provide trustworthy connectivity and interoperability by combining the IoT with AI into a distributed edge system for industrial applications. The project includes industry-driven use cases in domains such as digital life, digital industry, and smart mobility. One of the important use cases in DAIS is SER which is deployed on TV recommendation systems. The goal is to accurately capture users' emotions and provide personalized movie recommendations, leading to higher levels of user satisfaction. Achieving this requires a distributed, efficient, private, and accurate SER application. This was one of the main motivations for exploring the potential of LDP-FL with CSS in SER.

As part of this study, we evaluated the proposed method on one of the most widely used SER datasets, namely CREMA-D [25]. CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from various races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified). To train the SER model, we chose the four most commonly occurring emotion labels (neutral, sad, happy, and angry) based on the possible emotions expressed in the sentences.

For speech processing and feature extraction, we generate the Emo-Base feature set using the OpenSMILE toolkit [26]. The Emo-Base feature set is a widely used set of features for SER tasks. These features are extracted from the speech signal and capture various acoustic characteristics of the signal that are associated with different emotions. The features are designed to be highly discriminative for emotion recognition and have been shown to achieve state-of-the-art performance in various SER tasks. After extracting the features, we utilized a multilayer perceptron (MLP) architecture for the SER model and trained it using the FedSGD algorithms. The model consists of two dense layers with layer sizes of [256, 128] and ReLU activation function, along with a 0.2 dropout rate. We set a local training batch size of 20 and a learning rate of 0.1 to accelerate convergence in the FedSGD algorithm.

For the FL training on the CREMA-D dataset, each speaker serves as a unique client since there are 91 distinct speakers in the dataset. We employed 80% of the data for local training at each client and reserved the remaining 20% for validation. To ensure the robustness of our approach, we conducted five experiments with different test folds, and we reported the average results of the five-fold experiments. The FL scenarios were conducted over 200 global training epochs. Our experiments were conducted on a Windows 10 Pro environment, featuring an Intel(R) Core(TM) i7 CPU @1.80GHz 1.99 GHz processor and 16.0 GB of RAM.

B. SER accuracy results across different parameters: noise scale, failure probability and clipping threshold

We conducted an analysis to assess the accuracy of SER in LDP-FL by examining the impact of various LDP parameters on accuracy. Our evaluation involved 50 randomly selected clients and 120 training epochs, as depicted in Figure 2. Specifically, we investigated the influence of the noise scale σ on accuracy (Figure 2(a)), the effect of varying failure probability δ on accuracy (Figure 2(b)), and the impact of the clipping threshold C on accuracy (Figure 2(c)).

The experimental results illustrated in Figure 2(a) indicate that the accuracy of LDP-FL gradually stabilizes with an increase in the number of training epochs, indicating convergence of the method. However, higher noise scales, such as $\sigma = 10$, can impede convergence due to the injection of larger amounts of noise during training, resulting in an unstable system. Figure 2(b) demonstrates that higher failure probabilities δ lead to faster convergence and higher accuracy but weaker privacy protection. Conversely, lower failure probabilities provide stronger privacy guarantees at the expense of reduced accuracy. For instance, a failure probability of $\delta = 10^{-3}$ achieved the highest accuracy while sacrificing some privacy for utility.

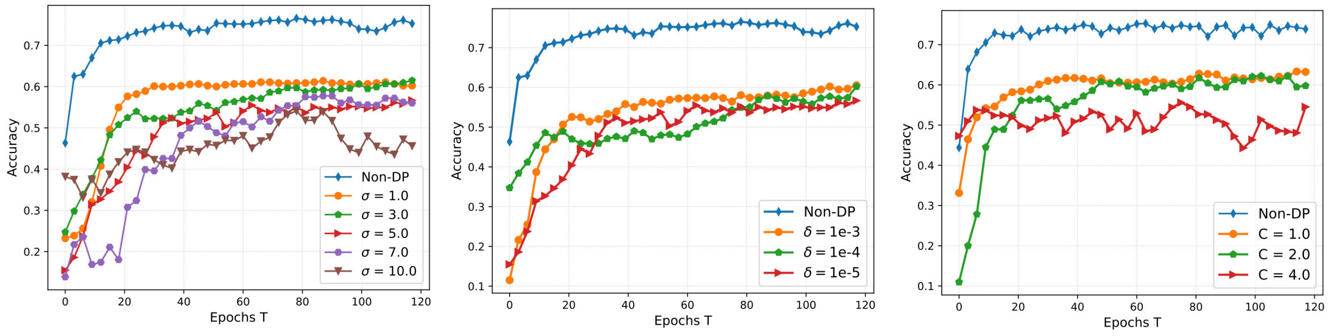
Our evaluation of LDP-FL's accuracy with different clipping thresholds showed that a threshold of 1.0 or 2.0 achieves high accuracy with fast convergence, as shown in Fig. 2(c). However, using a threshold beyond a certain point results in decreased accuracy due to excessive information loss during the clipping process. Hence, selecting the optimal clipping threshold is crucial to balance privacy preservation and model accuracy.

C. Effect of CSS on SER accuracy

To evaluate the effectiveness of LDP-FL with CSS for SER application, we conducted a comparative study between CSS and the commonly used random selection (RS) method in both LDP and non-LDP FL systems. Using parameters $\sigma = 1.0$, $C = 2$, $\delta = 10^{-5}$, and $K = 50$, we observed a significant improvement in accuracy from 60% to 70% when using CSS with LDP, as depicted in Figure 3 and meeting the accuracy requirements outlined in Section III-A. CSS proved to be an effective method for selecting clients, leading to more representative and larger datasets for training, resulting in more robust and accurate models. However, it is important to note that selecting clients with larger local datasets increases their exposure, potentially leading to data leakage. Therefore, a balance must be struck when employing CSS.

Interestingly, we observed that the choice of client selection method did not significantly impact the accuracy of non-LDP FL systems. This suggests that the accuracy improvement achieved by CSS is specific to the LDP-FL. Thus, adopting an

[1] https://dais-project.eu/



(a) (b) (c)

Fig. 2: Evaluation of the accuracy of the SER model using LDP-FL across different parameters: (a) noise scale σ, (b) failure probability δ, and (c) clipping threshold C.

TABLE I: MSE of reconstruction of speech features by model inversion attack in cases where FL has LDP or does not have LDP

Target model	**Clipping Threshold (C)**	**Mean Squared Error (MSE)**					
		LDP-FL					**Non-LDP-FL**
		$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\sigma = 7$	$\sigma = 10$	-
Client SER model	C=1	0.971	1.189	19.830	45.132	157.640	1.02
	C=2	1.028	9.189	78.910	1139.474	5807.308	1.02
	C=4	1.886	344.000	8508.620	59164.757	572765.191	1.02

efficient client selection strategy like CSS can be a valuable technique to enhance the performance of LDP-FL and mitigate the potential negative impact of LDP on accuracy.

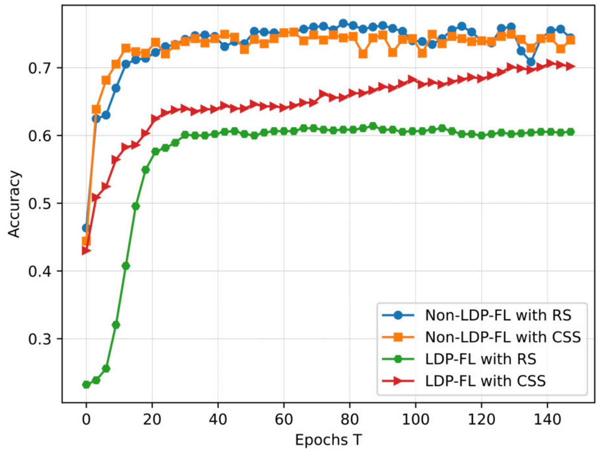


Fig. 3: Evaluation of different client selection methods based on accuracy.

D. Analyze the robustness of LDP-FL against model inversion attacks

We conducted a model inversion attack using the specified algorithms with the following settings: $T = 200$, $\eta = 0.1$, $\beta = 100$, and $\gamma = 0.99$. The attack was applied to two different settings in the system: LDP-FL with $C = [1, 2, 4]$ and $\delta = 10^{-5}$, and non-LDP in FL with $K = 7$. To ensure accurate results, we performed the attack on various client target models and target labels and reported the average outcomes.

The objective of model inversion attacks is to reconstruct the speech features of each client by exploiting the local SER model and its associated labels. To evaluate the effectiveness of these attacks on the FL system, we employed the Mean Squared Error (MSE) metric. The MSE was calculated by comparing the reconstructed speech features with the actual speech features of each specific client.

Table I illustrates the results obtained from the attack. When the noise scale σ was set to 1.0, the MSE values were similar for both LDP and non-LDP settings. However, as we increased the noise scale σ and the clipping threshold C, the MSE values significantly increased, indicating a decline in the attack effectiveness. These findings highlight the effectiveness of incorporating LDP as a robust privacy measure against model inversion attacks. Implementing LDP in the system significantly mitigates the risk posed by threat models and ensures compliance with the specified privacy requirements, particularly the 1.b privacy requirement. By introducing noise into the client models, the accuracy of predictions made by adversaries using these models is reduced, thereby impeding the reconstruction process of speech features associated with specific client labels.

E. Balancing privacy and accuracy

Achieving an optimal balance between privacy and accuracy is paramount when utilizing LDP for SER applications that require precise and accurate results. According to this reference [23], epsilon (ϵ) acts as a parameter that measures the level of privacy guarantee provided by the $(\epsilon, \delta) - LDP$

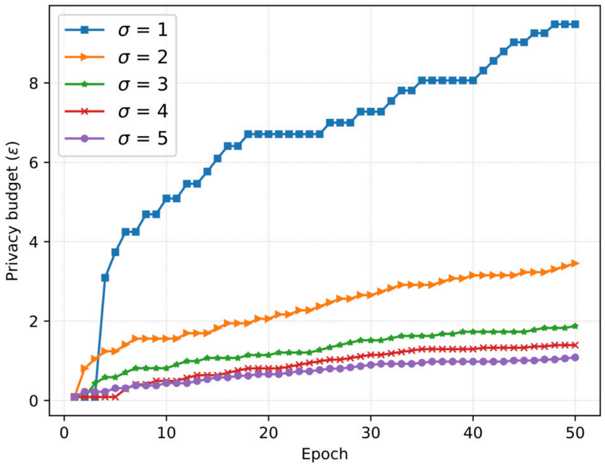


Fig. 4: With constant noise scale σ, changes in privacy budget (ϵ) with an increase in epochs

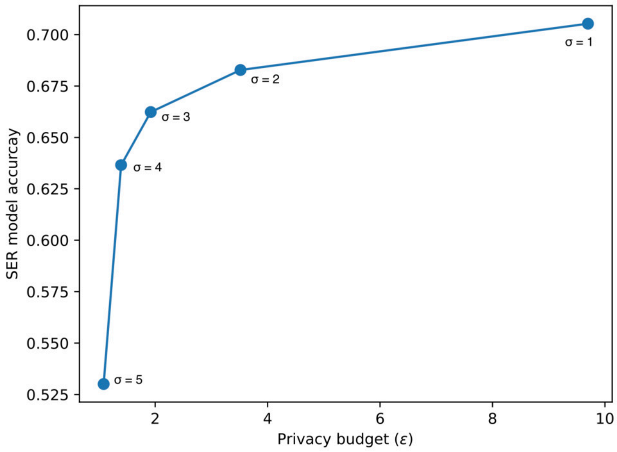


Fig. 5: An assessment of the accuracy of SER models based on privacy budgets and noise levels.

mechanism. It reflects the degree of privacy protection, with smaller epsilon values indicating stronger privacy guarantees. In our developed method, the value of ϵ is influenced by the number of epochs (T). Consequently, as the number of epochs increases, the value of ϵ changes, even when the noise scale remains constant. This association is illustrated in Figure 4.

In our evaluation of the LDP-FL with CSS mechanism for SER, we experimented with different noise scales σ, $k = 50$, failure probability parameter of $\delta = 10^{-5}$, clipping threshold $C = 2$ and a total of 50 epochs (T). The results, as illustrated in Figure 5 and Figure 4, revealed the following privacy levels and corresponding accuracy:

- For a noise scale of $\sigma = 5$, we achieved a privacy level of $(1.08, 10^{-5}) - LDP$, with an accuracy of approximately 54%.
- With a noise scale of $\sigma = 4$, we attained a privacy level of $(1.39, 10^{-5}) - LDP$, accompanied by an accuracy of around 64%.
- Employing a noise scale of $\sigma = 3$, we achieved a privacy level of $(1.92, 10^{-5}) - LDP$, while maintaining an accuracy of approximately 67%.
- A privacy level of $(3.51, 10^{-5}) - LDP$ was obtained by utilizing a noise scale of $\sigma = 2$, resulting in an accuracy of about 69%.
- Finally, with a noise scale of $\sigma = 1$, we achieved a privacy level of $(9.69, 10^{-5}) - LDP$, accompanied by an accuracy of roughly 70%.

Striking the right balance between privacy and accuracy is contingent upon specific system requirements. In the case of the SER application in the FL setup, where the specified acceptable accuracy range is 65-70% and privacy requirements are outlined in Section III-A, it is feasible to attain an acceptable level of privacy by utilizing a privacy parameter of $(1.92, 10^{-5})$-LDP, along with a noise scale of $\sigma = 3$, while maintaining the desired accuracy.

V. Conclusions and Future work

In this paper, we introduced LDP-FL with CSS, a novel approach for privacy-sensitive SER applications. Our objective is to ensure the privacy of clients' speech data while maintaining system accuracy. By combining LDP-FL and CSS, we mitigate the impact of noise scale on accuracy and improve it by selectively choosing clients based on their data size during each training round of FL. We evaluated our approach using the CREMA-D dataset. The evaluation results demonstrate that LDP-FL with CSS achieved an accuracy range of 65-70%, slightly lower than the initial SER model accuracy while maintaining a privacy level of $(1.92, 10^{-5})$-LDP. Our analysis highlights the importance of achieving a balance between privacy and accuracy, which aligns with the specific requirements of SER applications.

In the future, we plan to discuss personalized privacy with an adaptive noise scale of LDP mechanisms that are tailored to each client's privacy preference.

VI. Acknowledgements and Disclaimer

This work was partially supported by EU ECSEL project DAIS that has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No.101007273. The work reflects only the authors' views; the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] R. A. Khalil, E. Jones, M. I. Babar, T. Jan, M. H. Zafar, and T. Alhussain, "Speech emotion recognition using deep learning techniques: A review," *IEEE Access*, vol. 7, pp. 117 327–117 345, 2019.

[2] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication*, vol. 116, pp. 56–76, 2020.

[3] P. Chhikara, P. Singh, R. Tekchandani, N. Kumar, and M. Guizani, "Federated learning meets human emotions: A decentralized framework for human–computer interaction for iot applications," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6949–6962, 2020.

[4] J. L. Kröger, O. H.-M. Lutz, and P. Raschke, "Privacy implications of voice and speech analysis–information disclosure by inference," *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, Windisch, Switzerland, August 19–23, 2019, Revised Selected Papers 14*, pp. 242–258, 2020.

[5] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, no. 3152676, pp. 10–5555, 2017.

[6] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[7] S. Latif, S. Khalifa, R. Rana, and R. Jurdak, "Federated learning for speech emotion recognition applications," in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2020, pp. 341–342.

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[9] M. S. Jere, T. Farnan, and F. Koushanfar, "A taxonomy of attacks on federated learning," *IEEE Security & Privacy*, vol. 19, no. 2, pp. 20–28, 2020.

[10] Z. Xiong, Z. Cai, D. Takabi, and W. Li, "Privacy threat and defense for federated learning with non-iid data in aiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1310–1321, 2021.

[11] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy-based federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8836–8853, 2020.

[12] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[13] M. Kim, O. Günlü, and R. F. Schaefer, "Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2650–2654.

[14] M. A. Pathak, *Privacy-preserving machine learning for speech processing*. Springer Science & Business Media, 2012.

[15] T. Feng, R. Peri, and S. Narayanan, "User-level differential privacy against attribute inference attack of speech emotion recognition in federated learning," *arXiv preprint arXiv:2204.02500*, 2022.

[16] A. A. Alnuaim, M. Zakariah, A. Alhadlaq, C. Shashidhar, W. A. Hatamleh, H. Tarazi, P. K. Shukla, and R. Ratna, "Human-computer interaction with detection of speaker emotions using convolution neural networks," *Computational Intelligence and Neuroscience*, vol. 2022, 2022.

[17] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[18] V. Tsouvalas, T. Ozcelebi, and N. Meratnia, "Privacy-preserving speech emotion recognition through semi-supervised federated learning," in *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 2022, pp. 359–364.

[19] Y. Chang, S. Laridi, Z. Ren, G. Palmer, B. W. Schuller, and M. Fisichella, "Robust federated learning against adversarial attacks for speech emotion recognition," *arXiv preprint arXiv:2203.04696*, 2022.

[20] T. Tuncer, S. Dogan, and U. R. Acharya, "Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques," *Knowledge-Based Systems*, vol. 211, p. 106547, 2021.

[21] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, pp. 1–19, 2022.

[22] R. Bassily, "Linear queries estimation with local differential privacy," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 721–729.

[23] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[24] A. Balador, S. Sinaei, M. Pettersson, and I. Kaya, "Dais project - distributed artificial intelligence systems: Objectives and challenges," in *26th Ada-Europe International Conference on Reliable Software Technologies (AEiC'22)*, 2022.

[25] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.

[26] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1459–1462.

DOI: 10.15439/2023F6360
ISSN 2300-5963 ACSIS, Vol. 35

Back to the Essential: A Literature-Based Review on Agile Mindset

Necmettin Ozkan
Architecht Information Systems
İstanbul, Turkey
necmettin.ozkan@architecht.com

Karen Eilers
Institute for Transformation
Hamburg, Germany
karen.eilers@in-transformation.com

Mehmet Şahin Gök
Gebze Technical University
Kocaeli, Turkey
sahingok@gtu.edu.tr

Abstract— **Starting from software development, Agile approaches are spreading across a broad range of industries and functions, with many great challenges. Mindset, as one of the crucial human factors of individuals in Agile, influences people's decision-making and affects every aspect of behavior and action. However, many organizations and teams face big challenges in achieving an Agile Mindset of their individuals. In addition to an often-unclear understanding of the Agile Mindset, various aspects of it such as success factors and indications are largely unknown, which makes it extremely difficult to establish an Agile Mindset. Motivated by this, our study aims to conduct a literature review by answering comprehensive research questions related to the Agile Mindset regarding publication demographics, importance, definitions, characteristics, elements, critical success factors, indicators, activities for development of Agile Mindset, and future directions of research.**

Keywords—Agility, Agile Mindset, systematic literature review, SLR, project management, Scrum.

I. Introduction

While Agile approaches have their native grounding in software development, today they are spreading across a broad range of industries and functions as well as research fields [1]. At the same time, organizations are facing great challenges in transitioning to Agile [1-5] and some of the transitioning attempts fail [1, 6], simply because of the actor's mindset involved in the process [1]. It is a reminder that regardless of tools, methods, or frameworks, the adoption of Agile comes down to the people who make up the organizations [2].

People have a special position in Agile transformations, compared to technology and process. Developers develop applications and systems mostly for people, and always with people. They design and shape technology and processes and are always one of the significant factors directly affecting their organizations and the success of projects [1, 5, 7]. They may cause serious problems. Weinberg [8] briefly states their key role: "No matter how it looks at first, it's always a people problem".

Even though social aspects are at least as important as technical skills [8] and the significance of people is obvious, human factors continue to be ignored by many organizations [1, 10, 11] and it presents a serious problem to them [12]. This issue is also reflected in the literature on Agile, which focuses on engineering perspectives, practices, and processes [13, 14, 15]. On the technical side, Agile teams are more often doing Agile rather than leading to being agile [14]. Issues related to people are usually missing from Agile adoption journeys [2].

Mindset, as one of the crucial human factors, influences people's behavior [16], decision-making [17], forms their way of thinking, beliefs, and attitudes [18], and thus, shapes the way organizations act [17]. It reflects a set of beliefs, assumptions, perceptions, norms, attitudes, and notions held by people [17, 19]. Individuals transfer their (agile) mindset to organizations, processes, and tools they create [20], which makes it a key factor to consider. Considering its significance, various research disciplines, ranging from psychology to information systems, are focused on the construct of mindset and its underlining key roles and its implications [6].

The mindset aspect becomes very important for Agile approaches as well, which put people at the center. Agile Mindset settles at the core of agility [22, 23] and without a change in mindset, targets cannot be met [24]. Agility can be viewed at different levels including individuals, teams, processes, tools, strategy, or culture. However, Agile Mindset has a special position compared to any specific methodology, process, system, platform, or organizational structure [25, 26]. Durbin and Niederman [27] state, "Following agile approaches requires the spirit of agile as well as the mechanics of following its 'rules'". When considering these two parts, the way to agility should start with establishing a proper and right Agile Mindset, the spirit of agility, instead of directly applying any Agile method [14, 28], because methods and practices can only lead to a shift in a degree of agility and they alone do not guarantee being agile [9, 14, 29]. Moreover, "without the right mindset, the methods are often adapted in an incorrect way and lose their purpose" [9]. To face this successfully, agile individuals require an Agile Mindset, beyond the given set of procedures, techniques, and rituals [30].

On account of this, organizations should go far beyond "doing Agile" and seek ways to "be agile" [14]. For Agile initiatives in information systems and beyond, it is required to live the core values, and principles of agility [9, 14], which may come with an Agile Mindset of individuals [9]. However, many organizations and teams fail to build an environment, which enables Agile Mindset development of the individuals [16, 31]. Even though organizations want to enable Agile Mindset development, Agile team members face several challenges in doing so, such as a continuous paradox of 'doing Agile' (trying to 'perfect' the adoption of their chosen packaged/ customized practices coming with Agile methods) versus 'being Agile' (continuously endeavoring to enhance their work, to handle uncertainty and for improvement) [3].

Hence, it is crucial to investigate the construct of Agile Mindset that has such importance in information systems and beyond and is furthermore concerned with high application challenges. However, until today, the research regarding this topic is still in its infancy. Motivated by this need, our study aims to conduct a literature review by addressing several issues related to Agile Mindset and to find answers to comprehensive research questions. In terms of the disciplines of the included studies, it was preferred to go beyond the software development that makes use of this construct intensely and gather information from every field in order to have a generalized representation of the construct and to nurture it with more inputs from a wide area.

The remainder of this paper is organized as follows. In Section 2, we describe the overview of the research design with research questions and the paper selection process. Section 3 delivers the results of the literature review along with discussions of our findings. In Section 4, we deliver conclusions and limitations of the study.

II. RESEARCH DESIGN

This research process has been undertaken as a Systematic Literature Review (SLR) based on the guidelines proposed by Kitchenham et al. [32]. The following section describes the implementation of this SLR.

The research process starts with defining research goals and questions. After defining search queries and searching in the Scopus and Web of Science (WoS) digital libraries, we gathered 1850 potentially relevant publications. For scanning the retrieved studies, we developed and applied inclusion/exclusion criteria and obtained a final pool of 19 sources. In addition, the references in the identified 19 studies were examined (backward snowballing) and one other related study was added. Finally, 21 studies in Table IV were identified. After extracting the data from the sources, the results of the SLR were analyzed and the findings were discussed. The remainder of the section concerns the research questions, publication selection process, and data extraction and synthesis.

A. Research Questions

This study aims to review studies that focus on Agile Mindset. Thus, we set the main goals related to our research 1) identify the studies which focus totally or partly on Agile Mindset and 2) analyze and synthesize the studies' relevant results. We raise and investigate the research questions (RQs) under two main groups: (1) Publication demographic-related RQs to identify developments regarding interest and relevance in research. And (2) contribution-related RQs. In the latter we summarize relevant insights regarding the Agile Mindset. We thereby started by investigating the relevance of the Agile Mindset for different outcomes (e.g. productivity or motivation) (RQ2.1). To build a common ground, we further searched for definitions and conceptualizations of the Agile Mindset (RQ2.2). To go even deeper, we reviewed for insights, what characteristics describe the nature of the Agile Mindset (e.g. stability, latency) (RQ.2.3) and provided elements, that build the Agile Mindset as a construct (e.g. openness, collaborative exchange) (RQ2.4). The following two research questions address insights, which are necessary for organizations to build effective surroundings for Agile Mindset development. We thereby searched for critical success factors (RQ2.5) and concrete activities to achieve Agile Mindset (RQ2.6). The Agile Mindset should be reflected in special behavior, which indicate its presence. Those behavioral indicators are summarized in RQ2.7. Finally, we conclude with future directions for research (RQ2.8). Finally, the RQs were identified as:

(1) Publication demographic-related RQs (RQ1) including country of authors, publication year, publication venue, authors' affiliation type, domain of study, and paper citation data.

(2) Contribution-related RQs (RQ2):

RQ2.1 What is the importance of Agile Mindset?

RQ2.2 What are the definitions of Agile Mindset?

RQ2.3 What are the characteristics of Agile Mindset?

RQ2.4 What are the elements of Agile Mindset?

RQ2.5 What are the critical success factors for Agile Mindset?

RQ2.6 What are the activities for developing Agile Mindset?

RQ2.7 What are the indicators of Agile Mindset?

RQ2.8 What are the future directions for Agile Mindset research?

B. Publication Selection Process

The search process was a manual search of peer-reviewed studies in the well-known digital libraries, Scopus and Web of Science (WoS), without any filter in the year range to gather a full overview. Based on the scope of this study, the search string was developed by following the SLR protocol in Table I [32]. We did not add a "population" related keyword in the string referring to the application area, which could be any discipline for our study, to access the largest possible set of the data. Regarding the search place, and taking our inclusion criteria IC2 (Table II) into account, we searched in meta-data and titles instead of the full text. Finally, the search strings were finally formed as in Table I.

TABLE I. SEARCH STRINGS AND LIBRARIES

Library	**Place**	**Search strings**	**Number of Initial Results**
Scopus	TITLE-ABS-KEY	TITLE-ABS-KEY ("be* of agil*" OR "be* agil*" OR "agile mindset" OR "agile mind set" OR "agile mind-set" OR "agile mind" OR "agile mental" OR "agile mentality" OR "mental agility" OR "agility mindset" OR "agility mind set" OR "agility mind-set" OR "agility mind" OR "agility mental" OR "agility mentality") OR TITLE (("be " OR being OR becom* OR bec ame) OR (mind* OR mental*) AND agil*) AND (LIMIT-TO (DOCTYPE , "cp") OR LIM IT-TO (DOCTYPE , "ar") OR LIMI T-TO (DOCTYPE , "ch")) AND (LIMIT-TO (LANGUAGE , "English"))	1641
WoS	All Fields	("be " OR being OR becom* OR bec ame) AND agil* (Title) OR	1067

		(mind* OR mental*) AND agil* (Title) OR ("be* of agil*" OR "be* agil*" OR "agile mindset" OR "agile mind set" OR "agile mind-set" OR "agile mind" OR "agile mental" OR "agile mentality" OR "mental agility" OR "agility mindset" OR "agility mind set" OR "agility mind-set" OR "agility mind" OR "agility mental" OR "agility mentality") (All Fields) and English (Languages) and Article or Proceeding Paper or Book Chapters (Document Types)	
Total in Dictint			1850
Selected in Dictinct Only From Scopus			4
Selected in Dictinct Only From WoS			1
Selected in Dictinct From Scopus and WoS			15
Snowballing			1
Total Selected			21

Based on the scope and context of our study, for the selection of papers, the following propositions of inclusion criteria (IC) and exclusion criteria (EC) in Table II were specified and applied to the papers. Every Agile practice, value, and principle is supposed to be theoretically and practically related to Agile Mindset. Considering this, the content was excluded if it was not explicitly related to Agile Mindset in the paper. For example, although it is known that one of the main elements in the journey to being agile is the mindset, if Agile Mindset was not mentioned explicitly by the study, then the contents of the study were not included.

During the application of inclusion/exclusion criteria, the papers were examined through their titles and, where necessary, abstracts in order to identify whether they were within our scope. If the abstracts were not sufficient to decide to include or exclude the papers, then, a scanning through the full texts of the papers was done to identify relevant ones. The studies for which inclusion and exclusion decisions could not be clearly made by the first author, a separate joint evaluation step with the second and third authors was conducted to reach consensus about the decisions.

The search process was conducted in June 2023. The initial list included duplicate records. A total number of 1850 of distinct peer-reviewed studies were returned after removing the duplicate ones. Three identified studies were the extended versions of their previous ones. In this case, only the extended versions were included. All papers were accessible by the authors when their full text was needed. We applied the exclusion criterion for papers not available in English by filtering, via the libraries' relevant feature allowing the elimination of non-English studies, and we also applied it via a manual investigation. Consequently, 21 papers were identified as shown in Table IV.

C. Data Extraction and Synthesis

The first researcher ran the data extraction and synthesis process that served to answer the research questions by applying detailed and thorough examinations of the relevant studies. The second author directed and consulted on the paper and the third author coordinated the research process.

A data collection form was designed, to record the relevant information from the identified studies. The collected information ranged from general information about each study such as author, title, year, venue, author affiliation, and author country, as well as specific information to answer the research questions. The relevant content items were captured and taken “as-is” from the studies and copied to the Excel file manually. In addition to this, the parts of the studies including the relevant data were highlighted in the original papers for any future reference. Thus, it was the authors' aim to have as little bias as possible, to comment on the original data and develop their research answers gradually. Once the data extraction was complete, they were synchronized and grouped into the relevant RQs manually. Thus, an interpretation for data analysis was limited since the actual extraction of data was conducted. Even so, for cases that were open to subjective evaluation in the interpretations of the extracted data, the first and second authors jointly evaluated the data until a consensus was reached for a common understanding.

TABLE II. INCLUSION CRITERIA AND EXCLUSION CRITERIA

ID	Criterion
IC1	Papers fully or partially focus on Agile Mindset
IC2	Papers in any field including software development, business management, human resource management, etc.
IC3	Conference, workshop, journal, or book-chapter papers
IC4	Papers in english language
EC1	Papers not involving the Agile Mindset as a research construct
EC2	Papers published in non-peer-reviewed sources such as thesis, web pages, workshop proposals, tutorials, panels, proceeding information, and books.
EC3	Papers not accessible by the authors
EC4	Duplicate studies
EC5	Extended papers
EC6	Articles not in English

D. Quality assessment

The entire process relied on a search procedure that called for explicit criteria to validate the quality of the selected candidate papers by ensuring each paper was of adequate standard [32]. Accordingly, a custom quality-assessment-criteria-list and item descriptions were established as shown in Table III.

Each paper was then assessed against this given set of questions by the first author. A manual inspection was done through the full-text investigation carried out to identify each selected paper’s quality assessment score.

We set a score weight based on two values: Satisfactory (1) and Not Satisfactory (0). Accordingly, the evaluations of the papers have been made based on the two predefined values to set their scores yielding a score maximum of three. It was decided that the studies with a score below one point would be eliminated. After applying the determined quality criteria, the quality scores of each study were satisfied; No studies existed lower than the threshold score and no elimination regarding the quality assessment was done. This was most likely due to the venue of publications being well-qualified and generally well-known.

TABLE III. CRITERIA FOR QUALITY ASSESSMENT

Criteria- Statement	Descriptions

QA1- Are the contributions of methods clear?	The clarity and robustness of the method applied in the study are satisfactory
QA2- Are outcomes as results clear?	Outcomes are clearly delivered and relevant to the method applied
QA3- Is the discussion on results clear?	Discussion of the results is satisfactory and based on the results objectively. Validity threats are delivered.

III. RESULTS AND DISCUSSION

We present the results and findings of this SLR study concerning the identified RQs. Table IV lists the identified studies along with their demographic information (Regarding RQ1).

According to the results, 52% (11/21) of the studies are conference papers (C) and 48% (10/21) are journal articles (J). In terms of venues for the selected 21 papers, there are 21 different venues. The conferences are in general well-known in their respective fields. The names of the journals are Sustainability, Human Resource Development International, International Business Review, International Journal of Information Systems and Project Management, International Journal of Managing Projects in Business, Journal of Advances in Management Research, Journal of Software: Evolution and Process, Research-Technology Management, Industrial and Organizational Psychology, and Technological Forecasting and Social Change. Considering the quality of journals and conferences, it can be deduced that qualified venues and researchers are interested in this subject.

The fact that there are more conference publications than journal articles may indicate that the field is developing. Publishing conference papers is in general easier than publishing journal papers, especially in developing topics. One more reason to have such a ratio for conference papers is related to the preference to communicate the research at a conference to get feedback on first insights, rather than in journals on such a developing topic. The journal publication process can be longer than the conference publication process in general. In addition, to find a place for Agile-related papers, there may have been a tendency for conferences that can be considered more flexible in scope compared to journals. Moreover, the fact that there are some conferences dedicated to Agile topics but there is a lack of (active) dedicated journals, may be another interpretation of this tendency.

TABLE IV: IDENTIFIED STUDIES

ID	Reference Number	Country	Year	Conference (C)/Journal (J)	Industry/ Academia/ Collaborative	Discipline	Citation Count	Citation Count Per Year
P1	[5]	Slovenia	2022	J	Academia	Human Resources	2	2
P2	[26]	Germany	2022	C	Collaborative	Project Management	0	0
P3	[33]	India	2022	J	Academia	Human Resources	0	0
P4	[16]	India	2022	J	Academia	Information Technology	2	2
P5	[27]	USA	2021	J	Academia	Information Technology	5	2.5
P6	[34]	Turkey, UK	2022	C	Collaborative	Information Technology	0	0
P7	[35]	Sweden	2021	J	Academia	General	12	6
P8	[9]	Germany	2022	J	Academia	Information Technology	2	2
P9	[20]	Turkey	2020	C	Collaborative	General	3	1
P10	[3]	Germany, New Zealand	2020	C	Academia	General	9	3
P11	[23]	Italy	2022	C	Academia	Teaching & Education	0	0
P12	[36]	Poland	2020	C	Collaborative	General	9	3
P13	[37]	Sweden	2016	C	Academia	Information Technology	4	0.7
P14	[22]	Netherlands	2014	C	Academia	Information Technology	43	5.4
P15	[38]	Germany	2021	C	Academia	General	5	2.5
P16	[17]	Israel	2022	J	Academia	Business	0	0
P17	[39]	New Zealand	2013	C	Academia	General	48	5.3
P18	[1]	Switzerland	2022	J	Academia	Information Technology	18	18
P19	[6]	Germany	2020	C	Academia	Information Technology	21	7
P20	[46]	Denmark	2019	J	Industry	Enterprise Agility	78	19.5
P21	[21]	Germany	2020	J	Academia	Human Resources	26	8.7

In terms of the authors' affiliation types, 76% (16/21) of the papers are from academia, 19% (4/21) are from industry and academia collaboration, and 5% (1/21) are from industry. There is only one paper that has one sole author from the industry. It is obvious that the Agile Mindset issues attract the attention of the academic community more, a community that is more independent on Agile matters. This fact contradicts the common belief that Agile is mainly a practice-driven domain [40] and coincides with the view that Agile is commonly regarded as an object to sell [60]. This special case regarding the Agile Mindset poses an exception may be due to the fact that selling it cannot be feasible. A more non-profit-oriented and neutral part of the community may want to normalize and liberalize the Agile movement with the Agile Mindset publications.

The disciplines of the studies include information technology with seven papers at the top, General (with no specific domain stated) with six papers, Human Resource Management and Leadership with three papers, Project Management, Management, Enterprise Agility and Teaching and Education with one paper each. This distribution shows, again, Agile is popular in the Information Technology discipline, meanwhile, it has started to spread to other management areas as well.

Among the authors, Schoop M., Ozkan N., and Mordi A. have two papers in the list while other authors contribute with one paper. Only two papers have an international collaboration among their authors, including Turkey and the UK, and Germany and New Zealand as collaborating countries. The distribution of authors' countries is Germany (6), New Zealand (2), Sweden (2), Turkey (2), India (2), Netherlands (1), Israel (1), Italy (1), USA (1), UK (1), Switzerland (1), Slovenia (1), Poland (1), and Denmark (1). The reasons that separate Germany from other countries in this regard require further research. Like other Agile-related issues, European countries and Germany's positions stand out. Our study's outputs coincide with the bibliometrics study on Agile Software Development [15]. According to the study [15], the top two countries in terms of the number of Agile publications are the United States (257) and Germany (166). Thus, this remarkable position of Germany needs further investigation.

Figure 1 shows the number of papers per year. The acceleration in the paper numbers in recent years can be seen in the figure. The first published paper in the scope of our study [P17] is more about the effective and sustained usage of Agile methods rather than Agile Mindset, although it provides inputs for our study. The first paper focusing mainly on Agile Mindset, [P14], was published in 2014, 13 years later the Agile Manifesto was announced and 20 years later Scrum emerged.

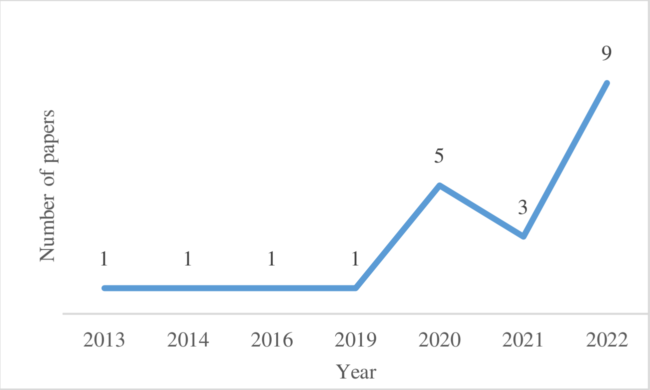


Fig. 1: Number of papers per year

With 287 total citations as shown in Table IV, the average citation count per publication is 13.6, the median of citation counts is 5, the h-index is 9, the i10-index is 7, and the i100-index is zero. There are studies that have received a high number of citations, as well as low numbers or none. The number of papers with at least one citation is 16 (76%). For the remaining 5 papers with zero citations yet, the average lifespan is almost one year (assuming all publications are published in the middle of the publication year). There is no paper published in 2021 or previous years with no citation. The studies that have never been cited seem to have a short lifespan.

Among the most influential papers include both journal and conference papers. With the highest citation count per year, [P20] and [P18] stand out. [P20] delivers the Agile transformation at LEGO Group. Eilers et al. [P18] focuses on Agile Mindset, discusses the construct and definitions of Agile Mindset and develop an instrument to measure it. Then, it investigates how Agile Mindset influences an organization's strategic agility and thus benefits its performance. This paper seems, to the authors, as one of the most comprehensive studies in the field focusing on Agile Mindset.

The remainder of this section was dedicated to RQ2 to address several dimensions including importance, definitions, characteristics, elements, critical success factors and indicators of Agile Mindset, ways to develop it, and future directions of research.

What is the importance of Agile Mindset? (RQ2.1)

The biggest issue when transitioning to an Agile organization is acquiring Agile Mindset by team members [27, 43, 44]. Many studies mention the key role of Agile Mindset and the necessity of internalizing it in order to succeed in various cases including the transition to agility [6, 14, 22, 41], effective and sustained usage of Agile approaches [34, 39], achievement of enterprise agility [6], effective teamwork [30], scaling Agile [6, 38], Agile team's productivity, responding to crises [16], onboarding for new-comers [42], helping team members in reducing negative behaviors [26], and having a proper culture, competitive advantage and project success [5, 26]. A team that does not adopt the Agile Mindset is likely to have less task responsibility, be disengaged, demotivated, and avoid challenges [16].

These studies show that Agile Mindset is important in many aspects. What is interesting is that, as far as we know, the number of papers focusing on deep understanding of Agile Mindset is highly limited.

What are the definitions of Agile Mindset? (RQ2.2)

We have found a very limited number of studies providing a definition of Agile Mindset. One study [6] states "Agile Mindset is a mindset based on the values and principles of the agile manifesto. It is underpinned by specific personal attributes on the individual level and an enabling environment on the organizational level... with the goal of achieving a state of being agile instead of merely doing agile." Another study states [30] "Agile team requires…a particular attitude, way of thinking and behavior of both the individuals and the entire team, a so-called 'Agile Mindset". Sathe and Panse [16] define Agile Mindset as "a way of thinking, that emphasizes collaboration among team members, and being adaptable to changing environments, to be a high-performing team." Study [21] produced an Agile Mindset definition derived from other studies: "the understanding of the workforce that agile behaviors are necessary for the organization to survive in a changing marketplace…a positive attitude toward learning and self-development, as well as a positive attitude toward change". Finally, [1, p.8] developed a definition of Agile Mindset: "An individual with a strongly developed AM [Agile Mindset] evaluates learning, exchanges with others, their own work organization, and value creation in terms of the customer in a highly positive way".

Regarding the main features of Agile Mindset, the most commonly used source is the Agile Manifesto [37], exemplified in the study of [6]. However, organizations should go beyond the ideas summarized in the manifesto [9], since the manifesto does not include a reference to mindset even though its values and principles contain a certain overlap with the Agile Mindset concepts [6, 34, 45]. Moreover, [34] proposes going beyond the manifesto and not solely relying on it for the Agile Mindset.

The definition by [30] resembles the general dictionary definition of a mindset: a mental and established set of attitudes, a cognitive understanding and interpretation of the environment, and a person's way of thinking and opinions [17, 47, 48]. To prove this, we just replaced the "Agile Mindset" definition with effectiveness or quality mindset. The definition still works for both (and possibly for many more). Another issue with previous Agile Mindset definitions is that people use different terms other than mindset to describe similar or identical construct, such as Agile culture [6].

Eilers et al. [1] provide in their work a sharp definition of an individual's Agile Mindset, which integrates previous definition approaches and synthesized them with new data. Consequently, in our study, we reached a similar Agile Mindset definition as compiled by [1]. While the definition of the term agility itself has no consistent, complete, precise, and agreed definition yet [49], it seems that it will take time to clarify the definitions of Agile Mindset from different perspectives and different levels. The current situation regarding the definitions of Agile Mindset implies a need for more studies on this topic and reinforces the findings of others; it remains unclear what Agile Mindset is on different levels and perspectives [1, 6, 9].

What are the Characteristics of Agile Mindset? (RQ2.3):

Agile Mindset is an abstract, vague, and latent (invisible) construct, thus, difficult to measure, even to observe, and demonstrate [P5, P6, P11, P15, P16], which makes the transformation and training of it the most difficult part [P6, P11]. Thus, it is hard to prove and show when the transformation and training of it is successful. This may cause both "sellers" and "buyers" of Agile to want to take less risk by acquiring "well-known" and "proven" products, such as Agile frameworks and tools, even though such aspects are on the "less valuable side" of the Agile Manifesto. Thus, Agile has penetrated the various sectors with relatively easy adoption of the "proven" products [50]. This abstract characteristic of Agile Mindset also brings a duality; its weakness comes with its strength [P15]. While it is vague and has a lack of prescriptive properties [P6, P15], it also presents a freedom and a "safe place" kept away from "Agile trading" for organizations.

Agile Mindset is a soft, dynamic, and intangible asset, resource, and capability and a kind of trigger that can influence various tangible assets of organizations [P11, P16, P18]. It is a prerequisite and starting point for a successful Agile transformation [P16]. It is inherently a psychological, socio-cultural, and human-related matter [P14, P15, P19]. Like other human-related assets, it presents complex interactions of social, cultural, and psychological perspectives of individuals with other people. This makes it challenging to understand, substitute, and emulate [P16]. The nature of it also creates challenges for organizations in terms of finding and developing their Agile Mindset as an individual endeavor [P8, P16].

The previous studies include a wide range of different behaviors, attitudes, and traits as elements of Agile Mindset. It seems that some of them appear to be specific forms of behavior (e.g. self-organization, self-managing, continually adapting, changing behavior, face-to-face conversations, mutual listening, not covering ups failures, continuous delivery, and so on). Such items that are more about the actual behavior can be results of Agile Mindset but are not elements of it. Agile Mindset in the mind of people is an intangible, invisible asset that influences various visible aspects [P11]. It is and should be a variation of "a way of thinking about things" [48] or "a person's way of thinking and their opinions" [47], not a way of doing in the apparent physical way.

What are the Elements of Agile Mindset? (RQ2.4):

We have found five main sources of studies identifying elements of Agile Mindset. One of these, [6] condensed 192 Agile Mindset elements into 27. Manen and van Vliet [22] identify factors that affect the expansion of Agile development in large organizations positively or negatively by using interviews. Those factors were then grouped into two categories: "Agile Mindset" and "Contextual Dependencies". Miler and Gaida [36] in their extended paper, focus on the elements of Agile Mindset and their importance to the effectiveness of Agile teams by reviewing the current literature and conducting interviews with experts, which results in 70 elements initially and 26 elements after applying their threshold criteria. Additionally, [16] mention two items covered by the current studies above; collaboration and focus on delivery. Finally, [1] identified four dimensions of Agile Mindset called a positive attitude towards learning spirit, collaborative exchange, customer co-creation, and empowered self-guidance and measured it on a scale of 20 items. Study [26] provides a list of Agile Mindset competencies for project leaders by conducting a semi-systematic literature review using a content analysis of current publications on the leadership role of project managers in Agile projects. Then, the identified competencies were rated by 40 respondents using a questionnaire.

We have found two more studies using the data from these primary studies. Ozkan and Gok [20] combine Agile Mindset elements from three primary studies [6, 22, 36], converged, categorized and examined their relationships. Secondly, [23] propose a training method to help trainers by focusing on the critical Agile Mindset elements based on the work of [6] that are grouped into ten topics.

After aligning Agile Mindset definitions, we need to identify and further extract Agile Mindset elements. "We need to know more about how these elements develop Agile Mindset and how they are connected with Agile Mindset" [1] and each other.

What are the Critical Success Factors for Agile Mindset? (RQ2.5):

The critical success factors for developing Agile Mindset are given in Table V and can be found in different levels in the environments of the individuals. According to the table, proper leadership and management mindset approaches are required to develop employees' Agile Mindset [P3, P16]. Agile talents should be supported to be proactive and resilient to cope with changing environments by having explorative activities such as creative ideas, risk-taking, and independent thinking [P3]. However, Agile Mindset should deal with both explorative and exploitative activities and attitudes of people [P3]. Agile teams are not completely free in their fields; they need monitoring and redirection from the management and strategy layers. This reminds us that agility is a matter of "how" that should serve a whole (what) and purposes, rather than being positioned or used in a way as to make room for unconscious acting. This inference is close to the expression that agility is a balancing act [51] in itself and with its environment.

Communication matters and continuous feedback systems are also crucial. Personal attitudes are the main driver for Agile Mindset, then, [P8] regards personal prerequisites of team members as a success factor and are a prerequisite for creating Agile Mindset. At the team level, Agile teams should have a common ground, understanding, norms, consensus, and team spirit as opposed to an "individualistic" mindset [P17]. Agile Mindset has a spirit supporting a continuous change in behavior, learning, and growth [P17].

TABLE V: CRITICAL SUCCESS FACTORS OF AGILE MINDSET

Critical Success Factors	Paper ID
A suitable leadership approach	P3
Building in-house agile talents by leaders	P3
Leadership behavior promoting employees' explorative activities	P3
Leaders facilitating employees to openly express their creative ideas, developing new competencies, and aiding in routine tasks	P3
A leadership approach where leaders provide clarity on employees' roles and responsibilities, communicate information timely and regularly, continuously provide feedback, set defined team goals, monitor their goal attainment, and promote entrepreneurial activities and innovative work behavior.	P3
Establishing a continued and consistent focus on value creation	P5
Personal prerequisites and attitudes of team members	P8
Having an open mind towards others and an Agile way of working	P8
Willingness to change	P8
Having a right management mindset	P16
Flexibility	P16
Becoming failure tolerant	P16
Having norms and consensus across different definitions	P17
Team spirit where team members display a strong sense of identification and commitment with the team as opposed to an "individualistic" mindset.	P17
A continuous change in behavior based on possibility thinking, learning, and growth	P17

What are the Activities for Developing Agile Mindset (RQ2.6):

Table VI shows the suggested actions from the identified studies about how to develop Agile Mindset. These items in total, partially, or in any form do not claim to be comprehensive, complete, or provide a method or framework to develop Agile Mindset, rather, list items from various sources.

According to the table, all stakeholders who want or will experience change, should know the reasons and value of change and should be involved in the transformation processes with a base of trust [P5, P7, P8, P16]. Although the change direction is recommended from the top-down by [P5], the change should be bi-directional and the two directions (from the top-down, from bottom-to-top) should be aligned. Managers should invest in training, building, and measuring their people/team/organization's Agile Mindset [P16]. Managers and Agile coaches should be role models for showing the Agile values [P8].

Organizations should take a holistic view of Agile implementations [P7] ranging from people aspects [P8] to tools [P5], from the individual level to the organizational level [P8]. As long as the trainings trigger behavioral transformation [P5, P16], they cannot go beyond being a weak start [P5]. In training, Serious Games can be used [P11] not only for teaching but also for learning for each individual. Highlighting the main features of Agile Mindset and in what way it differs from the mindset of a more traditional one is key to its internalization [P13]. A single teaching experience will not help. Teams should understand the vision and reasons behind the practices [P8]. After introducing and implementing the single Agile practice, some adjustments will be required according to need [P8]. During the transformation, combining existing elements into the current way of working can be used to facilitate a smooth and evolutionary process [P8].

Practices put aside, even Agile Mindset is not sufficient for being agile [P16]. As mentioned, the transformation of a mindset is challenging and takes considerable time. Therefore, it is crucial to be patient and give time to all entities involved in the transformation to mature [P7]. In this process, it is important to consider the context and unique nature of changing environments, let them fail, and try a new approach [P8]. The whole transformation process of the Agile Mindset should be meticulously followed by internal/external experts [P16, P19], and plans and progress should be monitored and measured [P13, P16]. Organizations should be aware of misconceptions and obstacles and remove them as soon as possible [P8].

TABLE VI: WAYS TO DEVELOP AGILE MINDSET

Ways to Develop Agile Mindset	Paper ID
The shift from top to bottom	P5
Integrate new tool use with Agile Mindset	P5
Establishing a continued and consistent focus on value creation throughout the development process consistent with Agile Mindset.	P5
Training sessions should be reinforced over time in a persistent manner until the mindset and practices become habitual	P5
Take a holistic view of Agile implementation	P7
Give time to mature	P7
Build trust	P7

Consider: (1) personal prerequisites and attitudes, (2) what the team has to provide for the collaboration with the coach, (3) problems on the team level, (4) the team's needs, and (5) what the team needs to learn.	P8
Consider the aspects related to the agile coach: (1) observing and understanding (the team), (2) activities of the coach, (3) making agile tangible, (4) perception of agile by the coach, and (5) experiences of the coach.	P8
Take care of (1) the collaboration between coach and management and (2) misconceptions and obstacles.	P8
Understand vision and reasons behind introducing an Agile way of working, get a brief theoretical training, including an explanation of why the respective practice is helpful and should be implemented, and start implementing single Agile practices, and adjust them.	P8
Integrate existing elements in the way of working to ease the transformation	P8
Observe teams to get an understanding of teams' dynamics, their current situation, and their way of working	P8
An Agile coach has to be a role model for showing Agile values	P8
Integrate teams into whole processes and listen to their ideas, concerns, and needs.	P8
Allow development teams to fail and to try something that may or may not work	P8
Serious Games can be used in training for Agile Mindset	P11
Find a basis on which to identify the main features of Agile Mindset and in what way it differs from traditional mindset	P13
Develop and plan a way of shifting Agile Mindset of a current team	P13
Measure the progress	P13
Build in advance resources that can be used at short notice	P16
Managers should invest in training, building, and measuring their organization's Agile Mindset	P16
Start by collaborating with expert institutions to train their managers to acquire an Agile Mindset	P16
Agile Mindset is not enough; organization must 'walk the talk'	P16
Place Agile Mindset Trainer roles	P19

Everything is influenced by people's mindset, even human-made artifacts such as tools, processes, and organizational structures. For instance, tools inherit a mindset from the person who produced them. Even such dummy entities should be aligned with the targeted Agile Mindset, which the organization desires for the individuals. We see such a need in the study of [P5] that proposes an integration of new tool use with an Agile Mindset and Agile resources management method which was suggested by [P16]. Agile practices should also be conduct with a proper mindset. 'Doing Agile' can be a step on the way towards fully embracing the Agile Mindset" [25, 52], but, starting blindly or only with Agile practices is not satisfactory; they alone do not guarantee being agile [9, 14, 29] and "without the right mindset, the methods are often adapted in a wrong way and lose their purpose" [9].

The findings indicate that Agile Mindset transformation, is a challenging, grueling, and long journey that requires patience and effort. It must focus on all dimensions of change, and in itself must be conducted in an agile way.

What are the Indicators of Agile Mindset? (RQ2.7):

When it comes to the indicators which reflect Agile Mindset in behaviors of people, we identified that those people collaborate with others [P4, P14] and use real-time planning [P1]. They behave with ownership, make decisions autonomously, and build connections between issues [P4]. They respond to changes [P4], focus on delivery [P4] and search for continuous improvements and new, unconventional, and better ways for organizations' management structures, methods, and systems [P14, P16], even when it is challenging [P16]. One of the fields that prove the existence of a strong Agile Mindset can be seen in flexible, quick, fluid, and successful resource management [P16]. Agile Mindset is indicated to be related to superior performance and higher innovativeness [P16]. While trustful interactions are a prerequisite and facilitating factor for the Agile Mindset, its presence is also an indicator of the Agile Mindset's existence [P14].

What are the Future Directions for Agile Mindset Researches? (RQ2.8):

Table VIII lists possible future work items extracted from the identified studies. While there are many studies investigating how the technical side of Agile can be agile, we see that reflections of them on the Agile Mindset need to be studied in the future. For instance, [P16] asserts that no study has examined how a mindset can be agile [at least until their work was conducted in the year 2022] or how it can be measured [P18].

While physical actions are observable, we need to find ways to observe and ensure that individuals are immersed in Agile Mindset [P5] and to remove the impediments which hinder achieving the Agile Mindset [P18]. Some personalities have more potential in terms of supporting Agile Mindset than others. Then, it would be interesting to combine research on Agile Mindset with research on personality, social aspects [P8], and experiences and maturity of practitioners [P19] to examine what type of organizations and people are able to utilize Agile Mindset more [P15]. Each unique individual, team, and organization should find an optimum level for their Agile Mindset by considering its tradeoffs and side effects. For instance, it can be interesting to research whether having an excessive Agile Mindset harms performance [P16], quality, or other aspects. We need to locate the responsibilities of Human Resource Management, Talent Development, other departments, and leaders in installing target-oriented initiatives and providing ways in which actors can develop their Agile Mindset [P18]. There is a need for valuable insights into Agile leaders' mindset and their effects on organizations [P18].

Agile Mindset should be integrated into a comprehensive network with other constructs [P18]. A satisfactory number of studies is missing in the literature to measure Agile Mindset [P18]. We also need more studies on how the state of having an Agile Mindset can be achieved at different levels starting from the individual level to the organizational level [P18, P19], by distinguishing differences between the perceived Agile Mindset at different organizational levels [P19].

Among the Agile Mindset elements, which of them are the most important and how and to what extent each element supports real agility and Agile Mindset can be sought [P8, P9] by categorizing them [P19]. Another research area can be what determines Agile Mindset and what organizational outcomes Agile Mindset and actors with Agile Mindset influence [P18, P19]. One of the interesting dimensions of the construct can be about time. Specifically, how the characteristic of the construct changes over time can be a matter of interest [P19].

TABLE VIII: FUTURE WORK ITEMS

Future Work Items	Paper ID
Are there ways to ensure that individuals are immersed in the Agile Mindset?	P5
Combining research on Agile mindset with research on personality and social aspects, for instance investigating what type of organizations and people are able to utilize Agile Mindset	P8, P15
What Agile elements are the most important and how and to what extent each element supports real agility and Agile Mindset?	P8, P9
Can an excessive Agile Mindset harm performance?	P16
Can Human Resource Management and Talent Development Departments install target-oriented initiatives and provide a framework in which actors can develop their Agile Mindset themselves?	P18
Investigating influences on Agile Mindset	P18
Investigating effects of managers' Agile Mindset on employees	P18
How actors with an Agile Mindset interact [with a specific subject]	P18
How managers with Agile Mindset empower their employees to develop their Agile Mindset	P18
Exploring nomological network of Agile Mindset, specific behaviors, and practices or social Agile practices	P18
How Agile Mindset improves other outcomes such as value to organizations	P18
Investigating impediments that hinder employees and teams to achieve Agile Mindset and transfer it into action	P18
Studying Agile Mindset on individual and organizational levels	P18, P19
Studying whether characteristics can be categorized and change over time	P19
Investigating whether there are differences in perception of mindset related to Agile experiences and maturity of practitioners	P19
Investigating whether there are differences between organizational levels in perceiving the Agile Mindset	P19

We point out that the Agile community and practitioners are aware that internalizing the Agile values and principles is key to being agile. However, it is remarkable that there has been very limited interest in Agile Mindset and the social aspects of agility [1], which is especially relevant for information systems as well organization studies. Even though Agile development is more about people and human factors from the onset [53], and people and human factors are an underlying foundation of agility as defined in the Agile Manifesto [54], the increased interest in Agile in the academic field is more on concrete entities such as practice, method, and frameworks. For instance, while there are plenty of cases investigating how practices can be agile, [17] asserts that no study has examined how a mindset can be agile [at least until their work was conducted in 2022] and studies are limited on how it can be measured [1]. In another instance, while our search with the keyword ("agile mindset" OR "agile mind set" OR "agile mind-set") brought 75 initial results in Scopus, the search in the same database within the same condition with the "Scrum" keyword brings 4,047 results (54 times of the former one). These, among others, indicate that Agile-related works focus on the practical, concrete, "easy-to-perceive" side of it.

On the "harder-to-implement side", Agile Mindset requires shifting to a whole new way of thinking, which manifests a challenge to unlearn old and traditional practices and to move towards new ones [55]. One of the other reasons for inhabiting the harder side to implement can be because Agile Mindset is a relatively hard-to-internalize-aspect of Agile in organizations [34, 35], and changing the mindset of employees and management seems to be more difficult than the mere implementation of Agile practices that is rather simple [9].

Gelmis et al. [34] state that because "these aspects are abstract, the transformation of a mindset is the most difficult part of the work and hard to prove and show; then, consultants do not prefer such a transformation [since the transition process can take several years and requires major resources] [35]. Rather, they prefer to transform only the concrete substances of the organizations. [50], [14], and [56] put forward that the industrialization effects driven by the Agile marketing and selling Agile™ products and "Fake Agile" to organizations have caused the overshadowing of Agile Mindset and prevented organizations to properly understand real and market-independent agility. Thus, Agile Mindset stays behind the "sold" practices because the market may want to sell "agility" for their economic interest [50]. This trading mostly ends with an illusion of "doing agile", which takes years to realize and overcome. It can also be seen with the first paper focusing mainly on Agile Mindset published in 2014, 13 years later the Agile Manifesto was announced and 20 years later Scrum emerged. We see a similar reflection of this case in agile trainings; the abstract nature of Agile Mindset leads to a limited study on explicit training for it [23]. Briefly stated, while there is an intense focus on the methods, the number of studies on the mindset part is very low [6, 20, 23], due to some reasons such as economic interests [57] and the abstract nature of it [23]. Consequently, many organizations fail to enable an Agile Mindset of the individuals [31].

However, an increasing number of researchers have started to focus on the internal aspects and human side of agility [1]. It seems that Agile Mindset aspects will be on the agenda of the organizations now and in the future [34]. This increasing interest of researchers in recent years can also be seen in our study results.

Even more, [34] foresees that the focus on the people side and having a proper Agile Mindset will be more important and the predefined practices will have relatively less place in the future. Similarly, [58] argues that after a while, Agile practices will be largely equalized for organizations, and organizations that make a difference will come to the fore with the people dimension. While the majority of organizations today prefer to focus on Agile frameworks, in the future, there can be some new frameworks tailored to each organization depending on organizational cultures and needs [34], which requires the intellectual capabilities of people's minds to make practices evolve more organically.

It is seen in our work that most of the excluded studies are satisfied by only mentioning the term Agile Mindset as a "fixed concept" without actual descriptions, details, explanations, or definitions. This case has also been witnessed by the study of [6] and [20]. Most of these publications investigate Agile Mindset either as a precondition or a relation to organizational culture [9]. When mentioned by some studies, Agile Mindset has been confined to being understood as only one category amongst many [59] or only a prerequisite for implementing Agile.

Regarding the RQ2, there are very limited resources especially involving definitions of Agile Mindset, its elements, methods to develop it, indicators, and the measurement of it. Inappropriate definitions lead to a variety of invalid measures [1]. Because of the missing consensus in existing Agile Mindset conceptualizations, an aligned comprehensive understanding of it must be conducted and conceptualizations from different perspectives and levels [1, 9]. Measuring instruments for this different conceptualization and progress on different levels are missing [1].

How can we decide the optimum level needed for Agile Mindset? What other capabilities should Agile Mindset be supported with? How can those who have a low Agile Mindset and those who have a high Agile Mindset work together? Or, do we need to have a common mindset understanding as stated by [35]? What are the effects of the tenets of Agile Mindset on agility? What are the relationships between a person's personality and Agile Mindset, which is currently absent from the academic literature?

With these and many more unanswered questions, it seems that there is a long way to go on this topic and that further studies should involve behavioral research, cognitive science, learning (rather than teaching), and other disciplines, to accelerate this relatively new concept.

IV. CONCLUSIONS AND LIMITATIONS

Although the importance of Agile Mindset is known by many studies and people, the construct interestingly seems to be underrated in the literature. Most of the studies involving the Agile Mindset term as a fixed term. For this reason, it seems necessary to further unbox the construct. To do so, it seems necessary to create projects to reflect the situation regarding Agile Mindset in practice. Although Agile Mindset is a new construct to study, it is recommended in the first stage to consider the studies that deal with the construct of mindset in general terms and to benefit from studies from multiple related disciplines.

Our study aims to deal with the important Agile Mindset construct comprehensively, using sources from many disciplines while doing this. We aim to open a door to this construct, which is worth researching in terms of practice and theory. In the future, we will study on the development and measurement of the Agile Mindset for individuals in organizations.

The procedures used in our study have limitations in several ways. Limitations of search terms and search engines can lead to an incomplete set of primary sources. It is possible that we may have missed some relevant studies as we did not include all possible libraries. In particular, we have missed the studies published in non-peer-reviewed resources. To minimize risks that may result from search engines process, we included two comprehensive academic databases and used a comprehensive search string developed through several iterative improvement processes. We recorded each paper that we found with its source in an Excel sheet. Therefore, we believe that an adequate and inclusive basis was established for this study.

Defining search terms in the source selection approach resulted in obtaining only the sources written in English and the peer-reviewed ones. However, the main issue regards whether the selected works represent all types of literature in the area of study. We ensure that the relevant studies collected in the study pool contained sufficient information to represent the entire related literature.

A single researcher extracted the data from the included studies. Also, the values of the quality assessment criteria are subjective but based on field experience. Moreover, the primary studies' results are context dependent and have thereby limited generalizability. However, when these processes were unclear a consensus session was applied with the second author. Additionally, the relevant data was taken as an actual extraction of terms from the identified studies and copied to the Excel file. To ensure the reliability of our study, the entire pool of the sources was analyzed carefully and the data were reviewed, extracted, and synthesized in iterations according to the research protocol and guideline applied.

REFERENCES

[1] K. Eilers, C. Peters, and J. M. Leimeister, "Why the agile mindset matters", Technological Forecasting and Social Change, 179, 121650, 2022.

[2] J. Ryskowski, "Revealing the Unobvious Social Norms and Traditional Development Fantasies that Impede Agile Adoption", 2018 International Conference on Computational Science and Computational Intelligence (CSCI) pp. 829-834, IEEE, 2018.

[3] B. Horlach and A. Drechsler, "It's not easy being agile: Unpacking paradoxes in agile environments", Agile Processes in Software Engineering and Extreme Programming–Workshops: XP 2020 Workshops, Revised Selected Papers, pp. 182-189, Springer International Publishing, 2020.

[4] S. Denning, "Agile's ten implementation challenges", Strategy Leadersh, vol.44, 15–20, 2016c.

[5] K. Crnogaj, P. Tominc, and M. Rožman, "A Conceptual Model of Developing an Agile Work Environment", Sustainability, vol. 14, no.22, 14807, 2022.

[6] A. Mordi and M. Schoop, "Making it tangible–Creating a definition of agile mindset", 28th Conference on Information Systems (ECIS2020), 2020.

[7] T. Cooke-Davies, "The "real" success factors on projects", International journal of project management, vol. 20, no.3, 185-190, 2002.

[8] M. G. Weinberg, Secrets of consulting: Dorset House, 1985

[9] J. Klünder, F. Trommer, and N. Prenner, "How agile coaches create an agile mindset in development teams: Insights from an interview study", Journal of Software: Evolution and Process, vol. 34, no.12, e2491, 2022.

[10] A. McDermid and K. H. Bennett, "Software Engineering Research: A Critical Appraisal", IEE Proceedings on Software, vol. 146, no.4, 1999.

[11] C.O. Melo, C. Santana, and F. Kon, "Developers motivation in agile teams", 38th Euromicro Conference on Software Engineering and Advanced Applications, 2012.

[12] H. C. Sharp et al., "The Role of 'Culture' in Successful Software Process Improvement", 25th EUROMICRO Conference, pp. 170-176, 1999.

[13] E. Whitworth and R. Biddle, "Motivation and cohesion in agile teams". Agile Processes in Software Engineering and Extreme Programming: 8th International Conference, XP 2007, Como, Italy, June 18-22, 2007, pp. 62-69, Springer Berlin Heidelberg, 2007.

[14] P. Hohl J. Klünder and A. van Bennekum, "Back to the future: origins and directions of the "Agile Manifesto"—views of the originators", J Softw Eng Res Dev., vol. 6, no.1, 2010, doi:10.1186/s40411-018-0059-z.

[15] N. Ozkan, T. G. Erdogan, and M. Ş. Gök, "A Bibliometric Analysis of Agile Software Development Publications", 3rd International Informatics and Software Engineering Conference (IISEC), pp. 1-6, IEEE, 2022.

[16] C. A. Sathe and C. Panse, "Analyzing the impact of agile mindset adoption on software development teams productivity during COVID-19", Journal of Advances in Management Research, 2022.

[17] Y. Asseraf and I Gnizy, "Translating strategy into action: The importance of an agile mindset and agile slack in international business", International Business Review, vol. 31, no.6, 102036, 2022.

[18] R. Kramer, "From skillset to mindset: a new paradigm for leader development", Вопросы государственного и муниципального управления, vol.5, 26-45, 2016

[19] A. R. Zablah, B. P.Brown, and N. Donthu, "The relative importance of brands in modified rebuy purchase situations", International Journal of Research in Marketing, vol.27, no.3, pp.248–260, 2010.

[20] N. Ozkan and M. S. Gok, "Investigation of Agile Mindset Elements by Using Literature Review for a Better Understanding of Agility", Turkish National Software Engineering Symposium (UYMS), pp. 1-6, IEEE, 2020.

[21] M. K Petermann and H. Zacher, "Agility in the workplace: Conceptual analysis, contributing factors, and practical examples", Industrial and Organizational Psychology, vol. 13., no. 4, pp. 599-609, 2020.

[22] H. van Manen, and H. van Vliet, "Organization-wide agile expansion requires an organization-wide agile mindset", Product-Focused Software Process Improvement: 15th International Conference, PROFES 2014, Helsinki, Finland, December 10-12, 2014. pp. 48-62, Springer International Publishing, 2014.

[23] I. Fronza and X. Wang, "Revealing Agile Mindset Using LEGO® SERIOUS PLAY®: Experience from an Online Agile Training Project", 34th International Conference on Software Engineering and Knowledge Engineering, SEKE 2022, pp. 428-433, 2022.

[24] P. Gregory, L. Barroca, H. Sharp, A. Deshpande, and K. Taylor, "The challenges that challenge: Engaging with agile practitioners' concerns", Information and Software Technology, vol. 77, 92-104, 2016.

[25] S. Denning, "How to make the whole organization "agile"", Strategy and Leadership, vol. 44, 10–17, 2016.

[26] O. Mikhieieva, R. Baumgartner, K. Stephan, and E. Lipilina, "Agile Mindset Competencies for Project Leaders", IEEE European Technology and Engineering Management Summit (E-TEMS), pp. 208-213, IEEE, 2022.

[27] M. Durbin and F. Niederman, "Bringing templates to life: overcoming obstacles to the organizational implementation of Agile methods", International Journal of Information Systems and Project Management, vol. 9, no.3, pp.5-18.

[28] N. Ozkan, M. Ş. Gök, and B. Ö. Köse, "Towards a better understanding of agile mindset by using principles of agile methods", 15th Conference on Computer Science and Information Systems (FedCSIS), pp. 721-730, IEEE, 2020.

[29] M. Kuhrmann,"What makes agile software development agile", IEEE Trans Softw Eng, 2021, doi:10.1109/TSE.2021.3099532.

[30] J. Miler and P. Gaida, "On the agile mindset of an effective team—an industrial opinion survey", Federated Conference on Computer Science and Information Systems (FEDCSIS), 2019.

[31] Z. Wang, S. L. Pan, T. H. Ouyang, and T. C. Chou, "Achieving IT-enabled enterprise agility in China: an IT organizational identity perspective", IEEE Transactions on Engineering Management, vol. 61 no.1, pp.182-195, 2013.

[32] B. Kitchenham, et al., "Systematic literature reviews in software engineering–a systematic literature review", Information and software technology, vol. 51, no.1, pp. 7-15, 2009.

[33] G. K. Gouda and B. Tiwari, "Ambidextrous leadership: a distinct pathway to build talent agility and engagement," Human Resource Development International, pp. 1-9, 2022.

[34] A. Gelmis, N. Ozkan, A. J. Ahmad, and M. G. Guler, "Perspectives on the Sustainability and Future Trajectory of Agile", Systems, Software and Services Process Improvement: 29th European Conference, EuroSPI 2022, pp. 443-458, 2022.

[35] C. Lindskog and J. Netz, "Balancing between stability and change in Agile teams", International Journal of Managing Projects in Business, vol.14, no.7, 1529-1554, 2021.

[36] J. Miler and P. Gaida, "Identification of the agile mindset and its comparison to the competencies of selected agile roles", Advances in Agile and User-Centred Software Engineering: Third International Conference on Lean and Agile Software Development, LASD 2019, and 7th Conference on Multimedia, Interaction, Design and Innovation, MIDI 2019, Leipzig, Germany, September 1–4, 2019, Revised Selected Papers, pp. 41-62, Springer International Publishing, 2020.

[37] I. Bider and O. Söderberg, "Moving Towards Agility in an Ordered Fashion", ICEIS, pp. 175-199, Springer International Publishing, 2017.

[38] A. Mordi and M. Schoop, "Scaling with an Agile Mindset-A Conceptual Approach to Large-Scale Agile", AMCIS, 2021.

[39] M. Senapathi and A. Srinivasan, "Sustained agile usage: A systematic literature review", 17th International Conference on Evaluation and Assessment in Software Engineering, pp. 119-124, 2013.

[40] K. Rolland, T. Dingsoyr, B. Fitzgerald, and K. J. Stol, "Problematizing agile in the large: alternative assumptions for large-scale agile development", 39th International Conference on Information Systems, pp. 1-21, Association for Information Systems (AIS), 2016.

[41] I. Bider and O. Söderberg, "Becoming Agile in a Non-disruptive Way-Is It Possible?", ICEIS, pp. 294-305, 2016.

[42] P. Gregory et al., "Onboarding: how newcomers integrate into an agile project team", Agile Processes in Software Engineering and Extreme Programming: 21st International Conference on Agile Software Development, XP 2020, pp. 20-36, 2020.

[43] H. Hajjdiab and A. Taleb, "Adopting Agile Software Development: Issues and Challenges", IJMVSC, vol. 2, no.3, pp.1-10, 2011.

[44] K. Conboy, S. Coyle, X. Wang, and M. Pikkarainen, "People over Process: Key Challenges in Agile Development," IEEE Software, vol. 28, no.4, pp.48-57, 2011.

[45] S. Denning, "What's Missing In The Agile Manifesto: Mindset", Forbes, Available Online: https://www. forbes. com/sites/stevedenning/2016/06/07/the-key-missingingredient-in-the-agile-manifesto-mindset, 2016b.

[46] A. F. Sommer, "Agile Transformation at LEGO Group: Implementing Agile methods in multiple departments changed not only processes but also employees' behavior and mindset", Research-Technology Management, vol. 62, no. 5, pp.20-29, 2019.

[47] Cambridge Dictionary, Meaning of mindset in English, URL: https://dictionary.cambridge.org/dictionary/english/mindset (visited on 16/05/2023), 2023.

[48] Macmillan Dictionary, Definition of mindset in English. URL: https://www.macmillandictionary.com/dictionary/british/mindset (visited on 16/05/2023), 2023.

[49] N. Ozkan and M. S. Gok, "Definition Synthesis of Agility in Software Development: Comprehensive Review of Theory to Practice", International Journal of Modern Education and Computer Science, vol.14, no.3, 2022.

[50] N. Ozkan and M. S. Gok, "Towards the End of Agile: Owing to Common Misconceptions in the Minds of Agile Creators", ICSOFT, pp. 224-232, 2021.

[51] R. Hoda, J. Noble, and S. Marshall, "Balancing acts: walking the Agile tightrope", ICSE Workshop on Cooperative and Human Aspects of Software Engineering, pp. 5-12, 2010.

[52] T. Dingsøyr and N.B. Moe, "Towards principles of large-scale agile development", In: T. Dingsøyr, N.B. Moe, R. Tonelli, S. Counsell, C. Gencel, K. Petersen (eds.) XP 2014, LNBIP, vol. 199, pp. 1–8, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-14358-3_1

[53] A. Cockburn and J. Highsmith, "Agile software development, the people factor," Computer, vol, 34, no.11, pp. 131-133, 2001.

[54] K. Beck, et al., The agile manifesto, 2001. https://agilemanifesto.org/

[55] S. Thangasamy, "Lessons learned in transforming from traditional to Agile development", Journal of Computer Science, Vol. 8 No. 3, pp. 389-392, 2012, doi: 10.3844/jcssp.2012.389.392.

[56] A. L. M. Calafat, A. Mas, and M. Pacheco, "Fake Agile: What Is It and How to Avoid It?", IT Professional, vol. 24, no. 2, 69-73, 2022.

[57] F. Yvette, "Is the Agile Manifesto dead? Not by a long shot", TechBeacon, [Online] Available: http://techbeacon.com/agilemanifesto-dead-not-long-shot, 2015.

[58] C. Alexander, "Beyond the Agile Manifesto: Epoch of the Team", Crosstalk: The Journal of Defense Software Engineering, vol. 29, no.4, 2016.

[59] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review", Journal of Systems and Software, vol. 119, pp. 87-108, 2016.

[60] P. Hohl, et al., "Back to the future: origins and directions of the "Agile Manifesto"–views of the originators", Journal of Software Engineering Research and Development, vol.6, no.1, 2018.

DOI: 10.15439/2023F2503
ISSN 2300-5963 ACSIS, Vol. 35

Sign language interpreting - relationships between research in different areas - overview

Barbara Probierz*†, Jan Kozak*†, Adam Piasecki* and Angelika Podlaszewska*
* Łukasiewicz Research Network – Institute of Innovative Technologies EMAG,
Leopolda 31, 40-189 Katowice, Poland
Email:barbara.probierz,jan.kozak,adam.piasecki,angelika.podlaszewska@emag.lukasiewicz.gov.pl
† Department of Machine Learning, University of Economics in Katowice,
1 Maja 50, 40-287 Katowice
Email:barbara.probierz,jan.kozak@ue.katowice.pl

***Abstract*—Translation from the national language into sign language is an extremely important area of research and practice, which aims to ensure communication between deaf or hard of hearing people and the hearing community. The article provides an overview of the most important research on sign language interpretation conducted in various research areas. The latest scientific and theoretical achievements were presented, which contribute to a better understanding of the subject of sign language translation and the improvement of the quality of translation services. Our main goal is to identify outstanding areas of interdisciplinary research related to sign language translation and to identify links between these studies conducted in different areas. The conclusions of the article aim to broaden the knowledge and awareness of sign language translation and to identify areas that require further research and development. The work is linked to a project related to the application of machine learning in increasing accessibility for deaf people.**

I. Introduction

SIGN language is an integral part of the lives of deaf people, enabling them to communicate, express their emotions and participate in society. Sign language interpreters are essential to ensure a balanced access to information and services for people with hearing impairments. Sign language interpreters play a key role in the transfer of information between deaf and hearing people, enabling full participation in various areas of social and professional life [1].

Sign language is a natural language that, like any other natural language, has its own grammar, vocabulary, and sentence structure [2]. There are many different sign languages in the world, each with its own unique characteristics. Therefore, translating from a national language into a sign language requires not only knowledge of the sign language, but also an understanding of the culture and social context of the people who use it. For this reason, the process of translating from the national language into a sign language is complicated and requires the interpreter not only to know both languages, but also to be able to interpret and translate the meaning. Sign language interpreters must consider not only the literal meaning of words, but also their connotations and cultural context. It is also important to skilfully use gestures, facial expressions and body movements to convey the speaker's emotions and intentions [3].

Increasing equal opportunities and participation of deaf people in society is important, therefore the need to develop solutions aimed at reducing the social exclusion of deaf people is the result of growing social awareness and legal obligations regarding equal access to services and information for all citizens. For this reason, research and development of solutions for automatic translation of natural language into sign language are of great importance to deaf people. Creating new solutions that automatically convert spoken language into sign language has great potential in removing communication barriers and enabling full participation of deaf people in various areas of social and professional life.

Research into automatic translation of natural language into sign language requires the use of advanced technologies such as artificial intelligence, machine learning and natural language processing. As a result, the development of such solutions contributes to technological progress and is also used in other fields, such as machine translation or speech recognition. The most important thing is that scientific research and creating solutions for automatic translation of natural language into sign language have a direct impact on improving the quality of life of deaf people. Eliminating communication barriers and enabling full participation in society contributes to greater equality and social inclusion for this community.

The aim of this article is to present an overview of research on sign language translation and to identify links between research conducted in various areas related to this field. Through our work, we aim to provide readers with a comprehensive view of contemporary research related to sign language interpretation, addressing a variety of aspects, such as translation efficiency, quality of interpreting services, technological support, and social and cultural contexts. Our work focuses on identifying common points and interconnections between research conducted in various areas that, having an interdisciplinary nature, relate to sign language translation. By analyzing these relationships, the article aims to develop understanding of the sign language translation process and on the potential benefits and challenges of an interdisciplinary approach to research in this field.

Our key goal is to implement the project of developing a virtual human figure presenting the content of public ad-

Topical area: Information Technology for Business and Society

ministration in Polish Sign Language. The project broadly aims to increase accessibility for deaf people – particularly those who use Polish Sign Language but do not understand Polish. It is a comprehensive solution in which it is necessary to prepare both a module for language translation and a corresponding avatar that is used to sign translated texts. To achieve this goal, it is necessary to familiarize yourself with the research and scientific and theoretical achievements in the field of sign language translation and their implications for translation practice. Although many of the analyzed studies are interdisciplinary in nature, one can distinguish the pursuit of scientists in specific directions. For this reason, we want to present our review of the work in the proposed research areas and encourage researchers to further research and development in the field of sign language interpretation. By identifying areas for further research and challenges, the article aims to stimulate conversations and innovations that will contribute to the continuous improvement of sign language services and the full participation of people with hearing impairments in society.

This article is organized as follows. The section I provides an introduction to the subject of this article. Section II provides an overview of related work on sign language interpretation methods. Section III describes a solution model that uses machine learning to automatically translate texts into Polish Sign Language. In the section IV, we present our proposed analysis of the literature review, where we indicate key research areas. In the sections VI, V and VII we present research into sign language translation broken down into the indicated areas. In the section IX, we present the results of identified links between research conducted in various areas. Finally, in the section X we conclude with general remarks about this work and indicate some directions for future research.

II. Related works

Traditional methods of obtaining information allow deaf people to become familiar with a very small percentage of digital information. In the context of the huge amount of material to be translated, it seems necessary to use semi-automatic and automatic tools to make at least some of it available in sign language on an ongoing basis [4]. Three basic barriers to automatic sign language translation have been distinguished, i.e. differences in modality, lack of a standard written form and a shortage of resources in the form of tools and technologies for translation as well as a limited number of sign language corpora [5]. In addition, the scope of sign language synthesis was defined by a detailed discussion of sign language dictionaries and repositories [6], [7]. A thorough review of the synthesis of sign language from the perspective of animation was also carried out, paying attention to the difference between the synthesis of single signs and the generation of full statements [7].

In some countries, public administrations are required to provide sign language interpretation for deaf people to enable them to fully enjoy public services. Translations are required to be available in places such as government offices, hospitals, schools and courts [8]. Public organizations must meet guidelines on the qualification and professionalism of sign language interpreters to ensure effective and accurate interpretation. The authors [9] focused on the criteria that public entities placed in virtual reality would have to meet, and pointed out that the appearance of avatars representing officials should be adequate to the situation of a fairly official nature. In [10], a framework of public values was developed that virtual advisors in offices should follow.

Many works indicate the interdisciplinary nature of the subject of sign language avatars and the great importance of non-manual signs, facial expressions and facial expressions, which is a great challenge for researchers dealing with the subject of sign language synthesis [11], [12]. The author [13] points to three interrelated threads, i.e. a linguistic approach to facial expressions that an avatar must have in order to convey a comprehensible message, computer graphics providing tools and technologies that are required to create avatars, and the third thread deals with the subject of sign language representation systems from the point of view of their ability to represent non-manual signs and facial expressions. The topic of non-manual signs, facial expressions and generating synthetic emotions was also addressed in works [3], [14]–[16]. Moreover, in [17], [18], a study of social trust in chat bots conducted was described. The study was based on theories of operators' trust in machines in industrial environments, showing that acceptance and trust can be related to the field in which the technology was introduced.

III. Project $Avatar2PJM$

The research is part of the $Avatar2PJM$ project (Project: Framework of an automatic translator into the Polish Sign Language using the avatar mechanism, The National Centre for Research and Development, GOSPOSTRATEG-IV/0002/2020). This project aims to develop a solution for translating speech from Polish into Polish Sign Language using avatar and artificial intelligence methods. The innovation of this solution lies in the inclusion of emotions and non-verbal elements in the visualisation of gestures.

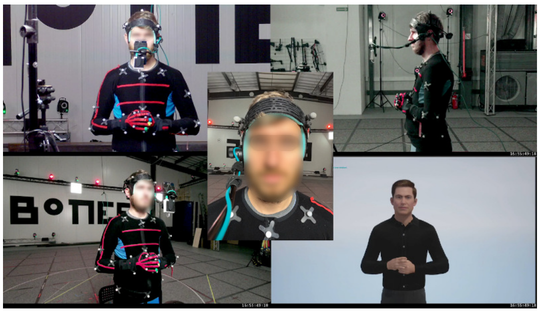


Fig. 1. MoCap recordings as part of the $Avatar2PJM$ project.

One of the key objectives of the project is to develop a method of translating Polish into Polish Sign Language using the avatar application control mechanism. A sign language avatar is a computer representation (animation) of linguistic phenomena. With appropriate reference material recorded in video form, it is possible to animate any described speech. To this end, Motion Capture (MoCap) sessions were conducted, which is a technique for capturing the actor's three-dimensional movements (see Fig. 1). Used in computer games, the MoCap technique imitates the natural movements of objects or people in a very realistic way to achieve a natural effect. In the case of sign language avatars, MoCap makes it possible to copy sign language signs and increase the understanding of the communicated content, since, from the animator's perspective, spoken sign language signs consist of geometric positions and movements.

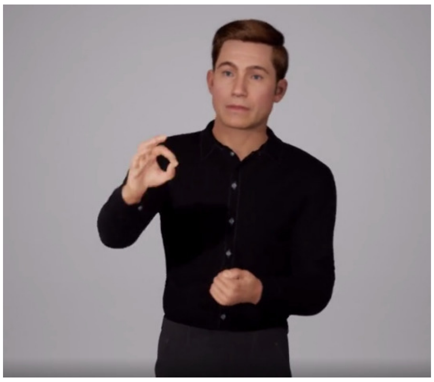


Fig. 2. Actual avatar prepared for testing the $Avatar2PJM$ project.

A sign language message consists of both sign language signs and various additional information, as what is physically expressed is the result of co-existing linguistic and extra-linguistic processes. In the process of creating computer-generated animations, the emotional context of the utterance is taken into account, as well as phenomena such as correct mouth movements or voiceless speech that occur during sign language communication. In particular, the sign language interpreter's facial expressions and the information they convey are important. Such elements are also important in the context of the data needed to develop the interpreting module. The material acquired during the MoCap session is used to feed the animation module and provide an input data set for the translation module based on machine learning methods. For this to be possible, it is necessary to subject the collection of recordings to an annotation process. The annotation process involves describing individual sign language sign elements at specific intervals. This includes sign units, lexical interpretations (lemmas, lexemes), as well as information concerning the non-manual elements of the sign. Since one of the key elements of annotation is the sign language interpreter's face, and this process is time-consuming, an attempt was made to explore the possibility of automatically recognising the interpreter's facial expressions. Automatic annotation would significantly improve and speed up the annotator's work. The article describes partial results of the research carried out in this field.

One of the expected outcomes of the project is pilot testing of selected online information services run by public administrations (the avatar used in the tests is presented in the Fig. 2). The widespread use of automatic translation mechanisms in public online systems is a constructive step towards improving the accessibility of digital public administration. In addition, the project team is investigating the professional potential of deaf people and their satisfaction with contact with public administration before and after the implementation of the virtual interpreter. This will identify the social and economic barriers that deaf people face in their contacts with the administration and in the labour market. The vocational potential of deaf people will also be explored, as well as methods of capturing data to maximise the effects of vocational activation. The results of the project will contribute to the sustainable elimination of barriers faced by users of Polish Sign Language.

IV. Proposed approach

When analyzing the current state of knowledge on the subject of sign language interpretation, one should pay attention to the interdisciplinarity of the subject of research, which combines various areas of knowledge in order to better understand this natural language of communication for the deaf. The interdisciplinarity of this research allows for a holistic approach to the issue of sign language translation. It requires collaboration between scientists and specialists in various fields to improve the translation process and develop new tools and strategies.

In order to provide a comprehensive view of contemporary research on sign language, we decided to analyze many scientific papers related to sign language translation, group them according to the nature of research, and then identify links between research conducted in different areas. To this end, we can identify three research areas related to sign language:

- the technological area where research focuses on developing and improving technological solutions related to sign language interpretation;
- the character animation area, where the proposed approaches relate to character animation for conveying information in sign language;
- the area of application, which presents solutions related to the use of automation in the field of providing information in sign language.

In the technological area, we want to classify works related to e.g. with translation into sign language, also in terms of the transition from the national language to the national sign language. In these works, researchers focus on developing innovative technological solutions for sign language translation. In this respect, classic methods can be distinguished, which include converting speech from the national language into sign language, often using motion capture techniques to generate animations. However, the development of artificial intelligence is also playing an increasingly important role, enabling the automatic translation of speech into sign language. In the technological area, research is focused on improving these methods and developing new tools and strategies that will improve the sign language translation process.

The character animation area focuses on ways to animate characters that convey information through sign language. Here too, motion capture techniques are often used to generate character animations that reflect flicker. It is also possible to create animations based on pre-recorded sequences performed by a lector. The aim of research in this area is to improve character animation techniques and search for new solutions that will allow even better transfer of information in sign language.

On the other hand, in the area of applications, research focuses on the use of automation in the transmission of information in sign language. Automation can be used in various contexts, for example in translation systems or online communication. The aim of the research is to improve these solutions so as to enable easier and more effective communication for deaf people using sign language.

In addition, there are many works related to with the analysis of accessibility [19], chaterbots [20], translations of sign languages [21] (e.g. from English (BSL) or American (ALS) to another sign language), or the development of applications related in some way to the implemented project [22]. Such a group in our research was named as other works and requires further analysis.

V. The technological area

The technological area is usually concerned with the translation of spoken language into sign language as input using text, sound or image. There are also studies on reverse translation, for example the solution described in [23], the system is able to recognize sign language poses and translate through avatars in the form of talking faces [24]. Many works also focus on the development of two-way communication by creating solutions that translate spoken languages into sign languages, and are able to recognize sign languages, as for example in works [8], [25], [26].

A. Recognition and translation of sign language into spoken languages

In the field of recognizing and translating sign language into spoken languages, we can distinguish the work [23] in which the authors proposed a solution for translating films with signers through a speaking avatar. This solution is able to generate videos with "talking faces" translating poses from sign language.

Another example is an AI-based novel approach to capturing and representing sign language [27]. A solution to the problem of unavailability of annotated sign language datasets is presented in the form of a crowdsourcing platform [28]. The authors [29] present an innovative approach to automatic sign language synthesis based on advances in the field of machine translation. A system has been proposed that is able to generate sign language videos from spoken language videos. Creating sequences of human poses is based on a combination of neural network-based machine translation (NMT) with motion graphs (MG). Kolejnym rozwiązaniem jest system rozpoznawania ruchu działający w czasie rzeczywistym z wykorzystaniem sygnału elektromiografii zaprezentowany w pracy [70]. Wyniki badań pokazały, że system może z dużą dokładnością rozpoznawać 20 znaczących i szeroko stosowanych ruchów ASL. Another solution is a real-time motion recognition system using the electromyography signal presented in [30]. Test results showed that the system could recognize 20 significant and widely used ASL movements with high accuracy.

B. Two-way communication

Recognition and translation of sign language into spoken languages and vice versa, i.e. two-way communication, is another important area of research related to sign language translation. Two-way communication is essential to ensure smooth interaction between deaf people using sign language and hearing people using spoken language.

The task of the European project [8] is to develop technology for automatic translation of sign languages into spoken languages and vice versa. The solution will be provided in the form of a mobile application for translation and communication. It focuses on several languages, i.e. English, Irish, Dutch and Spanish. In addition, the EXTOL project [25] was developed, which aims to develop the world's first system for translating British Sign Language into English and a functional machine translation system for any sign language. On the other hand, in [31] an automatic system for registering deaf patients was proposed, including a workstation with a computer, an RGB camera and a depth sensor (Kinect). Work related to sign language also applies to translations into Chinese. The proposed solution [26] is a framework-based framework for recognizing and generating Chinese Sign Language based on a recursive neural network. The algorithm has high accuracy in recognizing real and synthetic data, with reduced execution time.

C. Translation of spoken languages into sign languages

An important process among the aforementioned translations is the translation of spoken languages such as text, voice or film into sign languages. This is an important area of research and technological development that aims to facilitate communication between hearing people and deaf people who use sign language. This area includes paper [32] in which an Arabic sign language dictionary was developed using the HamNoSys notation and eSIGN editing software. Also presented is the SIGML sign language, where characters are presented using a 3D avatar, 3000 characters. The Mexican Sign Language avatar presented by the authors [33] was created based on a combination of natural language processing (NLP) techniques with the use of programming engines (Unity) to create animation. Using the structured language model of AZee, Paul's Avatar [34] was created, which is a hybrid system animated mainly with hand-made keyframes, and Kazoo's virtual avatar [35], which generates content from French to sign language.

On the basis of the results available in related literature, an analysis and evaluation of the Portuguese Sign Language (LIBRAS) translation system developed as part of the project was carried out in the context of generating signs and sentences considered ungrammatical by the Deaf community [36]. ProDeaf was developed as computer software for Portuguese Sign Language [4]. Translation takes place both from voice and from text to sign language. However, the proposed application is focused on one-way communication and is not able to translate sign language into text or sound.

On the other hand, for Hindi, a system was proposed that translates English text into Indian Sign Language (ISL) via a 3D avatar [37]. The basic component is an ISL parser that allows parsing sentences based on ISL grammar rules. The system does not use previously saved photos and videos, it displays representations of sentences and words in real time. The Indian Sign Language Dictionary [38], which is bilingual for both English and Hindi, uses the Hamburg notation system and markup language SIGML and the Web Graphics Library (WebGL) to animate 3D avatars. The presented dictionary has 2000 English words and 3286 Hindi words with 110 example sentences.

VI. The character animation area

Major sign language research focuses on the recognition and production of sign languages, as well as the improvement of sign language systems and tools. Research is focused on the development of advanced systems and tools that enable the generation of sign language animations [34], [39]. In this context, researchers are developing software that allows the creation of fluid gestures, facial expressions and body movements characteristic of sign language. For more authentic sign language animations, research is focused on developing motion capture techniques. Researchers are developing techniques that enable realistic facial expressions and emotional expression in sign language animation [40]. Improving the quality of animation and adding facial expressions contributes to better communication and understanding of the information conveyed [41].

A. Systems and tools in sign language animation

One of the proposed tools is Kazoo's virtual avatar [35], which generates content from French to sign language. This project offers the possibility of automatically animating a virtual avatar and content synthesis based on an abstract representation of the author's language model AZee. Paul's English Sign Language avatar was created using the AZee structured language model [34]. Developed at DePaul University by a team of scientists who are working on the avatar. The main goal is to create an avatar that would translate English to ASL in real time.

An interesting solution is a system that allows adding sign language translations in the form of a 3D avatar to digital mathematical educational materials [39]. Operation and construction of ASL System, which consists of 3 basic components: supporting the 3D model, supporting animations, supporting rendering. The SignGAN system [40] on the other hand, is a model for sign language production, a neural network-based translator between text and a synthesized skeletal pose, creating photorealistic sign language videos directly from spoken language. The proposed solution reduces the problems associated with motion blur.

B. Motion Capture Techniques for Sign Language Animation

Advanced motion capture, image processing, and virtualization techniques are often used to create a 3D avatar. An example is the avatar, which acts as a teacher of quite specific concepts in the field of electrical engineering in sign language [42] and the aforementioned Kazoo avatar [35] or Paul's avatar [34].

The key barriers to sign language generation, in particular differences in modality, lack of a standard written form and insufficient resources, are presented in paper [5]. The state of the art and challenges in presenting non-manual signs in avatar animation were also presented from the point of view of three areas, i.e. linguistic approach to facial expressions; Computer Graphics; sign language representation systems [13]. Solving the problem [27] of unavailability of annotated sign language datasets in the form of a crowdsourcing platform has been presented as a novel approach to capturing and representing sign language.

C. Efforts to improve the quality and realism as well as facial expression in sign language animation

Actions to improve the quality and realism of sign language animation are very important in order to better understand the information conveyed. For this reason, a new method [43] based on machine learning was proposed to automatically calculate three key values: selecting the location for inserting pauses, setting the differential speed of individual words, and setting the duration of pauses. In addition, in the years 2006 - 2014, models were worked on that combine language phenomena with specific facial movements in order to generate animations, and an infrastructure for animation synthesis using MPEG-4 facial animation parameters was developed [15].

On the basis of a review of existing lighting models and current progress and research efforts in the field of facial expression and facial expressions, an innovative technique was proposed [11] modifying the classic computer graphics techniques, which, according to the author, is the most efficient combination to present the smallest details. In addition to the automatic generation of complex facial expressions in 3D avatars, a new parametric model of facial expression synthesis using 3D avatars has been proposed [14].

VII. The area of application

The use of translation from the national language into sign language is especially important in situations where communication is necessary to understand and express thoughts, such as business meetings, school activities or conversations with a doctor. Thanks to appropriate translation, deaf people can actively participate in discussions, make decisions and express their views. The development of technologies, such as mobile applications or interactive screens, opens up new possibilities in the field of translation from the national language into the sign language. These innovations facilitate a faster and more precise translation process, thus enabling more effective communication between deaf and hearing people [44].

An important area of application for sign language avatars seems to be all issues related to meeting basic needs and ensuring proper functioning in various spheres of public and social life. Avatars should be used wherever access to information is crucial and it is not possible to employ a professional sign language interpreter. In particular, applications in the field of education, medicine and security and transport can be distinguished here.

A. Education Application

Automatic translation of natural language into sign language has the potential to improve the quality of education for deaf people. Access to translation tools enabling understanding of the content taught at school is invaluable for the intellectual and educational development of this group of people. The Mexican Sign Language Avatar, as a mobile application via a cloud server using NLP and automatic translation, will present limited content from a 4th grade primary Mexican history textbook in sign language [33].

In order to teach mathematics, an e-learning system was developed that was developed for Arab deaf sign language students using an Arabic Sign Language avatar [45], as well as a system for adding sign language translations in the form of a 3D avatar to digital math education materials [39]. A system facilitating independent learning of English Sign Language [46] has also been developed, in which the user has a graphical interface at his/her disposal. This system is based on a neural network that classifies signs flashed in the hand alphabet, recorded with a webcam.

Many efforts have been made in the field of education, e.g. in [47] a Turkish project was described, which presented the benefits of using 3D Avatar in the process of educating deaf children. For the experiment, an avatar was created and a test was performed using it to compare the educational effectiveness of the avatar with text-based learning tools. The results indicated that avatar-based tutoring was more effective in assessing a child's knowledge of certain words in sign language. In the field of education, dictionaries in English [34], Irish [48], Arabic [32] and Indian [37], [38] as well as Portuguese [42] dictionaries may also be considered.

B. Medical Application

The period of the pandemic significantly verified the accessibility of deaf people to medical care. Dutch researchers in [49], [50] studied the potential of automatic translation of text into sign language. Based on consultations with medical professionals, they built a corpus of the most frequently used expressions when diagnosing COVID. SIGML representation and JASignin avatar were used. Attention was drawn to the advantage of avatar over video translation in terms of flexibility and scaling. A definitely lower level of realism and difficulty of understanding was considered a disadvantage.

In a situation where physical well-being is at stake, patients are likely to feel more comfortable watching a human film than an animated avatar [51]. It was also found that users have greater acceptance of virtual advisors operating in general areas, e.g. answering questions in the field of waste management. However, in the case of more personal topics, e.g. parental support, they feel anxiety and distrust when the advisor is not a human but a virtual assistan [52], [53].

C. Security and transportation applications

In the area of ensuring safety, one of the examples of application can be the paper [54], which dealt with the subject of messages about disasters. The authors focus on voice notations that are widely used to transcribe video sequences in sign language. In the first steps, the authors created a corpus for disaster messages in Indian language to be presented by an avatar. In terms of character animation, two methods of Motion Capture and Video Tracing were investigated, for reasons of cost, it was decided to use Video Tracing, which creates 2D avatars. An avatar was created based on the video, the process of animating a 3D avatar based on a 2D video required a lot of effort and a lot of manual tweaking. The final generated corpus contained about 4,000 words on the subject of disasters and about 600 sentences. Several solutions have been identified to ensure smooth avatar movements during translation. However, collecting data on emotional expression and facial expressions remained a challenge during the work. Another example is the avatar of a machine translation system under development, built to translate Swiss Federal Railways' messages in real time [55]. The JASigning software was used to generate the avatar animations.

D. Public administration

Public administration is increasingly showing interest in AI technologies and their implementation. There are numerous publications showing all the research towards the acceptance of modern technologies and the way they are implemented, among others in offices [56], [57]. All actions taken in this direction and their results can be partly related to the implementation of sign language avatar technology. Sign language avatars should primarily be focused on ensuring accessibility to services, but they must also meet all other criteria and be appropriately adapted to the specific area of public administration [58].

For this purpose, a study of trust in virtual advisors in public administration was conducted [17] and the possibilities of using artificial intelligence techniques in public administration were discussed [59]. The methods of using modern technologies in employee training in the context of occupational health and safety were shown [60], and the advantages and potential threats in the use of virtual advisors in public administration were presented [61]. In the research on the introduction of virtual reality to public administration, the criteria that must be met were set, where an example is the appearance of avatars in offices [62], which should have a fairly official character [9], and a framework for public values was developed, which should include chat bots in public administration with examples of achievement [10]. In addition, research was conducted on the impact of implementing modern technologies in public administration in the context of changes in the relationship between employees and their tools, as well as changes in the ways of organizing work in the public sector [63].

The practical use of avatars was also reflected in the example of using avatars in libraries. Paper [64] presents the opportunities and possibilities of using spoken language in the avatar library. For the purpose of simulating crisis situations, the multi-agent multi-user architecture was used, which allows the use of virtual cities as environments for simulations with participants in the form of avatars [65].

E. General use

Works that are not strictly related to the application of research in one area can be indicated here as general application. The proposed solutions are often universal, and small transformations or refinements of parameters to a specific problem indicate their general applications. In the case of general applications, practical projects are most often created, such as applications, systems or other tools, but we also present here theoretical analyzes in the form of literature and systems reviews. An example of general applications may be a web application for generating sign language using the Kazoo virtual avatar [35]. This project is in progress, and the current version offers the ability to automatically animate a virtual avatar and synthesize content based on an abstract representation of the proprietary AZee language model. The same applies to the PE2LGP Animator tool [66], which was created as part of a wider project on the translation of Portuguese Sign Language. The tool allows users with no technical knowledge or animation experience to create LGP character animations for avatars using simple frame-by-frame poses [67].

Another project is a novel machine translation model [68] that translates English sentences into the Pakistani Sign Language equivalent. The system consists of an NLP pipeline and an external video rendering service for translated words based on avatars. In the aforementioned European Project [8], the task was to develop a technology for automatic translation of sign languages into spoken languages and vice versa. The solution will be provided in the form of a mobile application for translation and communication. It focuses on English Irish, Dutch, Spanish language. The Austrian project SIMAX [69] is being implemented as a semi-automatic system for interpreting into sign language. It is one of the most comprehensive systems and consists of several highly advanced ICT technologies. However, The benefits of using synthetic characters from the HamNoSys/SiGML notation instead of working with advanced and expensive motion capture technology were presented on the example of the eSIGN project [70]. It was a development of the ViSiCAST system, which was created as a new project.

VIII. The other works

In some of the analyzed works, it was not possible to indicate one main area of research, therefore we defined a group of other works. Our goal is to show that although much research is focused on one area, sign language research is interdisciplinary. Therefore, in this section we analyze papers providing an overview of solutions to various problems in the field of sign language interpretation.

The authors [36] developed a translation system for Portuguese Sign Language (LIBRAS) and presented research in the context of generating signs and sentences considered ungrammatical by the Deaf community [71]. However, in [67], various methods for finding the meaning of an unknown word in American Sign Language (ASL) were investigated. An overview of currently existing translation systems with their advantages and disadvantages as well as the approach they use is discussed in [4]. A new approach to the construction of sign languages has also been proposed, which significantly increases accuracy in translation. A systematic review of the literature [6] on the synthesis of sign language was carried out, in which dictionaries and repositories of sign languages were discussed in detail, emphasizing the importance of sign notation; translation systems and application areas [72]. Similarly, the authors of the paper [7] reviewed the synthesis of sign language from the perspective of animation, noting the difference between the synthesis of single signs and the generation of complete statements.

A review of existing sign language avatars in the context of details and facial expressions is presented in [11], [41]. An overview of modern techniques for generating facial expressions from the last 15 years was developed, based on 5 examples of avatar use in projects: HamNoSys-based, VComD, DePaul, SignCom, ClustLexical [16]. However, the authors [44] conducted research on the intelligibility of sign language avatars, in terms of methodology, it was proposed to combine a focus group with online research. In addition to determining the key aspects that the deaf community pays attention to in the avatar, it has been shown that the very conduct of research among the deaf community affects their positive perception of avatars [62]. The authors [73] developed a technique for automatically adding realism to animation without the need to manually animate details, and also identified issues related to avatar optimization that can reduce real-time rendering costs. It was also examined to what extent synthetically generated animations are understandable by the deaf community both in the form of skeletal visualizations and generated films [74]. The results show that the deaf community prefers synthetically realistic generated animations to skeletal visualization, it was pointed out that automatic methods of synthesis are not effective enough, the respondents had difficulties in recognizing some signs [75], [76].

Research [3], [12] focused on Irish Sign Language, where the impact of avatar facial expressions on better understanding and acceptance by the deaf community was analyzed and assessed. The reception of avatars' utterances with facial expressions enriched with 7 commonly accepted emotions was compared with the avatars' basic utterances. The results showed that the differences in understanding the content are small. In [27], however, the problem of unavailability of annotated sign language datasets in the form of a crowdsourcing platform was solved.

New challenges for sign language processing have also emerged, based on a discussion of the interdisciplinary nature and multidimensional approach based on Italian Sign Language [77], [78]. An overview of the most modern methods of interception, recognition, translation and representation in sign language, with an indication of their advantages and limitations, was made in [79]. In contrast, the authors [80] analyzed recent advances in the fields of deep learning recognition and production of sign language. The advantages, limitations and future directions for research are discussed, and key barriers to sign language generation such as differences in modality, lack of a standard written form, insufficient resources are presented [5].

IX. Results of literature review analysis

As part of the review of the literature on research on sign language, nearly several hundred scientific papers were reviewed. However, several dozen most closely related to the scope of work were selected for this study. A significant part of the works has been presented in the sections VI, V and VII, where they have been grouped according to the areas (character animation area, technological area and applications), with the exception that some works concern several areas. Such a situation is additionally presented in the figure 3 (together with other works), where all articles are in appropriate groups – it is possible to notice the permeability of works between the analyzed scopes of works.

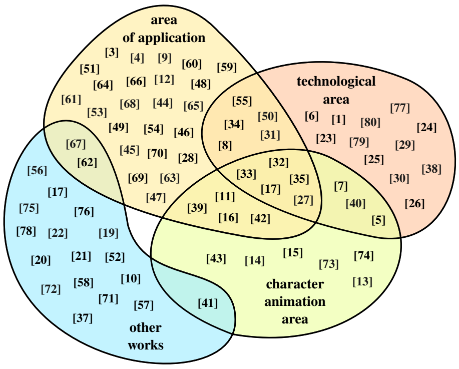


Fig. 3. Relationships between publications from different areas, based on the analyzed literature.

Over the dozens of analyzed works, it can be clearly seen that many of them concern only one area, but there are frequent connections between the application and technology or character animation. This means that there is a need to carry out scientific research combining the indicated areas.

Figure 4 presents boxplots for the number of publications by year and by area. It shows the current research related to the analyzed topic. As can be seen, the first articles were published in the years 2005 – 2008, since then the number of publications on sign language animation has increased significantly. The first quantile in the area related to animation and application is 2015, which is also a period of growth in publications in which there is a combination of sign language with technological possibilities. The median for most of the analyzed publications is 2019-2020. This shows that the analyzed subject matter is currently a very popular issue of research, application and development.

The observations resulting from the analysis of the drawing 3 also allow to justify the division adopted in this review of the current state of knowledge. It is important to thoroughly research both character animation and sign language applications, including - above all - from a technological point of view. The presentation of the multitude of works that have been analyzed requires an appropriate methodological approach. Hence the appropriate, original grouping of all works. On the other hand, the presentation of statistics related to publication dates (see fig. 4) is related to emphasizing the needs related to the discussed topic and its topicality. The needs are already visible on the example of similar works, which are related to e.g. with the analysis of accessibility for people with special needs and appear already in 2005. On the other hand, the topicality is indicated by a clear shift of the median and the third quantile to around 2020-2021 - primarily in the area of technology and character animation.

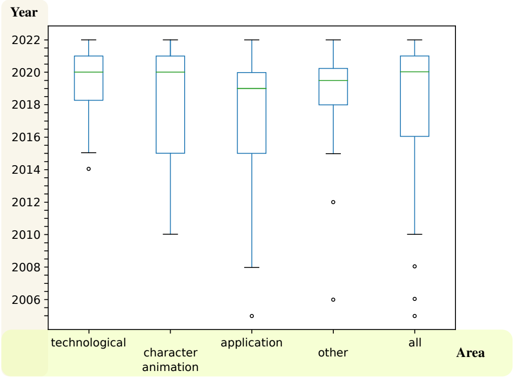


Fig. 4. Boxplots for the publication dates of the analyzed papers related to the subject of the project, in relation to the areas

X. Conclusions

This article focuses on the importance of national language to sign language translation as an important area of research and practice to enable communication between people with hearing impairments and the hearing community. The most important research on sign language interpretation conducted in various fields is reviewed. The article presents the latest scientific and theoretical achievements that contribute to a better understanding of this subject and the improvement of the quality of translation services.

The main aim of the article is to indicate the outstanding areas of interdisciplinary research related to sign language translation and to identify the links between these studies conducted in various fields. The authors try to broaden the knowledge and awareness of sign language interpretation, as well as identify areas that require further research and development.

The conclusions of the article aim to facilitate the improvement of interpretation practices and to promote further research and innovation in the field of sign language translation. This work is important to the hearing impaired community as it provides a better understanding of the communication needs of these people and inspires further efforts to improve the accessibility and effectiveness of sign language interpretation.

In conclusion, the subject of sign language animation is still quite a new and quite complicated field, because it combines many scientific disciplines, from linguistics to machine translation based on neural networks and advanced computer graphics. Today, professionals are collaborating and exploring many areas to develop acceptable and useful solutions to support deaf communities. Despite the great technological progress, the majority of works still discuss numerous challenges and barriers that must be faced in order to ensure full access to information that is so important for deaf people today. Ultimately, improving the quality of sign language interpretation services will contribute to strengthening the social inclusion and equality of people with hearing impairments, ensuring their full access to information and services in all spheres of life.

Acknowledgment

This research was funded by The framework of an automatic translator into the Polish Sign Language using the avatar mechanism, The National Centre for Research and Development, grant number GOSPOSTRATEG-IV/0002/2020.

References

[1] S. K. Liddell, "American sign language syntax," in *American Sign Language Syntax*. De Gruyter Mouton, 2021.

[2] A. Patil, A. Kulkarni, H. Yesane, M. Sadani, and P. Satav, "Literature survey: sign language recognition using gesture recognition and natural language processing," *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2021, Volume 1*, pp. 197–210, 2021.

[3] R. Smith and B. Nolan, "Manual evaluation of synthesised sign language avatars," in *Proceedings of the 15th international ACM SIGACCESS conference on computers and accessibility*, 2013, pp. 1–2.

[4] K. Shah, S. Rathi, R. Shetty, and K. Mistry, "A comprehensive review on text to indian sign language translation systems," *Smart Trends in Computing and Communications: Proceedings of SmartCom 2021*, pp. 505–513, 2022.

[5] R. Wolfe, "Sign language translation and avatar technology," *Machine Translation*, vol. 35, no. 3, pp. 301–304, 2021.

[6] U. Farooq, M. S. M. Rahim, N. Sabir, A. Hussain, and A. Abid, "Advances in machine translation for sign language: approaches, limitations, and challenges," *Neural Computing and Applications*, vol. 33, no. 21, pp. 14 357–14 399, 2021.

[7] L. Naert, C. Larboulette, and S. Gibet, "A survey on the animation of signing avatars: From sign representation to utterance synthesis," *Computers & Graphics*, vol. 92, pp. 76–98, 2020.

[8] H. Saggion, D. Shterionov, G. Labaka, T. Van de Cruys, V. Vandeghinste, and J. Blat, "Signon: Bridging the gap between sign and spoken languages," in *Alkorta J, Gonzalez-Dios I, Atutxa A, Gojenola K, Martínez-Cámara E, Rodrigo A, Martínez P, editors. Proceedings of the Annual Conference of the Spanish Association for Natural Language Processing: Projects and Demonstrations (SEPLN-PD 2021) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2021); 2021 Sep 21-24; Málaga, Spain. Aachen: CEUR Workshop Proceedings; 2021. p. 21-5.* CEUR Workshop Proceedings, 2021.

[9] I. Tozsa, "Virtual reality and public administration," *Transylvanian Review of Administrative Sciences*, vol. 9, no. 38, pp. 202–212, 2013.

[10] T. Makasi, A. Nili, K. Desouza, and M. Tate, "Chatbot-mediated public service delivery: A public service value-based framework," *First Monday*, 2020.

[11] R. Johnson, "Towards enhanced visual clarity of sign language avatars through recreation of fine facial detail," *Machine Translation*, vol. 35, no. 3, pp. 431–445, 2021.

[12] R. G. Smith and B. Nolan, "Emotional facial expressions in synthesised sign language avatars: a manual evaluation," *Universal Access in the Information Society*, vol. 15, pp. 567–576, 2016.

[13] R. Wolfe, J. McDonald, R. Johnson, R. Moncrief, A. Alexander, B. Sturr, S. Klinghoffer, F. Conneely, M. Saenz, and S. Choudhry, "State of the art and future challenges of the portrayal of facial nonmanual signals by signing avatar," in *Universal Access in Human-Computer Interaction. Design Methods and User Experience: 15th International Conference, UAHCI 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part I.* Springer, 2021, pp. 639–655.

[14] D. A. Gonçalves, M. C. C. Baranauskas, J. C. dos Reis, and E. Todt, "Facial expressions animation in sign language based on spatio-temporal centroid." in *ICEIS (2)*, 2020, pp. 463–475.

[15] M. Huenerfauth, "Learning to generate understandable animations of american sign language," 2014.

[16] H. Kacorri, "Tr-2015001: A survey and critique of facial expression synthesis in sign language animation," 2015.

[17] N. Aoki, "An experimental study of public trust in ai chatbots in the public sector," *Government Information Quarterly*, vol. 37, no. 4, p. 101490, 2020.

[18] C. Van Noordt and G. Misuraca, "New wine in old bottles: Chatbots in government: Exploring the transformative impact of chatbots in public service delivery," in *Electronic Participation: 11th IFIP WG 8.5 International Conference, ePart 2019, San Benedetto Del Tronto, Italy, September 2–4, 2019, Proceedings 11.* Springer, 2019, pp. 49–59.

[19] C. Geraci, "Language policy and planning: The case of italian sign language," *Sign Language Studies*, vol. 12, no. 4, pp. 494–518, 2012.

[20] V. Hristidis, "Chatbot technologies and challenges," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 2018, pp. 126–126.

[21] N. K. Kahlon and W. Singh, "Machine translation from text to sign language: a systematic review," *Universal Access in the Information Society*, vol. 22, no. 1, pp. 1–35, 2023.

[22] U. Farooq, M. S. M. Rahim, N. S. Khan, S. Rasheed, and A. Abid, "A crowdsourcing-based framework for the development and validation of machine readable parallel corpus for sign languages," *IEEE Access*, vol. 9, pp. 91 788–91 806, 2021.

[23] S. Mazumder, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, "Translating sign language videos to talking faces," in *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, 2021, pp. 1–10.

[24] N. S. Khan, A. Abid, K. Abid, U. Farooq, M. S. Farooq, and H. Jameel, "Speak pakistan: Challenges in developing pakistan sign language using information technology," *South Asian Studies*, vol. 30, no. 2, 2020.

[25] K. Cormier, N. Fox, B. Woll, A. Zisserman, N. C. Camgöz, and R. Bowden, "Extol: Automatic recognition of british sign language using the bsl corpus," in *Proceedings of 6th Workshop on Sign Language Translation and Avatar Technology (SLTAT) 2019.* Universitat Hamburg, 2019.

[26] Q. Xiao, M. Qin, and Y. Yin, "Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people," *Neural networks*, vol. 125, pp. 41–55, 2020.

[27] K. Stefanidis, D. Konstantinidis, A. Kalvourtzis, K. Dimitropoulos, and P. Daras, "3d technologies and applications in sign language," *Recent advances in 3D imaging, modeling, and reconstruction*, pp. 50–78, 2020.

[28] A. Soudi, K. Van Laerhoven, and E. Bou-Souf, "Africasign–a crowd-sourcing platform for the documentation of stem vocabulary in african sign languages," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 2019, pp. 658–660.

[29] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, "Text2sign: towards sign language production using neural machine translation and generative adversarial networks," *International Journal of Computer Vision*, vol. 128, no. 4, pp. 891–908, 2020.

[30] S. Tateno, H. Liu, and J. Ou, "Development of sign language motion recognition system for hearing-impaired people using electromyography signal," *Sensors*, vol. 20, no. 20, p. 5807, 2020.

[31] D. Szulc, J. Gałka, M. Másior, F. Malawski, T. J. Wilczyński, and K. Wróbel, "Studies on machine processing of sign language in the context of deaf support. application in health care–interactive service system for the deaf."

[32] A. H. Aliwy and A. A. Ahmed, "Development of arabic sign language dictionary using 3d avatar technologies," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 609–616, 2021.

[33] F. Barrera Melchor, J. C. Alcibar Palacios, O. Pichardo-Lagunas, and B. Martinez-Seis, "Speech to mexican sign language for learning with an avatar," in *Advances in Computational Intelligence: 19th Mexican International Conference on Artificial Intelligence, MICAI 2020, Mexico City, Mexico, October 12–17, 2020, Proceedings, Part II 19*. Springer, 2020, pp. 179–192.

[34] M. Filhol and J. C. McDonald, "Extending the azee-paula shortcuts to enable natural proform synthesis," in *sign-lang@ LREC 2018*. European Language Resources Association (ELRA), 2018, pp. 45–52.

[35] A. Braffort, M. Filhol, M. Delorme, L. Bolot, A. Choisier, and C. Verrecchia, "Kazoo: a sign language generation platform based on production rules," *Universal Access in the Information Society*, vol. 15, pp. 541–550, 2016.

[36] L. S. García, T. Felipe, A. Guedes, D. R. Antunes, C. E. Iatskiu, E. Todt, J. Bueno, D. d. F. Trindade, D. Gonçalves, R. Canteri *et al.*, "Deaf inclusion through brazilian sign language: A computational architecture supporting artifacts and interactive applications and tools," in *Universal Access in Human-Computer Interaction. Access to Media, Learning and Assistive Environments: 15th International Conference, UAHCI 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part II*. Springer, 2021, pp. 167–185.

[37] P. Kumar and S. Kaur, "Sign language generation system based on indian sign language grammar," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 19, no. 4, pp. 1–26, 2020.

[38] P. Kumar, S. Kaur *et al.*, "Online multilingual dictionary using hamburg notation for avatar-based indian sign language generation system," *International Journal of Cognitive and Language Sciences*, vol. 12, no. 8, pp. 1117–1124, 2018.

[39] K. Hayward, N. Adamo-Villani, and J. Lestina, "A computer animation system for creating deaf-accessible math and science curriculum materials." in *Eurographics (Education Papers)*, 2010, pp. 1–8.

[40] B. Saunders, N. C. Camgoz, and R. Bowden, "Everybody sign now: Translating spoken language to photo realistic sign language video," *arXiv preprint arXiv:2011.09846*, 2020.

[41] P. A. Angga, W. E. Fachri, A. Elevanita, R. D. Agushinta *et al.*, "Design of chatbot with 3d avatar, voice interface, and facial expression," in *2015 international conference on science in information technology (ICSITech)*. IEEE, 2015, pp. 326–330.

[42] T. Lima, M. S. Rocha, T. A. Santos, A. Benetti, E. Soares, and H. S. de Oliveira, "Innovation in learning–the use of avatar for sign language," in *Human-Computer Interaction. Applications and Services: 15th International Conference, HCI International 2013, Las Vegas, NV, USA, July 21-26, 2013, Proceedings, Part II 15*. Springer, 2013, pp. 428–433.

[43] S. Al-Khazraji, L. Berke, S. Kafle, P. Yeung, and M. Huenerfauth, "Modeling the speed and timing of american sign language to generate realistic animations," in *Proceedings of the 20th international ACM SIGACCESS conference on computers and accessibility*, 2018, pp. 259–270.

[44] M. Kipp, Q. Nguyen, A. Heloir, and S. Matthes, "Assessing the deaf user perspective on sign language avatars," in *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, 2011, pp. 107–114.

[45] S. M. Shohieb, "A gamified e-learning framework for teaching mathematics to arab deaf students: Supporting an acting arabic sign language avatar," *Ubiquitous Learning: An International Journal*, vol. 12, no. 1, pp. 55–70, 2019.

[46] R. Rajendran and S. T. Ramachandran, "Finger spelled signs in sign language recognition using deep convolutional neural network," *International Journal of Research in Engineering, Science and Management*, vol. 4, no. 6, pp. 249–253, 2021.

[47] R. Yorganci, A. A. Kindiroglu, and H. Kose, "Avatar-based sign language training interface for primary school education," in *Workshop: Graphical and Robotic Embodied Agents for Therapeutic Systems*, 2016.

[48] L. C. Galea and A. F. Smeaton, "Recognising irish sign language using electromyography," in *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2019, pp. 1–4.

[49] F. Roelofsen, L. Esselink, S. Mende-Gillings, M. De Meulder, N. Sijm, and A. Smeijers, "Online evaluation of text-to-sign translation by deaf end users: Some methodological recommendations (short paper)," in *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, 2021, pp. 82–87.

[50] F. Roelofsen, L. Esselink, S. Mende-Gillings, and A. Smeijers, "Sign language translation in a healthcare setting," in *Proceedings of the Translation and Interpreting Technology Online Conference*, 2021, pp. 110–124.

[51] P. Bouillon, B. David, I. Strasly, and H. Spechbach, "A speech translation system for medical dialogue in sign language—questionnaire on user perspective of videos and the use of avatar technology," in *3rd Swiss Conference on Barrier-free Communication (BfC 2020)*, 2021, p. 46.

[52] R. De Maria Marchiano, G. Di Sante, G. Piro, C. Carbone, G. Tortora, L. Boldrini, A. Pietragalla, G. Daniele, M. Tredicine, A. Cesario *et al.*, "Translational research in the era of precision medicine: Where we are and where we will go," *Journal of Personalized Medicine*, vol. 11, no. 3, p. 216, 2021.

[53] D. Kruk, D. Mętel, Ł. Gawęda, and A. Cechnicki, "Implementation of virtual reality (vr) in diagnostics and therapy of nonaffective psychoses." *Psychiatria Polska*, vol. 54, no. 5, pp. 951–975, 2020.

[54] P. M. Martin, S. Belhe, S. Mudliar, M. Kulkarni, and S. Sahasrabudhe, "An indian sign language (isl) corpus of the domain disaster message using avatar," in *Proceedings of the third international symposium in sign language translations and technology (SLTAT-2013)*, 2013, pp. 1–4.

[55] S. Ebling and J. Glauert, "Building a swiss german sign language avatar with jasigning and evaluating it among the deaf community," *Universal Access in the Information Society*, vol. 15, pp. 577–587, 2016.

[56] A. Androutsopoulou, N. Karacapilidis, E. Loukis, and Y. Charalabidis, "Transforming the communication between citizens and government through ai-guided chatbots," *Government information quarterly*, vol. 36, no. 2, pp. 358–367, 2019.

[57] Y. Wang, N. Zhang, and X. Zhao, "Understanding the determinants in the different government ai adoption stages: Evidence of local government chatbots in china," *Social Science Computer Review*, vol. 40, no. 2, pp. 534–554, 2022.

[58] A. Lommatzsch, "A next generation chatbot-framework for the public administration," in *Innovations for Community Services: 18th International Conference, I4CS 2018, Žilina, Slovakia, June 18-20, 2018, Proceedings*. Springer, 2018, pp. 127–141.

[59] P. Henman, "Improving public services using artificial intelligence: possibilities, pitfalls, governance," *Asia Pacific Journal of Public Administration*, vol. 42, no. 4, pp. 209–221, 2020.

[60] A. Grabowski, "Wykorzystanie współczesnych technik rzeczywistości wirtualnej i rozszerzonej do szkolenia pracowników," *Bezpieczeństwo pracy: nauka i praktyka*, no. 4, pp. 18–21, 2012.

[61] B. Kopka, "Theoretical aspects of using virtual advisors in public administration."

[62] S. Pauser and U. Wagner, "Judging a book by its cover: Assessing the comprehensibility and perceived appearance of sign language avatars."

[63] T. M. Vogl, C. Seidelin, B. Ganesh, and J. Bright, "Smart technology and the emergence of algorithmic bureaucracy: Artificial intelligence in uk local authorities," *Public Administration Review*, vol. 80, no. 6, pp. 946–961, 2020.

[64] B. Jaskowska, "Nie wiesz? zapytaj awatara: wirtualny doradca w bibliotece," in *Biblioteka-klucz do sukcesu użytkowników (ePublikacje Instytutu Informacji Naukowej i Bibliotekoznawstwa, nr 5)*. Instytut Informacji Naukowej i Bibliotekoznawstwa, Uniwersytet Jagielloński, 2008, pp. 104–110.

[65] H. Nakanishi, S. Koizumi, and T. Ishida, "Virtual cities for real-world crisis management," in *Digital Cities III. Information Technologies for Social Capital: Cross-cultural Perspectives: Third International Digital Cities Workshop, Amsterdam, The Netherlands, September 18-19, 2003. Revised Selected Papers 3*. Springer, 2005, pp. 204–216.

[66] P. Cabral, M. Gonçalves, H. Nicolau, L. Coheur, and R. Santos, "Pe2lgp animator: A tool to animate a portuguese sign language avatar," in *Proceedings of the LREC2020 9th Workshop on the Representation and Processing of Sign Languages: Sign Language Resources in the Service of the Language Community, Technological Challenges and Application Perspectives*, 2020, pp. 33–38.

[67] O. Alonzo, A. Glasser, and M. Huenerfauth, "Effect of automatic sign recognition performance on the usability of video-based search interfaces for sign language dictionaries," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 2019, pp. 56–67.

[68] N. S. Khan, A. Abid, and K. Abid, "A novel natural language processing (nlp)–based machine translation model for english to pakistan sign language translation," *Cognitive Computation*, vol. 12, pp. 748–765, 2020.

[69] G. Tschare, "The sign language avatar project. innovative practice 2016," 2016.

[70] R. San-Segundo, R. Barra, L. D'haro, J. M. Montero, R. Córdoba, and J. Ferreiros, "A spanish speech to sign language translation system for assisting deaf-mute people," in *Ninth International Conference on Spoken Language Processing*, 2006.

[71] A. Pardasani, A. K. Sharma, S. Banerjee, V. Garg, and D. S. Roy, "Enhancing the ability to communicate by synthesizing american sign language using image recognition in a chatbot for differently abled," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2018, pp. 529–532.

[72] A. Kizabekova and V. Chernyshenko, "E-government avatar-based modeling and development," in *Avatar-Based Control, Estimation, Communications, and Development of Neuron Multi-Functional Technology Platforms*. IGI Global, 2020, pp. 19–34.

[73] J. McDonald, R. Wolfe, J. Schnepp, J. Hochgesang, D. G. Jamrozik, M. Stumbo, L. Berke, M. Bialek, and F. Thomas, "An automated technique for real-time production of lifelike animations of american sign language," *Universal Access in the Information Society*, vol. 15, pp. 551–566, 2016.

[74] L. Ventura, A. Duarte, and X. Giró-i Nieto, "Can everybody sign now? exploring sign language video generation from 2d poses," *arXiv preprint arXiv:2012.10941*, 2020.

[75] R. Bartoszcze, Z. Bauer, E. Chudziński, M. DuVall, S. Dziki, B. Fischer, W. Furman, A. Hess, M. Jasionowicz, S. Jędrzejewski *et al.*, *Słownik terminologii medialnej*. Kraków: Towarzystwo Autorów i Wydawców Prac Naukowych Universitas, 2006.

[76] J. J. Bird, A. Ekárt, and D. R. Faria, "Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, pp. 3129–3144, 2023.

[77] S. Fontana and G. Caligiore, "Italian sign language (lis) and natural language processing: an overview." *NL4AI@ AI* IA*, 2021.

[78] D. Bragg, O. Koller, M. Bellard, L. Berke, P. Boudreault, A. Braffort, N. Caselli, M. Huenerfauth, H. Kacorri, T. Verhoef *et al.*, "Sign language recognition, generation, and translation: An interdisciplinary perspective," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 2019, pp. 16–31.

[79] I. Papastratis, C. Chatzikonstantinou, D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Artificial intelligence technologies for sign language," *Sensors*, vol. 21, no. 17, p. 5843, 2021.

[80] R. Rastgoo, K. Kiani, S. Escalera, and M. Sabokrou, "Sign language production: A review," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3451–3461.

DOI: 10.15439/2023F1402
ISSN 2300-5963 ACSIS, Vol. 35

Enhancing naive classifier for positive unlabeled data based on logistic regression approach

Mateusz Płatek
Warsaw University of Technology
Faculty of Mathematics and Information Science
Koszykowa 75, 00-662 Warsaw, Poland
Email: mateusz.platek.poczta@gmail.com

Jan Mielniczuk[0000−0003−2621−2303]
Institute of Computer Science
Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warsaw, Poland
and
Warsaw University of Technology
Faculty of Mathematics and Information Science
Koszykowa 75, 00-662 Warsaw, Poland
Email: jan.mielniczuk@ipipan.waw.pl

***Abstract*—It is argued that for analysis of Positive Unlabeled (PU) data under Selected Completely At Random (SCAR) assumption it is fruitful to view the problem as fitting of misspecified model to the data. Namely, it is shown that the results on misspecified fit imply that in the case when posterior probability of the response is modelled by logistic regression, fitting the logistic regression to the observable PU data which *does not* follow this model, still yields the vector of estimated parameters approximately colinear with the true vector of parameters. This observation together with choosing the intercept of the classifier based on optimisation of analogue of F1 measure yields a classifier which performs on par or better than its competitors on several real data sets considered.**

I. Introduction

In the paper classification problem is analysed for partially observable data scenario for which in the case of some observations class indicators assigned to them (positive or negative in the case of binary classification) are unknown. More specifically, for positive and unlabeled data considered here, it is assumed that some observations from the positive class are labeled, whereas the rest of the observations (either positive or negative) are unlabeled. Such scenario is called Positive Unlabelled (PU) scenario. Thus in the PU setting the true binary class indicator $Y \in \{0,1\}$ is not observed directly but only through binary label S. One knows that if $S = 1$ (labelled case), Y has to be 1 (positive), but for $S = 0$ (unlabeled case) Y may be either 1 or 0 (positive or negative). Besides, each object is described by the vector of features x. This setup encompasses a legion of practical situations, in which effective inference methods about class indicator Y are sought. Examples include disease data (diagnosed patients with a specific disease detected, and patients yet to be diagnosed who may be ill or not), web pages preferences of a specific user (pages bookmarked as of interest and pages not yet viewed, thus of unknown interest) and ecological examples when environments are labeled provided a specific specimen inhabits them, and unlabeled, where this specimen has not been yet looked for). Such scenario is also relevant for survey data, when questions concerning socially reproachable behaviour may not be answered truthfully.

One of the popular approaches to learn from PU data is to impose certain parametric assumptions on distribution of (X, Y) as it commonly done in classical classification task together with some assumptions on labeling mechanism S. This is partly necessitated by the fact that in general situation the posterior distribution of Y as well as prior probability $P(Y = 1)$ is not identifiable. It is thus common to consider logistic type of dependence for the posterior distribution $P(Y = 1|X = x)$ and assume that censoring mechanism acts indiscriminately of x and is described only by the label frequency $c = P(S = 1|Y = 1)$ (SCAR assumption discussed below). Majority of learning approaches has been developed under such assumptions; see [1] for an extensive review of the proposed methods. Recently the JOINT method has been proposed in [11] which consists in minimisation of empirical risk for the observed data $(X_i, S_i), i = 1, \ldots, n$ with respect to parameter of logistic distribution *and* label frequency. JOINT method can be considered as a generic method with specific algorithms depending on optimisation technique used. The issue is delicate as it turns out that the empirical risk is *not* a convex function of its parameters and thus it may posess multiple local minima. In particular [11] used BFGS algorithm, whereas approach in [6] has been based on Minorization-Maximization (MM) technique. Among other methods important group consists of approaches based on weighted empirical risk minimisation in which weights of observations depend on labeling frequency c (see [1], section 5.3.2).

In the present contribution attention is called to the fact that in order to construct a reasonable classifier one can use a logistic model fitted to observable data $(X_i, S_i), i = 1, \ldots, n$ in order to recover the direction of the separating hyperplane and then shift it to the optimal position by maximising observable analogue of $F1$ score. In this approach the direction is obtained by minimising the misspecified *convex* empirical risk (equal to minus log-likelihood) for the observed data. The justification of the method is based on properties of misspecified logistic regression which are valid for PU model under SCAR condition considered here. It is argued that considering fitting parametric models to PU data as the misspecification problem gives new insights to the established properties and leads to new solutions. In particular, results on behaviour of estimators under misspecification (see e.g. [14], [12]) can be used to assess the performance of the naive classifier and its modifications.

II. Notions and auxiliary results

We first introduce basic notations. Let X be a multivariate random variable corresponding to feature vector, $Y \in \{0,1\}$ be a true class label and $S \in \{0,1\}$ an indicator of an example being labeled ($S=1$) or not ($S=0$). We consider X as a column vector and let $X = (1, \tilde{X}^T)^T \in R^{p+1}$, where the first coordinate of X corresponds to an intercept and coordinates of $\tilde{X}$ relate to p collected characteristics of an observation. We assume that there is some unknown distribution $P_{Y,X,S}$ such that (Y_i, X_i, S_i), $i = 1, \ldots, n$ are independent observations drawn from this distribution. Observed data consists of (X_i, S_i), $i = 1, \ldots, n$. This is the single sample scenario as opposed to case-control scenario when the samples from positive class and the general population are given. Only positive examples ($Y=1$) can be labeled, i.e. $P(S=1|X, Y=0) = 0$. Thus we know that $Y=1$ when $S=1$ but when $S=0$, Y can be either 1 or 0. Our primary aim is to construct a classifier which predicts Y class based on PU data. Note that this corresponds to a specific censored data problem as we only observe samples from distribution of (X,S), where $S=Y$ with a certain probability.

To this end we define binary posterior probability of $S=1$ given $X=x$ equal $s(x) = P(S=1|x)$ and propensity score function $e(x) = P(S=1|Y=1, X=x)$. In this paper we adopt Selected Completely At Random (SCAR) assumption which stipulates that $e(x)$ does not depend on x, thus $e(x) = P(S=1|Y=1) := c$, where c will stand for labeling frequency. This means that labeling is not influenced by feature vector x and in this case labeled data is a random sample (of a random size) from a positive class. This commonly adopted assumption is restrictive but it serves as an useful approximation especially in situations when the possibility of labeling bias is recognised and one tries to avoid it. We note that as we have $P(S=1, Y=0|X=x) = 0$ it holds

$$\begin{aligned} s(x) &= P(S=1|x) = P(S=1, Y=1|x) \\ &= P(S=1|Y=1,x)P(Y=1|x) \\ &= e(x) \times y(x) = c \times y(x), \qquad (1) \end{aligned}$$

where we let $y(x) = P(Y=1|X=x)$ denote posterior probability of class 1 and the last equality follows from SCAR assumption. We note that SCAR is equivalent to the property that S and X are conditionally independent given Y. We stress, however, that it is valid only when the label value is assigned with a fixed probability regardless of characteristics of an item. Under this assumption it is easy to see that $P_{X|S=1} = P_{X|Y=1}$ whereas $P_{X|S=0}$ is a mixture

$$P_{X|S=0} = \frac{\alpha - \alpha c}{1 - \alpha c} P_{X|Y=1} + \frac{1-\alpha}{1-\alpha c} P_{X|Y=0}$$

and $\alpha = P(Y=1)$ is a prior probability of $Y=1$. We also note that $c = P(S=1|Y=1) = P(S=1)/P(Y=1) = P(S=1)/\alpha$. We do not assume any previous knowledge of c (although it is frequently imposed see, e.g. [1]) and thus we only know that $0 < c \leq 1$.We will adopt an parametric model for posterior probability $y(x)$ assuming that Y is governed by logistic response:

$$y(x) = \frac{\exp(x^T\beta)}{1 + \exp(x^T\beta)} = \sigma(x^T\beta), \qquad (2)$$

where $\sigma(s) = \exp(s)/(1+\exp(s))$ is a logistic function, β^T stands for transposed column vector β and $\beta = (\beta_0, \beta_{-0}^T)^T \in R \times R^p$ is an unknown but fixed vector value. Thus in view of (1) and (2) we have

$$P(S=1|x) = c \times \sigma(x^T\beta).$$

III. Misspecified logistic modelling

Assume that (2) holds and consider naive approach when the logistic model is fitted to (X,S) data using Maximum Likelihood method i.e. we maximise a log-likelihood

$$\mathcal{L}_n(b) = \sum_{i=1}^n S_i \log\big(\sigma(X_i^T b)\big) + (1 - S_i)\log\big(1 - \sigma(X_i^T b)\big). \qquad (3)$$

Maximisation of $\mathcal{L}_n(\cdot)$ is a concave optimisation problem. Note that this is equivalent to assuming (erroneously) that all unlabeled observations belong to the negative class and thus misspecified logistic model is fitted to the data for which posterior probability is governed by (1). Obviously, one can write down the complete correct log-likelihood for $(X_i, S_i)_{i=1}^n$:

$$\tilde{\mathcal{L}}_n(b,c) = \sum_{i=1}^n S_i \log\big(c\sigma(X_i^T b)\big) + (1 - S_i)\log\big(1 - c\sigma(X_i^T b)\big) \qquad (4)$$

and maximise it wrt (b,c). Such method, named JOINT, has been proposed and investigated in [11]. However, finding global maximum of (4) is hindered by the fact that due to the presence of multiplicative constant c in the form of posterior probability $P(S=1|x)$ given in (1) log-likelihood $\tilde{\mathcal{L}}_n(b,c)$ is no longer concave wrt b, in contrast to $\mathcal{L}_n(b)$. There are some attempts to account for this, either by using Minorization-Maximization algorithm or modelling $\tilde{\mathcal{L}}_n(\cdot, c)$ as the difference of two concave functions ([13]).

Frequently, our aim is not to approximate (β, c) but to construct a classification rule based on training data $(X_i, S_i)_{i=1}^n$. For review of such methods see e.g. [1]. In such a case one can ask whether the classifier based on maximiser of $\mathcal{L}_n(b)$ can not be modified to yield approximation of Bayes classifier of Y. The answer is affirmative and it relies on the crucial observation that $\mathcal{L}_n(b)$ can be viewed as log-likelihood of misspecified logistic regression fitted to data corresponding to posterior probability $q(x^T\beta) = c \times \sigma(x^T\beta)$. This was noticed already in the context of estimation of β in [11] using Ruud's theorem [9] stated below, however its useful consequences have been never explored for PU classification. Here we try to fill this gap by showing that the naive classifier can be improved by adjusting its intercept, the step which has significant influence on its performance. Below we state Ruud's theorem [9] for a logistic loss, for the general statement see [8].

A. Colinearity under misspecification: general case

Assume that the distribution of random vector (X, S) is such that posterior probability $P(S = 1|X = x) = q(x^T\beta)$ for some unknown response function $q : R \to (0, 1)$ which is possibly different from logistic function. Let β^* be the maximiser of expected normalised log-likelihood in (3) for such distribution:

$$\begin{aligned} n^{-1}E_{(X,S)}\mathcal{L}_n(b) &= E_X\{q(x^T\beta)\log\sigma(x^Tb) \\ &+ (1 - q(x^T\beta))\log\big(1 - \sigma(x^Tb)\big)\} \end{aligned} \quad (5)$$

We note that β^* can be interpreted as the minimiser of the averaged Kullback-Leibler (KL) divergence between binary distribution $(q(X^T\beta), 1 - q(X^T\beta))$ and family of logistic models $\{\sigma(X^Tb)\}_{b\in R^{p+1}}$ (see [3] for the definition and properties of KL divergence) and thus corresponds to the Kullback-Leibler projection of the true distribution on this family. It also follows that β^* satisfies the following vector equality

$$EXq(X^T\beta) = EX\sigma(X^T\beta^*). \quad (6)$$

The obvious consequence of (6) is that when $q(s) \equiv \sigma(s)$ and the projection is unique, then $\beta^* = \beta$.
We say that X satisfies Linear Regressions Condition $(LRC(b))$ for vector $b \in R^{p+1}$ if

$$E(\tilde{X}|\tilde{b}^T\tilde{X} = w) = \gamma w + \gamma_0 \quad (7)$$

for some $\gamma = \gamma(\tilde{b}), \gamma_0 = \gamma_0(\tilde{b}) \in R^p$. We note that $LRC(b)$ condition is satisfied for the multivariate normal distribution for any $b \in R^{p+1}$ and, more generally, by the class of eliptically contoured distributions.

Theorem 1: [9] Assume that X satisfies $LRC(\beta)$ condition and moreover covariance matrix of $\Sigma_{\tilde{X}}$ of vector $\tilde{X}$ is strictly positive definite. Additionally, $P(Y = 1|X = x) = q_0(x^T\beta)$ for some unknown function q_0 and for some $\beta \in R^{p+1}$. Then minimiser β^* of (5) satisfies

$$\beta^*_{-0} = \eta\beta_{-0},$$

where $\beta = (\beta_0, \beta^T_{-0})^T$ and $\beta^* = (\beta^*_0, {\beta^*_{-0}}^T)^T$. Moreover, $\eta > 0$ provided that $\mathrm{Cov}(Y, X) > 0$ and $LRC(\beta^*)$ holds.

For the proof of the first part see e.g. [8]. The second part follows from normal equations (6) and the fact that vector γ in (7) equals $(\beta^T_{-0}\Sigma_{\tilde{X}}\beta_{-0})^{-1}\Sigma_{\tilde{X}}\beta_{-0}$.

Theorem above implies that under the stated conditions despite the misspecification of the fitted model we still retain colinearity of true parameter β and the vector β^* of its Kullback-Leibler projection when the first coordinate in both vectors corresponding to intercept is omitted. This has an obvious relevance in classification if one recalls that Bayes classifier when logistic model is valid equals under conditions of Theorem 1:

$$\begin{aligned} &\hat{Y}(X) = I\{(\tilde{X}^T\beta_{-0} + \beta_0 > 0\} \\ = \ &I\{\eta\tilde{X}^T\beta_{-0} + \eta\beta_0 > 0\} = I\{\tilde{X}^T\beta^*_{-0} + \eta\beta_0 > 0\} \end{aligned} \quad (8)$$

Thus the direction of the optimal separating hyperplane $\tilde{X}^T\beta_{-0} + \beta_0 = 0$ is given by projection β^*_{-0} which is easily estimable and only the intercept $\eta\beta_0$ needs to be recovered. Let $\hat{\beta}^*$ denote maximiser of (3). As Maximum Likelihood estimator $\hat{\beta}^*$ consistently estimates β^* under mild conditions (see [14]) one can use $\hat{\beta}^*_{-0}$ as the vector defining the direction of the separating hyperplane $w^Tx + w_0$ and then adjust its intercept appropriately.

B. Collinearity under misspecification: PU case

Consider now Positive Unlabeled data case and assume that posterior probability of Y given X is given by logistic model defined in (2). Then in the view of (1) when logistic model is fitted to (S, X), the model is misspecified as $P(S = 1|X = x) = c \times \sigma(^T\beta)$. However, under conditions of Theorem 1 we have $\beta^*_{-0} = \eta\beta_{-0}$ and moreover (6) yields

$$cEX\sigma(X^T\beta) = EX\sigma(\beta^*_0 + \eta\tilde{X}^T\beta_{-0}).$$

This shows how parameter η depends on labeling frequency c and distribution of $X^T\beta$. When X is multivariate normal this can be restated more explicitly.

Theorem 2: Assume that $X \sim N(0, \Sigma)$ and conditions of Theorem 1 are satisfied. (i) Then we have for any $j = 1, \ldots, p$:

$$\frac{\eta}{c} = \eta\frac{EX_j\sigma(\beta_0 + \tilde{X}^T\beta_{-0})}{EX_j\sigma(\beta^*_0 + \eta\tilde{X}^T\beta_{-0})} = \frac{E\sigma'(\beta_0 + \tilde{X}^T\beta_{-0})}{E\sigma'(\beta^*_0 + \eta\tilde{X}^T\beta_{-0})} \quad (9)$$

(ii) If $c \leq 1/2$ then $\beta^*_0 < 0$ for any β_0.

Proof. The first equality in (9) is just a consequence of (6) when j^{th} coordinate is considered. The second equality follows from Stein's lemma, which states that $\mathrm{Cov}(h(Z_1), Z_2) = Eh'(Z_1)\mathrm{Cov}(Z_1, Z_2)$ for bivariate normal vector (Z_1, Z_2). It implies that

$$\begin{aligned} &EX_j\sigma(\beta_0 + \tilde{X}^T\beta_{-0}) = \mathrm{Cov}(X_j, \sigma(\beta_0 + \tilde{X}^T\beta_{-0})) \\ = \ &E\sigma'(\beta_0 + \tilde{X}^T\beta_{-0})\mathrm{Cov}(X_j, \beta_0 + \tilde{X}^T\beta_{-0}) \end{aligned} \quad (10)$$

and, analogously

$$\begin{aligned} &EX_j\sigma(\beta^*_0 + \eta\tilde{X}^T\beta^*_{-0}) = \mathrm{Cov}(X_j, \sigma(\beta^*_0 + \tilde{X}^T\beta^*_{-0})) \\ = \ &E\sigma'(\beta^*_0 + \eta\tilde{X}^T\beta_{-0})\mathrm{Cov}(X_j, \beta^*_0 + \eta\tilde{X}^T\beta^*_{-0}). \end{aligned} \quad (11)$$

Applying normal equations again we obtain the second equality.
In order to prove (ii) note that for any symmetric univariate random variable Z we have

$$E\sigma(a + Z) < 1/2 \iff a < 0.$$

Indeed

$$E\sigma(a + Z) = 1 - E\sigma(-a - Z) = 1 - E\sigma(-a + Z),$$

where the second equation is due to symmetry of Z. This, and the fact that $\sigma(a+Z) < \sigma(-a+Z)$ is equivalent (due to monotonicity of $\sigma(\cdot)$) to $a < 0$ justify the claim. However, note that normal equations for the first coordinate being 1 imply that

$$E\sigma(\beta_0^* + \eta\tilde{X}^T\beta_{-0}) = cE\sigma(\beta_0 + \tilde{X}^T\beta_{-0}) < c \leq \frac{1}{2}$$

and thus $\beta_0^* < 0$.

Remark 3.1: Part (ii) explains why the naive classifier applied to (S, X) data will work poorly, especially for small c: its intercept is likely to be negative regardless the sign of the intercept $\eta\beta_0$ in (8). Thus it has to be modified to enhance the performance of naive classifier.

Remark 3.2: The case when no intercept is included in both the true and the fitted model has been considered in [11]. It is shown there that then $0 < \eta \leq c < 1$. Thus in this case coefficients of logistic model corresponding to genuine predictors are shrunk towards 0.

C. Choice of the intercept

We propose to choose the intercept $\widehat{w}_0$ of the separating hyperplane $\tilde{x}^T\hat{\beta}^*_{-0} + \widehat{w}_0 = 0$, where $\widehat{w}_0$ is an estimator of $\eta\beta_0$ (see (8)), by maximising the analogue of $F1$ measure on training data. We let, for a given classifier $\hat{Y} = \hat{Y}(X)$ learnt on the training data $\mathcal{D}^{train}$:

$$r = P(\hat{Y}(X) = 1|Y = 1) \qquad p = P(Y = 1|\hat{Y}(X) = 1)$$

be population recall and precision of $\hat{Y}$, respectively. Here, (X, Y) stands for unobservable random variable having distribution $P_{X,Y}$ which is independent of $\mathcal{D}^{train}$. We define $F1$ measure as their harmonic mean

$$F1 = \frac{r \times p}{(r + p)/2}. \tag{12}$$

Thus in order to have large $F1$ value, both the precision and recall should be large. We also note that simple derivation yields $F1 = 2 \times P(Y = 1, \hat{Y} = 1)/(P(Y = 1) + P(\hat{Y} = 1))$. Moreover, note that for PU data under SCAR we have that $P(\hat{Y}(X) = 1|Y = 1, \mathcal{D}^{train}) = P(\hat{Y}(X) = 1|S = 1, \mathcal{D}^{train})$ as $\hat{Y}(X)$ given $\mathcal{D}^{train}$ depends on X only and $P(X|Y = 1) = P(X|S = 1)$.

This means that the recall r can be easily estimated from (X, S) sample. The precision, however is unobservable, and thus we consider the following analogue of $F1$ introduced in [7], Section 4, (see also [10]) defined as

$$F1_{PU} = \frac{r \times p}{P(Y = 1)}. \tag{13}$$

$F1_{PU}$ is proportional to squared geometric mean of the precision and the recall i.e. Fowlkes-Mallows index [5]. Note that one obtains

$$\frac{P(Y = 1|\hat{Y}(X) = 1)}{P(Y = 1)} = \frac{P(\hat{Y}(X) = 1|Y = 1)}{P(\hat{Y}(X) = 1)}$$

which in terms of the precision and the recall means that $p = r \times P(Y = 1)/P(\hat{Y}(X) = 1)$ and thus

$$F1_{PU} = \frac{r^2}{P(\hat{Y}(X) = 1)}. \tag{14}$$

Let $\hat{Y}_z(x) = I\{\tilde{x}^T\hat{\beta}^*_{-0} + z > 0\}$, where $\hat{\beta}^*$ is maximiser of (3) and define $\widehat{F1}_{PU}(z)$ to be a sample analogue of $F1_{PU}$ for the classifier $\hat{Y}_z(X)$. We propose to choose $\widehat{w}_0$ as maximiser of

$$\widehat{w}_0 = \text{argmax}_z \widehat{F1}_{PU}(z) \tag{15}$$

We will call the classifier $\hat{Y}(x) = I\{\tilde{x}^T\hat{\beta}^*_{-0} + \widehat{w}_0 > 0\}$ the enhanced naive classifier. The pseudo-code for enhanced classifier is given in Algorithm 1. We show below when analysing its behaviour on real data sets that modification of the intercept of the naive classifier is crucial for its performance.

Algorithm 1 Enhanced naive classifier

```
Input: Observed data $(x_i, s_i)$, $i = 1, \ldots, n$.
Step 1: Obtain estimator $\hat{\beta}^* = (\hat{\beta}^*_0, \hat{\beta}^*_{-0})$ by fitting logistic regression to observed data $(x_i, s_i)$.
Step 2: Calculate intercept $\hat{w}_0$ as $\text{argmax}_z \widehat{F1}_{PU}(z)$.
Result: Parameters $(\hat{w}_0, \hat{\beta}^*_{-0})$ of the separating hyperplane.
```

IV. Numerical experiments

In the numerical experiments we have considered the following classifiers:

- Naive classifier based on fitting logistic regression model to (X, S) data called Naive and the classifier Enhanced proposed here;
- Classifiers based on JOINT and MM estimators discussed above;
- Weighted classifiers introduced in [1], Section 5.3.1 using two alternative estimators of c: proposed in [4] (denoted by e_1, p.214) and TIcE estimator introduced in [2]. For the discussion of both estimators of c see e.g. [6]. They will be called EN and TIcE classifiers, respectively.

The implementation of Enhanced estimator is given in github directory[1]. Maximisation of $\widehat{F1}_{PU}(z)$ in (15) is achieved by looking for maximal value among the values of this quantity, noting that numerators of numerator and denominator of the ratio defining it may change by ±1 when moving along ordered values of intercept for which predictions of considered classifiers change, i.e. values $z_i = \tilde{x}_i^T\hat{\beta}^*_{-0}$.

A. Synthetic data

In order to check how Ruud's theorem works in practice and the performance of the proposed classifier, we considered a simple synthetic example where vector of predictors $\tilde{X}$ has three-dimensional normal distribution with mean $m = (1, 1, -1)^T$, variances equal to 1 and covariances $Cov(X_1, X_2) = 0.2$, $Cov(X_1, X_3) = -0.2$ and

[1] https://github.com/MateuszPlatek/PU_Enhanced_Naive_Classifier

$Cov(X_2, X_3) = 0$. Thus X_1 is positively correlated with X_2 and negatively correlated with X_3. Moreover posterior probability of $Y = 1$ given $X = x$ is logistic with $\beta = (-1, -1, 1, 1)^T$. We investigated the angle between $\hat{\beta}_{-0}$ and β_{-0} for all considered estimators, the performance of corresponding classifiers for $c = 0.3, 0.6$ and several values of n ranging from 500 to 5000. The results are shown in Figure 1 situated at the end of the paper. The first row of the panel exhibits goodness of fit of the considered estimators measured by the mean differences of their angles and the angle of β_{-0}. It indicates that in concordance with Ruud's theorem the direction of β_{-0} is approximately recovered by direction of naive estimator $\hat{\beta}_{-0}$ for sample sizes larger than 1000 and the accuracy increases with increasing sample size. Moreover the accuracy of $\hat{\beta}_{-0}$ measured by mean difference of angles for naive, MM and JOINT estimators approximately coincides and is consistently better than that of EN and TIcE estimators. In terms of F1 measure shown in the second row the introduced enhanced naive classifier works consistently better than its competitors and in terms of Balanced Accuracy (defined as the average of the recall and the specificity; the third row) it is only outperformed by EN classifier for $c = 0.3$.

B. Real datasets

We have analysed performance of the estimators on six data sets from UCI directory with sample sizes ranging from around 300 to 30 000 and number of features from 3 to 166 (the main characteristics of the data sets are given in Table I). The figures show mean performance with the regard of F1 measure (Figure 2) and Balanced Accuracy (Figure 3), for values of c ranging from 0.1 to 0.9, based on 200 random splits of the data into training and testing subsamples. Standard errors for the mean are smaller than 0.01 in most cases for both F1 and BA measure with the only exception of F1 measure on `credit-a` and `diabetes` data set and the maximal value of SE is 0.026 for JOINT estimator on `credit-a`. Note that the results for the naive classifier are truncated from below in Figure 2: F1 measure for naive classifier is very low for $c \leq 0.5$ and approach 0 for c close to 0. The first immediate observation is that the change of the intercept estimator, which is the only difference between the naive classifier and its enhanced version, has a huge impact on its performance with regard to both considered measures.

F1 measure In all cases but one the enhanced classifier works better (data sets `musk, credit-a, diabetes, adult`) or on par (`banknote`) with JOINT and MM estimators. In the case of *spam* it works marginally worse than JOINT and MM. This is interesting, especially in comparison with MM estimator which requires much more computing effort. It also outperforms TIcE and EN estimators on three data sets: `banknote, musk` and `spam`. On `adult` data set enhanced classifier works better than EN and on par with TIcE. Its excellent performance on `musk` data set is worth pointing out. The performance of enhanced estimator deteriorates for small values of c, possibly due to loss of accuracy of $\widehat{F1}_{PU}$ (note that the denominator of (14) becomes smaller for smaller c).

Name	Size	Features	Fraction of positive observations
adult	32561	57	0.24
banknote	1372	4	0.44
credit-a	690	38	0.44
diabetes	768	8	0.35
musk	6598	166	0.15
spambase	4601	57	0.39

TABLE I: Analysed datasets and their statistics

Algorithm	Oracle	Enhanced	JOINT	MM	EN	TIcE
Time	0.05s	0.22s	0.23s	201s	0.66s	0.9s

TABLE II: Mean training time in seconds on the largest dataset adult with $c = 0.5$.

Balanced Accuracy The performance of enhanced estimator with respect of Balanced Accuracy is similar to that with respect to F1 measure.

We have also analysed training times of the considered classifiers. Table II shows the training times for the largest data set `adult`. In the case of Enhanced and JOINT classifiers the times are approximately the same and 2-3 times shorter that the times for EN and TiCE classifiers. The most computation intensive is MM classifier as it requires inner loop of convex optimisation for each iteration of $\hat{\beta}$.

V. Conclusion

We have studied a novel modification of naive classifier for Positive Unlabeled data under SCAR assumption. The classifier has strong theoretical underpinnings following from Ruud's theorem which are are established in Theorem 1. These indicate that the coefficients of logistic classifier corresponding to genuine predictors are consistently estimated based on observed (X, S) data and the estimation problem boils down to consistent estimator of the intercept. We have proposed such an estimator based on maximisation of observable analogue of $F1$ measure. Moreover, we have shown analysing real data sets that the resulting enhanced naive estimator is a promising alternative to classifiers based on parametric models of posterior probability (JOINT and MM classifier) as well as nonparametric ones (TIcE and EN classifiers). Future research may include finding alternatives to the proposed method of estimating the intercept as well as extension of the considered method to the situation when SCAR assumption is violated. In particular, note that when posterior probability $y(x)$ satisfies (2) and $e(x)$ is an *arbitrary* function of $y(x)$, posterior probability $s(x)$ of $S = 1$ given $X = x$ is a function of $x^T\beta$ and it corresponds to misspecified logistic model. Thus the conclusion of Theorem 1 applies also to this more general situation which as its special case includes probabilistic gap assumption when $e(x)$ is an increasing function of $y(x)$.

REFERENCES

[1] J. Bekker and J. Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4):719–760, April 2020. http://dx.doi.org/10.1007/S10994-020-05877-5.

[2] Jessa Bekker and Jesse Davis. Estimating the class prior in positive and unlabeled data through decision tree induction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):2712–2719, April 2018. https://doi.org/10.1609/aaai.v32i1.11715.

[3] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, NY, 1991. http://dx.doi.org/10.1002/047174882X.

[4] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 213–220, August 2008. http://dx.doi.org/10.1145/1401890.1401920.

[5] E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of American Statistical Association*, 78:573–586, 1981. https://doi.org/10.2307/2288117.

[6] M. Łazecka, J. Mielniczuk, and P. Teisseyre. Estimating the class prior for positive and unlabelled data via logistic regression. *Advances in Data Analysis and Classification*, 15(4):1039–1068, June 2021. http://dx.doi.org/10.1007/S11634-021-00444-9.

[7] W. Lee and B. Liu. Learning with positive and unlabeled exampled using weighted logistic regression. In *Proceedings of the Twentieth International Conference on Machine Learning*, ICML '03, pages 448–455, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[8] K-C. Li and N. Duan. Regression analysis under link violation. *The Annals of Statistics*, 17(3):1009–1052, 1989. http://dx.doi.org/10.1214/aos/1176347254.

[9] P. Ruud. Sufficient conditions for the consistency of maximum likelihood estimation despite misspecification of distribution in multinomial discrete choice models. *Econometrica*, 51:225–228, 1983. http://dx.doi.org/10.2307/1912257.

[10] S. Tabatabaei, J. Klein, and M Hoogendoorn. Estimating the F1 score for learning from positive and unlabeled examples. In *LOD 2020*. Springer, Cham, 2020. https://doi.org/10.1007/978-3-030-64583-0_15.

[11] P. Teisseyre, J. Mielniczuk, and M Łazecka. Different strategies of fitting logistic regression for positive and unlabeled data. In *Proceedings of the International Conference on Computational Science ICCS'20*, pages 3–17, Cham, 2020. Springer International Publishing. https://doi.org/10.1007/978-3-030-50423-6_1.

[12] Q. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57:307–333, 1989. https://doi.org/10.2307/1912557.

[13] A. Wawrzenczyk and J. Mielniczuk. Strategies for fitting logistic regression for positive and unlabeled data revisited. *Int.J. Appl. Math. Comp. Sci.*, pages 299–309, 2022. https://doi.org/10.34768/amcs-2022-0022.

[14] H. White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–25, 1982. https://doi.org/10.2307/1912526.

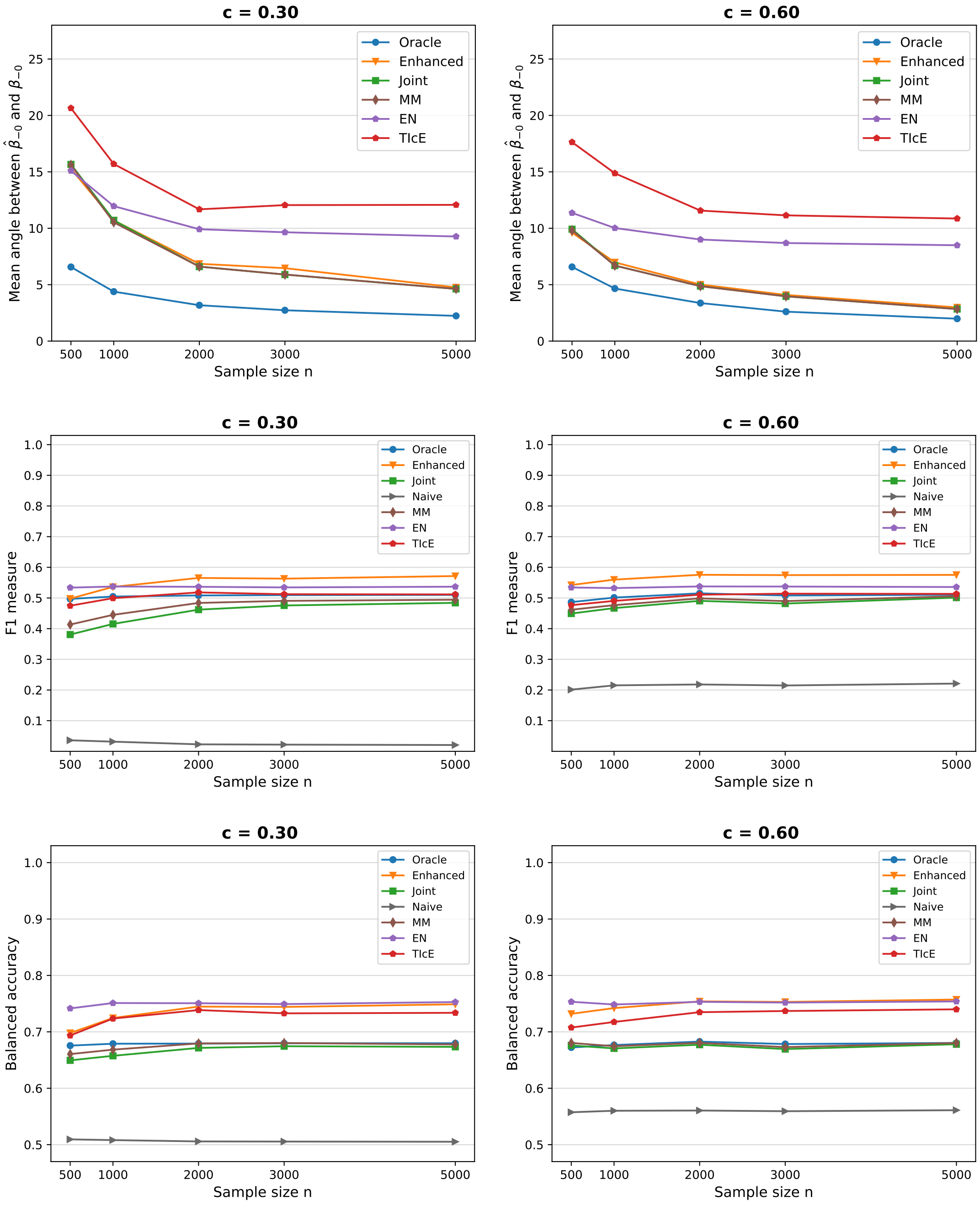


Fig. 1: Mean difference of angles, F1 and Balanced Accuracy against sample size for artificial data.

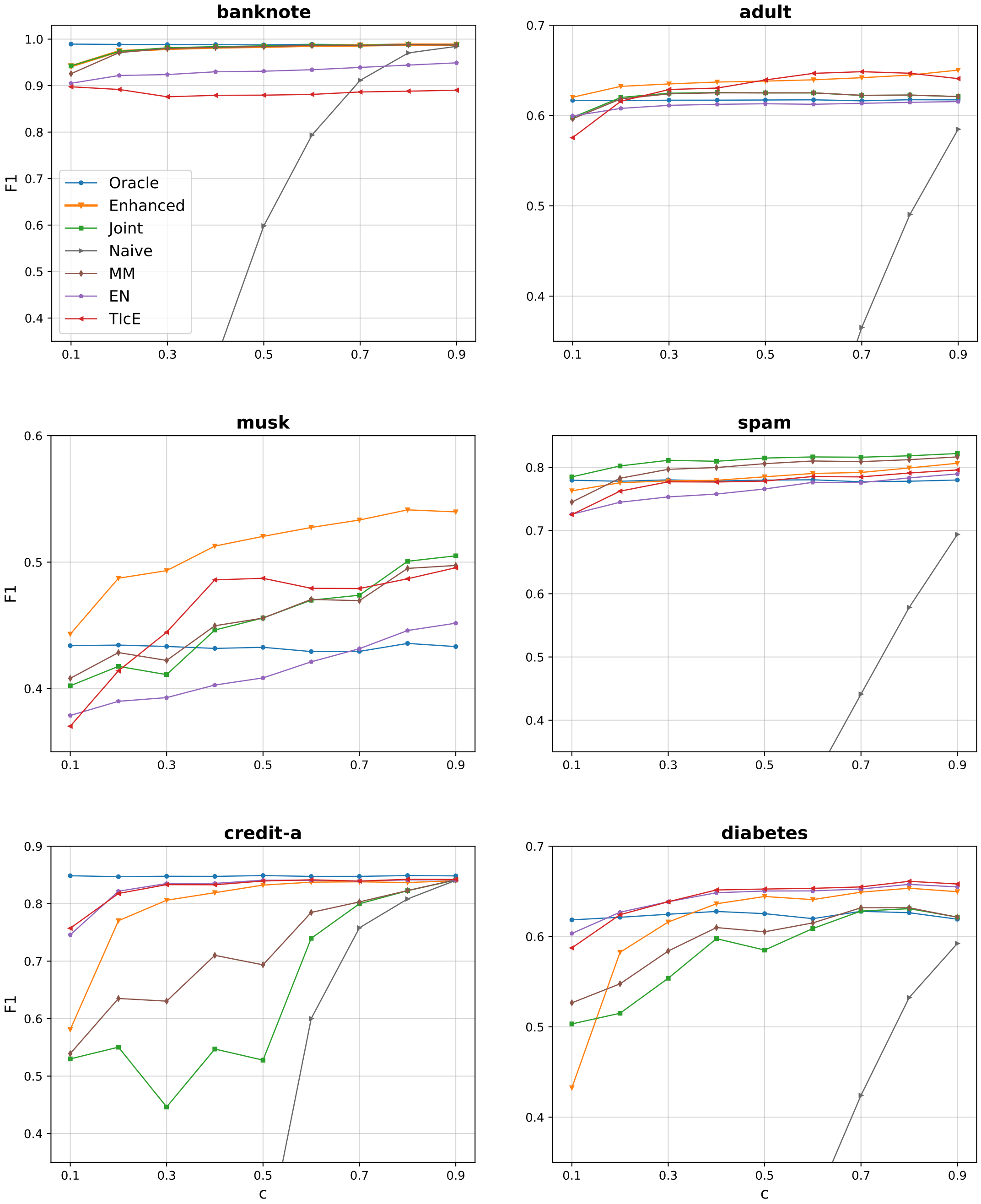


Fig. 2: F1 measure against values of c for the considered data sets.

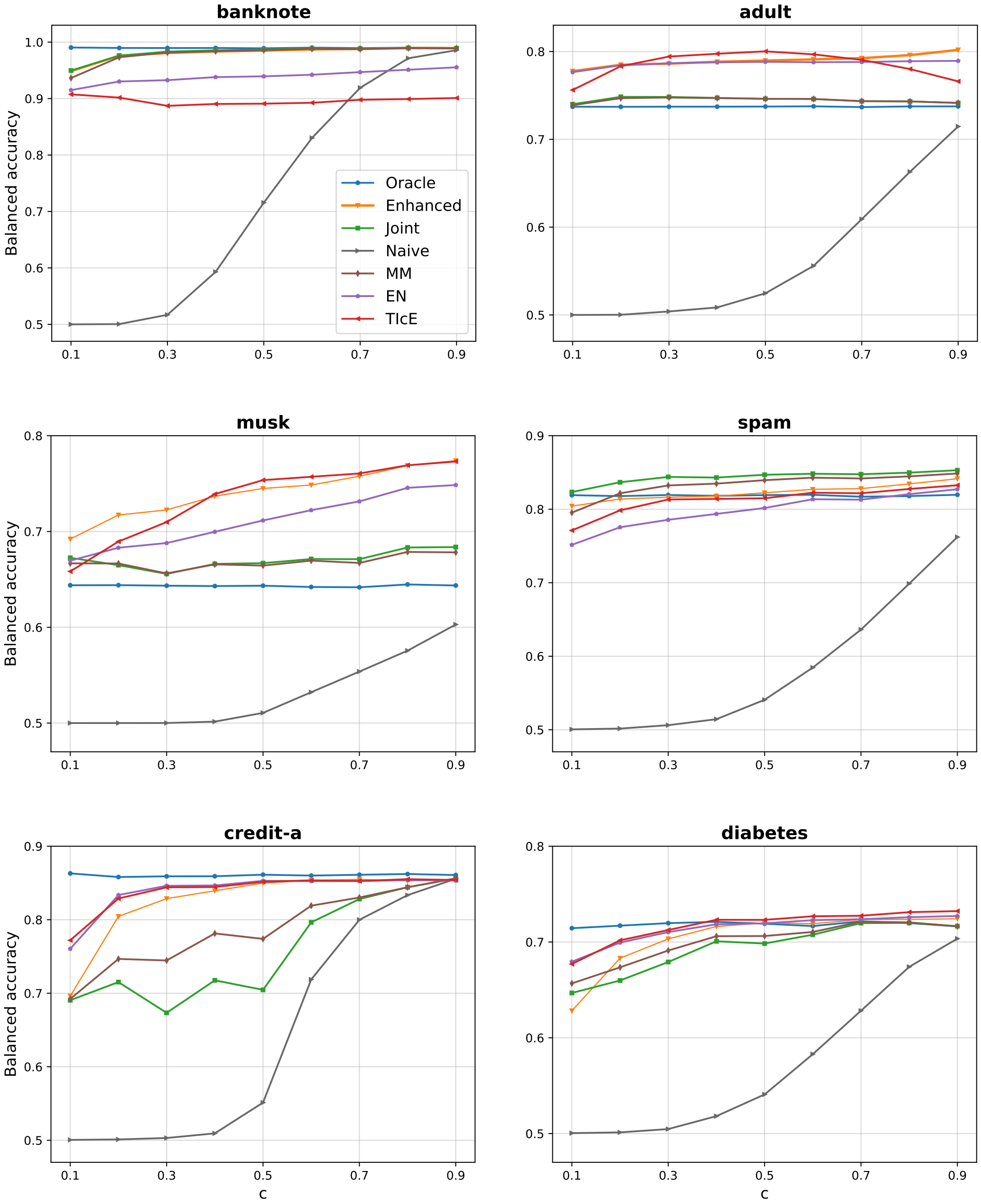


Fig. 3: Balanced Accuracy against values of c for the considered data sets.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 235–246
DOI: 10.15439/2023F708
ISSN 2300-5963 ACSIS, Vol. 35

Improving the Efficiency of Meta AutoML via Rule-based Training Strategies

Alexander Zender, Bernhard G. Humm, Tim Pachmann
0000-0002-6956-9049, 0000-0001-7805-1981, 0000-0003-4393-683X
Darmstadt University of Applied Sciences
Schöfferstr. 3, 64295 Darmstadt, Germany
Email: alexander.zender@h-da.de, bernhard.humm@h-da.de, tim.pachmann@icloud.com

***Abstract*—Meta Automated Machine Learning (Meta AutoML) platforms support data scientists and domain experts by automating the ML model search. A Meta AutoML platform utilizes multiple AutoML solutions searching in parallel for their best ML model. Using multiple AutoML solutions requires a substantial amount of energy. While AutoML solutions utilize different training strategies to optimize their energy efficiency and ML model effectiveness, no research has yet addressed optimizing the Meta AutoML process. This paper presents a survey of 14 AutoML training strategies that can be applied to Meta AutoML. The survey categorizes these strategies by their broader goal, their advantage and Meta AutoML adaptability. This paper also introduces the concept of rule-based training strategies and a proof-of-concept implementation in the Meta AutoML platform OMA-ML. This concept is based on the blackboard architecture and uses a rule-based reasoner system to apply training strategies. Applying the training strategy “top-3” can save up to 70% of energy, while maintaining a similar ML model performance.**

I. Introduction

MACHINE LEARNING (ML) is an important subdomain of artificial intelligence (AI), allowing programs to make predictions using models based on previous observations [1]. Creating effective ML models requires substantial knowledge and experience in the field of ML. Data scientists are a group of experts possessing those foundations. Generating an ML model involves several tasks, including data analysis, data preparation, feature engineering, model selection, validation, learning curve analysis and hyperparameter optimization.

The research field of *Automated Machine Learning (AutoML)* emerged to support data scientists and domain experts (professionals in a domain like medicine) with these tasks. AutoML aims to automate the model selection and hyperparameter optimization process leading to higher efficiency, and potentially better results [2]. Finding the best model and its hyperparameter optimization for a given problem is also known as the Combine Algorithm Selection and Hyperparameter Optimization (CASH) problem [3]. An AutoML solution programmatically searches for an ML pipeline by solving the CASH problem [4]. More progressive AutoML solutions also perform data preparation, feature engineering, and validation, allowing for the creation of entire ML pipelines [5]. There are a growing number of AutoML solutions available [6] offering automated solutions for ML tasks belonging to supervised learning (e.g. Auto-WEKA [7]) and unsupervised learning (e.g. AutoCluster [8]). Although AutoML aims to be accessible to users with and without ML and programming expertise, only a few AutoML solutions are targeted at domain experts [6]. Furthermore, due to the wide range of AutoML solutions and their constraint to mainly support only one major ML library (e.g. Auto-Keras [9] supporting Keras[1]), finding the most effective ML method for a given use case requires a trial and error approach.

Meta Automated Machine Learning (Meta AutoML) [10] is a concept that addresses such issues by integrating multiple Automated Machine Learning solutions into one ensemble. During a training session, the AutoML solutions search for their best ML model in parallel. Some AutoML solutions may use different approaches to automatically optimize the model search [3]. This internal optimization aims to reduce the training time and increase the effectiveness of the final ML model. Within the research field of AutoML, a wide range of different optimization approaches exist [11] e.g. meta-learning [12], or early-stopping [13].

Running multiple AutoML solutions in parallel is energy-inefficient. Therefore, improving the energy-efficiency of the Meta AutoML process is an important goal. AI approaches that use vast amounts of computation power to increase their performance can be labelled as red AI [14]. Meta AutoML falls into this red AI category, as it uses a massive amount of computation power to operate multiple AutoML solutions in parallel. It is important to optimize the Meta AutoML process and move it towards green AI [14]. However, on the Meta AutoML level, there has been no research on improving efficiency.

A simple Meta AutoML training strategy to improve energy efficiency is the “top-3” strategy. This strategy performs two successive training sessions. The first training session uses all AutoML solutions with a reduced sample set and training time. The second training session utilizes the full sample set and the remaining training time. This session uses the 3 best performing AutoML solutions found during the first training session. We use the top-3 strategy to demonstrate and evaluate the approach presented in this paper.

[1] https://keras.io

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

Topical area: Advanced Artificial Intelligence in Applications

The contributions of this paper are two-fold: (a) A survey of AutoML training strategies that can be applied to Meta AutoML; (b) A concept for a rule-based training strategy component for Meta AutoML and its prototypical implementation in the platform OMA-ML [6] as a proof-of-concept. This paper uses the prototypical implementation to evaluate the effectiveness of a selected training strategy, namely the top-3 strategy. The evaluation compares CO_2 emission equivalents produced by the entire Meta AutoML training and the ML model prediction performance with and without using the training strategy.

This paper is structured as follows. Section II presents related work. Section III shows the survey on training strategies. Section IV is the paper's core, introducing the concept of rule-based training strategies. Section V briefly indicates aspects of the prototypical implementation in the platform OMA-ML. Section VI evaluates the concept and prototypical implementation. Section VII concludes the paper and discusses future work.

II. Related Work

A number of surveys exist that focus on state-of-the-art research about AutoML [6][3][15][16][17][18][19]. These surveys focus on the algorithms and approaches used to solve the CASH problem. They also review individual AutoML solutions. However, only one survey [6] offers a broader survey of existing AutoML solutions. Furthermore, we are not aware of a survey on AutoML training strategies.

The next important aspect of research relates to AutoML training strategies. There are two different categories of AutoML strategies: strategies that are applied during preprocessing and strategies that are applied during training of an ML model. Some AutoML solutions preprocess the dataset only once; e.g., Autogluon infers the process from the initial dataset [20]. Others may use intricate data preprocessing approaches; e.g., TPOT uses genetic programming [21] to find the best preprocessing workflow. Additionally, some AutoML solutions apply strategies to the ML model training process, e.g. early stopping (e.g. H2O: AutoML [22]) which stops the fitting process of a ML model based on a user-defined termination criterion [13]. Another strategy is multi-fidelity (e.g. Auto-Pytorch [23]). This aims to optimize the ML model and hyperparameter search by training models using lower-fidelity (e.g. less time, computation, data) to determine the best configuration to run high-fidelity training [24]. AutoML solutions can implement frameworks like BOHB to use multi-fidelity optimization [25]. Finally, some AutoML solutions implement strategies such as meta-learning [26] to further improve their training by learning from previous ones. While early stopping and multi-fidelity training strategies are used by AutoML solutions, they are not limited to AutoML. Data scientists use the concept of applying optimization strategies to improve their manual search for the best ML model.

There exists a collection of foundation literature introducing different strategies to improve the ML training [27][28]. The literature provides different optimization strategies depending on the dataset and general ML training. There are scientific surveys that compare different approaches used to optimize the individual steps of preprocessing [29][30]. Additionally, there are publications that present new approaches to optimize or replace existing preprocessing approaches [31][32].

The concept of improving the efficiency of Meta AutoML via rule-based training strategies is novel. We are not aware of any publication dealing with this issue. Currently, we are aware of two Meta AutoML platforms: OMA-ML [6] and Ensemble Squared [33]. Both Meta AutoML platforms use a meta layer to administer the built-in AutoML solutions. This abstraction layer allows a user to leverage multiple AutoML solutions simultaneously, without requiring previous knowledge about individual AutoML solutions or data science.

Ensemble Square does not use training strategies to optimize its Meta AutoML process. It uses all supported AutoML solutions for every training session by default.

III. A Survey of ML Training Strategies

In this section we compare 14 training strategies from AutoML solutions regarding their applicability to Meta AutoML.

A. Methodology

The training strategies are evaluated using the following criteria:

- **Category** of the training strategy:
  - *Data cleaning*: Identification and correction of flaws in the data;
  - *Data transformation*: Changing the scaling or distribution of the data;
  - *Complexity reduction*: Reducing the feature or sample size to reduce complexity;
  - *Infrastructure*: Adjusting the available hardware and computation power;
  - *Spot checking*: Using trial training sessions to determine the most viable solution;
  - *Training observation*: Actively supervising the ML model performance during the fit process;
  - *Meta-learning*: Learning from past training sessions to improve future trainings.
- **ML process phase**: The phase during which the strategy is applied:
  - *Pre-processing*: The preprocessing phase occurs before the actual ML model training and focuses on preparing the dataset;
  - *Training*: The training phase where the ML models are fitted to the training data;
  - *Post-processing*: the phase after training proper;
  - *Meta-level*: The Meta-Level is not a phase as such but covers the entire process.
- **Advantage**: The benefits of applying the strategy:
  - *Effectiveness*: The ML model's effectiveness may increase;
  - *Efficiency*: The amount of computation power required by the training may decrease.

TABLE I
OVERVIEW OF AUTOML TRAINING STRATEGIES

Strategy	Category	ML process phase	Advantage	Feasibility	References
Handling outliers	Data cleaning	Preprocessing	Effectiveness	Yes	[27][34][29]
Imputing missing values	Data cleaning	Preprocessing	Effectiveness	Yes	[27][31][32]
Omitting redundant samples	Data cleaning	Preprocessing	Effectiveness, Efficiency	Yes	[27][35]
Data sampling	Data cleaning	Preprocessing	Efficiency	Yes	[27][36]
Text feature encoding	Data transformation	Preprocessing	Effectiveness	Yes	[37]
Numerical feature scaling	Data transformation	Preprocessing	Effectiveness	Yes	[38][39]
Feature extraction	Complexity reduction	Preprocessing	Effectiveness, Efficiency	Partial	[40][41]
Feature selection	Complexity reduction	Preprocessing	Effectiveness, Efficiency	Yes	[42][43][27]
Dimensionality reduction	Complexity reduction	Preprocessing	Effectiveness, Efficiency	Yes	[44][30][45]
Hardware optimization	Infrastructure	Training	Efficiency	Yes	
Multi-fidelity optimization	Spot checking	Training	Efficiency	Yes	[28][46][24]
Top 3 optimization	Spot checking	Training	Efficiency	Yes	
Early Stopping	Training observation	Training	Efficiency	Partial	[13]
Meta-learning	Meta-learning	Meta level	Effectiveness, Efficiency	Yes	[47]

- **Feasibility**: Can the strategy be applied to Meta AutoML:
  - *Yes*: Can be applied to Meta AutoML;
  - *Partial*: Can be partially applicable to Meta AutoML.
  - *No*: Cannot be applied to Meta AutoML.

B. Strategies

Table I gives an overview of the training strategies survey.

1) Data cleaning: Data cleaning aims to remove or repair dirty data within the dataset [27]. The following can be issues within a dataset:

- *Outliers*: A value is an outlier if it significantly deviates from the other values;
- *Missing values*: When no data is available for a feature in a sample;
- *Redundancy*: Identical samples present in a dataset.

a) Handling outliers: Detecting and handling outliers is one of the first steps of data cleaning. Outliers represent noise within data [27]. If outliers are improperly handled, they can decrease the prediction performance of the ML model [34]. Statistical methods can be used to automatically identify and handle outliers (e.g. interquartile range, standard deviation [29]). Some AutoML solutions (e.g. Pycaret[2]) offer support to handle outliers from datasets. Handling outliers may increase the effectiveness of ML models.

b) Imputing missing values: Imputing missing values is important. Missing values can negatively impact the quality and accuracy of ML models by introducing bias [27]. Statistical methods can automatically impute missing values (e.g. k-nearest-neighbor-based approaches [31], neural networks [32]). Some AutoML solutions (e.g. MLJAR[3]) implement automatic imputation of missing values. Handling missing values may increase the effectiveness of ML models, as the potential bias from missing values is not introduced.

c) Omitting redundant samples: Duplicated samples can introduce bias in the model. Identical samples can negatively impact the search time of the training session and the ML model's effectiveness. Statistical methods can automatically detect duplicate samples (e.g. probabilistic matching [35]). Removing duplicate samples may increase the effectiveness of the ML model by removing potential bias. It may also increase the training efficiency by reducing the size of the dataset.

d) Data sampling: Sampling can adjust the dataset size to the training configuration to allow the most effective training (e.g. limit time or computation power) [27]. By sampling the dataset the training efficiency may increase and the ML model performance may not be negatively impacted [27]. Some AutoML solutions (e.g. FLAML) apply sampling to adjust the dataset size on the training configuration (e.g. limited time and large sample set by applying e.g. holdout [36]).

2) Data transformation: Data transformation aims to change the type or distribution of the data in a dataset. This includes transforming data into a format an ML algorithm can process [27].

a) Text feature encoding: Text features often can not be processed by ML algorithms and require encoding [37]. Text features may be disregarded without proper encoding or lead to errors during the ML model training. When confronted with text features, three categories of encoding strategies exist:

- *Binary encoding*: encodes the textual values into binary values, e.g. yes/no;
- *Ordinal encoding*: encodes the textual values into a finite set of discrete values with a rank or ordering between them. e.g. low/medium/high;
- *Nominal encoding*: encodes the textual values into a finite set of discrete values with no relationship between them. e.g. married/single/widowed.

Some AutoML solutions (e.g. Autokeras [9]) automatically encode text features. Encoding text features may increase the

[2] https://github.com/pycaret/pycaret
[3] https://github.com/mljar/mljar-supervised

effectiveness of an ML model as it can consider those features while training instead of disregarding them.

b) Numerical feature scaling: Numerical features with a wide range of values may cause bias in the ML model. Some ML approaches are sensitive to the relative magnitude of features and may give more weight to features with larger values [38]. When confronted with numerical features displaying a wide range, two categories of scaling strategies can be applied:

- Normalization: This method resizes the data to a fixed value between 0 and 1;
- Standardization: This method resizes the data to have a median of 0 and a standard deviation of 1.

Some AutoML solutions (e.g. MLJAR) automatically scale numerical features using normalization or standardization functions. Scaling numerical features may increase the effectiveness of an ML model [39].

3) Complexity reduction: Complexity reduction aims to reduce the complexity of a dataset. The available methods can be divided into three categories:

- Feature extraction: Create new features from existing data that may have more meaningful information;
- Feature selection: Reduce the number of features by selecting the most relevant ones without changing the features themselves;
- Dimensionality reduction: Reduce the number of features by transforming the features into a lower dimensional space while preserving essential information.

a) Feature extraction: Feature extraction aims to create a subset of more meaningful features from the existing ones [40]. The construction of new features is highly specific to the data and data type of the dataset. Often, it is required to collaborate with domain experts who can group features correctly together. Automated feature engineering offers data scientists and AutoML solutions methods to automatically create candidate features derived from the original dataset e.g. Deep Feature Synthesis [41]. Using newly created features based on existing data may increase the performance of an ML model. Some AutoML solutions (e.g. MLJAR) apply automated feature engineering to create new and potentially more effective ML models. Multiple open-source libraries focus on automated feature engineering, e.g. Feature-engine[4]. Such tools can help data scientists quickly trial a wide range of different feature combinations. However, only a domain expert has the necessary understanding of the problem to determine the suitability of features.

b) Feature selection: Feature selection aims to reduce dataset complexity by removing non-useful features [42] and creating a feature subset that performs best under classification [43]. A dataset can contain irrelevant or noisy features (e.g. duplicated features) that may introduce bias into an ML model. By removing noisy features, the ML model's effectiveness and training efficiency may increase [27]. The existing methods can be classified into one of three categories:

- *Filter*: Filtering out undesirable features before learning by using heuristics based on the general data characteristics to evaluate the goodness of feature subsets;
- *Wrapper*: Methods that search the feature space for the best-performing subset. They assess the quality of features by training and evaluating a classifier with the subset;
- *Embedded*: Similarly to the wrapper method, the feature selection is performed during the learning process of the ML model.

Some AutoML solutions (e.g. MLJAR) apply automated feature selection to evaluate which feature is relevant for a given search.

c) Dimensionality reduction: Dimensionality reduction aims to reduce the dimensionality of the dataset. The number of features may be considered the dimensionality of the dataset. Dimensionality-reducing methods project the data into a lower-dimensional space that still preserves the most important properties of the original data. By reducing the complexity of the dataset the ML model's effectiveness and the training efficiency may increase. The existing methods can be divided into three categories:

- *Linear dimensionality reduction methods*: e.g. principal component analysis [44];
- *Non-linear dimensionality reduction methods*: e.g. multidimensional scaling [30];
- *Autoencoder methods*: e.g. using artificial neural networks [45].

C. Hardware optimization

Hardware optimization aims to choose the most suitable infrastructure for the task and use it in the most energy-efficient and resource-saving way possible. Optimizing the underlying hardware may reduce the amount of computation power invested during training by providing specialized hardware for ML. Another option is to limit the access to computation power to limit the used computation during training. Some AutoML solutions (e.g. MLJAR) can limit the maximum amount of RAM the solution uses.

D. Spot checking

Spot-checking aims to discover the approach that performs best for an ML task [28]. A spot-checking algorithm uses multiple trials to evaluate multiple ML algorithms on a given dataset to determine their performance. The spot-checking training aims to quickly assess the viability of a collection of ML models and decide which approach to use for further training.

a) Multi-fidelity optimization: The multi-fidelity strategy uses numerous training sessions with low-fidelity samples to evaluate the general trend of a system's behaviour, and a small number of high-fidelity samples to enhance the prediction accuracy in important regions [46]. Sequential multi-fidelity surrogate modelling is one multi-fidelity approach that limits the computational budget in addition to using low-fidelity

[4] https://feature-engine.trainindata.com/en/latest/

samples [24]. Some AutoML solutions (e.g. Auto-Pytorch) use multi-fidelity to improve their training.

b) Top-3 optimization: The top-3 strategy is a variation of the multi-fidelity strategy. The training is divided into two steps. The first is a pre-training with a low-fidelity configuration (limited data and time budget). After all candidates are trained, they are evaluated. The top 3 candidates are then trained with the full dataset and the remaining time. The benefit of top-3 is the increased efficiency of the training process, as only the best 3 ML models are fully trained. The top-3 strategy is not used by any existing AutoML solution, but it could be applied to AutoML.

1) Early-stopping: Early stopping is a strategy to intelligently terminate the training when a user-defined early stopping criterion is met. The standard early-stopping criterion is the loss on the validation set [13]. Some AutoML solutions (e.g. PyCaret) use early stopping to optimize training efficiency. On a Meta AutoML level, it is only partially possible to use early stopping. The main issue is that there is currently no interface available for Meta AutoML to extract the current status of the AutoML solution, except through text mining the console output. One issue with text mining is that some AutoML solutions produce limited or no relevant output during the training itself [6]. Additionally, no interface is available for the Meta AutoML process to halt the AutoML training. In most cases, the Meta AutoML could terminate the process using the underlying operating system, which leads to a complete loss of the ML model within the AutoML solution.

2) Meta-learning: Human domain experts derive knowledge from previous tasks by learning about the performance of ML algorithms. Meta-Learning mimics the perpetual process of "learning to learn" across similar tasks and transferring that prior knowledge to new tasks [47]. Some AutoML solutions (e.g. Auto-Sklearn) use meta-learning to form recommendations for ML algorithms and their configuration based on past training sessions. Meta-learning may increase training efficiency and the ML model effectiveness.

In the next section, we introduce a concept for implementing training strategies using a rule engine.

IV. A Concept for Rule-based Training Strategies

Before introducing the concept of rule-based training strategies, we define the optimization goals we aspire to achieve by applying training strategies:

- *Efficiency*: The energy consumption will be significantly reduced compared to a Meta AutoML training without applying a training strategy;
- *Effectiveness*: The ML model performance will not be significantly reduced.

The concept for achieving those goals uses a *blackboard architecture* [48]. The blackboard architecture is a problem-solving approach combining multiple specialized modules working collaboratively to solve a complex problem. The speech understanding system HEARSAY-II introduced and used this approach [49]. The blackboard architecture defines three components:

- *Blackboard*: a global data store that keeps the problem-solving state data;
- *Knowledge sources*: individual components with domain knowledge needed to solve a problem. Each component is represented as procedures, sets of rules or logic assertions and contributes information that will lead to a solution to the problem.
- *Controller*: A component that monitors the changes on the blackboard and decides what actions to take next. The controller decides on the order of invocation of the knowledge sources.

Fig. 1 shows a BPMN diagram [50] of the rule-based training strategies process. The process starts when a new Meta AutoML training is initiated.

A user initiates a new Meta AutoML training by configuring a new training within a Meta AutoML platform. A Meta AutoML platform is an application based on the Meta AutoML concept [10]. It unifies several AutoML solutions and allows the user to configure and start new training sessions without interacting with the AutoML solutions individually. One example of a Meta AutoML platform is OMA-ML [6]. The result of the user's training configuration is the *training configuration* data. For example, a training configuration could be represented as follows:

```
dataset : census_income.csv
task : tabular_classification
autoML_solutions_activated : [FLAML,
Autogluon, Autosklearn, TPOT, MLJAR,
AutoKeras, Pycaret, EvalML]
strategies_enabled : [top-3]
max_runtime : 60
```

In this example, the user configures the training for a tabular classification of the Census Income dataset [51] with a total runtime of 60 minutes. A total of 8 AutoML solutions are activated (FLAML, Autogluon, Autosklearn, TPOT, MLJAR, AutoKeras, Pycaret, EvalML) and one training strategy (top-3) is enabled.

When uploading new training data into the Meta AutoML platform, it is automatically analyzed. During this *dataset analysis*, the *analysis result* data is generated by extracting the properties of the *training data*. For example, the analysis result for the Census Income dataset could be as follows:

```
samples : 48000
features : 15
missing_values : none
duplicated_samples : 48
```

In this example, the Census Income dataset, is comprised of 48,000 samples and 15 features. It has no missing values and a total of 48 duplicated samples.

The training configuration and analysis result are added to the blackboard and represent the initial blackboard state data. The blackboard state data is the collection of the current state data of all the involved components and the current

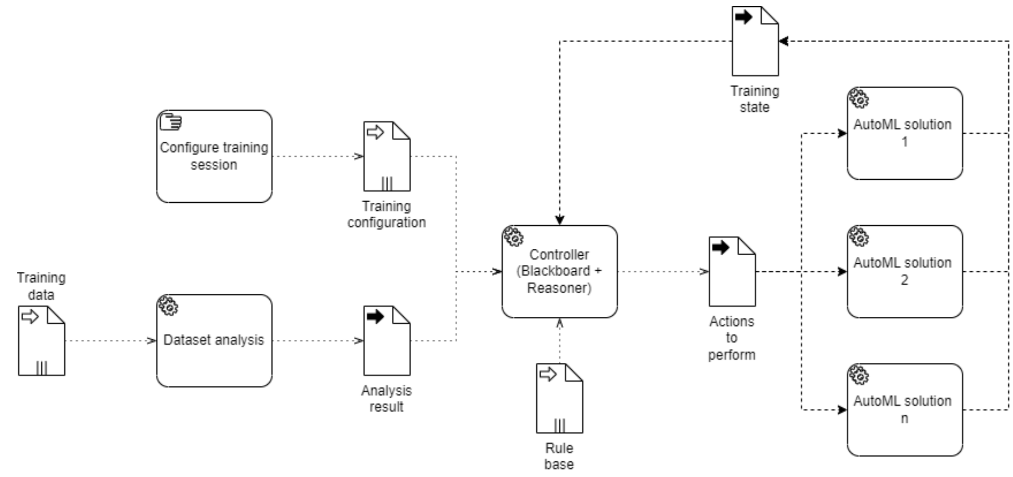


Fig. 1. Rule-based training strategies process as a BPMN diagram

training phase. When the blackboard is initialized the training phase is automatically set to 'preprocessing' and as the training progresses it is updated to 'training' and eventually to 'postprocessing'. For example, using the training configuration and analysis result from the previous examples, the initial blackboard state could be as follows:

```
phase: preprocessing
training_configuration:
    dataset : census_income.csv
    task : ...
analysis_result:
    samples : 48000
    features : ...
```

The blackboard and the reasoner component form together the *controller* module. The controller administers the Meta AutoML training process. It is responsible to advance the Meta AutoML training phase when the reasoner has no more rules to consider. During each phase, the reasoner will assess the *rule base* with the current blackboard state and execute matching rules. The rule base is a collection of all Meta AutoML training strategies supported by a Meta AutoML platform. A training strategy is composed of a condition and a collection of actions to perform if the condition matches. A training strategy is associated with one Meta AutoML training phase (see section III). For example, the definition of the top-3 strategy could be as follows:

```
if
phase == 'training' and
length(training_configuration.autoMl_
solutions_activated) > 3

then
do_top_3_training()
```

In this example, the condition for the top-3 strategy is that the Meta AutoML training phase is equal to 'training' and there are more than three AutoML solutions activated in the training configuration. If this is the case, the controller will execute the do_top_3_training function. During which, the controller instructs the *AutoML solutions* to begin the first training session. This initial training will use only 10% of the sample size, 10% of the max runtime and all activated AutoML solutions. For example, using the blackboard state from above, the *action to perform* by the top-3 strategy for the initial training could be represented as follows:

```
action : training
dataset : census_income.csv
task : tabular_classification
autoML_solutions_activated : [FLAML,
Autogluon, Autosklearn, TPOT, MLJAR,
AutoKeras, Pycaret, EvalML]
sample_size : 10%
max_runtime : 6
```

This action instructs the AutoML solution modules (FLAML, Autogluon, Autosklearn, TPOT, MLJAR, AutoKeras, Pycaret, EvalML) to perform tabular classification training, using 10% of the samples from the Census Income dataset, and a maximum runtime of 6 minutes.

During the training process, the AutoML solution modules notify the blackboard about their current *training state*. Each AutoML solution module represents the implementation of one AutoML solution used by the Meta AutoML platform. The training state represents information about the progression of the AutoML training process. For example, the information provided by the AutoML solution Autokeras during the initial training could be represented as follows:

```
AutoKeras:
    remaining_training_time : 2
    best_model_performance : 0.8
```

In this example, the remaining training time is 2 minutes, and the highest ML model performance is 80%. The ML model performance is measured using a standard ML metric for the current ML task. For example, the metric for tabular classification could be accuracy. This training state is updated throughout the AutoML solution training.

When the initial training for the top-3 strategy concludes, the controller evaluates the training state from all AutoML solutions and continues with the second training session using the complete sample set, the remaining 90% of the max runtime and the top-3 AutoML solutions from the preliminary training. For example, the final training state of the 8 AutoML solutions could be as follows:

```
FLAML:
    remaining_training_time : 0
    best_model_performance : 0.77
Autogluon:
    remaining_training_time : 0
    best_model_performance : 0.78
Autosklearn:
    remaining_training_time : 0
    best_model_performance : 0.87
TPOT:
    remaining_training_time : 0
    best_model_performance : 0.87
MLJAR:
    remaining_training_time : 0
    best_model_performance : 0.88
AutoKeras:
    remaining_training_time : 0
    best_model_performance : 0.92
Pycaret:
    remaining_training_time : 0
    best_model_performance : 0.95
EvalML:
    remaining_training_time : 0
    best_model_performance : 0.95
```

All AutoML solutions finished the preliminary training indicated by the remaining training time of 0. The controller based on the top-3 strategy decides that the best-performing AutoML solutions by best model performance are: AutoKeras, Pycaret and EvalML. The controller instructs these three AutoML solutions on further actions to perform. In this case, the action is to begin the second training using the remaining time and complete sample set. For example, the action to perform the second training could be as defined as follows:

```
action : training
dataset : census_income.csv
task : tabular_classification
autoML_solutions_activated : [AutoKeras,
Pycaret, EvalML]
sample_size : 1.0
max_runtime : 54
```

This action instructs the AutoML solutions AutoKeras, Pycaret and EvalML to perform a tabular classification training using the Census Income dataset with all the samples and a maximum runtime of 54 min.

As described above, the AutoML solution modules perform the training session and update their training state on the blackboard accordingly. When all three AutoML solutions conclude their training, the top-3 strategy ends. When the controller assesses that no more rules can be applied during this training, the Meta AutoML training concludes.

In the next section, we introduce the prototypical implementation of the rule-based training strategy within the OMA-ML platform.

V. Prototypical Implementation

The rule-based training strategy concept was implemented as a proof-of-concept in the Meta AutoML platform OMA-ML. OMA-ML is an open-source[5] platform providing users with a web application-based interface to configure the Meta AutoML training. Fig. 2 displays a screenshot of the training wizard used to configure the Meta AutoML training.

The training wizard displays the required and optional parameters for a Meta AutoML training. The minimal configuration OMA-ML requires comprises of the ML task (tabular classification), the target column (‘class’) and a maximum runtime (60 minutes).

For expert users, OMA-ML allows in-depth parametrization of the Meta AutoML training using optional parameters. The user may activate or deactivate individual ML libraries and associated AutoML solutions. For example, in Fig. 2 the following AutoML solutions are activated: Autogluon, Autosklearn, EvalML, FLAML, MLJAR, TPOT, Pycaret and Autokeras. Additionally, OMA-ML displays a collection of training strategies compatible with the current training configuration. For example, the top-3 strategy is available since more than 3 AutoML solutions are activated.

OMA-ML uses an ML ontology to display individual AutoML solution parameters. These are parameters AutoML solutions provide to fine-tune their search process. When the user clicks on the finish button, the Meta AutoML training begins.

The OMA-ML web application is developed in C# with the Blazor web framework[6]. OMA-ML follows a 3-layer architecture design. See Fig. 3 for an overview of the software architecture and technologies used.

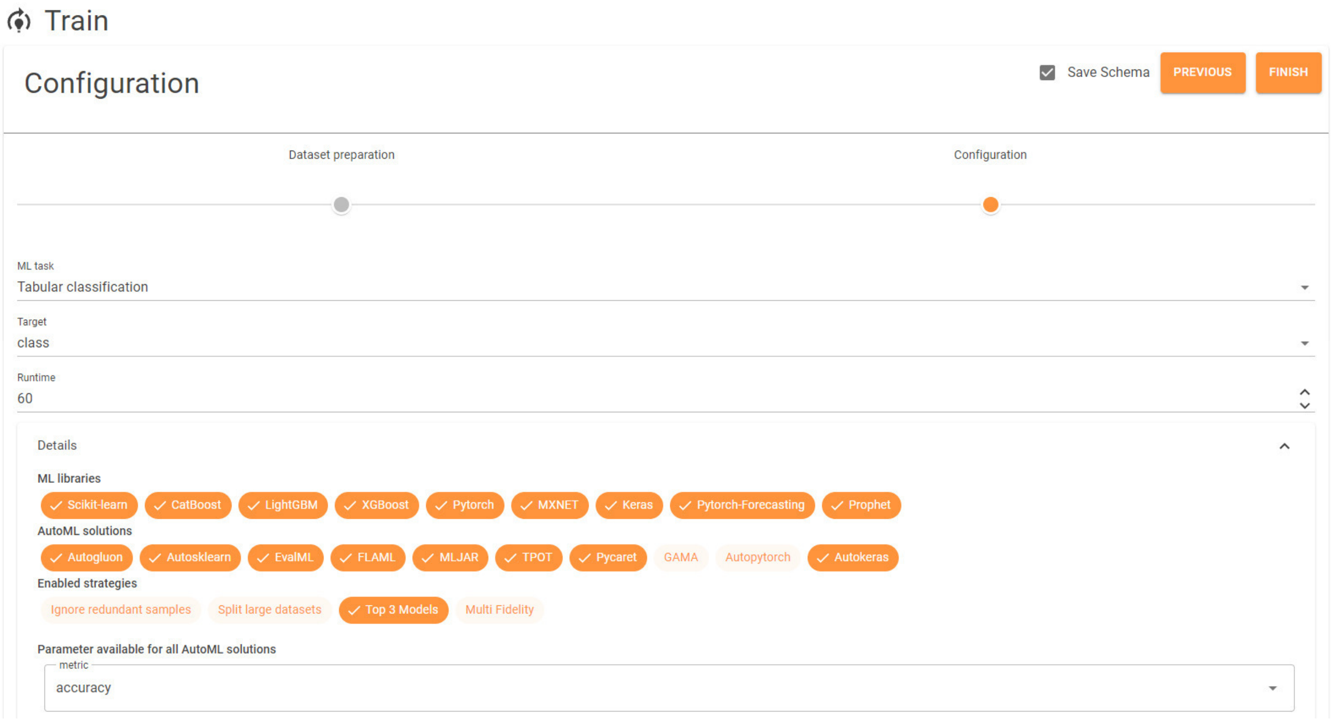
The presentation layer is connected to the logic layer using a gRPC[7] interface. The logic layer is developed in Python based on the blackboard architecture. The Controller component uses the library rule-engine[8] to reason over the rule base. The library rule-engine provides a grammar to create general-purpose rule objects from a logical expression that can be applied to arbitrary objects. The Meta AutoML training strategies conditions are modelled using this grammar. The training strategies are implemented within a rule base

[5] https://github.com/hochschule-darmstadt/MetaAutoML

[6] https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor

[7] https://grpc.io/

[8] https://pypi.org/project/rule-engine/



Train

Configuration

☑ Save Schema PREVIOUS FINISH

Dataset preparation ○ — Configuration ●

ML task: Tabular classification

Target: class

Runtime: 60

Details

ML libraries: ✓ Scikit-learn, ✓ CatBoost, ✓ LightGBM, ✓ XGBoost, ✓ Pytorch, ✓ MXNET, ✓ Keras, ✓ Pytorch-Forecasting, ✓ Prophet

AutoML solutions: ✓ Autogluon, ✓ Autosklearn, ✓ EvalML, ✓ FLAML, ✓ MLJAR, ✓ TPOT, ✓ Pycaret, GAMA, Autopytorch, ✓ Autokeras

Enabled strategies: Ignore redundant samples, Split large datasets, ✓ Top 3 Models, Multi Fidelity

Parameter available for all AutoML solutions

metric: accuracy

Fig. 2. Screenshot of the OMA-ML training configuration wizard

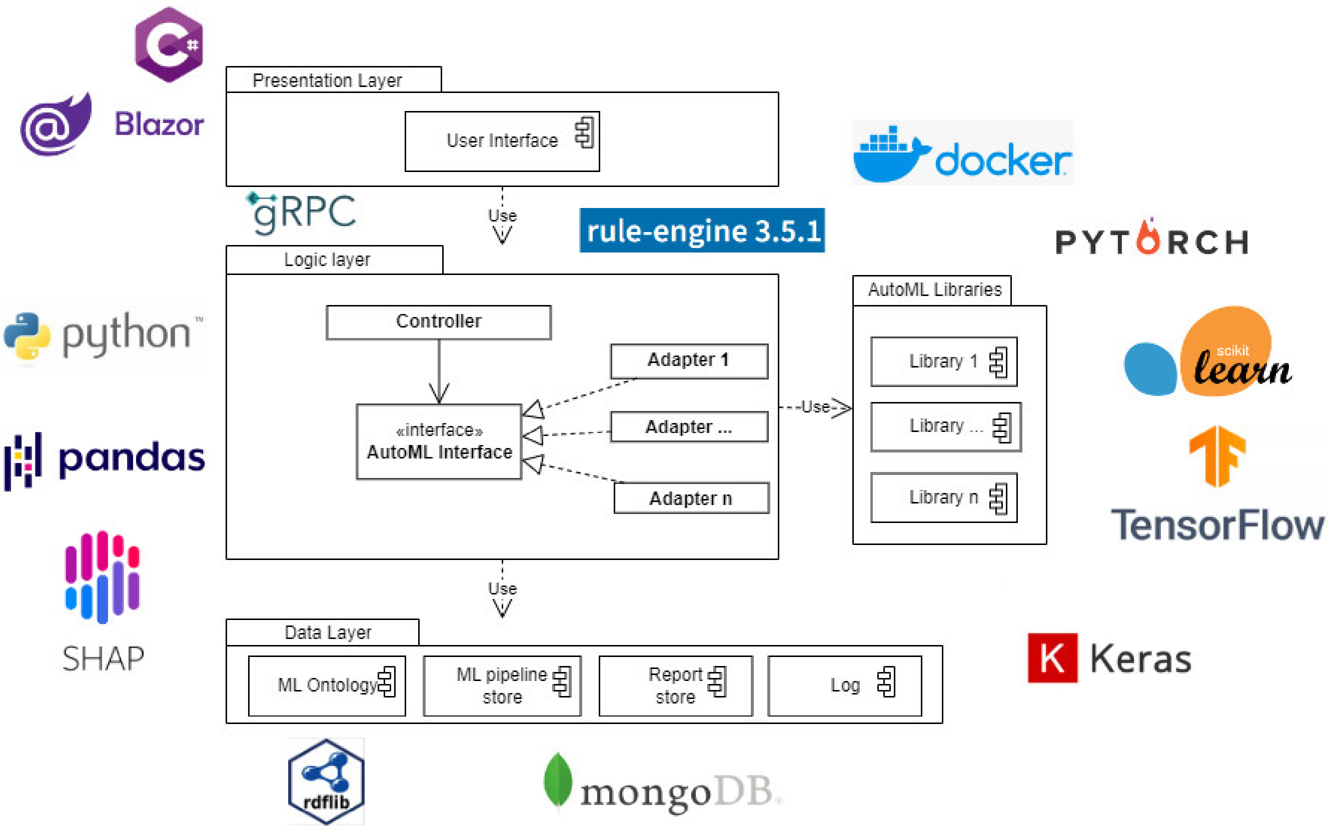


Fig. 3. OMA-ML software architecture and technologies (adapted from [6])

component. For example, the top-3 strategy implementation is as follows:

```
self.register_rule(
'training.top_3',
        Rule("phase == 'training'" and
        training_configuration['activated
        _auto_ml_solution'].length > 3,
        context=training_context),
        self.do_top_3
)
```

The register_rule functionality of the rule base permits rules to be registered. This method requires as parameters, the name, a condition in the rule-engine grammar and a function to execute when the condition matches. In the example above the top-3 strategy has:

- the name: 'training.top_3';
- the condition: the phase is training and the sum of all activated AutoML solutions must be greater than three;
- the function: 'do_top_3'.

Using the register_rule functionality, new training strategies can easily be added to the existing rule base. For example, the whole implementation of the top-3 strategy requires less than 50 lines of code. During a Meta AutoML training, the controller evaluates the rule base for any rule matching the state of the blackboard. Any matching rule invokes their respective function and interacts with the AutoML adapters. The AutoML adapters use the AutoML solutions (e.g. AutoKeras). They are based on the adapter-pattern and plug into the Controller, easing integration.

The logic layer uses the data layer to connect to various data stores. The ML ontology is located here and loaded using the Pyhton library RDFlib[9] into the Controller. SPARQL queries are used to interact with the ML ontology. Additionally, the document database MongoDB stores data generated by the Meta AutoML process. The ML pipelines generated by the AutoML adapters are saved in the ML pipeline store. Finally, any logs generated during the Meta AutoML training process are stored in a log storage.

VI. Evaluation

This section evaluates the concept and implementation of the rule-based training strategies within the Meta AutoML platform OMA-ML. By applying training strategies to Meta AutoML we aim to significantly reduce the energy consumption (efficiency) of the Meta AutoML process while avoiding a significant reduction in ML model performance (effectiveness). To evaluate these goals we compare two measures of quality:

- *Best ML model accuracy*: The highest accuracy of all the AutoML solutions found ML models;
- *Training CO_2-eq*: The sum of all the CO_2 equivalence produced by the AutoML solutions training.

To measure the CO_2 equivalence, the Python library codecarbon[10] was used. Codecarbon measures the amount of CO_2-eq emitted by the individual AutoML solutions. Codecarbon tracks the power consumption of the underlying computational infrastructure, measured in kilowatt-hours. This value is multiplied by the carbon intensity of the electricity consumed for the computation. The carbon intensity is calculated as a weighted average of the emissions of the different energy sources used to generate the used electricity (e.g. natural gas, coal, wind) in the respective country[11], here Germany.

During the course of the experiment, five datasets from the open-source AutoML benchmark by Gijsbers et al. [52] were used:

- *adult*[12]: This dataset contains samples of more than 48,000 individuals and their socioeconomic properties extracted from the Census database in 1994. This dataset is a binary classification and aims to predict if an individual earns over 50k a year;
- *amazon*[13]: This dataset contains samples of more than 32,000 resource requests with the associated Amazon employee meta information from 2010 and 2011. This dataset is a binary classification and aims to predict if access to a resource was approved;
- *sylvine*[14]: This dataset contains more than 5,000 numerical samples; there is no definition of the origin of the data of this dataset. This dataset is a binary classification;
- *credit-g* [15]: This dataset contains samples of 1,000 individuals, their economic properties and loan requests. This dataset is a binary classification and aims to predict the credit score of an individual;
- *kc1*[16]: This dataset contains samples of more than 2,000 software modules and their quality metrics. This dataset is a binary classification and aims to predict if a software module has a defect.

OMA-ML performed two training sessions with each dataset. During the first training session, no optimization strategy was activated. For the second training session, the top-3 optimization strategy was activated. After every training session, the best ML model performance by accuracy [53] was logged as well as the AutoML solution which produced this ML model. Finally, the total CO_2-eq was calculated by taking the total of all AutoML solutions training CO_2-eq.

The training sessions were performed on an AMD Ryzen 7 5800H @ 3.20 GHz CPU with 64GB of RAM. A result summary of all training sessions can be seen in Table II.

The Meta AutoML training sessions applying the top-3 optimization all display a significant saving in CO_2-eq. The difference ranges from 59% with the adult dataset to 70% with the sylvine dataset. The performance of the best ML model

[9] https://pypi.org/project/rdflib/

[10] https://github.com/mlco2/codecarbon

[11] https://mlco2.github.io/codecarbon/methodology.html

[12] https://openml.org/search?type=data&status=active&id=179

[13] https://openml.org/search?type=data&status=active&id=4135

[14] https://openml.org/search?type=data&status=active&id=41146

[15] https://openml.org/search?type=data&status=active&id=31

[16] https://openml.org/search?type=data&status=active&id=1067

TABLE II
EVALUATION SUMMARY

	Meta AutoML, no optimization			Meta AutoML, top-3 optimization			Comparision	
Dataset	**AutoML solution**	**accuracy**	**CO_2-eq [g]**	**AutoML solution**	**accuracy**	**CO_2-eq [g]**	**Difference accuracy**	**Saving CO_2-eq**
adult	FLAML	0.88	53.5	FLAML	0.87	22.1	-0.01	59%
amazon	Autogluon	0.95	45.8	Autogluon	0.95	16.6	0.00	64%
sylvine	Autosklearn	0.95	49.96	Autosklearn	0.95	15.0	0.00	70%
credit-g	FLAML	0.78	36.2	MLJAR	0.77	11.7	-0.01	68%
kc1	Autogluon	0.87	37.0	Autogluon	0.87	13.1	0.00	65%

is equal to or at worst one percentage point lower compared to the Meta AutoML training sessions without optimization. Additionally, the best AutoML solution is the same in both training sessions for four out of the five datasets. The only exception being the dataset credit-g, the best AutoML solution is FLAML in the training without optimization and MLJAR during the training with optimization. While FLAML did not produce the best ML model during the optimized training session, it was one of three AutoML solutions selected by the top-3 strategy to train a model in the second training session.

By using the top-3 training strategy it is possible to significantly (up to 70%) reduce the amount of CO_2-eq for a training session. While also achieving similar or slightly (1 percentage point) reduced results in the performance of the best ML model, the goals of this study can been regarded as achieved.

VII. Conclusion and Future Work

Meta AutoML provides users with or without data science knowledge access to automatically generated ML models. However, there is the major issue of massive computation power requirements for Meta AutoML. The concept of rule-based training strategies aims to optimize the energy efficiency and ML model effectiveness for Meta AutoML. The contribution of this paper is two fold. Firstly, we presented a survey of 14 AutoML training strategies, classifying them by their function category, ML phase, advantage and implementation feasibility for Meta AutoML. Most training strategies aim to optimize the dataset and only 3 strategies focus on the ML model fitting process. Twelve training strategies can be fully applied to the Meta AutoML process. The exception being the feature extraction and early-stopping strategy. While different approaches exist to automate feature extraction, only a domain expert can decide if the extracted features are relevant. Early stopping requires process information and an interface from the AutoML solutions to allow interaction by a Meta AutoML platform. Both of these are not supported by most AutoML solutions, making implementation of early stopping challenging. While AutoML solutions implement various training strategies already, on the Meta AutoML level however there has been no research on optimization using training strategies. The second contribution of this paper addresses this issue.

We presented the novel concept of rule-based training strategies. This concept uses the blackboard architecture to implement training strategies for Meta AutoML. This concept aims to significantly increase the efficiency of Meta AutoML by reducing the required computation power while not significantly reducing the best ML model performance. The Meta AutoML platform OMA-ML was used to implement a proof-of-concept. We evaluated the implementation by using the training strategy top-3. During the evaluation, five binary classification datasets were used. Using each dataset, two experiments were performed, one without an optimization strategy and one with the top-3 optimization strategy. Applying the top-3 optimization led to a saving of up to 70% CO_2-eq, with the best ML model having identical or slightly reduced performance (one percentage point).

The results show that rule-based training strategies can improve the Meta AutoML training efficiency significantly with only a slight reduction in ML model effectiveness. However, further evaluation is required. In future work, we aim to implement additional training strategies and perform extensive benchmark testing using a variety of ML tasks (e.g. regression, time series forecasting etc.) and various dataset types (e.g. texts, images, time series etc.).

Acknowledgment

This work is funded by the German federal ministry of education and research (BMBF) in the program Zukunft der Wertschöpfung (funding code 02L19C157), and supported by Projektträger Karlsruhe (PTKA). The responsibility for the content of this publication lies with the authors.

References

[1] S. J. Russell and P. Norvig, *Artificial intelligence: A modern approach*, fourth edition ed., ser. Pearson Series in Artificial Intelligence. Hoboken, NJ: Pearson, 2021. ISBN 9780134610993

[2] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, Eds., *Efficient and Robust Automated Machine Learning*. MIT Press, 2015. doi: 10.5555/2969442.2969547

[3] M.-A. Zöller and M. F. Huber, “Benchmark and survey of automated machine learning frameworks,” *Journal of Artificial Intelligence Research*, vol. 70, pp. 409–472, 2021. doi: 10.1613/jair.1.11854

[4] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, “Auto-weka: Combined selection and hyperparameter optimization of classification algorithms,” in *KDD ’13 : the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining : August 11-14, 2013, Chicago, Illinois, USA*, I. S. Dhillon, Ed. ACM, 2013. doi: 10.1145/2487575.2487629. ISBN 9781450321747 pp. 847–855.

[5] C. H. N. Larcher and H. J. C. Barbosa, "Auto-cve: a coevolutionary approach to evolve ensembles in automated machine learning," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. ACM Digital Library, M. López-Ibáñez, Ed. New York,NY,United States: Association for Computing Machinery, 2019. doi: 10.1145/3321707.3321844. ISBN 9781450361118 pp. 392–400.

[6] A. Zender and B. G. Humm, "Ontology-based meta automl," *Integrated Computer-Aided Engineering*, vol. 29, no. 4, pp. 351–366, 2022. doi: 10.3233/ICA-220684

[7] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka: Automatic model selection and hyperparameter optimization in weka," in *Automated machine learning*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Cham: Springer International Publishing, 2019, pp. 81–95. ISBN 978-3-030-05317-8

[8] Y. Poulakis, C. Doulkeridis, and D. Kyriazis, "Autoclust: A framework for automated clustering based on cluster validity indices," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020. doi: 10.1109/ICDM50108.2020.00153. ISBN 978-1-7281-8316-9 pp. 1220–1225.

[9] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. ACM Digital Library, A. Teredesai, Ed. New York,NY,United States: Association for Computing Machinery, 2019. doi: 10.1145/3292500.3330648. ISBN 9781450362016 pp. 1946–1956.

[10] B. G. Humm and A. Zender, "An ontology-based concept for meta automl," in *Artificial Intelligence Applications and Innovations*, ser. Springer eBook Collection, I. Maglogiannis, J. Macintyre, and L. Iliadis, Eds. Cham: Springer International Publishing and Imprint Springer, 2021, vol. 627, pp. 117–128. ISBN 978-3-030-79149-0

[11] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. ISBN 978-3-642-35288-1

[12] S. Thrun and L. Pratt, "Learning to learn: Introduction and overview," in *Learning to Learn*, S. Thrun and L. Pratt, Eds. Boston, MA and s.l.: Springer US, 1998, pp. 3–17. ISBN 978-1-4613-7527-2

[13] L. Prechelt, "Early stopping — but when?" in *Neural Networks: Tricks of the Trade*, ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7700, pp. 53–67. ISBN 978-3-642-35288-1

[14] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020. doi: 10.1145/3381831

[15] A. Doke and M. Gaikwad, "Survey on automated machine learning (automl) and meta learning," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021. doi: 10.1109/ICCCNT51525.2021.9579526 pp. 1–5.

[16] Y.-W. Chen, Q. Song, and X. Hu, "Techniques for automated machine learning," *ACM SIGKDD Explorations Newsletter*, vol. 22, no. 2, pp. 35–50, 2021. doi: 10.1145/3447556.3447567

[17] P. Ge, "Analysis on approaches and structures of automated machine learning frameworks," in *2020 International Conference on Communications, Information System and Computer Engineering*. Piscataway, NJ: IEEE, 2020. doi: 10.1109/CISCE50729.2020.00106. ISBN 978-1-7281-9761-6 pp. 474–477.

[18] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021. doi: 10.1016/j.knosys.2020.106622

[19] Q. Yao, M. Wang, Y. Chen, W. Dai, Y.-F. Li, W.-W. Tu, Q. Yang, and Y. Yu, "Taking human out of learning applications: A survey on automated machine learning." [Online]. Available: https://arxiv.org/pdf/1810.13306

[20] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data." [Online]. Available: https://arxiv.org/pdf/2003.06505

[21] T. T. Le, W. Fu, and J. H. Moore, "Scaling tree-based automated machine learning to biomedical big data with a feature set selector," *Bioinformatics (Oxford, England)*, vol. 36, no. 1, pp. 250–256, 2020. doi: 10.1093/bioinformatics/btz470

[22] E. LeDell and S. Poirier, "H2o automl: Scalable automatic machine learning," *7th ICML Workshop on Automated Machine Learning (AutoML)*, 2020. [Online]. Available: https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf

[23] L. Zimmer, M. Lindauer, and F. Hutter, "Auto-pytorch: Multi-fidelity metalearning for efficient and robust autodl," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 9, pp. 3079–3090, 2021. doi: 10.1109/TPAMI.2021.3067763

[24] A. I. Forrester, A. Sóbester, and A. J. Keane, "Multi-fidelity optimization via surrogate modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 463, no. 2088, pp. 3251–3269, 2007. doi: 10.1098/rspa.2007.1900

[25] S. Falkner, A. Klein, and F. Hutter, "Bohb: Robust and efficient hyperparameter optimization at scale," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 1437–1446. [Online]. Available: https://proceedings.mlr.press/v80/falkner18a.html

[26] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Auto-sklearn: Efficient and robust automated machine learning," in *Automated machine learning*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds. Cham: Springer International Publishing, 2019, pp. 113–134. ISBN 978-3-030-05317-8

[27] S. García, J. Luengo, and F. Herrera, *Data Preprocessing in Data Mining*. Cham: Springer International Publishing, 2015, vol. 72. ISBN 978-3-319-10246-7

[28] J. Brownlee, *Machine learning mastery with Python: Understand your data, create accurate models and work projects end-to-end*, edition: v1.20 ed. [Australia]: [Jason Brownlee], 2021. ISBN 979-8540446273

[29] E. Panjei, Le Gruenwald, E. Leal, C. Nguyen, and S. Silvia, "A survey on outlier explanations," *The VLDB journal : very large data bases : a publication of the VLDB Endowment*, vol. 31, no. 5, pp. 977–1008, 2022. doi: 10.1007/s00778-021-00721-1

[30] G. B. Rabinowitz, "An introduction to nonmetric multidimensional scaling," *American Journal of Political Science*, vol. 19, no. 2, p. 343, 1975. doi: 10.2307/2110441

[31] P. Keerin, W. Kurutach, and T. Boongoen, "Cluster-based knn missing value imputation for dna microarray data," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2012. doi: 10.1109/ICSMC.2012.6377764. ISBN 978-1-4673-1714-6 pp. 445–450.

[32] I. A. Gheyas and L. S. Smith, "A neural network-based framework for the reconstruction of incomplete data sets," *Neurocomputing*, vol. 73, no. 16-18, pp. 3039–3065, 2010. doi: 10.1016/j.neucom.2010.06.021

[33] J. Yoo, T. Joseph, D. Yung, S. A. Nasseri, and F. Wood, "Ensemble squared: A meta automl system." [Online]. Available: https://arxiv.org/pdf/2012.05390

[34] M. Kalisch, M. Michalak, M. Sikora, Ł. Wróbel, and P. Przystałka, "Influence of outliers introduction on predictive models quality," in *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery*, ser. Communications in Computer and Information Science, S. Kozielski, D. Mrozek, P. Kasprowski, B. Małysiak-Mrozek, and D. Kostrzewa, Eds. Cham: Springer International Publishing, 2016, vol. 613, pp. 79–93. ISBN 978-3-319-34098-2

[35] J. Asher, D. Resnick, J. Brite, R. Brackbill, and J. Cone, "An introduction to probabilistic record linkage with a focus on linkage processing for wtc registries," *International journal of environmental research and public health*, vol. 17, no. 18, 2020. doi: 10.3390/ijerph17186937

[36] C. Wang, Q. Wu, M. Weimer, and E. Zhu, "Flaml: A fast and lightweight automl library," in *Proceedings of Machine Learning and Systems*, A. Smola, A. Dimakis, and I. Stoica, Eds., vol. 3, 2021, pp. 434–447. [Online]. Available: https://proceedings.mlsys.org/paper/2021/file/92cc227532d17e56e07902b254dfad10-Paper.pdf

[37] K. Potdar, T. S., and C. D., "A comparative study of categorical variable encoding techniques for neural network classifiers," *International Journal of Computer Applications*, vol. 175, no. 4, pp. 7–9, 2017. doi: 10.5120/ijca2017915495

[38] J. Grus, *Data science from Scratch: First principles with Python*, 1st ed. Beijing and Köln: O'Reilly, 2015. ISBN 978-1-491-90142-7

[39] M. Ahsan, M. Mahmud, P. Saha, K. Gupta, and Z. Siddique, "Effect of data scaling methods on machine learning algorithms and model performance," *Technologies*, vol. 9, no. 3, p. 52, 2021. doi: 10.3390/technologies9030052

[40] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 Science and Information Conference*. IEEE, 2014. doi: 10.1109/SAI.2014.6918213. ISBN 978-0-9893193-1-7 pp. 372–378.

[41] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015. doi: 10.1109/DSAA.2015.7344858. ISBN 978-1-4673-8272-4 pp. 1–10.

[42] A. Zheng and A. Casari, *Feature engineering for machine learning: Principles and techniques for data scientists*. Beijing and Boston and Farnham and Sebastopol and Tokyo and Beijing and Boston and Farnham and Sebastopol and Tokyo: O'Reilly, 2018. ISBN 978-1491953242

[43] A. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 2, pp. 153–158, 1997. doi: 10.1109/34.574797

[44] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010. doi: 10.1002/wics.101

[45] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (New York, N.Y.)*, vol. 313, no. 5786, pp. 504–507, 2006. doi: 10.1126/science.1127647

[46] Q. Zhou, M. Zhao, J. Hu, and M. Ma, *Multi-fidelity Surrogates: Modeling, Optimization and Applications*, 1st ed., ser. Engineering Applications of Computational Methods. Singapore: Springer Nature Singapore and Imprint Springer, 2023, vol. 12. ISBN 978-981-19-7212-6

[47] S. Thrun and L. Pratt, Eds., *Learning to Learn*. Boston, MA and s.l.: Springer US, 1998. ISBN 978-1-4613-7527-2

[48] H. Penny Nii, "An introduction to knowledge engineering, blackboard model, and age," 03.1980. [Online]. Available: https://purl.stanford.edu/cq570jp5428

[49] L. D. ERMAN, F. HAYES-ROTH, V. R. LESSER, and D. R. REDDY, "The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty," in *Readings in Artificial Intelligence*. Elsevier, 1981, pp. 349–389. ISBN 9780934613033

[50] "Iso/iec 19510:2013(en), information technology — object management group business process model and notation," 31.03.2022. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso-iec:19510:ed-1:v1:en

[51] R. Kohavi, "Census income," 1996.

[52] P. Gijsbers, E. LeDell, J. Thomas, S. Poirier, B. Bischl, and J. Vanschoren, "An open source automl benchmark." [Online]. Available: https://arxiv.org/pdf/1907.00909

[53] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. doi: 10.1016/j.ipm.2009.03.002

DOI: 10.15439/2023F3838
ISSN 2300-5963 ACSIS, Vol. 35

Towards automated detection of adversarial attacks on tabular data

Piotr Biczyk*§, Łukasz Wawrowski†
* Silesian University of Technology, Faculty of Automatic Control, Electronics and Computer Science
Akademicka 16, 44-100 Gliwice, Poland, pbiczyk@gmail.com
† Łukasiewicz Research Network, Institute of Innovative Technologies EMAG
ul. Leopolda 31, 40-189 Katowice, Poland, lukasz.wawrowski@emag.lukasiewicz.gov.pl
§ QED Software sp. z o.o., Mazowiecka 11/49, 00-052 Warsaw, Poland

***Abstract*—The paper presents a novel approach to investigating adversarial attacks on machine learning classification models operating on tabular data. The employed method involves using diagnostic parameters calculated on an approximated representation of a model under attack and analyzing differences in these diagnostic parameters over time. The hypothesis researched by the authors is that adversarial attack techniques, even if attempting a low-profile modification of input data, influence those diagnostic attributes in a statistically significant way. Thus, changes in diagnostic attributes can be used for detecting attack events. Three attack approaches on real-world datasets were investigated. The experiments confirm the approach as a promising technique to be further developed for detecting adversarial attacks.**

I. Introduction

THE widespread adoption of machine learning (ML) algorithms in various fields, such as healthcare, finance, transportation, and industry [1], has revolutionized the way we process and analyze vast amounts of data [2]. However, the rapid proliferation of ML applications has also raised operational security concerns, as malicious actors increasingly target these models with adversarial attacks to undermine their reliability and compromise their performance [3]. These attacks pose a significant threat to the integrity and trustworthiness of ML models, necessitating the development of robust detection and mitigation techniques to protect the systems from potential threats [4], [5].

The motivation for our work is rooted in the observed disparity between machine learning implementations, which primarily emphasize traditional quality characteristics, and the security-focused mindset held by stakeholders responsible for operational security in businesses that incorporate machine learning solutions reinforced by real-world examples of adversarial machine learning attacks [6]. This gap highlights the need for a more holistic approach to designing and deploying machine learning systems, taking into account not only their performance but also their resilience to adversarial attacks and other security challenges [7], [8].

Furthermore, we have found that the field of rough sets theory (RST) has not been thoroughly explored when it comes to its capability in attack detection. One of defining characteristics of RST is that it can be used to handle uncertainty and vagueness of data [9]. By approximating the decision boundaries of a classifier model, rough sets can be used to identify regions in the input space where adversarial perturbations are likely to occur [10]. By monitoring these regions, unusual deviations or patterns in input data can be flagged as potential adversarial attacks. This approach, if proved to be working, can not only provide a robust mechanism for detecting adversarial examples but also offer insights into the underlying structure of the data and its susceptibility to manipulations, thereby informing the design of more secure machine learning models. In this work, we want to test the usefulness of RST methods in practical security applications in the domain of adversarial machine learning prevention.

The end goal of our work is to create a robust black-box-based method that can be utilized in real-world scenarios for the detection and prevention of misclassification adversarial attacks on machine learning models, increasing the safety and trustworthiness of machine learning applications in everyday scenarios.

II. Related Work

A. Advesarial Machine Learning (AML)

Starting with the pioneering works of Szegedy et al. [11] and Goodfellow et al. [12], the topic of adversarial machine learning has entered the spotlight of the research community. Those works demonstrated that it is possible to influence the operation of machine learning models, most notably image-based classifiers, by adding limited amplitude (undetectable to the human eye) perturbances to original images, causing spectacular cases of misclassification of images.

Since the concept's inception, it has left the walls of academia, and real-world adversarial machine-learning attacks have been proven possible in various areas [13], [14].

There are several possible ways to classify the diverse world of AML attacks [3], [8], [15]. The classification of AML attacks is based on a different axis:

- Knowledge-based classification — distinguishes attacks based on the amount of knowledge an attacker has about the target model.
- Capability-Based Classification — considers the capabilities of the attacker and the stage of the machine learning pipeline targeted.

- Goal-Based Classification — differentiates attacks based on the attacker's objectives.

A point of note is that most of the published papers refer to attacks and defenses on image data [15]. Only in recent years, the interest in attacks and defense on tabular data processing models has increased [16].

B. AML detection

Complementary to works dedicated to increasing the robustness of models against adversarial machine learning, significant effort is put into the detection of attacks against ML models. These techniques are primarily designed to identify inputs that have been modified with the intent of misleading a machine-learning model.

Some detection strategies attempt to detect adversarial examples by identifying instances that significantly deviate from the distribution of normal instances. An example of such a technique, specific for adversarial attacks on image classification models, has been described in [17]. The detection technique presented therein hangs on the realization that adversarial images place abnormal emphasis on the lower-ranked principal components from principal component analysis (PCA), which allows adversarial examples to stand out after PCA whitening. Most recently, salience-based methods have been used to analyze adversarial examples for NLP models — based on an observation that salience tokens have a direct correlation with adversarial perturbations [18].

Another approach to the detection of adversarial perturbations is to train a separate classifier used to classify inputs as normal or adversarial. In [19], such an approach was implemented using neural network classifiers. The method has been proven to be useful for the detection of small adversarial perturbances in images (below the human-detection threshold). This auxiliary classifier can be integrated with the main model and can provide a reasonable level of adversarial threat detection [20].

An interesting approach for the detection of perturbed images has been presented in [21], where a method has been presented that detects adversarial examples by comparing the output of a discriminator of a generative adversarial network (GAN) trained on the dataset — with the realization that adversarial examples are scored lower by the discriminator part of the GAN.

III. Notions and definitions

A. Adversarial attack types used

In this work, we have tested our attack detection method against three known attack techniques: HopSkipJump, PermuteAttack, and ZOO. These attack methods were chosen based on three criteria: a thorough description of the attack methodology in an academic paper, its applicability to attacks on classifier models operating on tabular data, and the availability of its source code. While the choice of attack methods to be used in our work was arbitrary, it was considered to be proper for the preliminary attack detection method verification presented in this work.

1) HopSkipJump Attack: The HopSkipJump attack, also known as the Decision-Based Boundary attack, is an adversarial attack on machine learning models designed to generate adversarial examples by directly manipulating the input data to cause misclassifications while minimizing the perturbation to the original input [22].

It is an iterative, decision-based attack, meaning that it only requires access to the model's output decisions (e.g., classification labels) rather than full access to the model's internal workings or gradients. The attack algorithm consists of three main steps:

- Hop: Initialization of the adversarial example by searching for a starting point near the decision boundary of the model.
- Skip: Binary search along the line connecting the original input and the initialized adversarial example to find a point that lies closer to the decision boundary.
- Jump: Gradient-free optimization to further perturb the adversarial example while keeping it within a predefined perturbation budget.

2) Permute Attack: PermuteAttack, described in [23], is a counterfactual example generation method capable of handling tabular data including discrete and categorical variables. The method is based on gradient-free optimization genetic algorithm, that permutes randomly selected features making sure that resulting values are within ranges that are not outstanding for a given data set. As a result, it produces adversarial data points that are modified, as compared to the original data points, in a way that can elude some anomaly-detecting methods. Resulting adversarial examples can be also used for the analysis of the robustness of the attacked model.

3) Zeroth-Order Optimization (ZOO) Attack: The Zeroth Order Optimization (ZOO) attack is a black-box adversarial attack proposed by Chen et al. [24] The key idea behind the ZOO attack is to approximate gradients of the target model using zeroth-order (derivative-free) optimization methods, allowing the attacker to generate adversarial examples without direct access to the model's gradients or architecture. The ZOO attack steps:

- Approximate the gradients using zeroth-order optimization, such as the coordinate-wise finite-difference method or the spherical coordinate-based method.
- Compute the adversarial perturbation using the approximated gradients.
- Apply the perturbation to the original input, ensuring that the adversarial example remains within a predefined perturbation budget.

B. Diagnostic attributes

The whole workflow connected with model approximation and diagnostic attributes was originally described in work [25]. Here we just shortly call the main idea. This approach focuses on building a surrogate model for origin model predictions using the rough sets theory [26]. Based on discretized input data set we construct the ensemble of approximate reducts. The next step is to create a neighborhood for every instance in the diagnosed data set as a set of instances from the train data set that is similar to a given instance in the diagnosed data set. The defined neighborhood is a basis for calculating the diagnostic attributes listed below.

- Target consistency with approximations in neighborhood — measuring the consistency of the target of the diagnosed instance with the approximations from the neighborhood of this instance.
- Prediction consistency with targets in the neighborhood — measuring the consistency of the prediction of the diagnosed instance with the targets from the neighborhood of this instance.
- Target consistency with targets in neighborhood — measuring the consistency of the target of the diagnosed instance with the targets from the neighborhood of this instance.
- Targets and approximations inconsistency in neighborhood — measuring the inconsistency of targets and approximations in the neighborhood of diagnosed instance.
- Targets diversity in the neighborhood — measuring the diversity of targets in the neighborhood of diagnosed instance in comparison to the diversity of targets calculated on the whole diagnosed data set.
- Approximations diversity in the neighborhood — measuring the diversity of approximations in the neighborhood of diagnosed instance in comparison to the diversity of approximations calculated on the whole diagnosed data set.
- Uncertainty — the measure of uncertainty of prediction based on the approximations.
- Neighborhood size — the number of instances in the neighborhood of diagnosed instance.

We used the Kolmogorov-Smirnov (KS) test [27] to compare the distribution of diagnostic attributes. Additionally, the Wilcox signed rank test [28] for paired two samples was conducted. The first test compares the distance between distributions while the second measure only changes in the location parameter.

IV. Experiments and results

To evaluate proposed diagnostic attributes in attack detection we prepare benchmark data sets. From OpenML[1] we gathered 22 data sets with classification task. Each data set was split into train and diagnosed parts assuming that the diagnosed data set should consist of at least 100 observations. A list of data sets is placed in the appendix in Table IV.

For each data set, we fitted a logistic regression model, support vector machine, and XGBoost. Afterward, three adversarial attacks were conducted at the diagnosed part of each data set.

Figure 1 shows the distribution of balanced accuracy measured at the diagnosed data set for the origin (base) model and how it changed after the given attack.

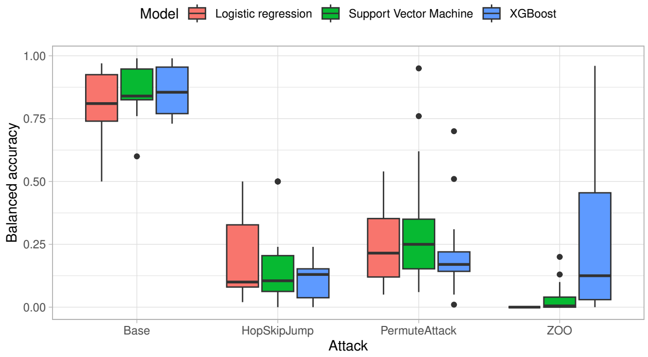


Fig. 1. Distribution of balanced accuracy in analyzed datasets across the type of model and attack

It can be seen that post-attack models in most cases result in worse performance than the base model. The median balanced accuracy for all base models is above 0.8 while in the case of the HopSkipJump attack, it is around 0.1. For Permute attack median value is slightly higher and equal to around 0.2. In the ZOO attack, these values are close to 0, but high dispersion of results for the XGBoost model can be observed.

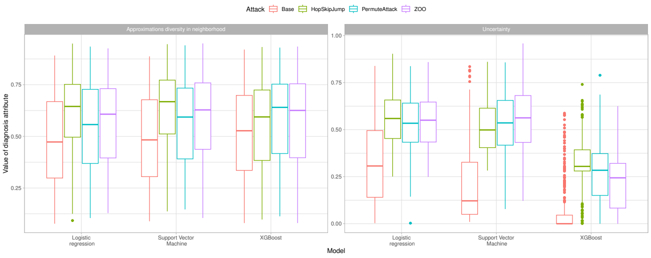


Fig. 2. Distribution of selected diagnostic attributes for the spambase data set

We calculated diagnostic attributes for each analyzed data set and attack, resulting in 264 tables with results (22 data sets × 3 model types × 4 attack variants (no attack + 3 others)). The distribution of two diagnostic attributes for the selected data set (spambase) is presented in figure 2. We used the Kolmogorov-Smirnov test to verify the null hypothesis that there is no difference between the distribution of the given diagnostic attribute before and after the attack. We also used the Wilcoxon test to examine the hypothesis that the median of differences between the paired attributes is zero. Both of these tests indicates whether there is a significant difference between diagnostic attribute before and after the attack. We summarize this data by calculating the fraction of cases in which the null hypothesis of a given statistical test was rejected at significance level $\alpha = 0.05$. Results are presented at three levels of aggregation — effectiveness of detecting attacks at the type of attack (table I), the type of model (table II), and diagnostic attribute (table III).

In the case of the HopSkipJump attack KS test rejected the null hypothesis in 88% cases while the Wilcox test in 95%.

[1] https://www.openml.org/

TABLE I
PERCENTAGE OF DETECTED DIFFERENCES BETWEEN DIAGNOSTIC ATTRIBUTES AT ATTACK LEVEL

Attack type	Kolmogorov-Smirnov	Wilcox
HopSkipJump	88.28	94.91
PermuteAttack	79.36	94.63
ZOO	81.25	94.83

The Permute attack and ZOO attack were slightly harder to detect — the effectiveness of the Kolmogorov-Smirnov test is 79% and 81% respectively. For the Wilcox test, values are close to 95%.

TABLE II
PERCENTAGE OF DETECTED DIFFERENCES BETWEEN DIAGNOSTIC ATTRIBUTES AT MODEL LEVEL

Model type	Kolmogorov-Smirnov	Wilcox
Logistic regression	84.47	92.59
Support Vector Machine	85.61	94.24
XGBoost	78.52	97.75

At the model type level, we detected 79% of attacks conducted on the XGBoost model, almost 84% on logistic regression, and 86% on SVM using the Kolmogorov-Smirnov test. With Wilcox test success rate is equal to 93% for Logistic regression, 94% for SVM, and 98% for the XGBoost model.

TABLE III
PERCENTAGE OF DETECTED DIFFERENCES BETWEEN DIAGNOSTIC ATTRIBUTES AT ATTRIBUTE LEVEL

Diagnostic attribute	Kolmogorov-Smirnov	Wilcox
Approximations diversity in neighborhood	87.76	98.96
Neighborhood size	72.45	96.35
Prediction consistency with targets in neighborhood	81.12	83.85
Target consistency with approximations in neighborhood	90.31	98.44
Target consistency with targets in neighborhood	87.24	95.83
Targets and approximations inconsistency in neighborhood	60.71	89.58
Targets diversity in neighborhood	88.27	96.35
Uncertainty	95.41	98.95

Another issue was the verification of diagnosis attributes effectiveness in attack detection. According to the Kolmogorov-Smirnov test, the highest detection rate was obtained for uncertainty (95%), target consistency with approximations in the neighborhood (90%), and targets diversity in the neighborhood (88%). In the case of the Wilcox test, we obtain similar high results for three attributes: uncertainty, approximations diversity in the neighborhood, and target consistency with approximations in the neighborhood.

V. CONCLUSIONS

In this paper, we have presented a novel approach to detecting adversarial attacks on machine learning classification models operating on tabular data. By analyzing differences in diagnostic parameters calculated on an approximated representation of the model under attack, we demonstrate that adversarial attacks can be detected in a statistically significant manner. Experiments performed on real-world datasets confirm the effectiveness of our method and its potential for further development as a detection technique for adversarial attacks.

A. Limitations

The method developed and presented in this paper has several limitations, which will be tackled in future works. Most notably:

- The robustness of the method to attack variability has not been subject to wider assessment — for the initial method validation a sample of three attacks was chosen, but it does not cover the range of currently known and published attacks on models designed for processing of tabular data.
- The method assumes that the model being monitored is replicable with a rough-sets-based method presented in previous work. In this paper, it was verified on three classification models, with the assumption that the underlying model replication method provides a layer of abstraction that is strong enough to consider our method model-agnostic. Verification of this hypothesis has not been a subject of this work.
- The computational efficiency optimization and scalability have not been, by design, within the scope o the work presented herein.
- The method has not been benchmarked against available AML detection techniques.

B. Future Work

Future work will be streamlined into three distinctive work streams.

First, we will broaden the range of scenarios on which the method is tested. The method will be verified on a larger representation of known attack methods and exploration of their attack parameter space. Special attention will be given to methods that attempt low-profile adversarial attacks, attempting to pass under the detection threshold of traditional monitoring tools. We also plan to compare our approach with other methods which aim to detect AML. Furthermore, we will verify the assumption of the method being model-agnostic, by checking how its effectiveness changes when used on a different original model being attacked.

The second workstream will be devoted to new features of the method:

- Concept drift detection - examining differences in diagnostic attributes behavior between changes in data resulting from malicious attacks and different types of concept drifts - both stochastic and deterministic in nature
- Exploration of possibility for new diagnostic attributes definition. Specifically - looking for diagnostic attributes that increase specificity and sensitivity of attack detection heuristics

In the third work stream, we intend to analyze the scalability of the method and prepare a thorough comparison of the presented attack detection method with available alternative adversarial attack detection methods.

We will also consider extending the set of diagnostic attributes with information obtained on the basis of approximation of diagnosed models with white-box models (e.g. rule-based models [29])

References

[1] M. Kozielski, M. Sikora, and Ł. Wróbel, "Disesor-decision support system for mining industry," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2015, pp. 67–74.

[2] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015. doi: 10.1126/science.aaa8415. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aaa8415

[3] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018. doi: 10.1109/ACCESS.2018.2807385

[4] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," pp. 506–519, 04 2017. doi: 10.1145/3052973.3053009

[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," pp. 39–57, 05 2017. doi: 10.1109/SP.2017.49

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[7] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[8] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," p. 2154–2156, 2018. doi: 10.1145/3243734.3264418. [Online]. Available: https://doi.org/10.1145/3243734.3264418

[9] Z. Pawlak, *Rough sets: Theoretical aspects of reasoning about data*. Springer Science & Business Media, 1991.

[10] A. Skowron and L. Polkowski, *Rough sets in knowledge discovery 1: Basic concepts*. CRC Press, 1998.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations*, 2013.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[13] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 07 2016.

[14] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *CoRR*, vol. abs/1707.08945, 2017. [Online]. Available: http://arxiv.org/abs/1707.08945

[15] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020. doi: https://doi.org/10.1016/j.eng.2019.12.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209580991930503X

[16] K. Kireev, B. Kulynych, and C. Troncoso, "Adversarial robustness for tabular data through cost and utility awareness," in *NeurIPS ML Safety Workshop*, 2022. [Online]. Available: https://openreview.net/forum?id=3ieyhWF1Hk

[17] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," *arXiv preprint arXiv:1705.07263*, 2017.

[18] L. Li, X. Chen, Z. Bi, X. Xie, S. Deng, N. Zhang, C. Tan, M. Chen, and H. Chen, "Normal vs. adversarial: Salience-based analysis of adversarial samples for relation extraction," in *Proceedings of the 10th International Joint Conference on Knowledge Graphs*, ser. IJCKG '21. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3502223.3502237. ISBN 9781450395656 p. 115–120. [Online]. Available: https://doi.org/10.1145/3502223.3502237

[19] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "Detecting adversarial perturbations with neural networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[20] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. Mcdaniel, "On the (statistical) detection of adversarial examples," *ArXiv*, vol. abs/1702.06280, 2017.

[21] G. K. Santhanam and P. Grnarova, "Defending against adversarial attacks by leveraging an entire gan," 2018.

[22] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," *arXiv preprint arXiv:1904.02144*, 2019.

[23] M. Hashemi and A. Fathi, "Permuteattack: Counterfactual explanation of machine learning credit scorecards," 2020.

[24] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 15–26.

[25] A. Janusz, A. Zalewska, Łukasz Wawrowski, P. Biczyk, J. Ludziejewski, M. Sikora, and D. Ślęzak, "Brightbox—a rough set based technology for diagnosing mistakes of machine learning models," *Applied Soft Computing*, p. 110285, 2023. doi: https://doi.org/10.1016/j.asoc.2023.110285. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623003034

[26] A. Skowron and D. Ślęzak, "Rough Sets Turn 40: From Information Systems to Intelligent Systems," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30, 2022. doi: 10.15439/2022F310 pp. 23–34. [Online]. Available: https://doi.org/10.15439/2022F310

[27] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[28] R. C. Blair and J. J. Higgins, "Comparison of the power of the paired samples t test to that of wilcoxon's signed-ranks test under various population shapes." *Psychological Bulletin*, vol. 97, no. 1, p. 119, 1985.

[29] A. Gudyś, M. Sikora, and Ł. Wróbel, "Rulekit: A comprehensive suite for rule-based learning," *Knowledge-Based Systems*, vol. 194, p. 105480, 2020.

Appendix

TABLE IV
BASIC CHARACTERISTICS OF DATA SETS USED IN EXPERIMENTS. THE COLUMNS N, $|A|$, AND $|L|$ SHOW THE TOTAL NUMBER OF INSTANCES, ATTRIBUTES, AND CLASSES, RESPECTIVELY.

name	N	$\|A\|$	$\|L\|$
Bioresponse	3751	1776	2
churn	5000	20	2
cmc	2000	47	10
cnae-9	1080	856	9
dna	3186	180	3
har	10299	561	6
madelon	2600	500	2
mfeat-factors	2000	47	10
mfeat-fourier	2000	76	10
mfeat-karhunen	2000	47	10
mfeat-zernike	2000	47	10
nomao	34465	118	2
optdigits	2000	47	10
pendigits	10992	16	10
phoneme	5404	5	2
qsar-biodeg	1055	41	2
satimage	6430	36	6
semeion	1593	256	10
spambase	2000	47	10
wall-robot-navigation	5456	24	4
wdbc	569	30	2
wilt	4839	5	2

DOI: 10.15439/2023F3585
ISSN 2300-5963 ACSIS, Vol. 35

Change of Brand Management in Social Media During the Russia-Ukraine War: Findings from Poland

Magdalena Grzanka
University of Economics in Katowice
ul. 1 Maja 50, 40-287 Katowice, Poland
Email: magdalena.grzanka@edu.uekat.pl

Artur Strzelecki
University of Economics in Katowice
ul. 1 Maja 50, 40-287 Katowice, Poland
Email: artur.strzelecki@ue.katowice.pl

***Abstract*—The research aims to explore the effects of the war in Ukraine in 2022 on social media and brand management, as well as the ways in which companies and users engage with each other on social media platforms. The study examines the strategies used by companies to engage with users on social media during the war, the role of social media in shaping public perceptions and responses to the war, and the impact of social media engagement on companies' relationships with users and consumers. The study employs a survey research method to collect data from a sample of Polish individuals using structured questionnaires. The results of the study provide insight into the changes in social media and brand management in response to the war and the ways in which companies and users engaged with each other.**

I. Introduction

IN RECENT years, social media has become an increasingly important platform for brands to reach and engage with their customers. With the proliferation of social media networks and the widespread use of mobile devices, consumers are now able to access and interact with brands in real-time, anywhere, and anytime. This has led to a shift in the way that brands manage their online presence and reputation, as well as the way that they communicate and interact with their customers.

In 2022, the outbreak of the war in Ukraine caused a significant shift in social media and brand management. By the end of February 2022, companies abruptly altered the content they were posting on social media. [14]. Companies that had not previously shared their social or political views on social media were required to take a side in the conflict - either the side of Ukraine or the side of the Russian state, which initiated the war [19]. The lack of involvement of a brand in helping Ukraine was often criticized by social media users, who had previously been willing to buy products from the company. During the war in Ukraine, social media users all over the world identified with the people affected by the war in Ukraine and demanded that brands withdraw from the Russian market.

In times of war, more and more social media users began to believe that companies should openly address social and political issues and share their own views on the actions of the Russian state. The start of the war in Ukraine forced many companies to change the way they conducted their social media activities [1]. The war conflict required companies to adopt a new strategy and change the way they communicated with social media users, where consumer views of a particular brand are most often expressed. Numerous companies decided to limit their social media activity and give up their standard Internet activity. Many of them showed immediate help to Ukraine, and expressed their support for the eastern community with posts published on their social media. Companies began to organize aid campaigns and money collections, which they informed about on their social media. Those who had not previously published posts on social media were now forced to do so in order to express their support for Ukraine and to inform about the actions taken in this regard. The start of the war in Ukraine led to a change in the way brands communicated with their customers and the way they conducted their social media activities [23].

According to Wirtualnemedia.pl, in the first days of the war conflict in Ukraine, about 900,000 posts appeared on social media calling for help for the Eastern neighbors. The majority of these publications appeared on the social networking site Facebook - about 43%, and on Twitter - about 38% [26]. With the start of the war, social media was dominated by posts related to help for the Ukraine. After the Russian attack on Ukraine, a significant decrease in the number of ads published on Facebook was observed - on February 28, 2022, the number of ads on this social networking site decreased by 73% compared to the previous highest result in that month [25]. At the end of February 2022, ads on social media were mainly suspended, but posts that had been planned several months earlier and had to be published by brands on the internet could be still observed. These were posts from companies that had already been contracted in the past and had to be completed according to the company's regulations. These posts included the results of organized contests, planned events, or webinars [15].

There were also companies that completely suspended their social media business during the war crisis. These actions mainly resulted from the concern of brands about the reactions of consumers to the company's further actions on the internet during the war in Ukraine. Some companies remained active on their social media platforms but chose to disable comments under their posts. Such behavior did not win the sympathy of consumers. These actions were perceived by social media users as insincere, and silence was treated as an attempt to mask the brand's opinion on the war in Ukraine. The situation changed at the end of March 2022, when companies began to publish posts on social media again, but they were mainly related to the company's involvement in activities supporting Ukraine. The war in Ukraine had a significant impact on the way brands communicated with their customers and changed their social media activities.

The war in Ukraine in 2022 had significant impacts on social media and brand management, as well as on how companies and users engaged with each other on social media platforms [21]. In this research, we will explore how social media and brand management changed in response to the war, how companies' social media posts and strategies evolved during this time, and how users' attitudes towards these efforts were impacted. Additionally, we will examine the role that social media played in shaping public perceptions and responses to the war and how companies' social media engagement during this time affected their relationships with users and consumers. The study relies on data from social media platforms such as Facebook, Instagram, and Twitter [13].

We have formulated the following research questions: 1. How did social media and brand management change in response to the war in Ukraine in 2022? 2. How did companies' social media posts change during the war in Ukraine in 2022? 3. How did users' attitudes toward companies' social media engagement change during the war in Ukraine in 2022?

II. Method

For this study, we used literature review and survey research. Survey research is a method of collecting information from a sample of individuals using structured questionnaires or interviews. In order to verify consumers' opinions about the actions taken by companies on social media during the war crisis in Ukraine, we decided to conduct a study among the Polish community. This study took the form of a questionnaire survey in Polish language and aimed to identify consumers' feelings and observations about daily brand image management compared to the one during the war crisis, which affected Ukrainian community in 2022. The survey questionnaire was prepared using Google Forms software, and made available from March to April 2022. The survey was fully anonymous, and anyone interested could fill it out. In order to gather as many responses as possible, we made the questionnaire available on social media, especially on themed groups on the Facebook social networking site. The aim of the survey was to conduct a public opinion review on the perception of companies on social media and the activities of companies on social media during the war in Ukraine.

The study was conducted in accordance with the Declaration of Helsinki and approved by the Faculty Research Ethics Committee of the University of Economics in Katowice, Poland; Approval code: 135890, and date: April 3, 2022. The informed consent statement of each participant was collected at the beginning of the survey. The statement was following: "By taking part in this study, you are agreeing to allow us to collect data about managing the brand image in social media during the war crisis in Ukraine. This data will be used to help us better understand brand management during this crisis and will be kept strictly confidential. You may withdraw from the study at any time by contacting us."

In the conducted survey, 150 respondents participated. The questionnaire was filled out by both women and men. The majority of the surveyed were women, 114 of them, which is 76% of the people. The questionnaire was published on social media platforms, which influenced a large percentage of the surveyed being young - they were mostly school or university students. The number of respondents aged under 18 and over 56 was minimal. The largest number of people who participated in the survey were those aged 18 to 26 - there were 85.3% of them, 128 respondents. The next largest group of people participating in the survey were those in the age range of 27-35 (16 respondents). Three people aged 36-55, two people under 18 - 1.3%, and one person over 56.

III. Results

A. Brands' activities in social media during the Ukraine war crisis

The onset of the war crisis in Ukraine changed the way brands were managed on social media. In the early days of the war, companies were constantly seeking new ways to build their brand image on social media. They tried to help their eastern neighbors and kept the public informed about it on social media. Companies chose different ways to support Ukraine. Brands supported refugees, organized financial and material collections, fought against disinformation, and used social media profiles as a place to publish reliable information from around the world. Some companies also decided to withdraw Russian products from their own range, suspend production, or ultimately cease operations in Russia [20]. As the war in Ukraine began, companies from various industries began to engage in assistance for their eastern neighbors. Brands on social media platforms called for support for refugees and documented their efforts to help Ukraine. Similar strategies were adopted by well-known celebrities and influencers, who were often associated with a particular brand as a result of a campaign or promotion of goods. While helping their eastern neighbors, companies did not forget about keeping the good reputation of their brand [22]. An important step during the war crisis in Ukraine was the fight against disinformation. Many companies decided to run a certain type of news service on their social media platforms [5]. This activity was intended to eliminate fake news and reduce panic among people [16]. A new service called VPolshchi.pl was created by Virtual Poland in response to the onset of the war crisis in Ukraine [24]. The VPolshchi.pl service was intended to correct false information. VPolshchi.pl features current information on the military actions being carried out in Ukraine and the most important news related to the ongoing war. The news in this service are conveyed in Ukrainian and are intended to be helpful to the Ukrainian community. This news focuse on delivering accurate information about the current situation in Ukraine and informing about the organized aid efforts [22].

During the war in Ukraine, a significant number of refugees sought shelter in various countries, including Poland. Both individuals and businesses extended their support to these refugees through financial and material assistance. Companies organized campaigns to collect funds, food, clothing, medical supplies, hygiene products, and children's accessories for their eastern neighbors. Special hashtags and discount codes were created, where the use of such codes resulted in a specified amount of money being donated for the benefit of aiding Ukraine. These campaigns effectively encouraged consumers to participate in helping the refugees. One notable example is Rossmann pharmacy, which actively supported its eastern neighbor since the war's onset. They organized an aid campaign by offering a special -40% coupon for selected products to those who wished to support Ukrainian refugees. The company emphasized that this campaign was not merely a promotion but aimed to raise awareness among consumers. The discount was provided to individuals who would donate the purchased products to the refugees. This coupon was valid until March 8, 2022, and covered various hygiene products and children's accessories. The coupon was exclusively available to users of Rossmann's mobile application and could only be used once. [10].

The coffee roaster KawePale also joined the aid action for Ukraine. A post appeared on its Instagram announcing the campaign organized by the brand. The company created a universal discount code for use in the company's online store [11]. The use of this discount code will contribute 15% of the sales of the company's ordered coffees to the Polish Humanitarian Action. The Polish Humanitarian Action supports both people in Ukraine during the ongoing war and refugees coming to Poland [18]. In order to help Ukraine, companies organized their own collections or transferred funds to the existing ones. Information about the ongoing campaigns was announced on the social media portals of companies and among employees of the network who had the opportunity to get involved in the assistance and show solidarity with Ukraine. Such activities were undertaken by one of the Polish banks - ING Bank Slaski SA. On their Facebook profile, the bank posted information about a fundraising campaign for the Ukrainian community in connection with the ongoing war [7]. In addition, the bank declared that it would not only transfer the collected funds to help Ukraine but also double the amount collected.

InPost, one of the main logistics and transportation operators in Poland, also decided to help the Ukrainian community. The company decided to use its resources to help with the delivery of products collected as part of the aid campaigns organized throughout the country for Ukraine. The brand informed about its decision on the Facebook social media platform [8]. The logistics operator InPost not only organized product deliveries to the eastern border but also participated in numerous charitable collections supporting refugees from Ukraine. The company entered into cooperation with the Polish Red Cross. InPost helped the PCK in transportation of medical equipment, dressings, medicines, hygienic and medical supplies, as well as food products. Together with the Melissa brand, the company developed aid packages that were available for purchase through the InPost mobile application. Those willing to help the Ukrainian community purchased these packages, and the company delivered the goods to those in need in Ukraine. This transport was free of charge and was intended to support the Ukrainian community. In addition, in order to support refugees' lives outside their homeland, the InPost brand created a Ukrainian language version of the mobile application - InPost Mobile [9]. Communication service providers also supported people from Ukraine, initiating the creation of free starters for people from Ukraine. One of the mobile operators that supported Ukraine was the Plus Poland company [22]. The company announced on Twitter information about the organized action related to free starters for every person from Ukraine. This starter included a free package of 500 minutes and 10 GB for 30 days and required the person interested to show their residence card or passport at the sales point [17].

During the Ukraine war crisis, a group of companies emerged, that, after about two months of the war, still did not declare the withdrawal of their brand from the Russian market. This group included Leroy Merlin, Auchan, and Nestle. In April 2022, the opinions of internet users about these companies were critical. These brands were regularly boycotted on the internet, and previous customers of these businesses stopped purchasing products from their offerings [3]. The consequences of not leaving the Russian market, for example, affected the Leroy Merlin brand. Internet users were calling for a continuous boycott of this company on the internet, and social media users formulated a special hashtag #boycottleroymerlin, calling for the cessation of using the services offered by this company. Newer graphics appearing on the internet showed the disapproval of social media users. The Russian market was not abandoned by the Auchan hypermarket chain either. As a result of the companies' approach to the events taking place in Ukraine, a petition was issued by the National Boycott of Leroy Merlin, calling for the dismissal of the Polish management of the Leroy Merlin and Auchan chains [4]. Internet users described the attitude of these companies as cowardly and unworthy of imitation. Strikes were being organized under the stationary stores of the companies, calling for help for the Ukrainian community and the withdrawal of brands from the Russian market. Similarly to the company Leroy Merlin, the company Auchan tried to alleviate the tense situation it is facing. On March 11, 2022, the Polish branch of the company informed the social media site Facebook about the assistance organized for the Ukrainian community and declared that it has no influence on the decision of the parent company regarding the conduct of business in the Russian Federation [2]. In view of the expression of solidarity with

Ukraine, the company hoped for a gradual easing of the consumer boycott. The effect of the message shared online was the opposite, and about twelve thousand negative reactions and comments directed at the Leroy Merlin company were recorded under the published post [13].

B. Survey study group

The respondents were asked about their attention to businesses' online activity during the war crisis in Ukraine and its impact on brand posts on social media. The survey items are presented in tables I to V. Out of the total number of respondents, 63 indicated paying attention to companies' posts during the war, with 36 showing significant interest. On the other hand, 31 respondents stated not paying attention to these posts, and one person explicitly ignored business posts during the war. Additionally, 19 respondents were uncertain and chose the "Difficult to say" response (Table I).

The respondents observed both the actions taken by companies on social media during the war crisis in Ukraine and the changes in their posts. Out of the respondents, 63 rather and 36 definitely noticed a difference in the behavior of brands on social media during the war. Only 32 respondents did not perceive any changes in the companies' social media posts following Russia's aggression against Ukraine. Additionally, 19 respondents found it challenging to provide a clear answer to this question (Table I).

Furthermore, the war in Ukraine led to the emergence of numerous negative consumer opinions about specific brands on the internet. Among the surveyed respondents, 104 individuals confirmed seeing criticism directed at companies on social media in relation to the ongoing war in Ukraine. Conversely, 46 respondents did not notice any negative opinions about companies on the internet concerning the war's impact on the Ukrainian community (Table II).

During the war in Ukraine, respondents noticed negative comments about companies on social media, but they rarely shared their own opinions or comments under brand posts. Only 14 respondents confirmed commenting on content posted by companies on social media regarding the war in Ukraine. On the other hand, 136 respondents stated that they did not express their views on the internet about the actions of companies during the war crisis in Ukraine (Table II).The respondents were also asked to provide their opinions on companies' social media activity during the war in Ukraine. A significant number of respondents (98) believed that brands should not engage in standard activity, while 21 had different views, and 31 marked the response as "Difficult to say" (Table III).

The respondents unanimously agreed that businesses should deliver continuous updates on social media regarding their actions in response to the ongoing war in Ukraine, including providing material and financial aid to refugees. This view was shared by the majority, with 97 respondents expressing this opinion. Additionally, 108 respondents believed that businesses should show support for Ukraine on social media, while only 12 disagreed with this statement (Table III).

The surveyed individuals also believed that businesses should inform on social media about the withdrawal of Russian products from their own offerings - 123 responses of this type were given. According to the respondents, it is also important for companies to involve their brand's consumers in actively helping Ukraine and its refugees. Companies should also pay more attention to comments made under posts by brands on social media, especially during the ongoing crisis in Ukraine (111 responses). As claimed by the respondents, businesses should feel obligated to help the Ukrainian community and appeal for help among the followers gathered on social media networks (96 responses) (Table IV).

The survey respondents showed a keen interest in the support demonstrated by companies on social media for Ukraine. Out of the respondents, 104 individuals expressed their concern and engagement with this issue. The respondents were aligned in their demand for businesses to provide assistance to Ukraine, but not all respondents were discouraged from purchasing products from brands that did not take decisive actions regarding the war in Ukraine. The responses were divided regarding the question of purchasing products from brands that lacked solidarity with Ukraine. Fifty-eight respondents indicated that they would still purchase products from such brands, while 51 respondents held the opposite opinion. However, the responses were almost evenly distributed regarding consumer reluctance to purchase products from companies that did not criticize Russia's aggression against Ukraine. For this question, 53 respondents answered affirmatively, while 72 respondents had a negative response. Furthermore, a decisive majority of the respondents expressed their intention to continue following the social media profiles of brands that do not show support for Ukraine, with 100 respondents confirming this. However, 28 respondents found it challenging to provide a definitive answer, and 22 respondents were prepared to stop following such businesses on social media (Table V).

IV. Discussion

The study addressed issues related to the change in brand image management due to the war crisis in Ukraine and the new reality that modern businesses had to find themselves in. The study on the activity of businesses and brand image management in social media was based on a developed questionnaire survey that was made available on social media portals. One hundred fifty people participated in the study, expressing their own opinions on social media and their impact on the brand image, even in the face of the war crisis in Ukraine. The analysis focused on the behavior and views of modern consumers in relation to the actions taken by companies in social media.

During the war crisis in Ukraine, businesses sought to provide assistance to their eastern neighbors and ensured support for refugees both materially and financially. The responses confirmed that consumers pay close attention to the activities of companies in social media, and these actions significantly impact the brand perception of customers. The surveyed individuals base their opinions on the products

TABLE I
RESPONDENTS' ASSESSMENT OF POSTS BY BRANDS IN SOCIAL MEDIA DURING THE WAR IN UKRAINE

Statement	**Answer**				
	No	Rather not	Difficult to say	Rather yes	Yes
I pay attention to the actions taken by companies in social media during the war in Ukraine	0.6%	20.7%	12.7%	42%	24%
I notice a change in the posts published by brands on social networks as a result of the war in Ukraine	1,3%	18.7%	14%	40.6%	25.4%

TABLE II
OPINIONS OF NETWORK USERS ABOUT COMPANIES IN SOCIAL MEDIA DURING THE WAR CRISIS IN UKRAINE

Statement	Answer	Count
I notice negative opinions about companies appearing in social media during the time of war crisis in Ukraine	Yes	69.3%
	No	30.7%
I comment on content posted by companies' relationship on social media with the war in Ukraine	Yes	9.3%
	No	90.7%

TABLE III
RESPONDENTS' ASSESSMENT OF THE STATEMENTS ABOUT THE COMPANIES IN SOCIAL MEDIA AND THE WAR IN UKRAINE

Statement	Answer				
	No	Rather not	Difficult to say	Rather yes	Yes
During the war in Ukraine, companies should not conduct standard activity in social media	34.7%	30.7%	20.6%	10%	4%
Companies should inform via social media about the current actions of the brand taken in the era of the war crisis	4.6%	12.7%	18%	40%	24.7%
Businesses should show support for Ukraine on social media	2.7%	5.3%	20%	37.3%	34.7%

TABLE IV
RESPONDENTS' ASSESSMENT OF THE STATEMENTS REGARDING THE ACTIONS OF BRANDS DURING THE WAR IN UKRAINE

Statement	Answer				
	No	Rather not	Difficult to say	Rather yes	Yes
Companies should inform on social networks about the withdrawal of Russian products from its own offer	2.7%	4.7%	10.6%	34.7%	47.3%
Companies should pay attention to comments posted by followers on social networks during the war crisis in Ukraine	4%	6%	16%	35.3%	38.7%
Companies should involve followers in aid for Ukraine	5.3%	6.7%	24%	40.6%	23.4%

offered by brands, and the reputation of the companies - on the opinions of other users. In addition, due to Russia's aggression towards Ukraine, respondents have experienced a significant change in the themes of posts published in social media and have encountered more negative comments directed toward contemporary businesses. These negative comments are mainly directed toward companies that have not yet decided to withdraw their brand from the Russian market and have not withdrawn Russian products from their own offerings (as of the date of the survey). Respondents confirmed that, in their opinion, businesses should not conduct their usual activities in social media during the ongoing war crisis in Ukraine, and some of those surveyed replied that they would refrain from purchasing products from a company that is not supportive of Ukraine.

We have found the following answers to the research questions. The war in Ukraine in 2022 had a significant impact on social media and brand management, as companies were required to take a side in the conflict and express their views on social and political issues. This led to a change in the way that companies conducted their social media activities and communicated with their customers (Research question 1). During the war in Ukraine in 2022, companies' social media posts changed as they were required to take a side in the conflict and express their views on social and political issues. Many companies also limited their social media activity or changed the way they communicated with their customers in order to show their support for Ukraine (Research question 2). Users' attitudes towards companies' social media engagement changed during the war in Ukraine in 2022, as they began to expect companies to openly address social and political issues and share their views on the actions of the Russian state. Social media users also demanded that brands withdraw from the Russian market and show their support for Ukraine through their engagement with companies on social media (Research question 3).

There are several limitations to consider for the conducted research. First is the time period. The study focuses on events that occurred in 2022, which may not be representative of the current state of social media and brand management. The second is geographical scope. The study focuses on the impact of the war in Ukraine on social media and brand management, which may not be applicable to other countries or regions. The third is the causal relationship. The study suggests that the

TABLE V
RESPONDENTS' ASSESSMENT OF THE STATEMENTS ABOUT THE WAR IN UKRAINE AND THE ACTIVITIES OF BRANDS ONLINE

Statement	Answer				
	No	Rather not	Difficult to say	Rather yes	Yes
I pay attention to the support shown in social media by companies for Ukraine	8.7%	11.3%	10.7%	40.6%	28.7%
No firm action by the company in the relationship with the conflict in Ukraine discourages the purchase of products of a given brand	20%	28%	16.7%	20%	15.3%
I will resign from following the profile of a company in social media that does not show support for Ukraine	35.4%	31.3%	16.7%	9.3%	5.3%
I will resign from buying products of a specific brand that is not in solidarity with Ukraine	16.7%	22%	27.3%	22%	12%

start of the war in Ukraine caused changes in the way brands managed their social media presence, but it is not clear if this is the only factor that contributed to these changes. Other factors, such as changes in consumer behavior or the adoption of new social media platforms, may also have played their role. The forth is data sources. The study relies on data from social media platforms such as Facebook, Instagram, and Twitter, which may not be representative of the entire population or all social media activity.

REFERENCES

[1] W. Abbassi, V. Kumari, and D. Pandey, "What makes firms vulnerable to the Russia–Ukraine crisis?," *J. Risk Financ.*, vol. 7, 2022. https://doi.org/10.1108/JRF-05-2022-0108

[2] AdMonkey, "Internauci wzywają do bojkotu firm, które nie wycofały się Rosji. Najczęściej w tym kontekście pojawia się Leroy Merlin," 2022. https://admonkey.pl/internauci-wzywaja-do-bojkotu-firm-ktore-nie-wycofaly-sie-rosji-najczesciej-w-tym-kontekscie-pojawia-sie-leroy-merlin-analiza-sentione/

[3] M. Badowski, "Bojkot firm, które nie wycofały się z Rosji. W tym kontekście najczęściej pojawia się market budowlany Leroy Merlin," 2022. https://strefabiznesu.pl/bojkot-firm-ktore-nie-wycofaly-sie-z-rosji-w-tym-kontekscie-najczesciej-pojawia-sie-market-budowlany-leroy-merlin/ar/c3-16240353

[4] Dlahandlu.pl, "Aktywiści domagają się dymisji polskich zarządów Leroy Merlin i Auchan," 2022. https://www.dlahandlu.pl/detal-hurt/wiadomosci/aktywisci-domagaja-sie-dymisji-polskich-zarzadow-leroy-merlin-i-auchan,107660.html

[5] Y. Golovchenko, "Fighting Propaganda with Censorship: A Study of the Ukrainian Ban on Russian Social Media," *J. Polit.*, vol. 84, no. 2, pp. 639–654, Apr. 2022. https://doi.org/10.1086/716949

[6] IKEA, "The war in Ukraine has both a huge human impact and is resulting in serious disruptions to supply chain and," 2022. https://twitter.com/IKEA/status/1499338854301642754

[7] ING Polska, "Wasza pomoc i zaangażowanie chwytają za serce! Zebraliście już ponad 5 mln zł na pomoc Ukrainie. My – zgodnie z obietnicą," 2022. https://www.facebook.com/INGPolska/

[8] InPost, "Wykorzystując nasze zaplecze logistyczne i flotę transportową pomagamy dostarczyć duże ilości produktów zebranych w ramach akcji i zbiórek organizowanych na," 2022. https://www.facebook.com/paczkomatkurier/posts/10159671676527999

[9] InPost.pl, "Pomoc dla Ukrainy – transport produktów samochodami InPost," 2022. https://inpost.pl/aktualnosci-pomoc-dla-ukrainy-transport-produktow-samochodami-inpost

[10] J. Jankowski, "Kupon -40 proc. dla osób, które chcą wesprzeć uchodźców z Ukrainy," 2022. https://www.rossmann.pl/firma/pl-pl/biuro-prasowe/informacja/kupon-40-proc-dla-osob-ktore-chca-wesprzec-uchodzcow-z-ukrainy,729051

[11] Kawepale, "Pomoc dla Ukrainy! Za naszą granicą rozgrywa się koszmar, którego boi się każdy z nas. Dlatego szczególnie teraz jest niezwykle," 2022. https://www.instagram.com/p/CbIJXqCod-A/

[12] M. Lenartowicz and A. Strzelecki, "Moderate Effect of Satisfaction on Intention to Follow Business Profiles on Instagram," *Int. J. Mark. Commun. New Media*, vol. 9, no. 16, 2021, pp. 4-24

[13] Leroy Merlin Polska, "Drodzy, jako Leroy Merlin Polska mamy wpływ na to co robimy w Polsce, dlatego nasze działania koncentrują się na pomocy," 2022. https://www.facebook.com/LeroyMerlinPolska/posts/4958165310968473/

[14] W. M. Lim *et al.*, "What is at stake in a war? A prospective evaluation of the Ukraine and Russia conflict for business and society," *Glob. Bus. Organ. Excell.*, vol. 41, no. 6, pp. 23–36, Sep. 2022. https://doi.org/10.1002/joe.22162

[15] A. Małkowska-Szozda, "Z powodu wojny firmy ograniczyły lub zmieniły aktywność w social mediach," 2022. https://www.press.pl/tresc/69789,z-powodu-wojny-firmy-ograniczyly-lub-zmienily-aktywnosc-w-social-mediach

[16] P. Majerczak and A. Strzelecki, "Trust, Media Credibility, Social Ties, and the Intention to Share towards Information Verification in an Age of Fake News," *Behav. Sci.*, vol. 12, no. 2, 2022, 51. https://doi.org/10.3390/bs12020051

[17] Plus Polska, "#Plus poszerza wsparcie. Darmowy starter dla każdego, kto rejestruje się w punkcie sprzedaży na ukraiński paszport lub kartę pobytu," 2022. https://twitter.com/Plus_Polska/status/1498999195252006919

[18] Polska Akcja Humanitarna, "UKRAINA. Pomóżmy ludziom w strefie konfliktu," 2022. https://www.pah.org.pl/sos-ukraina/

[19] V. Ratten, "The Ukraine/Russia conflict: Geopolitical and international business strategies," *Thunderbird Int. Bus. Rev.*, Nov. 2022. https://doi.org/10.1002/tie.22319

[20] D. P. Sakas *et al.*, "Social Media Strategy Processes for Centralized Payment Network Firms after a War Crisis Outset," *Processes*, vol. 10, no. 10, pp. 1995, Oct. 2022. https://doi.org/10.3390/pr10101995

[21] B. Smart *et al.*, "#IStandWithPutin Versus #IStandWithUkraine: The Interaction of Bots and Humans in Discussion of the Russia/Ukraine War," in *Soc. Informatics*, 2022, pp. 34–53. https://doi.org/10.1007/978-3-031-19097-1_3

[22] Ł. Sułkowski *et al.*, "Perception of patriotic entrepreneurship in Poland and Ukraine," *Entrep. Bus. Econ. Rev.*, vol. 10, no. 3, pp. 167–190, 2022. https://doi.org/10.15678/EBER.2022.100310

[23] F. O. Talabi *et al.*, "The use of social media storytelling for help-seeking and help-receiving among Nigerian refugees of the Ukraine–Russia war," *Telemat. Informatics*, vol. 71, pp. 101836, Jul. 2022. https://doi.org/10.1016/j.tele.2022.101836

[24] Wirtualna Polska, "Ruszył nowy serwis WP przygotowany dla społeczności ukraińskiej VPolshchi.pl, czyli 'w Polsce', w języku ukraińskim publikuje najważniejsze, wiarygodne informacje dotyczące," 2022. https://www.wirtualnemedia.pl/artykul/z-powodu-wojny-firmy-ograniczyly-lub-zmienily-aktywnosc-w-social-mediach

[25] Wirtualnemedia.pl, "Netto, Rossmann, Stokrotka, Polomarket i Topaz wycofują rosyjskie produkty ze swoich sklepów," 2022. https://www.wirtualnemedia.pl/artykul/bojkot-rosyjskich-produktow-sklepy-wycofuja-netto-rossmann-stokrotka-polomarket-topaz

[26] Wirtualnemedia.pl, "Prawie 900 tysięcy postów w social mediach o pomocy Ukrainie," 2022. https://www.wirtualnemedia.pl/artykul/prawie-900-tysiecy-postow-w-social-mediach-o-pomocy-ukrainie

DOI: 10.15439/2023F1856
ISSN 2300-5963 ACSIS, Vol. 35

Continual learning of a time series model using a mixture of HMMs with application to the IoT fuel sensor verification

Przemysław Głomb*, Michał Cholewa*, Paweł Foszner† and Jakub Bularz‡
*Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Bałtycka 5, 44-100 Gliwice, Poland, {pglomb,mcholewa}@iitis.pl
†Department of Computer Graphics, Vision and Digital Systems
Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology
Akademicka 2A, 44-100 Gliwice, Poland, pawel.foszner@polsl.pl
‡AIUT Sp. z o.o. ul. Wyczółkowskiego 113, 44-109 Gliwice, Poland

***Abstract*—This paper presents an application of a mixture of Hidden Markov Models (HMMs) as a tool for verification of IoT fuel sensors. The IoT fuel sensors report the level of fuel in tanks of a petrol station, and are a key component for monitoring system reliability (billing), safety (fuel/oil leak detection) and security (theft prevention). We propose an algorithm for learning a mixture of HMMs based on a continual learning principle, i.e. it adapts the model while monitoring a sensor over time, signalling unexpected or anomalous sensor reports. We have tested the proposed approach on a real-life data of 15 fuel tanks being monitored with the FuelPrime system, where it has shown a very good performance (average area under ROC curve of 0.94) of detecting anomalies in the sensor data. Additionally we show that the proposed method can be used for trend monitoring and present qualitative analysis of the short and long term learning performance. The proposed method has promising performance score, the resulting model has a high degree of explainability, limited memory and computation requirements and can be easily generalized to other domains of sensor verification.**

I. Introduction

A Key element of a recent change in the industry – dubbed 'Industry 4.0' or 'The Fourth Industrial Revolution' – is the proliferation of 'smart', connected Internet of Things (IoT) sensors, which have an ever increasing role in process monitoring, and as such require verification to achieve reliable, safe and secure systems [1]. In case of fuel tank sensors, which measure the state of large tanks e.g. within a petrol station, lack of sensor verification leads to problems being not detected, which results in reliability issues (billing errors) but can also have serious consequences for safety (not detecting a leak and subsequent environmental contamination) and security (facilitate theft of the fuel).

A sensor verification, or validation, is an internal, external or combined process to detect sensor faults and prevent control failures [2]. In case of IoT sensors, this consists of tasks like preparing models for denoising and missing data imputation, anomaly outlier detection, accuracy and semantic analysis [1]. The nature of the analysis can be complex, e.g. for subsequence outliers (set of consecutive data points that jointly behave unusually) [3]. On the other hand, sequential accumulation of data provides opportunity of continual learning of the properties of sensor behaviour over time; the challenge is to maintain learning ability without forgetting previously learned patterns [4]. Sensor verification method thus should be effective at its tasks [3], but at the same time provide adequate, explainable diagnostic data for the operator or maintenance engineer [2], support data processing within the technical constraints of the IoT sensor suite [1] while correctly dealing with new incoming data [4]. Recently, there has been an emphasis on proposing a approaches both explainable and comprehensive, which are able to deal with real world data, e.g. mobile networks scenarios [5] or oil well monitoring [6].

A large number of approaches have been proposed for application-oriented modelling of time series [7] and quality control [8], [9], including statistical tests, decomposition methods, autoregressive models, neural networks, and probabilistic models. Among them, the Hidden Markov Models (HMMs) [10], [11], [12] have proven to be versatile and effective across many fields, including fault diagnosis [13], [14]. HMM models are attractive, as they have three attractive properties: effective among many application fields; popular, thus well-studied; explainable, as their decisions can be easily traced to parameters of the underlying model. Original single-model HMM formulation have been extended with mixture or ensemble HMM models [15], [16], [17], [18], to further improve their modelling capability. However, while in continual learning setting incrementally learned HMMs can be almost as good as batch learned models [19], corruption of previously learned patterns is one of the main issue [20].

This paper proposes a mathematical model based on a mixture of Hidden Markov Models (HMMs), to be used as a tool for verification of IoT fuel sensors, together with

This work was partially funded by National Center for Research and Development no POIR.01.01.01-00-0376/17-00 'A system for gathering and analysis of streaming data for fuel stations, for optimization of distribution costs and fuel sales as well as on-line monitoring for leakages-related issues' (FuelPrime).

experiments documenting its performance. Our proposition leads to essentially nonparametric, lightweight (in terms of required computational resources, especially memory), continually learning modelling approach that is able to provide a verification of a sensor data series through the detection of structural changes, outliers and anomalies. Additionally, we present a case study, or experience report, of running the proposed approach through real-life historic data of 15 fuel tanks, with verification of detected patterns; the proposed method achieves a high average area under ROC curve of 0.94. While our study is, for the sake of clarity of presentation, limited to the case of IoT fuel sensor verification, the method can be easily generalized to other domains of sensor verification.

Our approach falls within the task-agnostic category described in [21], as we assume unknown both task boundaries – in our case changes in fuel tank usage characteristics over time – and task labels – in our cases the classes of anomalies to be detected. We note that this is the most general, and hence most desired in a practical application setting. Our learning setting from the point of view of anomaly detection is unsupervised, as all data is used without explicit consideration of the labels [22]. According to the taxonomy given in [23], our approach falls into the *Task-Free Continual Learning* (TFCL), with disjoint data label spaces and no task identities provided.

II. Method

A. Hidden Markov Models

A Hidden Markov Model is a model of a system that at any time is in one of n distinct states. At discrete time intervals, state switching occurs in time independent, first order Markovian dynamics (i.e. depends only on the current state). HMM states are not directly observable, however each state has an associated set of parameters describing the emission probability of observable symbols. For the fuel tank monitoring, the observed sequence is the volume of the fuel or its delta, while states correspond to the current situation (idle, refill from a tanker, distribution of fuel to clients, etc.).

A HMM λ of n states is described by initial state probability vector $\Pi = \left[\pi_i\right]_{n\times 1}$, state transition probability $A = \left[a_{ij}\right]_{n\times n}$, emission probability – typically Gaussian, with mean and standard deviation μ_i, σ_i defined for each state. Three main algorithms – Forward, Vitterbi, Baum-Welch – provide tools for finding a probability of a sequence for a given model, state sequence for given data, and learning a model from data [10]. Other parameter identification schemes are possible [24]. In this work, all HMMs are ergodic, i.e. transition from a state to all other states is possible at the start of the training.

B. Mixture of HMMs

Mixed (hidden) Markov models were originally introduced for latent class models [15] in social sciences, and further adapted e.g. for accelerometer measurements [16] or data imputation [17]. Those models introduce hierarchical structure, with class membership dictating Markov model parameters. A different approach has been proposed in [18], where an ensemble of HMMs is generated over time through incremental Boolean combination in the receiver operating characteristic (ROC) space.

In contrast to the above-mentioned propositions, we use a different approach. Our physical sensor model does not require latent class modelling, and absence of labels prevents from using ROC-based verification within the model operation. Our objective with using mixtures is to capture rare historical data patterns, and thus prevent them from being subject to catastrophic forgetting [4]. Our proposition is to model a IoT sensor time sequence with a set of m HMM models $\mathcal{H} = \{\lambda_1, \ldots, \lambda_m\}$. We assume that the sensor data $\mathbf{x} \in \mathbb{R}^d$ is processed in windows or batches (e.g. a day's worth of data, d can vary between windows). We propose the following algorithm for mixed HMM sensor verification:

1) Initialize $\mathcal{H} = \emptyset$.
2) Read next window of sensor data $\mathbf{x}$. Use the Baum-Welch algorithm to identify parameters of a model for this data, $\lambda_{\mathbf{x}}$, for a predefined range of number of states (see Section II-C).
3) If $\mathcal{H}$ is empty, then $\mathcal{H} = \{\lambda_{\mathbf{x}}\}$ and goto 2. Otherwise use the Forward algorithm to compute probabilities $P(\mathbf{x}|\lambda)$ and $N(\lambda)$ function to compute numbers of free parameters of the models
$$p_{\mathbf{x}} = P(\mathbf{x}|\lambda_{\mathbf{x}}) \quad n_{\mathbf{x}} = N(\lambda_{\mathbf{x}}) \tag{1}$$
$$p_{\mathcal{H}} = \max_{\lambda\in\mathcal{H}} P(\mathbf{x}|\lambda) \quad n_{\mathcal{H}} = \sum_{\lambda\in\mathcal{H}} N(\lambda) \tag{2}$$
4) Compute the information criteria values (e.g. AIC or BIC) for two cases: (C1) extending the current set of models with $\lambda_{\mathbf{x}}$ – possibly with better likelihood, but at the cost of expanding the total number of parameters – and (C2) staying with previous contents of the $\mathcal{H}$:
$$b_{\mathcal{H}+\lambda_{\mathbf{x}}} = IC(p_{\mathbf{x}}, n_{\mathcal{H}} + n_{\mathbf{x}}) \quad b_{\mathcal{H}} = IC(p_{\mathcal{H}}, n_{\mathcal{H}}) \tag{3}$$
5) If $b_{\mathcal{H}+\lambda_{\mathbf{x}}} < b_{\mathcal{H}}$ add the new model for the next iteration $\mathcal{H}' = \mathcal{H} \cup \{\lambda_{\mathbf{x}}\}$; if not, leave models as they were $\mathcal{H}' = \mathcal{H}$.
6) Regardless of decision in 5, calculate the anomaly score as $a_{\mathbf{x}} = p_{\mathbf{x}} - p_{\mathcal{H}}$. If there are remaining sequences, goto 2, otherwise stop.

The motivation for the algorithm presented above is as follows. While a HMM model has a very good performance in modelling time series, building a model of a long (monthly, yearly) series would require frequent, expensive re-training on a very long data history. For our case, initial observation of the data exposed dominant cycle or seasonality, in a similar way it is seen in energy consumption, water usage or weather patterns. Hence it's natural to treat the signal as a collection of cycle periods, in our case days, keeping in memory only an ensemble of HMM models, as the memory cost of a HMM model is much smaller than daily data sequence (see Section IV-A). Changes in the ensemble of HMMs occur between daily batches of data.

The proposed algorithm balances model complexity (number of parameters) and the ability to describe the signal (data fit). The collection of models retained on algorithm's progress over individual cycles serves additionally as a signal descriptor and source of diagnostic information.

C. Implementation of sensor verification

The verification system was incorporated into the fuel station tank monitoring system, which consists of the three main parts: the station part (implements software and hardware related to data acquisition, connects directly to devices and sends data to the central server); the server part (receives data from many stations, automatically processes this data and tries to draw and present preliminary conclusions [25]); the analytical part (responsible for analysing the results of the server part and for making decisions with human supervision).

For the actual verification, we focus on daily windows, as this corresponds to the rhythm of normal monitoring/verification applied in the system. A daily sequence of data $\mathbf{x}$ is fed into the system, and it's anomaly score $a_{\mathbf{x}}$ computed; if high enough, a 'require inspection' alert is generated. We note that there are additional possible ways to get information from the model, which are discussed in Section IV.

For each daily model identification (step 2), we use exhaustive search over a set of states $n \in \{1, \ldots, 10\}$, with Bayesian Information Criterion (BIC) [26] for selection of the final model. As the HMM identification algorithm (the Baum-Welch procedure) can end in local optimum, $k = 10$ independent searches for given n are performed, and the best model is evaluated. The BIC is also used for mixture extension decision ($b_{\mathcal{H}+\lambda_{\mathbf{x}}}$ and $b_{\mathcal{H}}$ values).

III. Results

A. Description of the sensor

Fuel and water level is measured by the Automatic Tank Gauging device (ATG). An ATG uses probes located in each tank or compartment to measure fuel and water levels. Each probe consists of a long rod with floats or sensors. The probe rod also has thermistors to measure the fuel temperature. The ATG sends an electrical impulse to both probes independently (product level float and water level float). The probe sends back the pulse and the ATG measures the time elapsed from sending to receiving. On this basis, the height is calculated. Measurements from all underground tanks are sent to a central unit located in the station building through wired or wireless connection. From here they are sent to the server (see Section II-C). The common risks with this type of device are: (1) suspension of the probe when it gets stuck at a certain height of the rod and (2) inertia of temperature measurements – especially important during the delivery of fuel with a significantly different temperature. Of the available data, we use fuel (product) level readout, which contains rich data about the tank situation; the remaining two (temperature and water) are used to diagnose specific, known problem conditions.

B. Selection of test cases

To test the method, data from $n_t = 15$ fuel tanks that have been previously known to have malfunctions and issues were selected for analysis. Both short and long term history sequences were selected, mean sequence length was 183 days ($8 - 646$ days) while mean sample count was $\approx 591\,189$ ($30\,151 - 2\,531\,218$ samples). The sequences were annotated by experts (analytical team in charge of verification of the sensors), with a list of days with erroneous or anomalous readings. There are two types of outliers in the data: (1) related to real probe disturbances (e.g. when the float hangs on the rod and does not represent the height of the liquid accurately; or in the middle of a delivery and there are significant fuel fluctuations causing the float to sink temporarily) and (2) virtual errors (incorrect translation of the pulse length to the real height, occurs when raw current measurement values are mismeasured or misinterpreted).

C. Experimental procedure

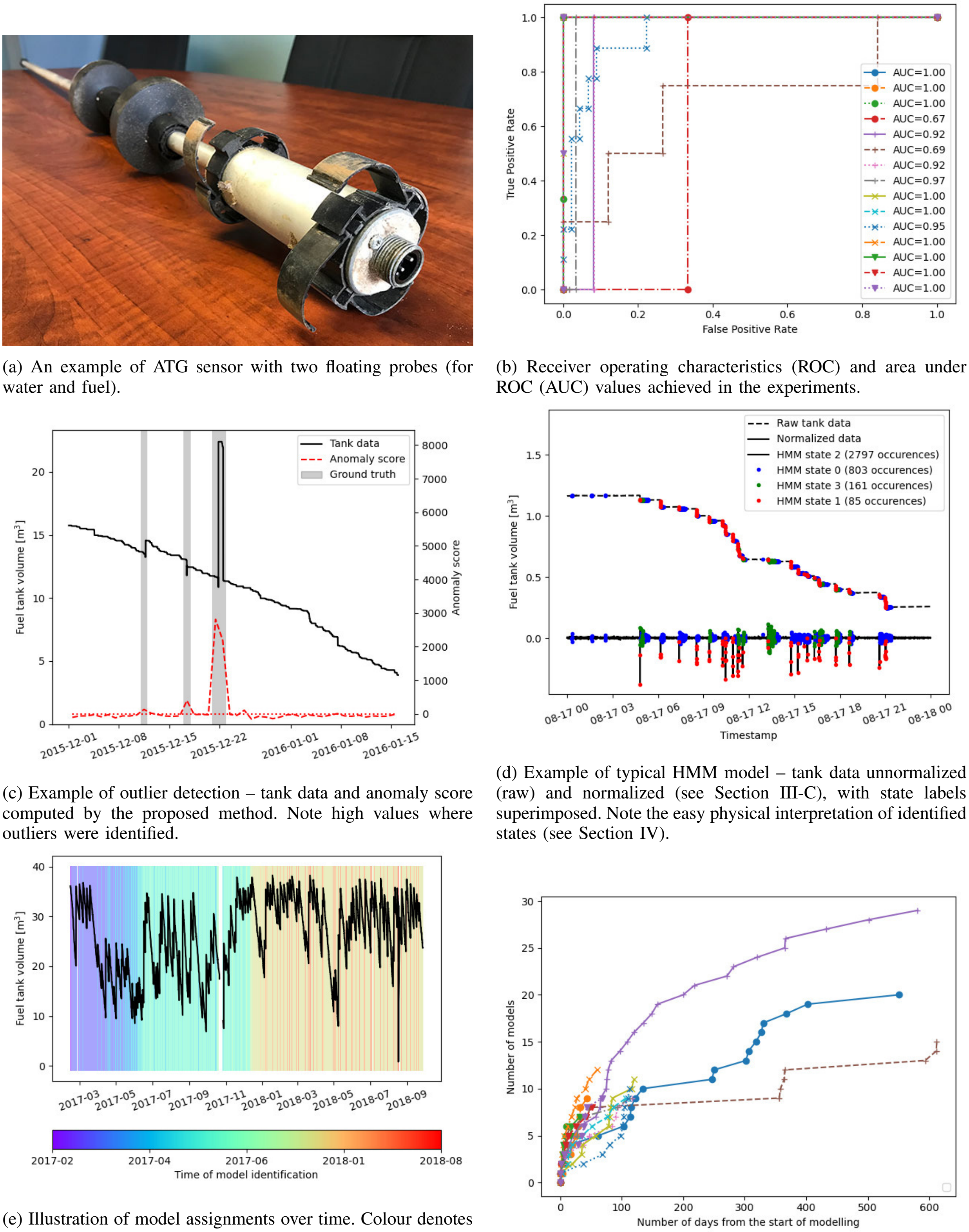
As each data sequence consists of fuel volume measurements at irregular intervals, the sequences were differentiated, normalized by time delta, and standardized[1] prior to inclusion in the experiments. Each normalized and standardized sequence has been cut into day's windows $\mathbf{x}$ and fed subsequently to the algorithm presented in Section II-B. For each sequence, the first half is treated as 'run-in' or training data, without using the labels (our case assumes they are unavailable during regular application). The resulting anomaly score $a_{\mathbf{x}}$ was recorded. The anomaly scores together with ground truth annotations were used to prepare ROC curves, with Area Under Curve (AUC) as the performance measure; for testing, positive labels were assigned to anomalies, while negative examples to normal levels.

Note that ground truth labels were generated especially for testing the proposed method, they are not required during the normal operation of the algorithm. We focus on evaluating the performance on the current set of data (current day), this is motivated by the performance measure of our underlying application setup, which is day to day monitoring of a fuel tank.

D. Algorithm performance

The average AUC score achieved by the method was 0.94 ± 0.11. In nine cases (no. 1-3, 9, 10, 12-15), the anomaly score was precise enough to correctly single out all anomalies, achieving maximum possible AUC value of 1.0. In two cases (no. 4 and 6) the results were strongly affected by difficulty of the problem, as the anomalies show with small changes in the signal, resulting in AUC values of 0.67 and 0.69. Remaining four cases (no. 5, 7, 8, 11) achieve $0.92 - 0.97$ (see Figure 1b). Almost all cases of mechanical failures (e.g. probe suspension) and faulty sensors were identified correctly. Anomalies (e.g. ripples or jumps within tank refill, especially in case 4)

[1] In actual application, the standardisation can be replaced by using mean and variance data from other tank with known history and similar characteristics, or estimated from physical tank properties, e.g. the total volume.



(a) An example of ATG sensor with two floating probes (for water and fuel).

(b) Receiver operating characteristics (ROC) and area under ROC (AUC) values achieved in the experiments.

(c) Example of outlier detection – tank data and anomaly score computed by the proposed method. Note high values where outliers were identified.

(d) Example of typical HMM model – tank data unnormalized (raw) and normalized (see Section III-C), with state labels superimposed. Note the easy physical interpretation of identified states (see Section IV).

(e) Illustration of model assignments over time. Colour denotes time of identification of a model assigned to given sample. Note two visible trend changes at 2017-04 and 2018-01; e.g. through the second half of the 2017 most assigned models were identified in the period right after 2017-04.

(f) Number of models as a function of days from modelling start, for given set of tanks. Consistent patterns are visible (see Section IV).

Fig. 1: Illustration of the proposed method and experiments' results.

were more difficult to spot, but the performance remained acceptable. In rare cases of tanks with a longer history, model adaptation (i.e. the process of conditional adding of a new HMM to an ensemble, see step 5 in Section II-B) had been seen interfering with a detection (outlier present within a new pattern was learned and not detected subsequently); however those cases could be isolated at the cost of increasing the sensitivity and potentially the number of false positives. A small portion of the errors were traced back to imprecise labelling of bad cases. Example detections are presented in Figure 1c. For our current results, a correlation study could not be carried out (correlation coefficients not statistically significant); an analysis of the results suggests that better AUC scores are achieved by bigger models, but not necessarily with longer training or test length, both when measured in number of days or number of samples.

IV. Discussion

A. Performance summary

Overall performance of the method was evaluated as very good, both in terms of quantitative score, and qualitative evaluation. Detailed inspection of the models revealed additional usage patterns, beyond the use of anomaly score. Adding a new model is usually connected with trend change of the series. If a high $p_{\mathcal{H}}$ value comes from a rare of old model (not matched lately, and previously matched to only a few cases), it may be additional signal of an anomaly. Finally some of the learned models could be tagged as interesting by the human observer and alert could be generated when they appear. The method has excellent aggregation properties; average number of parameters at the end of modelling was $n_p = 334$, which is less than 0.1% of the number of original samples.

B. Observed model behaviour

As expected, individual HMM within the ensemble represent daily behaviours of the tank being monitored. Example is presented in Figure 1d, note how the identified states (four in this example) correspond to known physical phenomena: stable level (s. 2), fuel unloading (s. 1), level oscillations after unloading (s. 3) and general or temperature induced oscillation (s. 0). Typically, the number of states is found in the range $3-7$, and most of the time they can be assigned some interpretation based on what happens inside the tank. We consider this correspondence a qualitative validation of our approach. Individual HMM inspection revealed that they contain features common mainly for the fuel type (e.g. diesel, gasoline, premium); this may make it possible to produce a tank-independent dictionary of HMMs that could be used as an initialization of the algorithm.

The step 5 of the algorithms prevents adding new models if previous adequately explain current data – until there's a trend change, when a set of new models must be added to keep the model accurate. In the example presented in Figure 1e two trend changes can be easily observed. Those trend changes are usually explainable by process or physical change (e.g. reassignment of fuel type for the tank, seasonal changes, general station usage type change resulting from roadworks). This trend change could be identified through analysis of model addition times, and provide additional monitoring information. Another view of this phenomenon can be seen in Figure 1f, where model count at given number of days from the start is presented. Often addition of models is seen in batches, on the beginning or when some trend change occurs.

C. Conclusions and future work

The proposed method has promising performance score, the resulting model has a high degree of explainability, limited memory and computation requirements and can be easily generalized to other domains of sensor verification.

As the objective of this work was a case study of the proposed algorithm, based on a mixture of HMMs, to the application of IoT sensor verification, further work will focus on extended analysis of the proposed approach, including: comparison with other approaches; detailed investigation of parameters, e.g. the effect of training sequence length; pruning the set of models $\mathcal{H}$ to remove rarely used, old models.

Acknowledgements

Authors wish to thank Bartosz Grabowski and Wojciech Masarczyk for insightful cooperation on research issues with neural networks sensor models, which led to the development of the method presented here. Authors also wish to thank anonymous reviewers for their insightful comments and remarks, which were basis for improvement of this paper and plans for future work.

References

[1] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of IoT sensor data processing, fusion, and analysis techniques," *Sensors*, vol. 20, no. 21, 2020. doi: 10.3390/s20216076

[2] M. Henry and G. Wood, "Sensor validation: principles and standards," *atp International*, vol. 3, no. 2, 2005.

[3] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Computing Surveys*, vol. 54, no. 3, 2021. doi: 10.1145/3444690

[4] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446

[5] J. M. Ramírez, F. Díez, P. Rojo, V. Mancuso, and A. Fernández-Anta, "Explainable machine learning for performance anomaly detection and classification in mobile networks," *Computer Communications*, vol. 200, pp. 113–131, 2023. doi: 10.1016/j.comcom.2023.01.003

[6] N. Aslam, I. U. Khan, A. Alansari, M. Alrammah, A. Alghwairy, R. Alqahtani, R. Alqahtani, M. Almushikes, and M. A. Hashim, "Anomaly detection using explainable random forest for the prediction of undesirable events in oil wells," *Applied Computational Intelligence and Soft Computing*, vol. 2022, p. 1558381, 2022. doi: 10.1155/2022/1558381

[7] F. Dama and C. Sinoquet, *Time Series Analysis and Modeling to Forecast: a Survey*. arXiv:2104.00164 [cs.LG], 2021.

[8] M. Staniszewski, A. Skorupa, Ł. Boguszewicz, M. Sokół, and A. Polański, "Quality control procedure based on partitioning of NMR time series," *Sensors*, vol. 18, no. 3, p. 792, 2018. doi: 10.3390/s18030792

[9] M. Staniszewski, A. Skorupa, Ł. Boguszewicz, A. Michalczuk, K. Wereszczyński, M. Wicher, M. Konopka, M. Sokół, and A. Polański, "Application of reiteration of hankel singular value decomposition in quality control," *AIP Conference Proceedings*, vol. 1863, no. 1, p. 400006, 2017. doi: 10.1063/1.4992575

[10] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989. doi: 10.1109/5.18626

[11] E. Lach, D. Grzechca, A. Polański, J. Rutkowski, and M. Staniszewski, "Analysis of semestral progress in higher technical education with HMM models," in *Computational Science – ICCS 2021*. Springer, 2021. doi: 10.1007/978-3-030-77967-2_18 pp. 214–228.

[12] M. Cholewa and P. Głomb, "Natural human gestures classification using multisensor data," in *Proceedings 3rd IAPR Asian Conference on Pattern Recognition ACPR 2015*. IEEE, 2015. doi: 10.1109/ACPR.2015.7486553 pp. 499–503.

[13] J. Ying, T. Kirubarajan, K. Pattipati, and A. Patterson-Hine, "A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 30, no. 4, pp. 463–473, 2000. doi: 10.1109/5326.897073

[14] W. Zhao, T. Shi, and L. Wang, "Fault diagnosis and prognosis of bearing based on Hidden Markov Model with multi-features," *Applied Mathematics and Nonlinear Sciences*, vol. 5, no. 1, pp. 71–84, 2020. doi: 10.2478/amns.2020.1.00008

[15] F. van de Pol and R. Langeheine, "Mixed Markov latent class models," *Sociological Methodology*, vol. 20, pp. 213–247, 1990. doi: 10.2307/271087

[16] M. D. R. de Chaumaray, M. Marbac, and F. Navarro, "Mixture of hidden Markov models for accelerometer data," *Annals of Applied Statistics*, vol. 14, no. 4, pp. 1834 – 1855, 2020. doi: 10.48550/arXiv.1906.01547

[17] D. Vidotto, J. K. Vermunt, and K. V. Deun, "Multiple imputation of longitudinal categorical data through bayesian mixture latent Markov models," *Journal of Applied Statistics*, vol. 47, no. 10, pp. 1720–1738, 2020. doi: 10.1080/02664763.2019.1692794

[18] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "Adaptive ROC-based ensembles of HMMs applied to anomaly detection," *Pattern Recognition*, vol. 45, no. 1, pp. 208–230, 2012. doi: 10.1016/j.patcog.2011.06.014

[19] P. R. Cavalin, R. Sabourin, C. Y. Suen, and A. S. Britto Jr., "Evaluation of incremental learning algorithms for HMM in the recognition of alphanumeric characters," *Pattern Recognition*, vol. 42, no. 12, pp. 3241–3253, 2009. doi: 10.1016/j.patcog.2008.10.012

[20] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "A survey of techniques for incremental learning of HMM parameters," *Information Sciences*, vol. 197, pp. 105–130, 2012. doi: 10.1016/j.ins.2012.02.017

[21] K. Faber, R. Corizzo, B. Sniezynski, and N. Japkowicz, "Vlad: Task-agnostic vae-based lifelong anomaly detection," *Neural Networks*, vol. 165, pp. 248–273, 2023. doi: 10.1016/j.neunet.2023.05.032

[22] D. Samariya and A. Thakkar, "A comprehensive survey of anomaly detection algorithms," *Annals of Data Science*, vol. 10, no. 3, pp. 829–850, 2023. doi: 10.1007/s40745-021-00362-9

[23] L. Wang, X. Zhang, H. Su, and J. Zhu, *A Comprehensive Survey of Continual Learning: Theory, Method and Application*. arXiv:2104.00164 [cs.LG], 2023.

[24] P. Głomb, M. Romaszewski, A. Sochan, and S. Opozda, "Unsupervised parameter selection for gesture recognition with vector quantization and hidden markov models," in *Human-Computer Interaction – INTERACT 2011*, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds. Springer Berlin Heidelberg, 2011. doi: 10.1007/978-3-642-23768-3_14 pp. 170–177.

[25] P. Foszner, A. Gruca, and J. Bularz, "Fuel pipeline thermal conductivity in automatic wet stock reconciliation systems," in *Advances in Data Mining. Applications and Theoretical Aspects: 16th Industrial Conference*. Springer, 2016. doi: 10.1007/978-3-319-41561-1_22 pp. 297–310.

[26] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–464, 1978. doi: 10.1214/aos/1176344136

DOI: 10.15439/2023F3797
ISSN 2300-5963 ACSIS, Vol. 35

Towards an HPC cluster digital twin and scheduling framework for improved energy efficiency

Alexander Kammeyer*†, Florian Burger*, Daniel Lübbert* and Katinka Wolter†
*Physikalisch-Technische Bundesanstalt
Abbestraße 2-12, 10587 Berlin, Germany
Email: {alexander.kammeyer, florian.burger, daniel.luebbert}@ptb.de
†Freie Universität Berlin,
Takustraße 9
14195 Berlin, Germany
Email: {a.kammeyer,katinka.wolter}@fu-berlin.de

***Abstract*—Demand for compute resources and thus energy demand for HPC is steadily increasing while the energy market transforms to renewable energy and is facing significant price increases. Optimizing energy efficiency of HPC clusters is therefore a major concern. Different possible optimization dimensions are discussed in this paper. This paper presents a digital twin design for analyzing and reducing energy consumption of a real-world HPC system. The digital twin is based on the HPC cluster at PTB. The digital twin receives information from multiple internal and external data sources to cover the different optimization opportunities. The digital twin also consists of a scheduling simulation framework that uses the data from the digital twin and real-world job traces to test the influence of the different parameters on the HPC cluster.**

I. Introduction

THE Physikalisch-Technische Bundesanstalt (PTB) operates a compute cluster at its Berlin site that is used by various departments within PTB. The cluster is the backbone of numerous research activities. The HPC cluster currently consists of 60 CPU nodes with CPUs from two generations and two special GPU nodes with 10 GPUs. This amounts to an energy consumption of approximately 30 kW. PTB plans to extend the installed compute power by another 50 kW in Berlin and, perspectively, to install a new cluster with over 200 kW installed power at its Brunswick site.

The PTB relies on HPC for many research activities in different departments including physics, mathematics and medicine. The new AI strategy of PTB will require more compute resources for AI applications and model training. Energy efficiency becomes a concern with this increased demand. While the energy efficiency of modern CPUs has improved over the years, the total energy consumption of HPC systems has increased at an even faster rate [1]. With this increasing energy usage, the total cost of ownership is impacted by increasing energy prices in a volatile market environment. Additionally, PTB, as a federal institution, is bound by the climate protection plan 2050 [2] and climate protection program 2030 [3] of the German federal government. New data centres, like the planned Brunswick site, need to meet the standards defined by the Blue Angel certificate for data centres [4].

Currently, the influence of different internal and external factors on the energy efficiency of the HPC cluster at PTB are not well understood. A digital twin can help in understanding those factors in detail and test changes to the system configuration without adverse effects to the production system and is commonly used for this purpose [5]. However, optimization goals and systems differ from every site and cluster. Thus, a specific, tailored solution is needed for this multi-factor optimization problem. Other HPC centres are also making efforts to improve their energy efficiency and are reporting on those efforts [6].

This paper introduces a design specification for digital twin of the HPC cluster at PTB. The digital twin shall help the operators of the cluster to improve the cluster operation with regards to several optimization goals. The digital twin collects data from various data sources for that purpose. In addition, the digital twin contains a simulation framework with a scheduling model. This simulation framework is used to test parameters and settings with regard to the combined optimization goals.

The next chapter presents the optimization dimensions that have to be taken into consideration. Chapter III presents the design specification of the digital twin and its different data sources. Finally, Chapter IV introduces the scheduling framework created to simulate the HPC cluster with data from the digital twin.

II. Background

Increasing the energy efficiency and energy usage of a compute cluster is a multi-dimensional optimization problem. Energy consumption, CO_2 emissions, energy cost, cooling and energy limitations have been identified to be of particular interest of PTB [7]. Optimizations for each of these goals are possible. Each of these dimensions needs a way to be monitored and a representation within the digital twin is required. The rest of this chapter describes these dimensions in detail:

A. Energy consumption

The most obvious optimization is the overall reduction of energy consumption by the HPC cluster. Different mechanisms

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

Topical area: Information Technology for Business and Society

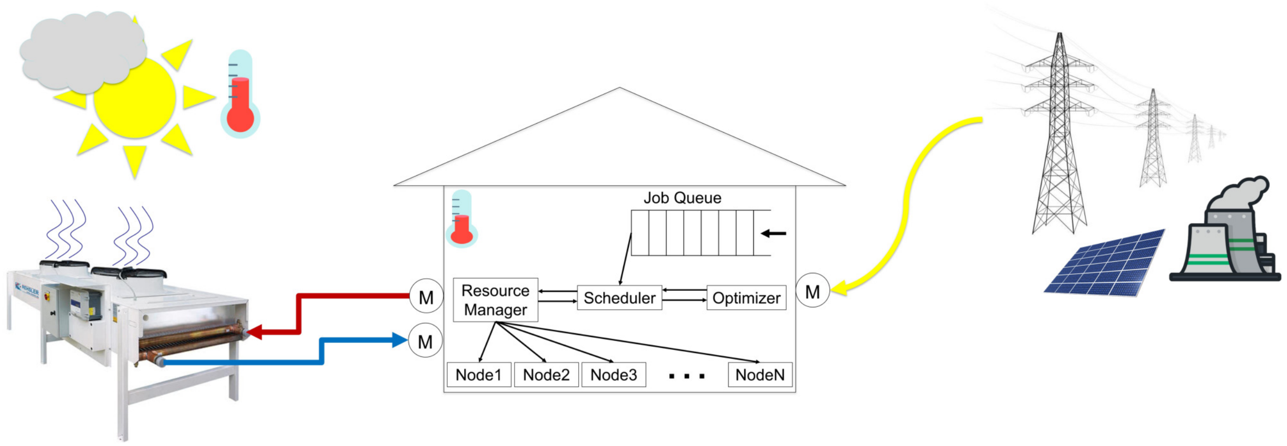


Figure 1. Schematic representation of the digital twin for PTBs HPC cluster

have been proposed like Dynamic Voltage and Frequency Scaling (DVFS) [8], turning off idle nodes [1], adapting jobs to the energy budget and running nodes at reduced frequencies [9]. The digital twin can be used to test different parameters and their effect on the cluster and monitor the long-term effect of configuration changes.

B. CO_2 emissions

The energy in the energy grid comes from different sources with different CO_2 emissions associated to them. Tracking the equivalent CO_2 emissions when using energy and moving jobs to times of low CO_2 emissions can help to reduce the CO_2 equivalent associated with the operation of the cluster. Since each energy source has different emissions, days with strong winds or low cloud coverage reduce emissions while sources like natural gas and coal used to cover base load in the energy grid increase emissions.

C. Energy cost

Energy prices for large consumers are dynamically determined at energy spot markets. These prices are volatile and shifting compute jobs to low-cost times can directly reduce energy costs. Optimizing for this goal can directly save energy costs. Using the data collected by the digital twin, it may be possible to identify a correlation between pricing and high energy availability, e.g. due to a lot of available renewable energy.

D. External influences

The weather has a direct influence on energy efficiency, hence most metrics, such as Power Usage Effectiveness [10], are averages over a year-long period to average certain weather and season related effects. PTB intends to move the cluster cooling to a free cooling system, which use temperature differences to cool the cluster. These machines are more efficient compared to classic compressor-based cooling machines but only work up to a certain outside temperature. Therefore, they might not provide sufficient cooling on extremely warm days. The cluster needs to adapt to such conditions, e.g. by limiting the amount of active nodes. Monitoring system temperatures and integrating weather forecast data into the digital twin, the HPC system can be configured to stay within defined operational parameters, e.g. by reducing system load.

Another problem might be limited energy availability. Other industries with high energy usage have developed so called demand response mechanisms to reduce energy consumption in cooperation with energy service providers when not enough energy is available. With increasing power demands of HPC clusters such strategies might be required for HPC as well and some strategies have been proposed [11]. Similar to the limitations imposed by insufficient cooling capacities, the cluster can be adjusted to using less energy when not enough energy can be supplied by the energy provider.

III. Digital Twin for HPC

A digital twin is used to reason about a real-world object with data available digitally. In order to do so, real world data needs to be collected and models about the object have to be created. The previous section introduced the relevant dimensions for the optimization problem. These dimensions need to be represented in the digital twin with data and models.

The HPC cluster receives compute jobs from the users in a job queue. The scheduler decides which jobs are run on which nodes at a certain time. A resource manager module manages the available system resources. The compute nodes execute the jobs submitted to the system. Each node has its own CPU, RAM and network connection. Some nodes also have GPUs. The general operation of the cluster is simulated via a scheduling simulation. This simulation is key to understand the cluster operation. Chapter IV introduces the scheduling simulation. The nodes operation can be estimated with energy estimates. No actual jobs need to run to simulate the scheduling simulation.

The simulation results and data is used by the optimizer component that influences the scheduler and the resource manager to change the operation parameters of the HPC

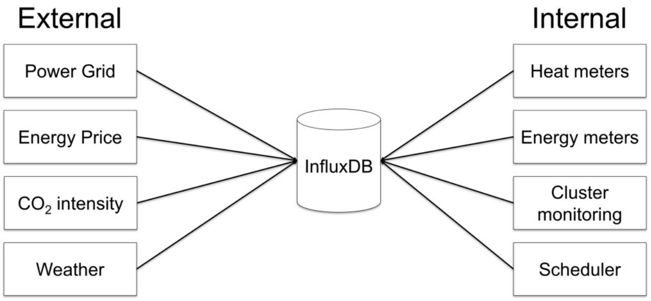


Figure 2. Software components of the digital twin with all data sources and models

cluster. This component is not yet functional but planned for future development with the results obtained by scheduling simulations from the simulation component.

The collected data generally falls into two categories: internal data from the HPC cluster and data centre, and external data about the outside world. The core component of the digital twin is an InfluxDB database. InfluxDB is a time-series database, optimized for processing time-dependent data. A collector program for each data source continuously writes the data to the Influx database. A Grafana dashboard provides access to and visualizations of the data for the user. All data collectors, Grafana and InfluxDB are managed via Docker containers. The scheduling simulator also uses the data from the InfluxDB directly.

The cluster status, including job and node information, is collected with ClusterCockpit [12]. Energy usage is collected via energy meters installed on site. This includes energy used by the cluster and the cooling infrastructure. Heat meters are used to monitor the amount of heat removed from the cluster by the cooling system.

All external data sources have been chosen specific to PTBs location and requirements as well as the optimization goals described in Chapter II. The external data sources for the digital twin include energy grid information, weather forecast data, energy prices and the $\mathrm{CO_2}$ emission associated with the current energy mix in the energy grid. Information about the energy grid is obtained from Bundesnetzagentur [13]. This federal agency offers information about the German energy grid and the energy sources used at any given time. Energy prices are also retrieved from Bundesnetzagentur. $\mathrm{CO_2}$ estimations are retrieved from Electricity Maps [14]. Finally, weather information are retrieved from the German Meteorological Service (DWD) [15] via the Bright Sky API [16].

Figure 1 shows the architecture described in this Chapter in a schematic representation. The individual data collectors described in this Chapter can be found in figure 2. They are grouped into internal and external data sources with InfluxDB being the core component.

IV. Scheduling simulation framework

This section gives a brief overview of existing scheduling simulators and introduces a new scheduling simulator that uses the digital twin data of the PTB HPC cluster as input for its simulations and models the operation of the cluster.

Energy aware scheduling has been of interest in the community with various goals, metrics and proposed solutions [17]. Scheduling simulators have been used to test different scheduling algorithms or additional parameters without changing production systems or the need of running actual jobs. Slurm [18] is a scheduler used in production HPC systems. Simakov et. al. developed multiple Slurm simulator versions [19], [20] that follow the steady Slurm development. All versions allow to test different Slurm parameters without altering the corresponding production system. Slurm can be combined with other schedulers like NQSV and digital twins [5]. However, since every setup is different, a custom solution is necessary. Yang et al. [21] proposed a scheduling simulator that offers two pricing options for scheduling but does not support dynamic pricing based on constantly changing energy prices.

The data of the digital twin is used for the scheduling simulations. The simulator connects to the database in order to get both the internal and external data. Because the digital twin is a purpose-build solution, no suitable simulator exists. Therefore a new solution was developed.

For development, testing and validation purposes the simulator component presented in this paper can generate synthetic job traces. For longer, realistic job traces the simulator can read the standard workload format (swf) from the Parallel Workloads Archive [22] and can use job traces from this archive as input. Additionally, job traces from the PTB compute cluster can also be used as input.

Furthermore, the data from the digital twin database as well as the job traces are time dependent. The scheduling simulation can run at arbitrary time points. The simulator was designed in such a way that it can run a given job trace at a chosen start time and match the data from the digital twin database accordingly. This allows to test varying internal and external factors on the same job trace or the same internal and external factors on different job traces.

The results of the simulations can be used to tune the system parameters to meet the optimization goals from Chapter II. An automatic optimizer can be implemented as a component of the HPC cluster for automatic tuning and adjustment of parameters.

V. Conclusion

With increasing compute demand and thus energy demand of HPC clusters, energy consumption and availability becomes a concern for HPC cluster operators. Possible goals for optimizations like energy consumption, $\mathrm{CO_2}$ emissions, energy cost and external influences that affect energy and cooling have been discussed in this paper.

A design of a digital twin, a representation of a real-world HPC cluster at PTB, has been proposed. It has a time-series database at its core and is connected to data sources for the internal cluster aspects as well as external factors relevant for the optimization goals. The digital twin makes data about the cluster available to the administrators. This data helps to asses and monitor the efforts towards meeting desired efficiency goals over time.

As part of the digital twin, a scheduling simulator has been developed that simulates the operation of the cluster. It uses the data from the digital twin and allows to test different job traces. Because of the time dependence of the data, the simulator can map different job traces with multiple data points to test traces with different data. The digital twin can be used by the administrators to test various configuration parameters and algorithms for their effect on the optimization goals.

So far, the digital twin is only used for simulations. It has not yet been integrated with the scheduler of the real-world HPC system. However, this shall be done in the future.

With the scheduling simulation framework, as part of the digital twin, further empirical studies of the optimization goals are planned. These simulations are a first step towards more efficient cluster operation at PTB and are the basis for future improvements of the real-world HPC cluster, especially toward an automatic optimizer component as part of the scheduler.

References

[1] O. Mämmelä, M. Majanen, R. Basmadjian, H. De Meer, A. Giesler, and W. Homberg, "Energy-aware job scheduler for high-performance computing," *Computer Science - Research and Development*, vol. 27, no. 4, p. 265–275, 2012. [Online]. Available: https://doi.org/10.1007/s00450-011-0189-6

[2] Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit (BMU), "Klimaschutzplan 2050," https://www.bmwk.de/Redaktion/DE/Publikationen/Industrie/klimaschutzplan-2050.pdf, 2019.

[3] ——, "Klimaschutzprogramm 2030 der Bundesregierung zur Umsetzung des Klimaschutzplans 2050," https://www.bundesregierung.de/resource/blob/974430/1679914/e01d6bd855f09bf05cf7498e06d0a3ff/2019-10-09-klima-massnahmen-data.pdf, Oct. 2019.

[4] R. UMWELT, *Energieeffizienter Rechenzentrumsbetrieb DE-UZ 161*, 2nd ed., https://produktinfo.blauer-engel.de/uploads/criteriafile/de/DE-UZ%20161-201502-de%20Kriterien.pdf, Fränkische Straße 7, 53229 Bonn, Feb. 2015.

[5] T. Ohmura, Y. Shimomura, R. Egawa, and H. Takizawa, "Toward building a digital twin of job scheduling and power management on an hpc system," in *Job Scheduling Strategies for Parallel Processing*, D. Klusáček, C. Julita, and G. P. Rodrigo, Eds. Cham: Springer Nature Switzerland, 2023, p. 47–67.

[6] M. Ott and D. Kranzlmüller, "Best practices in energy-efficient high performance computing," in *Workshops der INFORMATIK 2018 - Architekturen, Prozesse, Sicherheit und Nachhaltigkeit*. Bonn: Köllen Druck+Verlag GmbH, 2018, p. 167–176.

[7] A. Kammeyer, F. Burger, D. Lübbert, and K. Wolter, "Optimization of energy efficiency of an hpc cluster: On metrics, monitoring and digital twins," in *Sensor and Measurement Science International*, ser. SMSI 2023. AMA Service GmbH, May 2023, p. 378–379. [Online]. Available: https://doi.org/10.5162/SMSI2023/P51

[8] K. Ahmed, J. Liu, and X. Wu, "An Energy Efficient Demand-Response Model for High Performance Computing Systems," in *2017 IEEE 25th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2017, p. 175–186.

[9] A. Krzywaniak, J. Proficz, and P. Czarnul, "Analyzing Energy/Performance Trade-Offs with Power Capping for Parallel Applications On Modern Multi and Many Core Processors," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, p. 339–346.

[10] V. Avelar, D. Azevedo, A. French, and E. N. Power, "Pue: a comprehensive examination of the metric," *White paper*, vol. 49, 2012.

[11] K. Ahmed, "Energy Demand Response for High-Performance Computing Systems," Ph.D. dissertation, Florida International University, Miami, 2018.

[12] J. Eitzinger, T. Gruber, A. Afzal, T. Zeiser, and G. Wellein, "Clustercockpit — a web application for job-specific performance monitoring," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*, 2019, p. 1–7.

[13] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen (BNetzA), "SMARD - Strommarktdaten, Stromhandel und Stromerzeugung in Deutschland," https://www.smard.de/home/marktdaten, May 2023.

[14] Electricity Maps ApS, "Electricity Maps," https://www.electricitymaps.com/, May 2023.

[15] Deutscher Wetterdienst (DWD), "Open Data Server of the German Meteorological Service," https://opendata.dwd.de/, May 2023.

[16] Bright Sky Developers, "Bright Sky JSON API for DWD's open weather data," https://brightsky.dev/, May 2023.

[17] B. Kocot, P. Czarnul, and J. Proficz, "Energy-aware scheduling for high-performance computing systems: A survey," *Energies*, vol. 16, no. 2, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/2/890

[18] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Job Scheduling Strategies for Parallel Processing*, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, p. 44–60.

[19] N. A. Simakov, M. D. Innus, M. D. Jones, R. L. DeLeon, J. P. White, S. M. Gallo, A. K. Patra, and T. R. Furlani, "A slurm simulator: Implementation and parametric analysis," in *High Performance Computing Systems. Performance Modeling, Benchmarking, and Simulation*, S. Jarvis, S. Wright, and S. Hammond, Eds. Cham: Springer International Publishing, 2018, p. 197–217.

[20] N. A. Simakov, R. L. Deleon, Y. Lin, P. S. Hoffmann, and W. R. Mathias, "Developing accurate slurm simulator," in *Practice and Experience in Advanced Research Computing*, ser. PEARC '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3491418.3535178

[21] X. Yang, Z. Zhou, S. Wallace, Z. Lan, W. Tang, S. Coghlan, and M. Papka, "Integrating dynamic pricing of electricity into energy aware scheduling for HPC systems," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, 2013.

[22] D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with using the parallel workloads archive," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, p. 2967–2982, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731514001154

DOI: 10.15439/2023F9557
ISSN 2300-5963 ACSIS, Vol. 35

Perception of vector and triangle representations of fuzzy number most possible value changes

Dorota Kuchta, Jerzy Grobelny, Rafał Michalski, Jan Schneider
0000-0002-9747-0759, 0000-0001-9791-9395, 0000-0002-0807-1925, 0000-0001-6738-1790
Wroclaw University of Science and Technology, Faculty of Management
27 Wybrzeże Wyspiańskiego st., 50-370 Wrocław, Poland,
Email: {dorota.kuchta, jerzy.grobelny, rafal.michalski, jan.schneider}@pwr.edu.pl

***Abstract*— The aim of the study is to investigate and evaluate user preferences regarding two visual representations of uncertainty estimates for decision-making purposes. The research is concerned with the perception of fuzzy numbers, which are depicted either as triangles or as specifically constructed vectors. The study involves a series of pairwise comparisons in which participants must determine which representation reflects the change in the most possible value in a more salient way. The results are then analyzed and formally verified statistically. The study shows that there are specific circumstances where vector representations are more desirable than their triangle-based counterparts. The findings also suggest that there may be some differences in assessing these representations depending on gender. This examination expands our understanding of how subjects perceive different graphical methods for presenting change in a selected parameter uncertainty feature. From a practical standpoint, the findings offer suggestions for designing graphical user interfaces that present fuzzy data to users.**

***Index Terms*—Fuzzy number visualization, Fuzzy number vector representation, Visual processing, Project uncertainty, Usability.**

I. Introduction

UNCERTAINTY is a pervasive issue that must be addressed in numerous fields both those typically associated with precision such as physics or engineering, as well as areas where rather soft computing is prevalent, e.g., management, economics, and other social sciences. In particular, uncertainty needs to be handled while analyzing risk, building various models and making decisions [1]–[3], and not taking into account uncertainty can lead to often negative or even catastrophic results such as project failures [4], [5]. It is, thus, indispensable to control them and to analyse their behaviour on an ongoing basis.

The efficiency and effectiveness of managing uncertainty and solving related problems, for instance, in production, or project management can strongly depend how it is presented to persons analyzing the data. The importance of data visualization in the process of decision making has been stressed in many papers, for instance, showing its substantial positive influence on communication process [6]. There are also a number of various ways of graphically presenting uncertainty and they have been described and reviewed in numerous scientific publications (e.g., [7]–[10]).

In our previous papers [11], [12], we proposed the use of vectors to graphically visualize uncertainty defined by triangular fuzzy numbers. We have initially compared the usability and potential applications of triangle- and vectors-based representations of triangular fuzzy numbers. By performing experimental tasks, we found out that their capability to convey information may not be worse than that of triangles and is seen as even better by some of the users. This type of relatively simple fuzzy number requires to provide only three, point-type estimates and is commonly used in practice. For example, in project management, uncertainty of a time or cost estimate, encoded by a triangular fuzzy number, is fully defined by three parameters: the optimistic, the most possible, and the pessimistic value [13], [14]. This approach closely resembles the well-known PERT approach [15]–[17] that, albeit based on probability theory, also requires similar three parameters. In the PERT approach the beta distribution [18] is used with the corresponding *arithmetic* of random variables, while with triangular fuzzy numbers, the representation of the estimates and the mathematical operations are more straightforward and intuitional. Although in the literature one can find phrases "*fuzzy vector*", however it should be emphasized that these papers refer usually to fuzzy vectors that are different in nature from the current paper definition – compare, for example, the following works [19]–[23].

In the current study, we investigate the uncertainty visualizations as the traditional graphs of membership functions, and our own proposals of depicting them as vectors. Since the previous results have shown that the effectiveness and efficiency of both approaches are comparable, we decided to pursue this subject in more detail. Here, we focus particularly on the individuals' perception of the saliency of changes in the most possible value depicted both as triangles and vectors.

The focus on the most possible values results from the fact that in many cases the most possible values are taken as the basis for current decision making such as setting deadlines in

· The work of Jerzy Grobelny and Rafał Michalski was partially financially supported by the Polish National Science Centre grant no. 2019/35/B/HS4/02892: Intelligent approaches for facility layout problems in management, production, and logistics.

project planning. Therefore, the changes in the most possible values should be carefully controlled and their most suitable graphical representation is of great importance here. Moreover, monitoring variability in various aspects seems to significantly increase the chances for achieving project management success [24]. To extend our understanding how individuals perceive these changes portrayed in a different visual form, we designed and performed an experiment. The obtained outcomes are analysed and discussed in this paper. Since previous results [11] suggest that there can be some differences between men and women while assessing triangle- and vector-based representations, the gender effect was also included into our study.

The outline of the paper is as follows: In Section II, we present basic information about the triangular and vector representations of triangular fuzzy numbers. Section III includes the all the details about the experimental design and study subjects' characteristics which is followed by the results analysis section. The paper ends with a discussion and conclusions.

II. Membership functions versus vectors – two uncertainty visualization approaches

Let us suppose that a project cost or time item has been estimated by experts in the form of three values $\underline{r}, \hat{r}, \overline{r}$ – the optimistic, most possible and pessimistic values, respectively. The respective triangular fuzzy number will be denoted as $\tilde{R} = \left(\underline{r}, \hat{r}, \overline{r}\right)$. Its membership function $\mu_R(\mathrm{x})$ is defined on the set $\mathfrak{R}$ of real numbers and represents the possibility degrees of the respective real numbers. An example of $\tilde{R}$ (2,3,5) is shown in Fig.1. The alternative representation, put forward and examined in our previous study [11], [25], is based on vectors (Fig. 1). The vector $\vec{R}\left(\underline{r}, \hat{r}, \overline{r}\right) = \{m_R, s_R, \gamma_R\}$ will be defined by its starting point $(m_R, 0)$, where $m_R = \left(\overline{r} + \underline{r}\right)/2$, its length $s_R = \overline{r} - \underline{r}$, the angle in relation to the line $x = m_R$ computed as $\gamma_R = arc\,tan(\hat{r} - m_R)$; positive angles denote the inclination to the right and negative ones – to the left.

It is important to notice that the angle γ_R will be zero only if the most possible value $\hat{r}$ is equal to m_R, the arithmetic mean of the pessimistic and optimistic values $\overline{r}$ and $\underline{r}$. These two representations described above are investigated in an experiment described in following sections.

A. Study Subjects

In total, 88 individuals participated in the survey. However, five persons (four females and one male) were excluded from further analysis due to incomplete data.

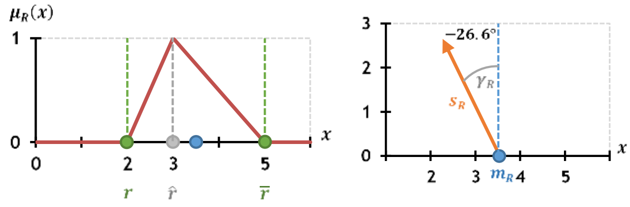


Fig. 1. Membership function based visualisation of parameter R determined by numbers 2, 3, 5 and its vector visualization.

III. Method

Thus, all the presented in this study results refer to 83 participants. They were primarily volunteer students aged between 18 and 46 years old from Wroclaw University of Science and Technology in Poland. The mean age was 23.42 years, with a standard deviation of 3.45. The group was highly homogenous in this regard, as the 25th age centile amounted to 23 and 75th – 24 years. Out of the participants, the majority were women, specifically 61 individuals, making up 73.5% of the total. All of them provided their informed consent to participate in the study.

A. Experimental Design and Task Description

The experimental design aimed to gain a deeper understanding of how individuals perceive changes in fuzzy number most possible value in both triangular and vector visualizations. Participants' task was to give subjective assessment of different variants of these representations in terms of their saliency of fuzzy number feature change. In particular, the effects described in the next subsection were examined.

Examined factors

The present study investigated two graphical representations of triangular fuzzy numbers: traditional triangle and vector-based visualizations. They were the first factor examined in the study, and their mathematical properties were concisely explained in Section II. The study specifically focused on how participants perceived changes in a single property related to fuzzy numbers: the information about the most possible value of the imprecise parameter in question. Graphically, this feature is associated with either the position of the maximum of the triangular fuzzy number membership function or the inclination angle of the vector. The study explored three distinct levels of change in the most possible value, which included (i) a change of two units from zero to two, (ii) a change of four units from zero to four, and (iii) a change of two units from two to four. The factors investigated and their corresponding levels are depicted in Fig. 2.

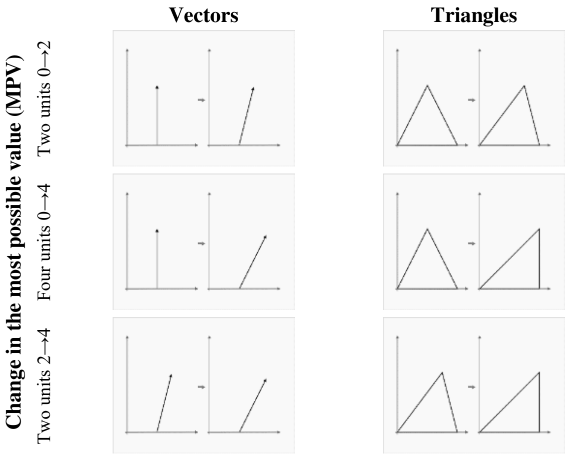


Fig. 2. Factors and their levels examined in the current study: visual representation (vectors, triangles), the most possible value change (MPV: Two units 0→2, Four units 0→4, Two units 2→4).

Dependent measures

We utilized two dependent measures to assess the subjective opinions of the respondents. To assess preferences, we obtained relative weights through pairwise comparisons of all experimental conditions. Pairwise comparisons have been demonstrated to enhance the accuracy of evaluations [26], [27] and have been successfully implemented in numerous studies for establishing hierarchies of preferences see, e.g., [28]–[32]). In this study, we utilized this approach within the Analytic Hierarchy Process (AHP) framework [33] to determine stimulus subjective perceptions and calculate consistency ratios for each participant.

To determine which figure showed a more noticeable increase in the most possible value of the fuzzy number, participants were asked to provide responses on a 5-point, two-directional linguistic scale recommended in the AHP approach.

By combining different levels of the two factors examined, we were able to identify six distinct experimental conditions. These conditions were generated by varying two types of graphical representations of triangular fuzzy numbers and three levels of indeterminacy changes (as shown in Fig. 2). To test all six experimental conditions, we applied a within-subject design where each subject participated in every condition.

B. Experimental Procedure

The data collection process for the study was conducted entirely over the internet. Participants were provided with general information about the research and a hyperlink to a slideshow containing a detailed audio explanation of the research. The final slide contained a hyperlink to the experimental application based on React.js, which opened in their default web browser upon clicking. After opening the software, the first page presented participants with the informed consent form, which they were required to read and accept before starting the examination. After providing their gender and age, the software displayed all necessary pairwise comparisons of the experimental conditions one by one, in a random order. During this process, the data were collected locally in the web browser's internal variables and were subsequently sent to a remote server after the completion of the entire procedure.

IV. Results

A. Descriptive statistics

The experimental data that was gathered was brought together and transferred to TIBCO Statistica version 13.3 software. The analysis included both consistency ratios and relative weights and was carried out taking advantage of typical descriptive statistics and analysis of variance. The outcomes of this examination are exhibited in the following sub-sections.

Consistency ratios

The consistency ratios of all the examined individuals were equaled, on average, 0.381 with a standard deviation amounting to 0.296. The median value of 0.263 was much lower than the mean, which suggest that the distribution was positively skewed. The consistency ratio ranged from a minimum value of 0.0421 to a maximum of 1.42.

Relative weights for studied stimuli

Table I displays the key descriptive statistics for the relative weights computed for all the stimuli that were investigated.

The greater the relative weight values, the stronger the perception of saliency of the change in the fuzzy number most possible value in a particular experimental condition. Pictures illustrating the change by four units exhibit the highest mean and median values, signifying the most salient perception of the changes. This observation was consistent for both triangle and vector representations.

In all cases, the median value was slightly smaller than the mean value, which indicates a somewhat positively skewed distribution. Furthermore, these experimental conditions had the highest variability, as evidenced by the larger standard deviations and mean standard errors.

B. Analysis of variance

The analysis of variance technique was employed to formally verify if the observed differences in average values were statistically significant. We have conducted this method to both consistency rations and relative weights. The results of these two analyses of variance are presented in the following subsections.

Consistency ratios

There were differences in the CR mean values for men and women with males being on average more consistent (0.356) than females (0.390). However due to the considerable standard deviations, the discrepancy occurred to not be meaningful, which was supported by performing one-way analysis of variance. Its results for gender differences in consistency ratios showed statistical insignificance at the level of $p = 0.65$ $[F(1, 81) = 0.21]$.

Relative weights for studied stimuli

To determine the statistical significance and extent of the differences in the mean relative weight values for the studied

TABLE I.
KEY DESCRIPTIVE STATISTICS OF RELATIVE WEIGHTS FOR ALL EXPERIMENTAL CONDITIONS

Graphic Represen-tation	Change Type	Mean	Median	Min	Max	Std Dev	Mean Std Error
Triangle	CT_0→2	0.101	0.082	0.024	0.523	0.072	0.0080
	CT_0→4	0.327	0.323	0.032	0.593	0.149	0.0163
	CT_2→4	0.128	0.105	0.019	0.348	0.090	0.0099
Vector	CT_0→2	0.122	0.098	0.021	0.423	0.095	0.0105
	CT_0→4	0.247	0.234	0.051	0.543	0.125	0.0138
	CT_2→4	0.075	0.058	0.013	0.283	0.058	0.0064
All		0.167	0.115	0.013	0.593	0.137	0.0061

effects, we performed a three-way analysis of variance, specifically analyzing the *Change Type* and *Graphical Representation* factors. We have also included the *Gender* effect, since our previous study [11] suggests that this may differentiate the results regarding the investigated stimuli. The results show that two of the three factors investigated were statistically significant. Specifically, the factors of *Change Type* and *Graphical Representation* had significant effects with: $F(2, 486) = 154.7$, $p < 0.0001$, and $F(1, 486) = 5.24$, $p = 0.0225$, respectively. The gender effect alone was statistically irrelevant, but its interactions with both *Graphical Representation* and *Change Type* were meaningful. The former one (*Graphical Representation* × *Gender*) at the level of $\alpha < 0.05$ [$F(2, 486) = 7.78$, $p = 0.0055$], whereas the latter *Change Type* × *Gender* at the level of $\alpha < 0.1$ [$F(2, 486) = 2.87$, $p = 0.0578$]. Additionally, the interaction between *Change Type* and *Graphical Representation* was statistically significant at the level of $\alpha < 0.05$ [$F(2, 486) = 7.13$, $p = 0.0009$].

Fig. 3 presents a visual representation of the mean relative weight values for the *Change Type* effect. The figure indicates that the study subjects perceived the most salient change in the most possible value for four-unit changes (CT_0→4). On the other hand, the difference between one-unit changes (CT_0→2 and CT_2→4) appears to be less clear-cut.

In order to explore the distinctions between the levels of the *Change Type* effect, a set of pairwise LSD post-hoc statistical tests were conducted. The results of these calculations indicate that the sole discrepancy that is not statistically meaningful pertains to two levels that entail changes of two units in the fuzzy number most possible value (CT_0→2 vs. CT_2→4 with $p = 0.368$). In other cases, differences were significant at $\alpha < 0.0001$.

Fig. 3 also displays the average relative weights for the two levels of the *Graphical Representation* effect. These findings corroborate the initial analysis graphically shown in the key descriptive statistics section. Specifically, the study subjects evidently recognized that changes in the fuzzy number most possible values visualized as triangles were more noticeable than those presented as vectors.

It seems that the most interesting results are associated with the interaction between *Change Type* and *Graphical Representation* [$F(2, 486) = 7.13$, $p = 0.0009$]. Fig. 4 graphically presents the differences in mean relative weights for this effect.

These data suggest that triangles were better suited for visualizing the change in the most possible value for CT_0→4 and CT_2→4 change types. However, the situation was reversed for the CT_0→2 level. In this case, vector representations were better rated than its triangular counterparts.

To further explore which of these differences were statistically significant a series of pairwise LSD post-hoc tests were carried out. The outcomes indicate that the suitability of triangle-based representation for the fuzzy number change in the most possible value is statistically significantly higher for CT_0→4 and CT_2→4 *Change Type* levels ($p < 0.001$ in both cases). Although, according to participants, the mean relative weights for vector representations were bigger for CT_0→2, the difference was statistically inconclusive ($p = 0.176$).

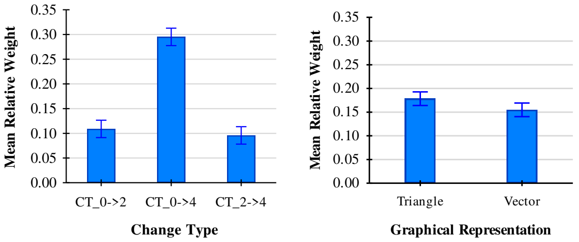


Fig. 3. Mean relative weights for *Change Type* [$F(2, 486) = 154.7$, $p < 0.0001$] and *Graphical Representation* [$F(1, 486) = 5.24$, $p = 0.0225$]. Bars denote 0.95 confidence intervals.

We also further examined the *Change Type* × *Gender* interaction effect [$F(2, 486) = 2.87$, $p = 0.058$], which is illustrated in Fig. 5. This graph suggest that women were more prone to perceive the changes in the most possible value as more salient than men if the changes were smaller, that is, amounted to two units. This phenomenon was inverted for the much bigger change involving four units.

Additional pairwise tests were employed to check which of the differences were statistically meaningful. The results of the LSD post-hoc tests, revealed that gender differences for smaller changes in most possible factors were irrelevant ($p > 0.15$). However, the difference between female and male study subjects for the bigger change was statistically significant at $p = 0.057$.

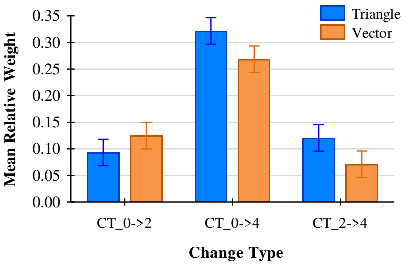


Fig. 4. Mean relative weights for *Change Type* × *Graphical Representation* interaction. Bars denote 0.95 confidence intervals [$F(2, 486) = 7.13$, $p = 0.0009$].

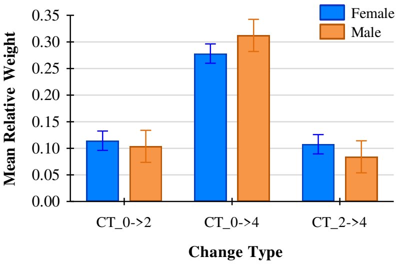


Fig. 5. Mean relative weights for *Change Type* × *Gender* interaction. Bars denote 0.95 confidence intervals [$F(2, 486) = 2.87$, $p = 0.058$].

The last significant effect from the performed analysis of variance, namely the *Graphical Representation* × *Gender* interaction, is visually demonstrated in Fig. 6. The graph shows that women considered triangular representations as better fitted to exhibit changes in most possible values than men did. On the other hand, males rated higher vector visualizations than women.

Again, we used pairwise LSD post-hoc analysis to verify the significance of the observed differences in mean relative weights. The findings, put together in Table VI, indicate that the observed gender discrepancies both for triangle and vector representations are statistically considerable ($p = 0.049$ and $p < 0.001$, respectively). Moreover, females significantly better perceived the change saliency if the fuzzy number change was presented as triangles than vectors ($p < 0.001$). For males, such an outcome was not detected ($p = 0.770$).

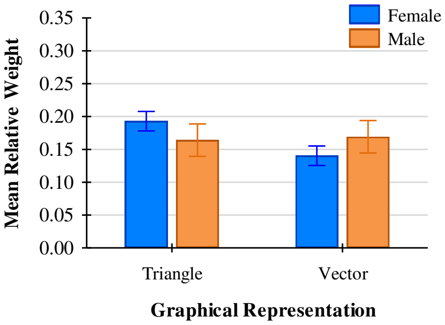


Fig. 6. Mean relative weights for *Graphical Representation* × *Gender* interaction. Bars denote 0.95 confidence intervals [$F(1, 486) = 7.78$, $p = 0.0055$].

V. Discussion of the Results and Conclusion

Our experimental study presented in this paper aimed to expand our understanding of how users perceive changes in triangular fuzzy numbers that are commonly used for expressing uncertainty. Specifically, we investigated two different visual representations of these fuzzy numbers, namely vector and classic, triangle-based ones along with various conditions concerned with their most possible values. The changes were depicted graphically through variations either in the vector angle or location of the maximum value of the membership function, and depended on the form of representation used. To gather study subjects' preferences towards the perceived saliencies regarding the change in the most possible value, we utilized pairwise comparisons within the AHP framework. Such an approach provided us with relative weights for all examined experimental conditions and allowed for comprehensive detailed formal statistical analysis. Gender differences in consistency ratios, computed according to this methodology, occurred to be statistically irrelevant. Furthermore, we identified statistically significant differences in the average relative scores for the two out of three factors studied, and three two-way interactions. The triangular-based fuzzy number representation, in general, were assessed as more appropriate than vectors for presenting changes in the most possible value. This factor considerably interacted with the *Change Type* effect. Triangles were better perceived in this experimental setup than vectors for large changes, and for changes more distant from the symmetrical case, that is vertical vectors and isosceles triangle. However, for smaller changes starting from that symmetrical situation, the reverse tendency was noticed suggesting that vectors could be better suited for detecting changes in such a case. Although this phenomenon was not statistically significant alone, the significance of the whole interaction certainly indicates that the application of vector representations should be studied in more detail in future research.

The general picture of the results obtained is further complicated by two additional gender interactions with *Graphical Representation* and *Change Type*. Females rated triangles considerably better than vectors, whereas for males the difference between these representations was unnoticeable. Moreover, triangles were relatively worse than vectors in terms of change saliency for men than women, but vectors were better perceived by male than female participants. As to the interaction with *Change Type* effect, men perceived big changes as more salient than women did. On the other hand, smaller changes of the most possible values were subjectively more noticeable for females than males. This suggest that females may be more sensitive in detecting smaller changes and less sensitive in identifying larger changes than males. This hypothesis, naturally, requires further empirical evidence. The discussed findings indisputably show that prospective research regarding graphical representations of uncertainty must involve gender-related analysis. This should be paid attention to already while designing and conducting the experiment, for instance, by ensuring similar number of man and women taking part in the study.

There are several possibilities of extending the presented study. Here, we confined only to the changes in one uncertainty feature of fuzzy numbers, that is, the most possible value. It is not clear, what would be the study subjects' perception of the saliency of changes if also the indeterminacy would be involved in the experimental setup. Thus, it should be subject to examination in future works as well. Since this study results showed that subjects' opinions depend considerably on the interaction between *Graphical Representation* and *Change Type* factors, another extension could include more levels of the *Change Type* effect to obtain a more comprehensive view of this outcome.

The results of this study contribute to the existing knowledge on how people perceive graphical representations of triangular fuzzy numbers. With the increasing use of artificial intelligence methods for handling inexact or ambiguous data, it has become crucial to develop suitable recommendations for user interfaces in computer programs that assist in solving problems with uncertainties.

The current investigation outcomes extend our understanding of individuals' opinions on the suitability of fuzzy number graphical visualizations in demonstrating uncertainty changes.

This can translate to provide appropriate recommendations for developing better graphical user interfaces with applications in such areas as production or project management. Given the presented results such interfaces should be carefully tailor-made individually for men and women.

REFERENCES

[1] M. Masmoudi and A. Haït, "Project scheduling under uncertainty using fuzzy modelling and solving techniques," Eng Appl Artif Intell, vol. 26, no. 1, pp. 135–149, 2013, doi: 10.1016/j.engappai.2012.07.012.

[2] E. W. Larson and C. F. Gray, Project management: the managerial process, 8th ed. McGraw-Hill Education, 2021.

[3] P. M. Rola and D. Kuchta, "Application of fuzzy sets to the expert estimation of Scrum-based projects," Symmetry-Basel, vol. 11, no. 8, pp. 1–22, 2019, doi: 10.3390/sym11081032.

[4] D. T. Hulett, Integrated cost-schedule risk analysis, 1st ed. London: Routledge, 2011. Accessed: Jun. 06, 2022. [Online]. Available: https://www.routledge.com/Integrated-Cost-Schedule-Risk-Analysis/Hulett/p/book/9780566091667

[5] M. A. Ajam, "Leading Megaprojects : A Tailored Approach," Leading Megaprojects, Jan. 2020, doi: 10.1201/9781003029281.

[6] I. Dikmen and T. Hartmann, "Seeing the risk picture: Visualization of project risk information," in EG-ICE 2020 Workshop on Intelligent Computing in Engineering, Proceedings, 2020.

[7] D. Streeb, M. El-Assady, D. A. Keim, and M. Chen, "Why Visualize? Untangling a Large Network of Arguments," IEEE Trans Vis Comput Graph, vol. 27, no. 3, pp. 2220–2236, 2021, doi: 10.1109/TVCG.2019.2940026.

[8] G.-P. Bonneau et al., "Overview and state-of-the-art of uncertainty visualization," Math Vis, vol. 37, pp. 3–27, 2014, doi: 10.1007/978-1-4471-6497-5_1.

[9] C. Ware, Information Visualization. Elsevier, 2013. doi: 10.1016/C2009-0-62432-6.

[10] R. Mazza, Introduction to information visualization. 2009. doi: 10.1007/978-1-84800-219-7.

[11] D. Kuchta, J. Grobelny, R. Michalski, and J. Schneider, "Vector and Triangular Representations of Project Estimation Uncertainty: Effect of Gender on Usability," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12747 LNCS, pp. 473–485, Jun. 2021, doi: 10.1007/978-3-030-77980-1_36.

[12] J. Schneider, D. Kuchta, and R. Michalski, "A vector visualization of uncertainty complementing the traditional fuzzy approach with applications in project management," Appl Soft Comput, vol. 137, p. 110155, Apr. 2023, doi: 10.1016/j.asoc.2023.110155.

[13] S. Chanas and P. Zieliński, "Critical path analysis in the network with fuzzy activity times," Fuzzy Sets Syst, vol. 122, no. 2, pp. 195–204, Sep. 2001, doi: 10.1016/S0165-0114(00)00076-2.

[14] D. Kuchta, "Use of fuzzy numbers in project risk (criticality) assessment," International Journal of Project Management, vol. 19, no. 5, pp. 305–310, Jul. 2001, doi: 10.1016/S0263-7863(00)00022-3.

[15] S. Chanas and J. Kamburowski, "The use of fuzzy variables in pert," Fuzzy Sets Syst, vol. 5, no. 1, pp. 11–19, Jan. 1981, doi: 10.1016/0165-0114(81)90030-0.

[16] B. Gładysz, "Fuzzy-probabilistic PERT," Ann Oper Res, vol. 258, no. 2, pp. 437–452, Nov. 2017, doi: 10.1007/S10479-016-2315-0/TABLES/3.

[17] J. W. Chinneck, "PERT for Project Planning and Scheduling," in Practical Optimization: a Gentle Introduction, Ottawa, Canada, 2016, pp. 1–11. Accessed: Feb. 24, 2023. [Online]. Available: https://www.optimization101.org/

[18] M. F. Shipley, A. de Korvin, and K. Omer, "BIFPET methodology versus PERT in project management: fuzzy probability instead of the beta distribution," Journal of Engineering and Technology Management, vol. 14, no. 1, pp. 49–65, Mar. 1997, doi: 10.1016/S0923-4748(97)00001-5.

[19] O. Pavlačka and J. Talašová, "Fuzzy vectors as a tool for modeling uncertain multidimensional quantities," Fuzzy Sets Syst, vol. 161, no. 11, pp. 1585–1603, Jun. 2010, doi: 10.1016/J.FSS.2009.12.008.

[20] O. Pavlačka, "Modeling uncertain variables of the weighted average operation by fuzzy vectors," Inf Sci (N Y), vol. 181, no. 22, pp. 4969–4992, Nov. 2011, doi: 10.1016/J.INS.2011.06.022.

[21] M. Arana-Jiménez, A. Rufián-Lizana, Y. Chalco-Cano, and H. Román-Flores, "Generalized convexity in fuzzy vector optimization through a linear ordering," Inf Sci (N Y), vol. 312, pp. 13–24, Aug. 2015, doi: 10.1016/J.INS.2015.03.045.

[22] J. Schneider and R. Urban, "Lévy Subordinators in Cones of Fuzzy Sets," J Theor Probab, vol. 32, no. 4, pp. 1909–1924, Dec. 2019, doi: 10.1007/S10959-018-0853-X/METRICS.

[23] J. Schneider and R. Urban, "A Proof of Donsker's Invariance Principle Based on Support Functions of Fuzzy Random Vectors," https://doi.org/10.1142/S0218488518500022, vol. 26, no. 1, pp. 27–42, Jan. 2018, doi: 10.1142/S0218488518500022.

[24] S. Zaleski and R. Michalski, "Success Factors in Sustainable Management of IT Service Projects: Exploratory Factor Analysis," Sustainability, vol. 13, no. 8, p. 4457, Apr. 2021, doi: 10.3390/SU13084457.

[25] J. Schneider, D. Kuchta, and R. Michalski, "A vector visualization of uncertainty complementing the traditional fuzzy approach with applications in project management," Appl Soft Comput, p. 110155, Feb. 2023, doi: 10.1016/J.ASOC.2023.110155.

[26] W. W. Koczkodaj, "Statistically Accurate Evidence of Improved Error Rate by Pairwise Comparisons," Percept Mot Skills, vol. 82, no. 1, pp. 43–48, Dec. 1996, doi: 10.2466/pms.1996.82.1.43.

[27] W. W. Koczkodaj, "Testing the accuracy enhancement of pairwise comparisons by a Monte Carlo experiment," J Stat Plan Inference, vol. 69, no. 1, pp. 21–31, Jun. 1998, doi: 10.1016/S0378-3758(97)00131-6.

[28] R. Michalski, "Examining users' preferences towards vertical graphical toolbars in simple search and point tasks," Comput Human Behav, vol. 27, no. 6, pp. 2308–2321, Nov. 2011, doi: 10.1016/j.chb.2011.07.010.

[29] R. Michalski, "The influence of color grouping on users' visual search behavior and preferences," Displays, vol. 35, no. 4, 2014, doi: 10.1016/j.displa.2014.05.007.

[30] J. Grobelny and R. Michalski, "The role of background color, interletter spacing, and font size on preferences in the digital presentation of a product," Comput Human Behav, vol. 43, pp. 85–100, Feb. 2015, doi: 10.1016/J.CHB.2014.10.036.

[31] J. Grobelny and R. Michalski, "Various approaches to a human preference analysis in a digital signage display design," Human Factors and Ergonomics in Manufacturing & Service Industries, vol. 21, no. 6, pp. 529–542, Nov. 2011, doi: 10.1002/HFM.20295.

[32] M. Płonka, J. Grobelny, and R. Michalski, "Conjoint Analysis Models of Digital Packaging Information Features in Customer Decision-Making," Int J Inf Technol Decis Mak, Nov. 2022, doi: 10.1142/S0219622022500766.

[33] T. L. Saaty, "A scaling method for priorities in hierarchical structures," J Math Psychol, vol. 15, no. 3, pp. 234–281, Jun. 1977, doi: 10.1016/0022-2496(77)90033-5.

DOI: 10.15439/2023F8647
ISSN 2300-5963 ACSIS, Vol. 35

Emotion-Based Literature Books Recommender Systems

Elena-Ruxandra Luţan
0000-0001-5363-9930
Department of Computers and Information Technology
University of Craiova, 200585, Craiova, Romania
Email: elena.ruxandra.lutan@gmail.com

Costin Bădică
0000-0001-8480-9867
Department of Computers and Information Technology
University of Craiova, 200585, Craiova, Romania
Email: costin.badica@edu.ucv.ro

***Abstract*—In this paper we propose two book recommendation methods based on emotions extracted from user reviews, using content-based filtering and collaborative filtering. The methods were experimentally evaluated on our own dataset that we collected from *Goodreads* – a popular website with large database of books and readers reviews. We created an experimental setup where the recommendation algorithms for carrying out the evaluation using two proposed evaluation metrics: coverage and average recommendations similarity.**

I. Introduction

PEERS feedback from interactions mediated by social media plays an increasingly important role in how we choose to approach different aspects of our lives. The lack of personal experience when taking decisions often leads to seeking the experience of peers by means of recommendations. Such recommendations can emerge in various forms including word of mouth, surveys, printed or online reviews, thus actively supporting our day-to-day life [13].

The aim of a recommender system is to provide meaningful recommendations to the users based on the products which might interest them, the recommendations trustworthiness being a mandatory characteristic. The design of a recommender system varies depending on the nature of products for which recommendations will be issued [9].

In this paper, we are focusing on a specific category of products, literature books. We propose and evaluate two recommender systems that incorporate emotion information based on two different recommendation techniques: content-based filtering and collaborative filtering.

Content-based filtering refers to recommending products which are similar to the product that is being watched [2]. Our proposed content-based filtering recommendation algorithm must observe the user interaction with a book and identify similar books based on certain book characteristics, such as book title or book author(s).

Collaborative filtering aims to mine the most similar users with the user of interest and to observe their preferences. Then, these preferences can be used to make predictions about what the user of interest might enjoy [3]. This allows the recommender system to also focus on products that the user of interest has not yet interacted with.

We appreciate that the outcome of our research is both fascinating and useful, because our methods use social generated data for identifying the similarities between the books, in addition to general publisher details about a given book (book title, author or genre).

The experimental dataset was collected from a popular book-oriented website, *Goodreads*, using our own customized web scraper.

The paper is structured as follows. In Section II, we present related works. Section III describes our proposed book recommendation algorithms, using content-based filtering and collaborative filtering. In Section IV, we provide an overview of the dataset and then we discuss the experimental results. The last section presents our conclusions.

II. Related Works

Chhavi Rana and Sanjay Kumar Jain [12] propose a system which makes content-based book recommendations based on the user navigation pattern. The system analyzes user's behaviour and then it predicts the category of books that would interest the user using content-based filtering. The authors observe the lack of accuracy of content-based recommendations, as after a certain amount of time, the users will be recommended the same similar items. Therefore they introduce a temporal dimension, which means that user navigation and most visited links are periodically analyzed and revised when using the content-based approach to make recommendations.

In [8], Jessie Caridad Martin Sujo and Elisabet Golobardes i Ribe present a system which recommends the book that best suits the reader based on the semantics of his or her writing style. They use posts from Twitter social network in order to determine the psychological profile of the user. The authors use a database consisting in characters text, associated personality type and corresponding book. Their proposed method computes the similarity between the Twitter post text and the cases database in order to recommend the most suitable book to the user.

An Enhanced Personalized Book Recommender System (EPBRS) is described in [15]. The proposed system uses the a similarity function based on Euclidean distance in order to identify users with similar interests. The recommendations are done using collaborative filtering by considering the books preferred by similar users. A dataset of reviews, users and associated ratings from Amazon bookstore was used for experiments. The book ratings are considered as features when making the predictions.

In [16], authors propose a system which is able to provide replies to queries regarding products details. The answer that is returned to the query is actually a review available for the product, which contains the relevant details. For experiments, they use two neural models, a simple model (NNQA) and a Transformer-based model (BERTQA). These models are evaluated regarding their ability to find the relevant reviews.

Anil Kumar and Sonal Chawla [6] make an analysis of the recommendation techniques which are most frequently used for book recommender systems. They also propose a new book recommender system based on Hybrid recommendation technique. The Hybrid recommender system works as follows: when the user searches for a book, the system computes the list of book recommendations using collaborative filtering on book ratings. Then the positive and negative user reviews for each book are identified such that the recommendation list will be sorted based on the number of positive reviews. The user is displayed the book recommendation list together with the details of the searched book.

Harsh Dubey and Suma Kamalesh Gandhimathi [4] propose a recommender system which uses Deep Learning GPT3 (Generative Pre-trained Transformer). The project refers to building an application which finds books that are similar to a certain book provided as input. On a web interface, the user must describe a book that he or she has enjoyed reading. OpenAI API module is used for generating the recommendations of books that are similar with the input book description, and the top 3 recommendations are displayed together with details about the books availability obtained using Google Books API.

III. System Design

Three main recommender system techniques can be identified: content-based filtering, collaborative filtering and a combination of both [2]. They differ in their data sources, as well as in how these data sources are interpreted, analyzed and processed for building the recommendations [9].

In this paper we propose two recommender system algorithms for literature book recommendation, corresponding to the two distinct techniques: content-based filtering and collaborative filtering. Both algorithms incorporate the emotion categorization of each book as an important feature for determining similarities between books.

The emotions are extracted from online book reviews and then used for creating an emotion-based categorization of books using the system we previously proposed in [7]. In total, there are 35 emotions considered: ’cheated’, ‘singled out’, ‘loved’, ‘attracted’, ‘sad’, ‘fearful’, ‘happy’, ‘angry’, ‘bored’, ‘esteemed’, ‘lustful’, ‘attached’, ‘independent’, ‘embarrassed’, ‘powerless’, ‘surprise’, ‘fearless’, ‘safe’, ‘adequate’, ‘belittled’, ‘hated’, ‘codependent’, ‘average’, ‘apathetic’, ‘obsessed’, ‘entitled’, ‘alone’, ‘focused’, ‘demoralized’, ‘derailed’, ‘anxious’, ‘ecstatic’, ‘free’, ‘lost’, ‘burdened’.

Field Name	**Field Description**
Book Id	The id which uniquely identifies the book
Book URL	The Goodreads URL of the book
Book Title	The title of the book
Book Series	The book series name
Book Author	The author(s) of the book
Book Overall Rating	The book rating, a number in interval [1, 5]
Book Ratings Number	The number of ratings available on Goodreads for the book
Book Reviews Number	The number of reviews available on Goodreads for the book
Book Full Description	The description of the book
Book Genres	Top 10 genres available for the book on Goodreads website
Book Pages	The number of pages of the book
Book Year	The year in which the book was published
Emotions	The book emotions computed using [7]

TABLE I
Book Entity Description

Field Name	**Field Description**
Review Id	The id which uniquely identifies the review
Review URL	The Goodreads URL of the review
Book Id	The id of the book for which the review is given
Author Id	The id of the user who wrote the review
Review Stars	The rating given by the review author, as integer in interval [1, 5]
Review Date	The date when the review was written
Review Tags	Review tags or keywords given by the review author
Review Content	The review (text) provided by the review author

TABLE II
Review Entity Description

The emotion extraction workflow takes as input the review, performs standard NLP text preprocessing techniques (tokenization, lower casing, removal of stop words) and determines the emotions present in the text by making word-matching with a list of adjectives and their corresponding emotion.

Our proposed recommendation algorithms were validated on the experimental dataset previously introduced in [7]. This data set was collected by us from Goodreads website using our own customized web scraper.

The dataset contains tabular data describing two entities, Book and Review, which are interrelated by a one-to-many relationship. For both entities, several parameters available on Goodreads website were extracted and captured as separate columns. They are described in Tables I and II.

A. Content-Based Filtering

Content-Based Filtering approach recommends items considering user preferences. The hypothesis of Content-Based Filtering is that users are usually more interested in those items that are similar to items they liked in the past [3].

We analyzed which fields of each book item can be used to better define its characteristics. We decided to use the Book Title, Book Series, Book Author, as well as the main emotions triggered by the book reading, which are computed during the extraction of sentiments from the book reviews.

The recommendation algorithm takes as input a review of a given user for a given book which is available in the database. The review consists of two components: a number in range $[1, 5]$ which represents a scaled value capturing how much the user liked the book (which will be referred as Review Stars) and the opinion of the user expressed in natural language text (which will be called Review Content).

The general idea of the algorithm is to use the input review in order to decide how much the user enjoyed the current book in order to recommend other relevant books to the user.

Algorithm 1 Content-Based Filtering Algorithm

```
1: if Review Stars $< 3$ then
2:     Extract the emotions from the Review Content
3:     if Review Content Emotions match Book Emotions $> threshold$ then
4:         Recommend a book which differs from the rated book
5:     else
6:         Recommend a book similar with the rated one
7:     end if
8: end if
9: if Review Stars $\geq 3$ and Review Stars $<5$ then
10:    Recommend a better book similar with the rated one
11: end if
12: if Review Stars $= 5$ then
13:    Recommend a book similar with the rated one
14: end if
```

The Review Stars is used to classify the level of satisfaction that the book provided to the user, as follows:

- The user did not like the book (Review Stars = 1 or 2).
- The user liked the book, but was not over-joyed (Review Stars = 3 or 4).
- The user loved the book (Review Stars = 5).

We detail each case of the algorithm used for Content-Based Filtering (Algorithm 1). The algorithm contains 3 main IF clauses that deal with each one of the possible three satisfaction levels extracted from the input review.

The first IF clause (line 1) refers to the case when the user did not like the book (Review Stars = 1 or 2). In this case, we need to know what caused this dissatisfaction. We will take into consideration the Review Content and extract the emotions. In order to decide what kind of books to recommend, we decided to compare the Review Content Emotions with the Book Emotions. In case they match with high value, we considers this indicates that the user did not like the overall idea of the book, the kind of emotions that the book made him or her feel. In this case, we will recommend a completely different book emotions-wise, because it is most likely that the user will prefer something different. If the Review Content Emotions and the Book Emotions do not match, we interpret this as indicating that the user did not perceive the book as expected; maybe he or she did not actually understand the meaning of the book. In this case, we will recommend a book that is similar with the current one, as we guess that the user is likely to enjoy a new book which provides emotions rather close with the ones present in current book.

The second IF clause (line 9) refers to the case when user liked the book, but was not over-joyed by it. The aim of the recommender system is to provide recommendations for products which are likely to offer the greatest experience. For this reason, we will recommend books which provide similar sentiments, but are higher in ratings than the current book.

The last IF clause (line 12) refers to the case when the user loved the book. In this case, we recommend to the user a book which is very similar to the current one, because he or she is likely to enjoy it as much.

When recommending new books, we include only books that the user has not seen, i.e. books to which the user has not yet given reviews.

By applying the content-based filtering algorithm (Algorithm 1) we obtain a list of books which are considered the user might enjoy, and the top 5 books are displayed to the user as recommendations.

B. Collaborative Filtering

Collaborative Filtering method aims to find similarities between users based on the user-item interaction [1]. The system divides the users into clusters by considering their past interactions and makes recommendations according to the preference of the cluster the user belongs [3].

Similarly to the Content Based Filtering method, the Collaborative Filtering algorithm takes as input a review of a user for a given book which is available in the database, with its two components, Review Stars and Review Content. Its pseudocode is presented as Algorithm 2.

Algorithm 2 Collaborative Filtering Algorithm

```
Compare the user of interest with all the users who provided reviews for the given book
if A similar user which matches $> threshold$ is found then
    The users are similar, recommend a book that the similar user liked
end if
```

So, we are interested in evaluating the similarity of the user of interest with other users from the database. In our model, the similarities between the user of interest and each of the users in database are computed based on the emotions that exist in their reviews given for the given book. Then we determine the 5 topmost users similar with the user of interest and we analyze their preferences. This means that we analyze their reviews to determine which other books these top 5 users rated.

We consider that a user liked a book if he or she provided 4 or 5 stars. Therefore, from all the books rated by the top 5 users, we will select only those which got 4 or 5 stars in the reviews. This will lead to a set of books which we consider the user of interest might enjoy.

In order to offer the greatest experience, we decided to filter the set of books according to the Book Overall Rating, assuming that higher rating means better book. Book Overall Rating is an attribute present for each book in our dataset and it represents the book rating as it is recorded on the *Goodreads*.

When a new review is given, the first step is to compute the emotions present in the review. The content of the review is pre-processed by removing unnecessary text from the review, tokenizing the review text into words and removing the stop words. Then we extract the emotions from the pre-processed text, using our own Emotions Extraction Algorithm introduced in [7]. Our emotion model is based on a list of maximum 35 emotions and their weights.

Following, we extract from the reviews dataset the set of reviews available for the rated book. This subset will be used with the purpose of finding similar users with the user who provided the review. Two users are considered similar if they provided review for the same book and the emotions which are available in their reviews match at least 50%.

The next step of the collaborative filtering algorithm is to identify the books that similar users liked in order to recommend them to the user of interest. In order to make recommendations, for each of the matching users we identify the rated books which received more than 3 stars (as we assume that the similar users liked these books) and were not yet rated by the user of interest, and we add them to the list of recommendations.

At this stage, we have obtained a list of recommendations which can be provided to the user of interest. Initially, we considered the default ordering of this list according to how were the books appended to the list. According to this ordering, the books preferred by the most similar users are located as topmost entries of this list. However, after a deeper analysis, we realized that this might not be the best possible ordering, because we would rely only on the most similar users in order to make recommendations, and this would restrict too much the space of possible recommendations. Therefore, we decided to define a better way to order the recommendations such that to not rely only on the preferences of the single topmost similar user. Consequently, we considered that a possibility is to order the recommendations list by the Book Overall Rating value, before providing the top recommendations to the user.

If the recommendation list does not contain the minimum number of 5 recommendations, the list is completed by adding the books with the highest rating available in the dataset.

IV. Experiments and Discussions

A. Dataset Preparation

The data set was pre-processed before the application of the recommendation algorithm. The aim of the pre-processing is to compute the books similarity matrix that contains the similarity value for each pair of books in the dataset.

We did not use all the fields of a book entity (see Table I) for our recommendation algorithms. Therefore we selected only those book features which are relevant and we combined them into a single text field. We consider to be relevant the following fields: Book Title, Book Series, Book Author and Emotions. The resulting string is stored into the books dataset as an additional column named “Combined Features”.

Then we converted each “Combined features” field of a book into a vector of token counts. We applied this processing for each book of the dataset, thus obtaining a matrix T of token counters with elements natural numbers. The total number of tokens is equal to the size of the vocabulary that is found by analyzing the “Combined Features” field of each book. So, if there are n books and the size of the vocabulary is m the resulting matrix of token counts will have size $n \times m$. The count matrix was created using *CountVectorizer* class of *Scikit-learn* library available in Python [10].

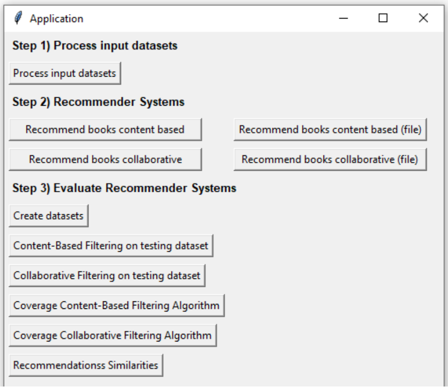

Application
Step 1) Process input datasets
Process input datasets
Step 2) Recommender Systems
Recommend books content based
Recommend books content based (file)
Recommend books collaborative
Recommend books collaborative (file)
Step 3) Evaluate Recommender Systems
Create datasets
Content-Based Filtering on testing dataset
Collaborative Filtering on testing dataset
Coverage Content-Based Filtering Algorithm
Coverage Collaborative Filtering Algorithm
Recommendationss Similarities

Fig. 1. Application Main Panel

Each row i of matrix T is a vector of counters describing book i. The similarity of two books i and j can be determined by applying a similarity measure to the vectors represented by rows i and j of T. In our implementation we have used the cosine similarity measure. If there are n books then the similarity matrix is a squared and symmetric matrix S of size n with real values in interval $[0, 1]$. We determined the books similarity matrix and we saved it into variable *cosine_similarity_matrix* [5].

For each input book $1 \leq i \leq n$, the books that are most similar with it can be determined by examining the row i of matrix S consisting of elements $S_{i,j}$ for all $1 \leq j \leq n$ of higher value.

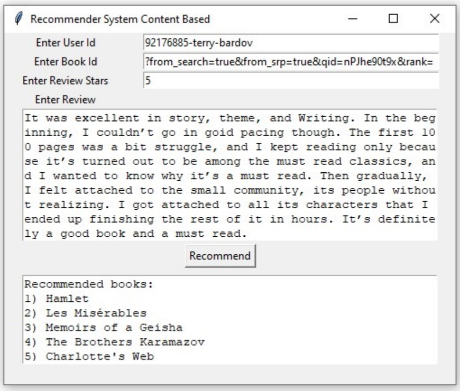
B. Application Interface

In order to simplify the use of the proposed system, we developed a convenient application interface using *Tkinter* Python Library, which can be seen in Figure 1.

The button “Process input dataset” from Step 1 refers to processing the reviews and books dataset by extracting the emotions from reviews and categorizing the books emotions based using our approach previously introduced in [7].

Step 2 refers to applying one of the recommendation algorithms. For both recommendation algorithms, we have created two approaches: the first that takes into consideration only one review manually inserted by the user, and the second that takes as input a list of reviews provided in a *CSV* file. From the main console of the application, the user can choose which function to execute, by using the corresponding button on Step 2.

Using the first approach (a manually inserted review), another panel will appear on the screen (Figure 2). The user has to insert using the keyboard the following information: the user id, the book id, the number of stars and the review content. If the Recommender System would be used in a real setting, the information about the user id and book id would be automatically collected from the context (current



Enter User Id: 92176885-terry-bardov

Enter Book Id: ?from_search=true&from_srp=true&qid=nPJhe90t9x&rank=

Enter Review Stars: 5

Enter Review:

```
It was excellent in story, theme, and Writing. In the beg
inning, I couldn’t go in goid pacing though. The first 10
0 pages was a bit struggle, and I kept reading only becau
se it’s turned out to be among the must read classics, an
d I wanted to know why it’s a must read. Then gradually,
I felt attached to the small community, its people withou
t realizing. I got attached to all its characters that I
ended up finishing the rest of it in hours. It’s definite
ly a good book and a must read.
```

```
Recommended books:
1) Hamlet
2) Les Misérables
3) Memoirs of a Geisha
4) The Brothers Karamazov
5) Charlotte's Web
```

Fig. 2. Application Panel for insertion of review details and results obtained using Content-Based Filtering Recommendations Algorithm

book selection and authentication information), but since our project is focused on the experimental evaluation of our algorithms (not on the actual graphical user interface of a Web-based deployed system), this information needs to be manually inserted by the user.

After filling in all the required fields, the user must press the button “Recommend” thus triggering the recommendation process. The Recommender System processes the input fields, applies the selected recommendation algorithm and displays the top 5 recommendations in the lower part of the panel.

The second implementation approach uses a list of reviews given in an input CSV file and applies the selected recommendation algorithm to each given review, rather than using a single review which was provided by the user as input.

Since using the second approach the results cannot be displayed in the same way as for the first approach, we had to find a meaningful way to display the output recommendations. We decided to store the results as a list inside an output text file. For readability, we also added to this file the information about the processed review, respectively Author Id, Book Id, Review Stars, Review Content and Emotions, together with the recommendations themselves.

C. Experimental Results

The dataset consists in 78 books and 6566 associated reviews, collected from *Goodreads* website. For majority of books (71 books) the reviews dataset contains 90 reviews, while for 7 books there are less than 90 reviews available.

These reviews were written by a total of 2658 users. 1755 users have written only 1 review, 795 users have written between 2 and 10 reviews, while 108 users have written more than 10 reviews (between 11 and 74 reviews).

The experimental setup is configured on Step 3 of the application workflow: Evaluate Recommender Systems (Figure 1). Firstly, the training and testing datasets have to be defined.

We have split the *Goodreads* dataset of 6566 reviews as 80% for training and 20% for testing. As different number of reviews are contained in the data set for each separate book, training - testing split was done for each book. Following this splitting procedure, the training dataset contains 5267 reviews and the testing dataset contains 1299 reviews.

The training reviews dataset was used for defining the book emotions feature, which means that the emotions were extracted from the review content and are attached to the book using our procedure previously introduced in [7].

The testing reviews dataset contains those reviews based on which the system will provide recommendations in order to perform the experimental evaluation of our proposed recommendation algorithms. Each entry in the testing dataset can be seen as a new review that is currently added by a user who expects to receive book recommendations.

Let us define the following parameters that are used for the rigorous definition of our proposed evaluation metrics:

- *Recommendation space* R refers to the total number of possible recommendations, i.e. the total number of books available in the books dataset (in our case 78).
- *User input space* U refers to the total number of user inputs u. A user input is a new review added for a certain book from the dataset.

$$u = (book, review), where\ book \in R$$

- *Test space* T refers to the subset of the input space $T \subset R$ used for experimental evaluation.
- A *recommendation* $f_i(u)$ refers to the output recommendation obtained when applying recommendation algorithm i. The output is a set of 5 books $r_i \in R$. $i = 1$ denotes the Content-based Filtering Algorithm, while $i = 2$ denotes the Collaborative Filtering Algorithm.

$$f_i \colon U \to R^5$$

$$f_i(u) = (r_1, r_2, r_3, r_4, r_5)$$

- *Total number of unique recommendations* $TNUR_i$ refers to the amount of unique books from the dataset which are returned as recommendations by algorithm i. In our case, $TNUR_1$ refers to the books returned as recommendations by Content-based Filtering algorithm and $TNUR_2$ refers to the books returned as recommendations by Collaborative Filtering algorithm.

$$TNUR_i = \bigcup_{u \in U} \{f_i(u)\}$$

- *Recommendations similarity* s refers to the similarity between recommendations $f_1(u)$ and $f_2(u)$ provided for the same user input u using the Content-based Filtering algorithm, respectively Collaborative Filtering algorithm.

$$s \colon R^5 \times R^5 \to [0, 1]$$

s is determined using Jaccard index.

$$s(f_1(u), f_2(u)) = \frac{|f_1(u) \cap f_2(u)|}{|f_1(u) \cup f_2(u)|}$$

Considering that each of the two algorithms provides a list of 5 recommendations, it follows:

$$s(f_1(u), f_2(u)) = \frac{|f_1(u) \cap f_2(u)|}{10 - |f_1(u) \cap f_2(u)|}$$

We propose two performance measures for evaluating our recommendation algorithms, as follows:

- *Coverage* C_i determines the proportion of books from R that the system was able to recommend using recommendation algorithm *i*.
$$C_i = \frac{|TNUR_i|}{|R|}$$
- *Average Recommendations Similarity* ARS is the average of the similarity between recommendations provided using Content-based Filtering and Collaborative Filtering algorithms.
$$ARS = \frac{1}{|T|} \sum_{u \in T} s(f_1(u), f_2(u))$$

For the Content-based Filtering algorithm, we have obtained a coverage of 96.153%, which means that that when suggesting book recommendations for the 1299 testing reviews, 75 books from the dataset were recommended.

The same coverage 96.153% is obtained for Collaborative Filtering algorithm (just simple coincidence).

Note that even if we obtained equal coverage values for both recommendation algorithms, the books which are not recommended by these algorithms are different. For the Content-Based Filtering algorithm the 3 not-recommended books from the dataset were: index 24 ("The Handmaid's Tale" by Margaret Atwood), 54 ("Charlie and the Chocolate Factory" by Roald Dahl) and 77 ("The Story of My Life" by Helen Keller), while for the Collaborative Filtering algorithm the books from the dataset which were not recommended are: index 63 ("The Good Earth by Pearl S. Buck | Summary & Study Guide"), 64 ("Sidekick to Mockingjay by Suzanne Collins" by Katherine R. Miller) and 70 ("The Road by Cormac McCarthy | Summary & Study Guide").

The *Average Recommendations Similarity* between the books recommendations received using the two algorithms is 5.71%. This rather low value was somehow expected. It shows that applying both recommendations algorithms on the same user input review generates rather different recommendations. In total, for our 1299 input reviews, 6495 recommendations were obtained using Content Based Filtering and 6495 were obtained using Collaborative Filtering, as both recommendations algorithms provide to the user the top 5 recommendations. Out of the 6495, only 602 were identical.

V. Conclusions

In this contribution, we presented two different algorithms for making valuable book recommendations considering also the emotions extracted from online book reviews submitted by book readers. The proposed recommendations algorithms are based on content-based filtering and collaborative filtering.

The emotions present in online book reviews are used to create an emotion-based categorization of books. Then the emotion categorization is considered as an additional book feature when computing the similarity between two different books, together with the book title and the book author(s).

We created an experimental setup using a books and reviews dataset that we collected from *Goodreads* website using our customized web scraper. We divided the reviews dataset into two groups: training and testing. The training consists in extracting the emotions present in the reviews and using them to categorize the books, while the testing refers to giving the reviews one by one as input to our system and receiving recommendations.

We proposed two performance metrics *Coverage* and *Average Recommendations Similarity*. Our experimental evaluation shows a good books dataset coverage on both recommendation algorithms, as almost all books from the dataset are given as recommendations for all the possible user inputs. On the other hand, the *Average Recommendations Similarity* metric provides low similarity values of recommendations generated using Content-based Filtering and Collaborative Filtering. This is expected, considering the different nature of the recommendation methods involved.

References

[1] Aggarwal, C.: Recommender Systems The Textbook (2016) Springer International Publishing

[2] Agrawal, R.: How to Build a Book Recommendation System (2021) https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-system-unsupervised-learning-project/

[3] Dey, V.: Collaborative Filtering vs Content-Based Filtering for Recommender Systems (2021) https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/. Last accessed 10 Feb 2023

[4] Dubey, H., Gandhimathi, S. K.: Book Recommendation System Using Deep Learning (GPT3) International Research Journal of Engineering and Technology (IRJET), vol. 9(5) (2022)

[5] Karbhari, V.:What is a cosine similarity matrix? (2020) https://medium.com/acing-ai/what-is-cosine-similarity-matrix-f0819e674ad1. Last accessed 10 Feb 2023

[6] Kumar, A., Chawla, S.: Framework for Hybrid Book Recommender System based on Opinion Mining. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Vol.8(4) (2019) 10.35940/ijrte.D7518.118419

[7] Luţan, E.-R., Bădică, C.: Emotion-Based Literature Book Classification Using Online Reviews. Electronics 2022, 11, 3412. https://doi.org/10.3390/electronics11203412

[8] Martín, J.,Ribé, E.: BRAIN L: A book recommender system (2023) 10.48550/arXiv.2302.00653

[9] Melville, P., Vikas, S.: Recommender systems. Encyclopedia of machine learning 1 pp. 829-838 (2010)

[10] Movie Recommendation Model Using Cosine_Similarity and CountVectorizer: Scikit-Learn (2019) https://regenerativetoday.com/movie-recommendation-model-using-cosine_similarity-and-countvectorizer-scikit-learn/ Last accessed 31 Mar 2023

[11] Polignano, M., Narducci, F. de Gemmis, M. Semeraro, G.: Towards Emotion-aware Recommender Systems: an Affective Coherence Model based on Emotion-driven Behaviors. Expert Systems with Applications 2021, 170, 114382, https://doi.org/10.1016/j.eswa.2020.114382

[12] Rana, C., Jain, S. K.: Building a Book Recommender system using time based content filtering. WSEAS Transactions on Computers 11.2 (2012): 27-33.

[13] Resnick, P., Hal R. V.: Recommender systems. Communications of the ACM 40.3 pp. 56-58 (1997)

[14] Roy, D., Dutta, M.: A systematic review and research perspective on recommender systems. Journal of Big Data 9, 59 (2022). https://doi.org/10.1186/s40537-022-00592-5

[15] Usman, A., Roko, A., Muhammad, A.B. Almu, A.: Enhancing Personalized Book Recommender System. Int. J. Advanced Networking and Applications, vol.14(03), pp. 5486–5492 (2022)

[16] Zhang, S., Lau, J. H., Zhang, X. J., Chan, J., Paris, C.: Discovering Relevant Reviews for Answering Product-Related Queries. 2019 IEEE International Conference on Data Mining (ICDM) 10.1109/ICDM.2019.00192

DOI: 10.15439/2023F3561
ISSN 2300-5963 ACSIS, Vol. 35

BERT-CLSTM model for the classification of Moroccan commercial courts verdicts

Taoufiq El Moussaoui
0000-0003-4879-7111
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz
Sidi Mohamed Ben Abdellah University
Fez, Morocco
Email: taoufiq.elmoussaoui@ucmba.ac.ma

Loqman Chakir
0000-0002-8261-9370
LISAC Laboratory, Faculty of Sciences Dhar El Mahraz
Sidi Mohamed Ben Abdellah University
Fez, Morocco
Email: loqman.chakir@usmba.ac.ma

Abstract—**The exponential growth of data generated by the Moroccan commercial court system, coupled with the manual archiving of legal documents, has led to increasingly complex information access. As data classification becomes imperative, researchers are exploring automatic language processing techniques and refining text classification methods. In this study, we propose a BERT-CLSTM model for the classification of Moroccan commercial court verdicts. By adding a Convolutional Long Short-Term Memory Network to the task-specific layers of BERT, our model can get information on important fragments in the text. In addition, we input the representation along with the output of the BERT into the transformer encoder to take advantage of the self-attention mechanism and finally get the representation of the whole text through the transformer. The proposed model outperformed the compared baselines and achieved good results by getting an F-measure value of 93.61%.**

I. Introduction

TEXT classification is a machine-learning task that assigns a document to one or more predetermined categories based on its content. It is a key problem in natural language processing, with diverse applications such as sentiment analysis, email routing, offensive language detection, spam filtering, news classification, and language identification.

Text mining [1] is one of the most important approaches for analyzing massive volumes of textual data. Also, it is used to discover previously unknown relationships and propose solutions to aid decision-making. Many technologies are utilized in the text mining process to attain these aims. Text summarization, translation, categorization, and information extraction are a few examples. This paper's content is limited to text classification.

Despite the advances made in text categorization performance, there is still significant potential for improvement, particularly in the Arabic language. According to Internet World Stats, Arabic is the fourth most common language online, with over 225k users, representing 5.2% of all Internet users as of April 2019. Arabic NLP is still a challenging task due to the Arabic language's richness, complexity, and complicated morphology. Arabic features are assorted in abundance aspects compared to other languages. There are several forms of grammatical, variations of synonyms word, and numerous meanings of words which differ based on factors such as the order of the word.

Traditional text classification methods use sparse vocabulary features to represent documents and treat words as the smallest unit. Documents represented by such approaches typically have high dimensionality and sparse data, so the classification accuracy is low. Later, with the rise of distributed representation, the usage of high-dimensional dense vector representation documents such as the word2vec or Glove models gradually becomes mainstream. The word vectors trained using this type of method represent the contextual semantic information of the text. Recently, with the emergence of deep learning, more researchers use deep learning neural networks for text classification, such as convolutional neural networks (CNN) and recurrent neural networks (RNN). The method using deep learning for text classification involves feeding text into a deep network to generate a representation of the text and then feeding the text representation into the softmax function to calculate the probability of each category. CNN-based models [2], [3], [4] may generate text representations with local information, whereas RNN-based models [5], [6] generate text representations with long-term information.

Nowadays, the process of classifying verdicts of Moroccan commercial courts is done manually. Court staff read each verdict and classify it into a predefined category. This process has many drawbacks. First, the processing time is long. Second, court staff may make mistakes while filing the document, and third, the confidentiality of citizens' data is not respected. Based on the disadvantages of the current system, we propose a BERT-CLSTM model that can provide the same service in a short time. The proposed model combines the advantages of CNN and RNN.

The rest of the paper is structured as follows: Section II presents the related literature. In section III, we give details on the dataset created for this task of classification, the data preprocessing, and the model. Section IV presents the experimental setup, evaluation measures, and results, and we discuss them. Finally, conclusions are drawn in Section V.

II. Related Work

Over the past few years, several researchers have addressed the issue of automatic categorization of legal data and the exploitation of these huge amounts of court-generated data to assist in decision-making. The study presented in [7] aims to classify Arabic news articles based on their vocabulary features. They employed multi-label classifiers like Logistic Regression and XGBoost, with XGBoost achieving the highest accuracy at 84.7%, while Logistic Regression scored 81.3%. Additionally, ten neural networks were constructed, and CGRU proved to be the top-performing multi-label classifier with an accuracy of 94.85%.

Research [8] introduce AraLegal-BERT, a bidirectional encoder Transformer-based model that has been thoroughly tested and carefully optimized to amplify the impact of NLP-driven solution concerning legal documents. They fine-tuned AraLegal-BERT and evaluated it against three BERT variations for the Arabic language. The results show that the base version of AraLegal-BERT achieves better accuracy than the general and original BERT over the Legal text.

El-Alami et al [9] propose an Arabic text classification method based on Bag of Concepts and deep Autoencoder representations. It incorporates explicit semantics relying on Arabic WordNet and exploits Chi-Square measures to select the most informative features. To produce a high-level representation, they applied successive stacks of Restricted Boltzmann Machines (RBMs). Experiments showed that using the Autoencoder as a text representation model combined with Chi-Square and classifier outperformed state-of-the-art techniques.

Hazm et al [10] evaluated Arabic user comments on Twitter using a common form of Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM). Experiments revealed that LSTM outperforms standard approaches in terms of accuracy while requiring less parameter computation, less working time, and more efficiency.

Alhawarat and Aseeri [11] implemented a CNN multi-kernel architecture with word embedding (n-gram) to classify Arabic documents of news. Regarding the current studies on Arabic text classification, their approach achieves very high precision using 15 of the publicly available datasets.

Galal et al [12] concentrated on classifying Arabic Text using a convolution neural network (CNN). They implemented GStem a new algorithm focused on extra Arabic letters and word embedding distances to group related Arabic words. Their studies have shown that it improves the accuracy of the CNN model when using it as a preprocessing stage.

Boukil et al [13] proposed a technique for categorizing Arabic datasets. They utilized an Arabic stemming algorithm to select and reduce features and employed Term Frequency Inverse Document Frequency (TFIDF) for feature weighting. Their study compared the CNN model and standard machine learning methods on a benchmark dataset. The authors found that the CNN model outperformed traditional methods, especially for large and complex datasets.

Al-khurayji and Sameh [14] proposed a novel method for Arabic text classification using Kernel Naive Bayes (KNB) classifier. Their approach involved preprocessing documents through tokenization, stop word removal and word stemming. They utilized Term Frequency-Inverse Text Frequency (TF-IDF) for feature extraction and represented terms as vectors. Experimental results on the collected dataset demonstrated the superiority of their methodology, showing excellent precision and efficiency compared to other baseline classifiers.

III. Research Methodology

In this section, we present the corpus created to train and evaluate our model, also the data preprocessing process, and finally, we explain the model architecture.

A. Dataset

The dataset created to train and evaluate our proposed classifier is a collection of Arabic verdicts issued from the Moroccan commercial courts. It consists of 2821 documents and 66900015 words. The average document size is 23715 words. Documents are categorized under four main classes: Unfair competition, Arbitration, Insurance, and Commercial lease.

The dataset was split into a training dataset (75% of each class) used to build the model and a testing dataset (25% of each class) used to evaluate the model's performance. The distribution of documents in each class is presented in Table I.

TABLE I
Distribution of documents in each class.

Class	Train	Test	Total
Unfair competition	523	175	698
Arbitration	509	163	672
Insurance	511	183	694
Commercial lease	557	200	757
Total	2100	721	2821

B. Data preprocessing

The first step in our preprocessing process was removing stop words, removing foreign characters, and punctuation by applying basic functions. Then, we execute the stemming algorithm, which transforms all words into their stems.

C. Model

The proposed BERT-CLSTM model exhibits two key characteristics. Firstly, it employs CLSTM to transform the task-specific layer of BERT, allowing our model to capture local and long-term text representations. Secondly, the output of BERT, along with the CLSTM representation, is fed into the transformer encoder. This facilitates the use of self-attention to focus the final text representation on essential segments. The architecture of the BERT-CLSTM model is illustrated in Figure 1.

1) CLSTM encoder: In order to make the representation of text focus on the information in the text. We use a convolution filter to extract features from T. Assume that the size of the convolution window is 1 × k, the output of the CLSTM encoder is:

$$P = CLSTM_k(T) \tag{1}$$

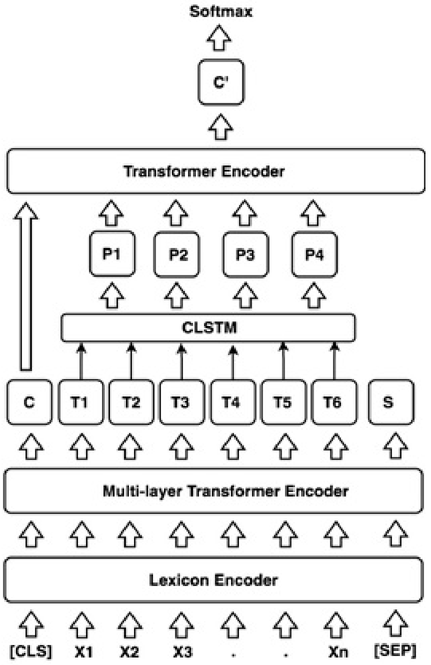


Fig. 1. Model architecture.

P is $\{P_1, P_2, P_3, ..., P_r\}$ and $r = mk + 1$. Through convolution operation, we can obtain the representation of sentences in windows with size K. For example, P_r is the representation of $T_{(r-1)}, T_r, T_{(r+1)}$.

2) Transformer encoder: Similar to multi-layer transformer encoder, to integrate information in P, we adopt transformer encoder to map the local representation P into the representation of whole text. The input of the transformer encoder is $\{C, P_1, P_2, P_3, ..., P_r\}$, and we use C' which corresponds to C as the representation of the whole text.

3) Output layer: The output of the model is represented as follows:

$$y = softmax(tanh(WC' + b)) \tag{2}$$

Where $\tanh$ presents the hyperbolic tangent function:

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3}$$

IV. Results And Analysis

This section presents the experimental setup, including the parameters used for training and testing the model. We detail the evaluation measures, highlight the results, and provide a discussion of the findings.

A. Experimental setup

To evaluate our classifier, we used the created dataset (See section III.A). Table II shows the model hyper-parameters.

TABLE II
Major hyper-parameters of the model.

Step	Parameter	Value
BERT Embedding	Size of BERT	1024
BERT Embedding	BERT layer	Last 4
CLSTM Layer	Number of filters	128
CLSTM Layer	Filter sizes	3,4,5
CLSTM Layer	Pooling function	Max Pooling
CLSTM Layer	Dropout probability	0.5
CLSTM Layer	Number of epochs	30
Training	Optimizer	Adam
Training	Learning rate	1e-3
Training	Loss	Cross Entropy

B. Evaluation measures

We used a variety of metrics to evaluate the performance of the model. We began by calculating the **Precision** measure, which indicates the number of documents correctly predicted by the model out of all documents predicted. The precision measure may be calculated using the equation 4:

$$Precision = \frac{True\_Positives}{True\_Positives + False\_Positives} \tag{4}$$

Second, the **Recall** measure, which indicates the number of documents accurately predicted by the model out of the total of documents in the dataset. The recall measure can be computed using this equation 5:

$$Recall = \frac{True\_Positives}{True\_Positives + False\_Negatives} \tag{5}$$

Third, **F1-measure** which is a metric that can be calculated based on the precision and recall using the following equation 6:

$$F1\_measure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{6}$$

The last metric that we use is **Accuracy**. It is a metric used in classification problems used to tell the percentage of accurate predictions. We calculate it by dividing the number of correct predictions by the total number of predictions.

$$Accuracy = \frac{Number\_of\_correct\_predictions}{Total\_number\_of\_predictions} \tag{7}$$

C. Results

We applied the baseline models which are: Logistic Regression (LR), Extreme Gradient Boosting (XGBoost), Naive Bayes (NB), and Random Forest (RF). Then, for deep learning models, we use the CNN algorithm with the fastText embedding and of course, our proposed model which is BERT-CLSTM. Table 3 illustrates the details of the model's evaluation findings. Precision, Recall, F1-measure, and Accuracy are denoted by the letters 'P', 'R', 'F', and 'Acc'. respectively.

TABLE III
PERFORMANCE OF OUR MODEL AGAINST THE BASELINE MODELS.

Models	P (%)	R (%)	F (%)	Acc (%)
LR + count vectors	86.07	86.07	86.07	86.07
LR + word level (TF-IDF)	86.07	86.07	86.07	86.07
LR + N-gram vectors	86.07	86.07	86.07	86.07
LR + CharLevel vectors	84.81	84.81	84.81	84.81
XGBoost + count vectors	87.34	87.34	87.34	87.34
XGBoost + word level (TF-IDF)	85.44	85.44	85.44	85.44
NB + count vectors	85.44	85.44	85.44	85.44
NB + word level (TF-IDF)	84.81	84.81	84.81	84.81
NB + N-gram vectors	87.97	87.97	87.97	87.97
NB + CharLevel vectors	79.74	79.74	79.74	79.74
RF + count vectors	84.17	84.17	884.17	84.17
RF + word level (TF-IDF)	87.97	87.97	87.97	87.97
CNN + FastText	90.98	90.52	90.75	90.68
BERT + CLSTM (our model)	**93.81**	**93.42**	**93.61**	**93.55**

D. Analysis and discussion

Table III shows that our model outperformed the baseline models and the CNN model in terms of Precision, Recall, F1-Measure and Accuracy. In term of F1-Measure, our proposed model outperforms the CNN+FastText model by **2.86**, the RF+TF-IDF model and the NB+N-Gram by **5.64** as well as the XGBoost+count vectors model by **6.27** and the LR+N-Gram model by **7.54**.

Figure 2 shows accuracy scores for models. In term of Accuracy, our proposed model outperforms the CNN+FastText model by **2.87**, the RF+TFIDF model and the NB+N-Gram by **5.58** as well as the XGBoost+count vectors model by **6.21** and the LR+N-Gram model by **7.48**.

The proposed method has a better classification effect, and the reason is that the text representation matrix has strong feature representation ability and is more representative, which can provide more category information for text classification. This also highlights the advantage of using CLSTM, which gives the representations of text with local and long-term information, unlike the traditional methods that are based on the bag-of-words model.

V. CONCLUSION

Arabic is considered one of the most difficult languages to process, due to its high morphological ambiguity, writing style, and lack of capitalization. Therefore, every NLP task involving this language requires a lot of feature engineering and preprocessing. In this paper, we present our model that classifies the Moroccan commercial court verdicts into four categories. The model outperformed the compared baselines and achieved good results by getting an F-measure value of 93.61%.

ACKNOWLEDGMENT

This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco [Alkhawarizmi/2020/36].

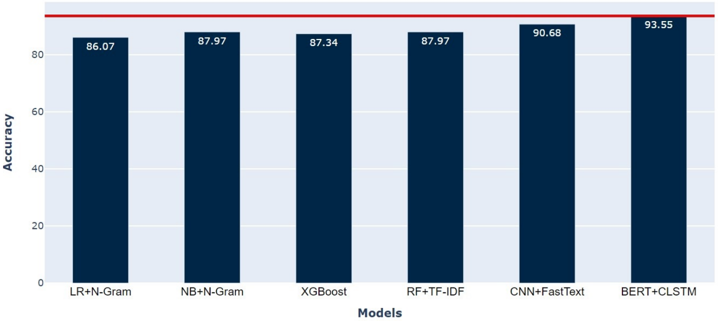


Fig. 2. Accuracy scores for models.

REFERENCES

[1] C. Blake, “Text Mining,” *Annual review of information science and technology,* vol. 45, 2011, pp. 121–155.

[2] X. Zhang, J. Zhao, Y. LeCun, “Character-level convolutional networks for text classification,” *Advances in Neural Information Processing Systems,* 2015.

[3] A. Conneau, H. Schwenk, L. Barrault, Y. LeCun, “Very deep convolutional networks for text classification,” *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,* vol. 1, 2017, pp. 1107–1116.

[4] Y. Kim, “Convolutional neural networks for sentence classification,” *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* vol. 1, 2014, pp. 1746–1751.

[5] K. Sheng Tai, R. Socher, C.D. Manning, “Improved semantic representations from tree-structured long short-term memory networks,” *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing,* vol. 1, 2015, pp. 1556–1566.

[6] M. Huang, Q. Qian, X. Zhu, “Encoding syntactic knowledge in neural networks for sentiment classification,” *ACM Transactions on Information Systems,* vol. 35, 2017, pp. 1–27.

[7] H. El Rifai, L. Al Qadi, A. Elnagar, “Arabic text classification: the need for multi-labeling systems,” *Neural Computing and Applications,* vol. 34, 2022, pp. 1135–1159.

[8] M. AL-Qurishi, S. AlQaseemi, R. Soussi, “AraLegal-BERT: A pretrained language model for Arabic Legal text,” *Proceedings of the Natural Legal Language Processing Workshop 2022,* 2022, pp. 338–344.

[9] F. El-Alami, A. El Mahdaouy, S.O. El Alaoui, N. En-Nahnahi, “A deep autoencoder-based representation for Arabic text categorization,” *Journal of Information and Communication Technology,* vol. 3, 2020, pp. 381–398.

[10] W.H.G. Gwad, I.M.I. Ismael, Y. Gultepe, “Twitter Sentiment Analysis Classification in the Arabic Language using Long Short-Term Memory Neural Networks,” *International Journal of Engineering and Advanced Technology,* vol. 9, 2020, pp. 235–239.

[11] M. Alhawarat, A.O. Aseeri, “A Superior Arabic Text Categorization Deep Model (SATCDM),” *IEEE Access,* vol. 8, 2020, pp. 24653–24661.

[12] M. Galal, M.M. Madbouly, A. El-Zoghby, “Classifying Arabic text using deep learning,” *Journal of Theoretical and Applied Information Technology,* vol. 97, 2019, pp. 3412–3422.

[13] S. Boukil, M. Biniz, F. El Adnani, L. Cherrat, A.E. El Moutaouakkil, “Arabic Text Classification Using Deep Learning Technics,” *International journal of grid and distributed computing,* vol. 11, 2018, pp. 103–114.

[14] R. Al-khurayji and A. Sameh, “An Effective Arabic Text Classification Approach Based on Kernel Naive Bayes Classifier,” *International Journal of Artificial Intelligence and Applications,* vol. 8, 2017, pp. 01–10.

DOI: 10.15439/2023F6245
ISSN 2300-5963 ACSIS, Vol. 35

Calculating and comparing solar radiation results using GIS in the City Sarajevo area

Nedim Mujić
Faculty of Electrical Engineering,
University of Sarajevo
71000 Sarajevo, Bosnia and Herzegovina
Email: n.mujic@hotmail.com

Almir Karabegović
Faculty of Electrical Engineering,
University of Sarajevo
71000 Sarajevo, Bosnia and Herzegovina
Email: akarabegovic@etf.unsa.ba

***Abstract*—Citizens of the city of Sarajevo and of other industrial cities are faced with a record number of days of increased pollution. In the winter months, the city of Sarajevo faces a large number of days of pollution caused mainly by the use of fossil fuels in individual houses for heating purposes. The current situation can be changed by the massive use of energy from renewable sources such as solar energy. This paper aims to evaluate the potential of solar energy in the city of Sarajevo. The use of Geographic Information Systems (GIS) represents the most significant technological and conceptual approach to spatial data analysis. Using existing models for calculating incoming solar radiation integrated in the GRASS GIS and SAGA GIS software, we achieved the goal and calculated the results for solar energy potential in the city of Sarajevo and presented them for the specific settlements. The model was implemented on the basis of created Digital Elevation Model (DEM) from Google Earth – free datasets, using techniques to collect and convert data with different software. Comparative results of selected model research are evaluated using the collected solar irradiance values from the meteorological stations, other research results, and the solar energy potential estimated via the Photovoltaic GIS Information System (PVGIS).**

Keywords: QGIS, DEM, PVGIS, GRASS, SAGA, Solar energy potential

I. Introduction

THE COUNTRIES of the European Union are signatories to the Kyoto Protocol of 1997, which deals with the reduction of global warming of the planet by reducing the emission of greenhouse gases. GHG emissions decreased in the majority of sectors between 1990 and 2018, and particularly in energy supply, industry and the residential sector [1]. The European Union provides great support for electricity generation from renewable energy sources and gradually reduces the capacity consumption of fossil fuels. Solar energy is one form of renewable energy addressed by this research. The European Union significantly encourages the population to use this energy source to generate electricity.

Pollution in the city of Sarajevo is mainly caused by burning fossil fuels to heat individual housing units in settlements where individual housing prevails, and relatively more expensive prices for gas and electricity for heating. In most urban zones of collective housing (buildings) thermal energy is used for heating, consuming gas as a basic energy source, which is an environmentally friendly source of energy for combustion.

Insolation is considered the direct solar radiation that reaches the earth's surface. The duration of insolation correlates with the latitude of the measurement location, elevation, land relief, and cloud cover. The terrain (relief) is the main factor that changes the distribution of solar radiation regionally and locally. Variability in terrain altitude, the slope of the terrain, aspect terrain, as well as the effect of shading, give special strong local gradients. Accurate and spatially distributed solar irradiance data are desirable for various applications (environmental science, climatology, ecology, photovoltaic installations, land management, etc.).

The growing availability of geospatial data and the demand for better analytic insight have helped to move location analysis from departmental (GIS) experts to other users [2], i.e. organizations for decision-making or users who want to take decisions about photovoltaic installations.

This study is based on the calculation of the solar potential using GIS methods and using free resources such as datasets from Google Earth and free software Quantum GIS - QGIS calculation [3], Geographic Resources Analysis Support System – GRASS [4], System for Automated Geoscientific Analyses - SAGA [5]. The aim of this study is to use certain GIS methods to obtain the results of solar potential in the city of Sarajevo. Validation and comparison of the results with previous ones and those obtained from Photovoltaic GIS Information System PVGIS [6] could provide information about the accuracy of GIS methods for solar potential calculation.

The study work of the region Abruzzo in Italy provides a framework for analyzing the sustainability of renewable energy sources using GIS. They present a GIS-based methodology to support decision-making in energy supply from renewable energy sources. One of the primary data required for the study was Digital Elevation Model - DEM [7]. The r.sun [8] model is a complex and flexible solar radiation model, fully integrated within the open-source environment of GRASS GIS [9]. The implemented equations follow the latest European research in solar radiation modeling. Integration in GRASS GIS enables to use interpolation tools that are necessary for data preparation [8]. The assessment of photovoltaic potential in the urban area of Bardejov in Slovakia was performed by using r.sun and a 3D city model. It was emphasized that the city has the potential to generate about 45% of its current electricity consumption by installing PV systems, although the study area

This work was not supported by any organization

included only two-thirds of the entire city. This confirms that solar potential estimates performed by the r.sun model can be used to assess solar potential as well as in urban areas [10]. In the research and estimation of the solar potential of roof surfaces in Baton Rouge (USA) on the basis of LiDAR data, information about the buildings (slope and aspect of roofs) and information concerning the roof surfaces using GRASS GIS software results indicate the annual solar potential of approx.1598,7 (kWh/m^2) [11]. Gorički M. with the group uses DSM to calculate the solar potential in the small town of Sveti Križ Začretje in northwestern Croatia. The data for the study were provided by an unmanned aerial vehicle. To evaluate the solar potential, they used the SAGA GIS software annually with a solar potential of 1900 (kWh/m^2) on the southern parts of the house roofs [12].

Validation of the obtained results could be done on the basis of solar radiation measurements, which are measured daily by hydrometeorological stations. In Sarajevo there are stations that measure insolation but not solar irradiance.

In Europe, there are several hundred weather stations that directly or indirectly measure solar radiation [8].

In this paper, we will present the methodology for estimating solar potential using available GIS tools and compare the obtained results using the city of Sarajevo as an example. In the following, we will first introduce the selected area and the method of data collection, then model the data, choose the appropriate method for the budget, and use it with free GIS tools. At the end, we will discuss the obtained results.

A. Data analysis and preparation area

The city of Sarajevo was selected for the area of analysis due to the frequent pollution of the city and its population during the calendar year and the potential opportunity to make better decisions towards electricity generation from renewable energy sources, especially solar energy.

The area of the City of Sarajevo is located in the Mountain Valley macro-region in the Sarajevo-Zenica Valley, which is the southern end of the larger morpho structure. The lowest parts of the city area are located at an altitude of about 500 meters (zone of Rajlovac and Reljevo), while some of the settlement slopes in Stari Grad reach over 800 meters. The city center is located at an altitude between 540 and 550 meters. Mount Trebević (1627 meters) is a dominant mountain elevation near the city center. The Sarajevo Valley, where most of the city of Sarajevo is located, is in the east-west direction in a length of about 12 (km) and 1-1.5 (km) wide, with geographical coordinates 43.8563°N, 18.4131°E. The area of analysis is shown in Fig. 1.

Free software QGIS version was used to prepare and process data. Due to the widespread data availability and the possibility to repeat the results and the possibility to use free data for processing, the Google Earth application is used to download data from which DEM is created.

The area bounded by a red rectangle with dimensions (11.5 km long and 7.5 km wide) is an area for analysis. This is a very wide area for the creation of DEMs and an area of about 82 km^2. For this survey, we are collecting data for the following 6 settlements:

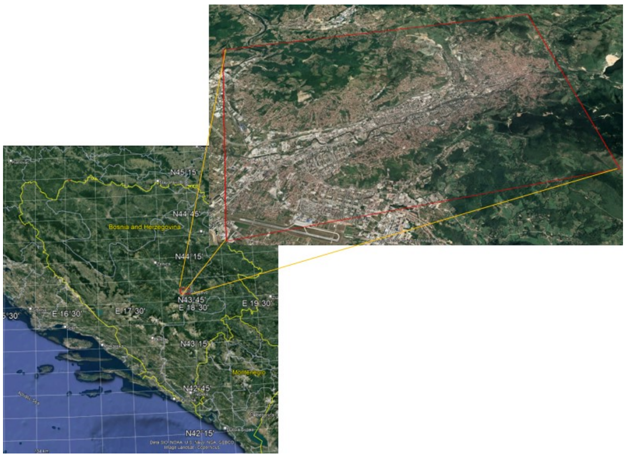


Fig. 1. A view of wider surroundings of the city of Sarajevo

1. Nedžarići, residential houses, lowland part of the city,
2. Grbavica, blocks of flats, lowland plant of the city,
3. Buća Potok, residential houses, hillside area, south –orientation,
4. Širokača-Krka, residential houses, hillside area, north orientation,
5. Šip, blocks of flats and meadows, mountainous area, south-west orientation,
6. Sedrenik-Grdonj, residential houses, south-west orientation.

B. Insolation and insolation data for Sarajevo

Insolation is the time for which the earth's surface receives direct solar radiation. The duration of solar radiation correlates with the latitude of the measurement location, altitude, land relief, and cloud cover. A heliograph is a measuring device consisting of a glass ball 9-12 (cm) in diameter recording the sun rays and focus them onto a strip of paper that burns under the influence of heat. Since older versions of heliographs did not indicate values below 120 (W/m^2), this threshold was retained by the World Meteorological Organization Convention and in more modern measuring instruments "unpublished" [13].

The term cloud cover refers to the degree to which the sky is covered with clouds, i.e. the size of the cloud cover relative to the entire sky. Smaller values mean brighter days, so the value of 1 means coverage of up to 10% of the sky with clouds, cloudiness 10 means that the sky is completely covered with clouds. According to [14] the annual insolation from the Bjelave hydro-meteorological station for a period of 20 years amounts 1919 hours i.e. 5(h) per day. The monthly sum of insolation (I) is calculated according to (1):

$$I = i_1 + i_2 + \cdots + i_n \tag{1}$$

$i_1, i_2, \ldots, i_n$ - the daily sum of sunshine duration for the first, second and i^{th} day of the month.

The annual insulation (I_g) is calculated according to (2):

$$I_g = I_1 + I_2 + \cdots + I_{11} + I_{12} \qquad (2)$$

$I_1, I_2, I_3, \ldots, I_{12}$ - represent the insolation for the first, second, third,..., and twelfth month, respectively [14].

C. Irradiation on a horizontal plane

The term irradiance is used to consider the solar power (instantaneous energy) falling on unit area per unit time (W/m^2). The term irradiation is used to consider the amount of solar energy falling on unit area over a stated time interval (W/m^2) [8]. According to [15] integrating the obtained value by time, the total energy of radiation the unit area in the observed interval is obtained. By integrating the intensity of solar radiation at different time intervals, we obtain hourly, daily, monthly, and annual irradiation (H):

$$H = \int_{t_1}^{t_2} G_{0n} cos\theta_z dt \qquad (3)$$

G_{0n} – extraterrestrial radiation intensity on a surface perpendicular to the radiation (W/m^2); G_0 – solar constant 1367 (W/m^2); n – ordinal number of days in a year.

$$H_0 = \int_{-\omega_s}^{\omega_s} G_0(1 + 0,34cos\frac{n}{356}360)$$

$$(sin\varphi sin\delta + cos\omega cos\delta cos\omega)d\omega \qquad (4)$$

$$H_0 = \frac{24}{\pi} G_0(1 + 0,364cos\frac{n}{356}360)$$

$$(\frac{\pi}{180}\omega_\mathbf{s} sin\varphi sin\delta + cos\varphi cos\delta sin\omega_\mathbf{s}) \qquad (5)$$

H_0 – the irradiation on a horizontal plane; (Wh/m^2); $\omega_\mathbf{s}$ – angle of sunrise (°); $-\omega_\mathbf{s}$ – angle of sunset (°); δ – solar declination (°); φ – latitude (43°52 for Sarajevo) (°).

The average annual extraterrestrial irradiation in the city of Sarajevo is obtained by calculating the average daily extraterrestrial irradiation for each day of the year. Table I shows the average monthly irradiation of a horizontal plate area 1 (m^2) at the latitude (43°52 for the city of Sarajevo). Calculated values of extraterrestrial irradiation do not take into account the influence of absorption, diffuse radiation and reflection of solar radiation, effects of clouds, pollution, height of the measuring plate, etc., which strongly influences the measurement results "unpublished" [13].

TABLE I
THE EXTRATERRESTRICAL IRRADIATION ON A PLANE (kWh/m^2) [13]

January	February	March	April	May	June
112	142	222	260	339	348
July	**Augus**	**Sept**	**Octob**	**Novem**	**Decem**
348	307	237	176	118	99

D. Data modeling

The digital elevation model (DEM) is considered one of the most important input data for the purpose of terrain's (relief) surface representation. DEM system [16]. There is no harmonized terminology about the name in the literature. Digital terrain model (DTM), this concept, which includes relief as well as other general geographic objects, refers to the part of the terrain that has certain distinctive features.

The specificity of this paper is reflected in the generation of DEM from free data sources. The quality and quantity of data as well as the method of collecting these data are the most important for obtaining quality DEM. The access was done through the free online application Google Earth and the data offered in it on the Internet. Fig. 2 shows the sequence diagram of steps that were performed to achieve the final DEM product. Creating a vector file in the Google Earth web platform, i.e. when such a trajectory is drawn over the desired area with sufficient desired density, a file with a .kml extension is formed. The trajectory is clearly marked with latitude, longitude, and altitude. A larger number of points in the trajectory represents a more detailed DEM. After creating the trajectory, data extraction was started using certain free-type applications. To convert a .kml file into a .gpx file, the GPS Visualizer was used, where a certain conversion method saved .gpx files. The next step was to create a .scv file where you can visually see the numerous values of each point from the trajectory with latitude, longitude and altitude and this process was done using TCX Converter - free software. By processing .csv file and inputting it into QGIS [3], we created a set of points from which we obtained DEM shown in Fig. 3 by post-processing and applied algorithm.

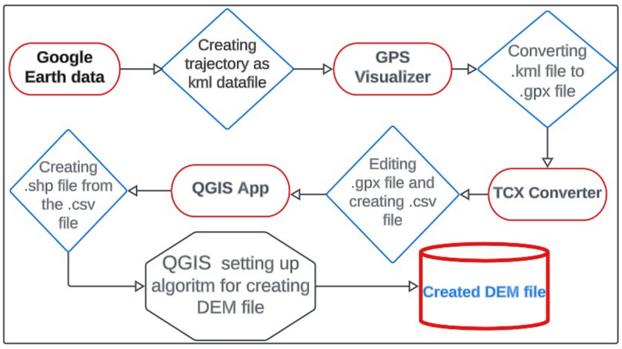

Google Earth data
Creating trajectory as kml datafile
GPS Visualizer
Converting .kml file to .gpx file
TCX Converter
Editing .gpx file and creating .csv file
QGIS App
Creating .shp file from the .csv file
QGIS setting up algoritm for creating DEM file
Created DEM file

Fig. 2. The sequence diagram of steps for creation of DEM

II. OVERVIEW OF GIS METHODS FOR ESTIMATING SOLAR POTENTIAL

To calculate the potential of solar radiation with GIS methods, certain raster data such as DEM or DTM terrain models are needed. Several methods have been developed and are used in different GIS tools. Each of these methods has its own specific parameters.

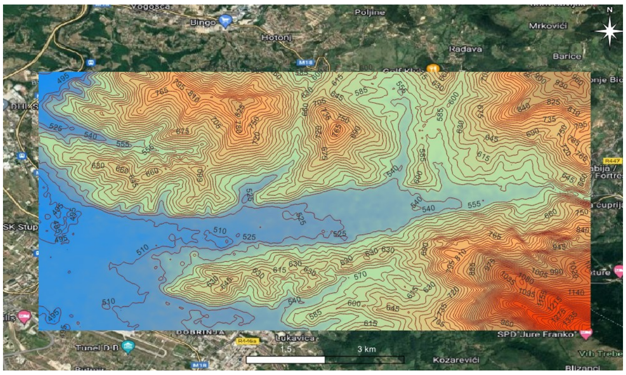


Fig. 3. Created DEM – City of Sarajevo

The free software GRASS GIS [4] and SAGA GIS [5] were used for calculations and analyses, and the data used were also obtained on a free basis or created by the authors.

The algorithm for the Satellite-Based Retrieval of Solar Surface Irradiance in Spectral Bands [17] has been implemented in PVGIS and is based on solar irradiance results. PVGIS [6] is often used as a basis for informative display of monthly and daily solar potential.

Hofierka and Šuri created an open-source solar radiation model called r.sun [8], GRASS GIS [4] is the best known open source software based on a methodology that uses equations published in the European Solar Radiation Atlas (ESRA) [18] and applies the r.sun model to calculate the solar potential. According to [5] SAGA is a specialized digital terrain analysis tool on a comprehensive and widely used GIS platform for scientific analysis and modeling. It is designed for easy and efficient implementation of spatial algorithms and thus serves as a framework for the development and implementation of geoscientific methods and models. Today, SAGA is a modular programmable GIS software that provides raster analyses of DEM and DSM substrates to estimate solar potential, with the ability to input specific data such as solar constant, atmospheric pressure, atmospheric height, humidity, etc. Solar radiation models integrated into GIS systems provide rapid, cost-effective, and accurate estimates of radiation over large areas, considering surface slope, aspect, and shading effects. Significant progress has been made in the development of solar radiation models over the past two decades [8].

III. SOLAR ENERGY POTENTIAL CALCULATION

A. Solar radiation and photovoltaic data

PVGIS [6] is an information system that allows the user to get data on solar radiation and photovoltaic system energy production, at any place in most parts of the world. It is completely free to use, with no restrictions on what the results can be used for, and with no registration necessary. PVGIS can be used to make several different calculations. As an example of solar potential in the city of Sarajevo, we used the PVGIS SARAH2 database, which uses predefined samples of satellite image resolution ($5km$ x $5km$) for the period from 2005 to 2020 [6]. PVGIS provides monthly solar potential values, from which we calculated the mean annual potential. Table II. shows the solar radiation for each settlement and refers to the radiation at an abduction angle with a calculated optimal plate angle. Table III. represents the solar radiation on a horizontal panel, i.e.the panel at an angle of 0^o.

TABLE II
IRRADIATION ON OPTIMALLY INCLINED PLANE (kWh/m^2)

Sett:	**Nedžari**	**Grbavi**	**Buca Pot**	**Širok**	**Šip**	**Sedre**
Lat:	43.837	43.851	43.864	43.850	43.884	43.876
Lon:	18.337	18.395	18.36	18.431	18.402	18.429
Ang:	37^o	37^o	34^o	31^o	35^o	35^o
Jan	98,21	96,54	101,9	49,7	100,9	102,21
Apr	185,3	186,04	190,9	183,1	189,4	189,33
July	180,5	185,13	182,4	188,1	184	183,53
Oct	122,6	122,42	125,7	102,5	124,4	125,16
Dec	55,7	57,56	58,2	28,2	60,93	62,96
Sum	1572	1582,3	1593,4	1412	1579	1584,8

TABLE III
SOLAR IRRADIATION ON HORIZONTAL PLANE (kWh/m^2)

Sett	Nedžar	Grbavi	Buća Pot	Široka	Šip	Sedren
Jan	54,43	54,11	54,99	37,03	55,4	55,57
Apr	164,2	164,7	168,9	161,78	167	167,6
July	185,88	189,6	188,99	189,11	188,1	188,1
Oct	86,79	86,47	87,59	77,55	87,26	87,37
Dec	35,12	35,71	35,69	25,37	36,93	37,33
Sum	1349	1355	1360,1	1274,25	1348	1347

B. Geographic Resources Analysis Support System

The installed GRASS GIS [4] software with version 7.6.1 was used to analyze the terrain of the observed area and then calculate the solar radiation. In the first case, a database was created with the basic element of DEM resolution ($30m$ x $30m$), shown in Fig. 3. The DEM was used to create slope and aspect, shown in Fig. 4. These two graphical representations are very important for us to calculate the solar potential in the selected area with the setting of certain parameters (Linke coefficient of atmospheric turbidity and albedo). GRASS does not count the annual solar radiation for the whole year but for each day separately. Fig. 5 illustrates solar irradiance for 01 July and the locations from which it was collected for records and analysis. Given the width of the workspace shown in Fig. 5 on the left and the resolution of ($30m$ x $30m$) and the diameter of a point from Fig. 5 from which the data was collected, which is about 200 (m), it is understandable that when collecting the solar potential from one of these points and repeated procedures, there is a possibility that we will not hit the previous pixel, but neighboring ones. Thus, there is a possibility that the value of the potential will also be different, but not drastically. The solar radiation map for the middle day of December is shown in Fig. 5 on the right.

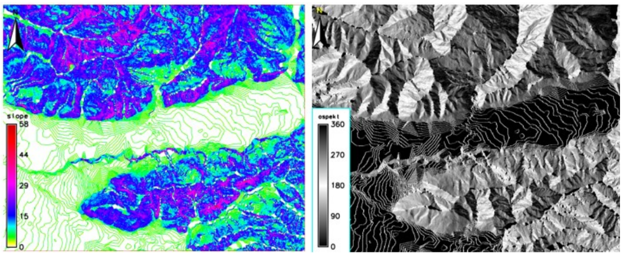


Fig. 4. Graphical presentation of slope on the left side and aspect on the right side

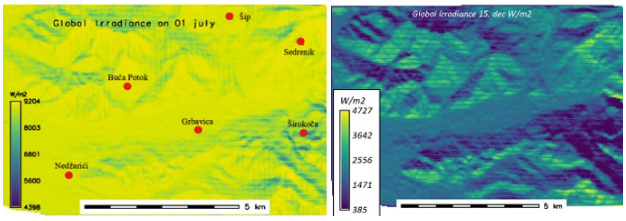


Fig. 5. Locations for collecting data for Solar irradiation for the 01. July on the left, and Solar irradiation data for 15. December on the right

The mean value of solar potential for each month for a given location is determined from the daily values, which are shown in Table IV.

TABLE IV
AVERAGE SOLAR IRRADIATION ON THE MONTHLY BASIS IN DIFFERENT SETTLEMENTS (kWh/m^2) [4]

Sett:	Nedzar	Grbav	Buca Pot	Sirok	Sip	Sedren
Jan	67	67	110	14	74	97
Apr	188	185	213	69	194	208
July	242	241	246	186	244	247
Nov	72	69	109	15	80	102
SUM:	1798	1776	2113	944	1879	2045

Very interesting are the areas of the city located on the slopes of mountain Trebević, where the solar potential ranges from about 0.4 (kWh/m^2) in winter to 7 (kWh/m^2) in summer. The results obtained with this method range from about 950 (kWh/m^2) in areas of the city facing north and northwest, located on the sides and foothills of the mountain Trebević, to 1800 (kWh/m^2) in areas located in the lowland part of the city, to 2100 (kWh/m^2) on the slopes of the city facing south, southeast and southwest.

C. System for Automated Geoscientific Analyses

SAGA GIS [5] was used to calculate the solar potential in a wide area of the city of Sarajevo. Unlike GRASS, it can calculate solar potential for the whole year. The algorithm for calculating the solar potential also uses as a basis the created DEM of areas, based on which the Sky view factor is calculated, and accordingly the algorithm calculates the solar radiation with the adjustment of certain parameters. For this calculation, we used a created example of DEM resolution $(30m$ x $30m)$. To display the average value of solar potential on monthly basis in Fig. 6, we set the algorithm sampling for each day of the month and each half hour during the day, which is a rather detailed calculation.

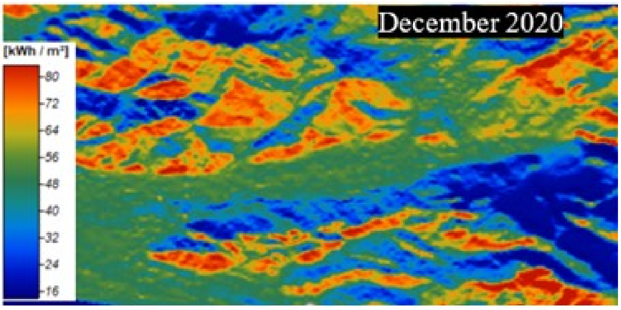


Fig. 6. Solar radiation map for December

The SAGA GIS results for 2020 range from 1449 to 2099 (kWh/m^2). A more detailed overview of the selected sites/locations is shown in Table V.

TABLE V
SOLAR POTENTIAL RESULTS IN SPECIFIC SETTLEMENTS (kWh/m^2)

Sett:	Nedzar	Grbavic	Buca Pot	Sirok	Sip	Sedre
Jan.	59	54	66	33	67	93
April	191	187	199	173	198	216
Aug.	211	208	218	182	218	230
Dec.	49	55	44	15	58	82
SUM:	1777	1729	1875	1449	1861	2099

It is expected that the result of solar potential in the settlement of Sedrenik is greater than the potential in the lowland parts of the city due to the higher location and orientation of the settlement to the south and southwest. The lowest solar potential is illustrated in the area of Širokača, which is located on the northern side of the Trebević Mountains, adding extra shade in the morning hours Fig. 6.

The dark shades on the map show fairly shaded areas and slopes facing north and northwest, where we get the lowest values of solar radiation.

IV. DISCUSSION AND COMPARISON OF RESULTS

The estimated results of solar potential in Sarajevo Canton "unpublished" [13] amount ca. 2700 (kWh/m^2). The annual irradiation on a horizontal plane of 1 (m^2), based on insolation, was estimated in range of 1100 to 1550 (kWh/m^2) according to the study "unpublished" [19]. The annual solar potential PVGIS, GRASS and SAGA for the Sarajevo area is shown in Table VI. Using PVGIS does not require extensive foreknowledge and expertise to obtain results, which makes PVGIS very easy to use. To use GRASS and SAGA GIS, it is necessary to have good knowledge in managing GIS applications and a good knowledge of the factors that affect the amount of solar radiation. There is a noticeable difference in solar potential results, especially in methods (GRASS and SAGA) in which we used DEM ($30m$ x $30m$) compared to PVGIS results which uses satellite image resolution ($5km$ x $5km$) from its database, shown in Table VI.

TABLE VI
AVERAGE ANNUAL SOLAR POTENTIAL (kWh/m^2)

Settlement	PVGIS	GRASS	SAGA
Nedžarići	1349	1798	1777
Grbavica	1355	1776	1729
BućaPotok	1360	2113	1875
Širokača	1274	944	1449
Šip	1348	1879	1861
Sedrenik	1347	2045	2099

The differences are especially noticeable in parts of the city that are in the shadow of mountain Trebević in winter months. Colleagues from Croatia, who used SAGA in research [12] concluded that the solar potential results appear to be quite large and that the actual energy of solar energy that can be obtained annually from solar radiation is much lower. Although Sveti Križ Začretje (Croatia) is 200 (km) farther north than Sarajevo, SAGA provided them results for solar potential of about 1900 (kWh/m^2).

V. Conclusion

The aim of this study was to obtain solar potential results in the city of Sarajevo using certain GIS methods. We found that GIS methods provide the user with detailed insight into solar potential with the availability of certain datasets and the level of GIS management expertise required to perform such an analysis. Based on the size of the satellite image resolution sample that makes up the PVGIS database, it can be concluded that PVGIS is more commonly used for large-area solar potential calculations, such as states. Since GRASS and SAGA do not have predefined datasets, they can use detailed DEM for input data. Due to the level of detail of the DEM created ($30m$ x $30m$), we have a more pronounced relief of the area we analyzed, so the shading effects are more visible, especially in winter months, which was reflected in the calculation of solar potential at these locations. Thus, we conclude that GRASS and SAGA are better suited for calculations of smaller areas. Based on the results, we get the impression that in micro-sites, the detail of DEM significantly increased the difference of solar potential in certain areas compared to the results of PVGIS. Looking at the solar potential results through the details of DEM samples, we can conclude that GRASS and SAGA provide more accurate results, which has been confirmed by some previous works. The results of GRASS and SAGA differ very little with maximum deviations up to 0.96%, except for the results for settlements at the foothill of mountain Trebević. Unfortunately, the validation of the results cannot be established due to the lack of measuring stations that record solar radiation (the nearest registered stations are located in Budapest and the Austrian Alps). A possible improvement in terms of more accurate solar potential results could be achieved by creating a digital surface model LIDAR record or satellite image resolution (1m x 1m).

References

[1] "Trends and drivers of EU greenhouse gas emissions," European Environment Agency, Copenhagen, Danmark, 2020. Accessed: Feb. 28, 2023. https://dx.doi.org10.2800/19800

[2] A. Karabegović and M. Ponjavić, "Geoportal as Decision Support System with Spatial Data Warehouse," in Proceedings of the Federated Conference on Computer Science and Information Systems, Wroclaw, Poland, 2012, no. 978–8360810–514, pp. 915–918. Accessed: Jan. 03, 2023. https://ieeexplore.ieee.org/document/6354304

[3] QGIS, "Welcome to the QGIS project!" Qgis.org, 2017. https://www.qgis.org/en/site/ (accessed Mar. 14, 2023).

[4] GRASS Development Team, "Geographic Resources Analysis Support System (GRASS GIS) Software, Version 8.2," https://grass.osgeo.org (accessed Mar. 14, 2023).

[5] O. Conrad et al., "System for Automated Geoscientific Analyses (SAGA) v. 2.1.4," Geoscientific Model Development, vol. 8, no. 7, pp. 1991–2007, July 2015, doi: https://dx.doi.org/10.5194/gmd-8-1991-2015

[6] [6] European Commission, "Photovoltaic Geographical Information System (PVGIS) - European Commission," https://re.jrc.ec.europa.eu/pvg_tools/en/

[7] E. Caiaffa, A. Marucci, and M. Pollino, "Study of Sustainability of Renewable Energy Sources through GIS Analysis Techniques," Computational Science and Its Applications – ICCSA 2012, pp. 532–547, 2012, https://doi.org/10.1007/978-3-642-31075-1_40

[8] J. Hofierka and M. Šury, "The solar radiation model for Opensource GIS: Implementation and applications," GRASS users conference 2002, Trento, Italy, Sep. 2002. Accessed: Mar. 15, 2023. https://www.researchgate.net/publication/2539232_The_solar_radiation_model_for_Open_source_GIS_Implementation_and_applications

[9] M. Neteler, H. Mitasova, and Springer link (Online Service, Opensource GIS: A GRASS GIS Approach. New York, Ny: Springer Us, 2004.

[10] J. Hofierka and J. Kaňuk, "Assessment of photovoltaic potential in urban areas using open-source solar radiation tools," Renewable Energy, vol. 34, no. 10, pp. 2206–2214, Oct. 2009, https://doi.org/10.1016/j.renene.2009.02.021

[11] WEYRER, T. N., "GIS Based Analysis of the Potential of Solar Energy of Roof Surfaces in Baton Rouge, Louisiana," Unpublished Bachelor of Science Thesis, Carinthia University of Applied Sciences, Austria., 2011. Available: https://www.marshallplan.at/images/All-Papers/MP-2011/Weyrer.pdf

[12] M. Gorički, V. Poslončec-Petrić, S. Frangeš, and Ž. Bačić, "Analysis of Solar Potential of Roofs Based on Digital Terrain Model," The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-4/W3, pp. 37–41, Sep. 2017, https://doi.org/10.5194/isprs-archives-xlii-4-w3-37-2017

[13] I. Gakić, "Analysis of the possibility of using the space for the construction of solar power plants," Master's thesis, Faculty of Electrical Engineering – University Sarajevo, 2015. "unpublished".

[14] Bjedić, S. Hodžić, B. Krajinović, Dž. Zulum, N. Voljevica, and A. Tucaković, "Meteorological annual reports," Federal Hydrometeorological Institute, Sarajevo, 2022. Accessed: Mar. 15, 2023. http://www.fhmzbih.gov.ba/latinica/KLIMA/godisnjaci.php

[15] L. Majdandžić, Solarni sustavi, Graphis d.o.o., Zagreb, 2010.

[16] A. Šiljeg, "Digital relief model in the analysis of geomorphometric parameters," Doctoral Dissertation, Faculty of Science, University of Zagreb, 2013.

[17] R. Mueller, T. Behrendt, A. Hammer, and A. Kemper, "A New Algorithm for the Satellite-Based Retrieval of Solar Surface Irradiance in Spectral Bands," Remote Sensing, vol. 4, no. 3, pp. 622–647, Mar. 2012, https://doi.org/10.3390/rs4030622

[18] K. Scharmer and J. Greif, The European Solar Radiation Atlas. Paris, France: Vol.2., 2000.

[19] CETEOR BH, CEDES BA, and EKONERG Cro, Eds., "Optimal energy supply study of Sarajevo Canton," 2010., "unpublished".

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 291–296
DOI: 10.15439/2023F677
ISSN 2300-5963 ACSIS, Vol. 35

Towards Industry 4.0: Machine malfunction prediction based on IIoT streaming data

Dragana Nikolova*, Petre Lameski*, Ivan Miguel Pires†, Eftim Zdravevski*
*Faculty of Computer Science and Engineering,Ss Cyril and Methodius University, Skopje, Macedonia
Email: dragana.nikolova.1@students.finki.ukim.mk, petre.lameski@finki.ukim.mk, eftim.zdravevski@finki.ukim.mk
†Instituto de Telecomunicações, Universidade da Beira Interior, Covilhã, Portugal
Email: impires@it.ubi.pt

***Abstract*—The manufacturing industry relies on continuous optimization to meet quality and safety standards, which is part of the Industry 4.0 concept. Predicting when a specific part of a product will fail to meet these standards is of utmost importance and requires vast amounts of data, which often are collected from variety of sensors, often reffered to as Industrial Internet of Things (IIoT). Using a published dataset from Bosch, that describes the process at every step of production, we aim to train a machine learning model that can accurately predict faults in the manufacturing process. The dataset provides two years of production data across four production lines and 52 stations. Considering that the data generated from each production part includes more than four thousand features, we investigate various feature selection and data preprocessing methods. The obtained results exhibit Area Under the Receiver Operating Characteristic Curve (AUC ROC) of up to 0.997, which is remarkable and promising even for real-life production use.**

***Index Terms*—Industrial Internet of Things, Industry 4.0, Machine malfunction prediction, Machine failure prediction**

I. Introduction

THE ONGOING progression of the fourth industrial revolution, accompanied by a fundamental shift towards digitization, referred to as Industry 4.0, is advancing at an exponential rate [1].

In Industry 4.0 systems, the goal is to utilize different temperature sensors, pressure sensors, audio sensors, camera devices, etc., as Industrial Internet of Things (IIoT) devices for machine monitoring and operation control in industrial environments [2]. However, performing machine fault diagnosis and failure prediction is challenging, especially considering the explainability and interoperability requirements.

Introducing predictive maintenance to production environments can provide many benefits, albeit with a few challenges. Some benefits include heightened productivity, decreased system faults, minimized unplanned downtimes, optimized utilization of financial and human resources, and improved planning of maintenance interventions as stated in [3] [4]. In addition, employing machine learning is an effective means of accomplishing prognostics and predicting failures [5]. Predicting when a part of a product will fail is paramount in identifying and preventing defects, thereby improving product quality and safety [6]. By leveraging data generated from each production part, manufacturers can determine whether a part has a weakness and take appropriate action.

Bosch, a leading manufacturer, recognizes this need and has started recording data at every step of the production process. In 2016, they published an anonymized dataset on Kaggle that provides valuable insights into two years of production data across four production lines and 52 stations [1]. Each workstation in the production process performs a variable number of tests and measurements on each part, generating 4,264 features. This experiment aims to train a machine learning model that can accurately predict faults in the manufacturing process using this dataset.

Given the vast number of features in the dataset, data preprocessing and feature selection is a critical step in the model development process [7], [8]. The enormous data growth requires using big data architectures for efficient, robust, and timely processing of it [9]. In turn, it requires the use of efficient algorithms for optimizing hardware resources and minimizing computational cost [10].

In this paper, we perform analyses and feature extraction techniques for numerical, date, and categorical data types to ensure only the most relevant data are used to train the model. With this approach, we aim to create a highly accurate machine-learning model that can aid the manufacturing industry in predicting faults in the manufacturing process using a training dataset where 6,879 parts out of 1,183,747 were labeled as failed, which is a 0.58 error rate. This relatively low error rate presents a significant challenge in creating an accurate predictive model. Figure 1 shows the number of failed parts per line on the left and the number of failed parts per station, indicating that station S32 has a significantly higher number of failures than the rest.

We aim to solve a classification problem to predict whether a product part will fail to meet quality and safety standards during manufacturing. To achieve this, we trained separate machine learning models such as Random Forest, Decision Tree, GradientBoostingClassifier, AdaBoostClassifier, and XGBoost. After evaluating the performance of each model, we found that the XGBoost model had the highest accuracy in predicting faults. Therefore, it was chosen as the final model for the task.

This paper is organized as follows. Section II extensively reviews the machine-learning approaches used for machine

[1] https://www.kaggle.com/c/bosch-production-line-performance

malfunction prediction. In Section III, we introduce our proposed method and provide a detailed explanation of the approach that we are using. Section IV describes the dataset used in our study and presents the results obtained from our experimental analysis. Finally, in Section V, we summarize our findings and conclusions from our research, discussing the potential impact of our work on the manufacturing industry.

II. Related Work

The foundation of this study lies in the preprocessing and feature extraction of numerical and time series data, as well as in classification for malfunction prediction. In the following, we delve into related research on these topics.

A paper focuses on methods for fault prediction [11] and using raw sensor data elaborates on the differences between the Support Vector Machine (SVM) and the Multilayer Perceptron (MLP) for fault prediction. This paper presented an initial development of a supervised machine learning algorithm for diagnosing faults in rotating machinery in the oil and gas industry. They aim to create a simple, easily implementable model that enables quick, informed decision-making. Some preprocessing steps explained in the study are filling in the missing values using linear interpolation, feature engineering performed to introduce the correlation between a data sample and preceding samples in chronological order, and data related to downtime and start-up periods filtered out. As a result, the SVM algorithm demonstrated higher precision than MLP but lower recall for the positive class.

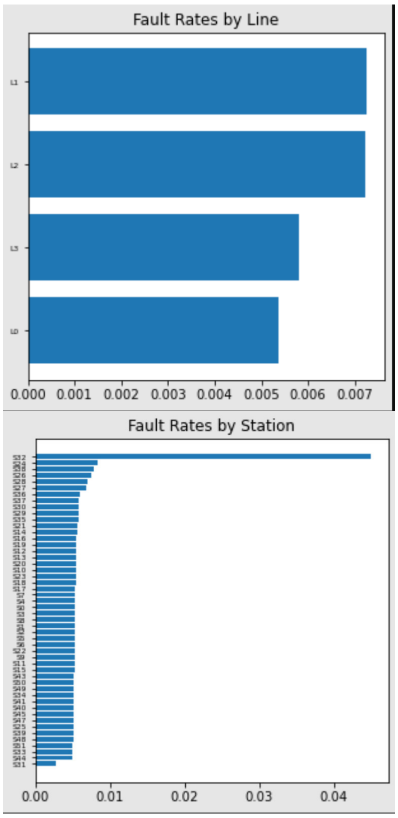


Fig. 1. (above) Number of failed parts by a production line in the training dataset (below) Number of failed parts per production station in the training dataset

[12] discusses the problem of hardware failures in circuits due to aging or variations in circumstances. While self-healing and fault tolerance techniques can recover circuitry from a fault, fault prediction can be used as a pre-stage to these techniques. The proposed method for early fault prediction of circuits uses Fast Fourier Transform, Principal Component Analysis, and Convolutional Neural Network to learn and classify faults. The approach was validated by testing it on two different circuits (comparator and amplifier) using 45 nm technology, providing a fault prediction accuracy of 98.93% and 98.91%, respectively.

Not only hardware failures but, in an article from [13], software failures were also analyzed. The quality of software depends on its bug-free operation, and identifying bugs in the early stages of development can reduce the cost of testing and maintenance. Software defect prediction models can identify bugs before release using historical data from software projects for training. The study used software change metrics for defect prediction, and the performances of machine learning and hybrid algorithms were compared. This study uses different machine learning techniques to create defect prediction models, including Random Forest, Multilayer Perceptron, Fuzzy-AdaBost, and Logitboost. With Logitboost, was reached the best accuracy.

Focusing on the feature extraction part, [14] proposes new damage classifiers for locating and quantifying damage based on a supervised learning problem. A new feature extraction approach using time series analysis is introduced to extract damage-sensitive features from auto-regressive models. The coefficients and residuals of the AR model obtained from this approach are used as the main features in the proposed supervised learning classifiers, which are categorized as coefficient-based and residual-based classifiers. These classifiers are validated using experimental data for a laboratory frame and a four-story steel structure. They are shown to be able to locate and quantify damage, with the residual-based classifiers yielding better results than the coefficient-based classifiers. Furthermore, comparative analyses show that these methods are superior to some classical techniques.

The following related work uses the same dataset in our paper and solves a classification problem for faulted parts. The authors of [15] used the Bosch dataset uploaded on Kaggle. First, they trained a model that predicts which parts are most likely to fail. Then, to manage many categorical features, they employed the FTRL(Follow The Regularized Leader) algorithm to train a model using solely categorical features. Afterward, they stacked the probability predictions with numerical and date features as a new column. This technique serves as a means of reducing features, in which all the categorical features are condensed into one feature column. The top 200 features were used to train an XGBoost model on the entire training dataset, which consists of approximately 1 million samples. The training data were randomly divided into three subsets, each containing 33% of the data. Three separate XGBoost models with the same hyperparameters were trained on 67% of the data and evaluated on the remaining 33% of the data.

Authors in [16] propose a systematic feature engineering and selection methodology considering data from a variety of sensors. From the originally recorded time series and some newly generated time series, a variety of time and frequency domain features are extracted and then selected. Such approaches can be also used in the machine malfunction problems where there is abundancy of time-series data originating from IoT devices and sensors.

The main focus in many related papers is typically on data preprocessing, and this was also our primary focus. Our work dealt with numerical features and extracted additional features from the date values, emphasizing capturing the time dependence between different parts. For example, in [15], the feature extraction was made directly on the features combined, while we deal with numerical and date features separately with different techniques for each. However, we also devoted significant attention to analyzing the raw data, as the time series data provided by Bosch were anonymized and normalized, which made it necessary to create estimations of the duration in the time series data. We were able to leverage this information to develop additional features based on the date values.

Additionally, we paid careful attention to null values during feature extraction, given the large number of numerical values we were dealing with (970 in total). Finally, we addressed the challenge of imbalanced data by performing downsampling and oversampling. We use a gradient-boosted decision tree as our primary model for malfunction prediction. This robust algorithm can handle complex data structures and identify the most valuable features for predicting the desired outcome.

III. Methods

The primary objective of this paper is to develop a fault prediction model using the Bosch dataset. The dataset was released by Bosch in 2016 as a challenge to improve its future defect reduction efforts. Furthermore, we aim to solve a classification problem using the dataset. We trained different models separately, Random Forest, Decision Tree, GradientBoostingClassifier, AdaBoostClassifier, and XGBoost. After evaluation, we found that the XGBoost model produced the highest accuracy.

The numerical data in the dataset contains many zero values, with 929,125,166 fields or 80.91% of the total being empty. However, this is not surprising, as each part goes through a specific set of stations and measurements, and empty cells indicate that a part did not undergo a particular measurement at a certain station or line. As such, these zero values are not considered missing data, and filling them using standard methods like mean imputation or forward/backward filling is inappropriate.

The dataset comprises 970 numerical columns, including the Id and Response columns. There are no columns with all null values, so such columns cannot be dropped. However, 227 columns have 99% null values, and their relevance must be determined to decide whether to keep or discard them. Upon closer inspection, we found that 11 of these columns have non-zero values for parts that were not classified as failed, and these can be immediately removed. We calculated Pearson correlation coefficients between the 11 columns and the Response variable to ensure these columns are irrelevant. That indicated correlation values close to zero, meaning they are not significantly correlated with the outlier parameter and can be safely removed.

Our next step is to use the XGBoost model to identify the most and least significant numerical features to reduce the set of features. First, we ran the model on the remaining 227 columns from the previous step and found that 107 features had a significance of 0.005 or less, so we removed them. This process left us with 852 numerical features.

Next, we applied the XGBoost model to the remaining 852 columns and removed those with more than 90% null values and a significance of 0, resulting in the removal of 218 columns. It brought the total number of numerical features down to 634, reducing the percentage of empty fields to 47.58% from the initial 80.91%.

Of the remaining 534 columns, 100 had a significance greater than 0 and more than 90% non-zero values, so we used them directly in the model. For the remaining 434 columns, we found that, on average, 49 columns had over 50% non-zero values. Therefore, for each of these 49 columns, we merged 10 zero columns and used them in the combining process, where the first non-zero value is taken. As a result of these steps, we ended up with 149 numerical features and 13.58% null fields.

Moving next to the date features, we have extracted some features from the existing date features, and then we have added additional date features with a focus on time dependence between parts.

To include time dependence, we added the following time domain features:

1) Number of parts in one takt (6 minutes). To calculate the number of parts that pass through the measuring stations in one takt, we look for the consecutive parts with the same starting takt where the first column is the takt and the second column is the number of parts that pass in the same takt.
2) Number of failures in the next 1, 10, 24 hours
3) Number of failures in the last 1, 10, 24 hours

The date features represent the date and time the measurements were taken. These features can be important because they capture temporal patterns and trends that may be relevant for predicting the target variable.

Since the test and training data are consecutive parts with indices from 1 to 2,367,494, the specified features are appropriate for the training and unlabeled data.

Of the 4,258 categorical features, 1,913 are duplicates and will be removed. Among the remaining ones, 1,549 have a single value, and 428 have multiple values. Any empty feature will also be discarded. Categorical feature values are represented as classes denoted by T followed by a number, which labels different processes. For instance, column L1_S24_F1269 contains four classes: 'T1372', 'T618624', 'T83888', and 'T8389632'. After removing duplicates and empty features, we use one-hot encoding to represent each category with an integer. One-hot encoding transforms a single variable with d distinct values into d binary variables, where each observation indicates a particular binary variable's presence (1) or absence (0). This results in a vector of size 988 for each row, but since most values are zero, we end up with a sparse matrix. To avoid overfitting, we will compare the performance of the model with and without categorical features.

Moreover, we will reduce the dimensionality of the resulting matrix using a dimensionality reduction algorithm. Sparse PCA is an unsupervised learning method used in statistical analysis to identify sparse features that can reconstruct the data. We will replace the 988 features with 5 features obtained from Sparse PCA.

After processing the features, we have 1,183,747 rows and 173 features from the training set, of which 19 are date, 149 are numeric, and 5 are categorical.

Classification models aim to assign data to different classes, but in an unbalanced dataset, one class may have a much larger number of samples than the other classes. It creates a majority class and a minority class, which can be problematic during model training because the model may not learn enough about the minority class. In our case, the defect class is the minority class, with only 6,879 parts, or 0.58% of the training data being defects.

One effective approach is to reduce the sample size of the majority class and increase the weight of the minority class. It can be achieved by either downsampling the majority class or oversampling the minority class. Downsampling may lead to a loss of information, while oversampling can result in overfitting. [17] elaborate on the sampling approaches, and they suggest that downsampling approaches give a better overall performance on all datasets. Thus, we tested the model with both techniques, and the best results were achieved by oversampling the minority class and downsampling the majority class.

IV. Experiments

In this section, we describe the datasets and the experimental results obtained in our study.

A. Data description

Bosch, a leading manufacturing company, has made a dataset available on Kaggle as part of a research project to assess the quality and safety of its manufacturing recipes. The dataset tracks parts as they move through the production lines, each with a unique identifier that serves as a row in the dataset. The dataset is a time series, with each row representing a specific section and the date attribute providing the time when each measurement was taken.

The dataset contains three types of characteristics: numeric, categorical, and date characteristics. Each feature is named according to a specific convention, including the line, station, and feature number. For example, the feature L1_S25_F2202 was measured at line 1, station 25, and has a sequence number of 2202. Therefore, this feature is measured in column L1_S25_D2203, since each feature Lx_Sy_Fn is calculated at time Lx_Sy_D(n+1).

By processing the column names, we concluded that there are four lines and 52 segment stations. Each line performs a specific production process, and one line can have multiple stations where different operations are performed, such as machining, turning, and welding. Line L0 has stations S0 to S23, line L1 has stations S24, S25, line L2 has stations S26, S27, S28, and line L3 has stations S29 to S51. Each workstation performs various tests and measurements on a given part, resulting in 4,264 features. We have 969 numeric features, 1,156 date features, and the rest are categorical.

The date features are normalized and given as takt time, which is a value for how long it takes for a process to fulfill the demand. To determine the period the data represent, we analyzed the unique values in the date features. There are 105,413 unique values, ranging from 0 to 1,718.48, with a rate of 0.01. Figure 2 shows a graph of the values and their frequencies, indicating space in the middle, meaning no measurements were made during that period.

We conducted an autocorrelation analysis using a lag function to understand the time dependence between the parts further. The results are presented in Figure 3, where the most significant values are at 1,675, with 7 local maxima corresponding to 7 days of the week. Therefore, 16.75 in the normalized data correspond to one week, and the data have a granularity of 6 minutes. It means that 0.01 corresponds to 6 minutes, and 1 corresponds to 600 minutes or 10 hours, indicating that the data correspond to two years.

For each part, we determined the maximum and minimum date, their difference, and the station with the maximum and minimum time. We also calculated the path size for each part by counting the number of non-zero values. Additionally, we found the week for the maximum and minimum dates by calculating a module of 16.75 on such values. Given the week, we labeled each day of the week, where 1 is Monday, 2 is Tuesday, 3 is Wednesday, 4 is Thursday, 5 is Friday, 6 is Saturday, and 7 is Sunday. Therefore, there are a total of 10 date features. In addition, as described previously, we added 9 features based on time dependence. Overall, in addition to focusing on the preprocessing of the data, we also analyzed the raw data to extract useful features for our model.

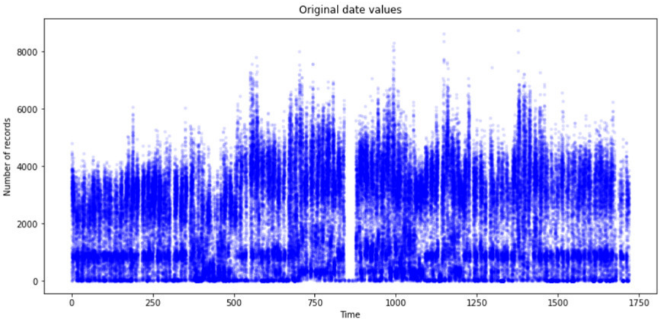


Fig. 2. Date values with frequencies

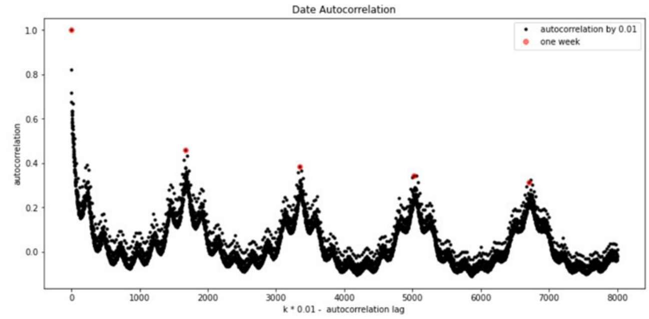


Fig. 3. Autocorrelation of date features. Autocorrelation for the number of observations recorded daily as a function of the time lag between them.

B. Experimental setup

After extracting features from the dataset, we train multiple classifiers on a balanced dataset. The classifiers include Random Forest, Decision Tree, Gradient Boosting, Ada Boosting, and XGBoost. Next, we will compare the accuracy results between 20% of testing data and the entire training dataset. Then, using the unlabeled dataset, we will predict the number of faults.

To obtain the best results, we have trained models using different combinations of features. We found that including all features (date, numerical, and categorical) resulted in the worst accuracy. Therefore, we removed the categorical features, leading to the models' highest accuracy. We also compared the results using standardized features and raw features. Standardized features yielded better results, and we used Z-Score Standardization. Z-Score Standardization is a widely used method to standardize data in machine learning. It transforms each value of a given feature in the dataset to a representative number of standard deviations away from that feature's mean. The resulting standardized value measures how far the raw value is from the mean in standard deviation units.

C. Accuracy results

Because we are dealing with an unbalanced dataset, for correct accuracy results, we have to consider true positives, true negatives, false positives, and false negatives. AUC-ROC (Area Under the Receiver Operating Characteristic Curve) plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings and calculates the area under the curve. The AUC-ROC score ranges from 0 to 1, with higher values indicating better performance. MCC (Matthews Correlation Coefficient): MCC is a correlation coefficient used in binary classification that considers true positives, true negatives, false positives, and false negatives. MCC ranges from -1 to +1, with +1 indicating a perfect classification, 0 indicating a random classification, and -1 indicating a completely wrong classification. F1-Score (F-Measure): F1-score is the harmonic mean of precision and recall. It provides a single score that balances precision and recall and is often used to measure a model's performance in binary classification. F1-score ranges from 0 to 1, with higher values indicating better performance.

After several tests of the parameter values, the following obtained the best result using the XGBoost classifier: learning_rate=0.2, n_estimators=100, max_depth=16, min_child_weight=3, colsample_bytree=0.9, gamma=1, subsample=0.9, booster='gbtree', objective='binary:logistic'.

Table I reports results regarding AUC ROC, MCC, Precision, Recall, and F-Score on the training dataset. Table II provides accuracy scores on 20% testing data.

TABLE I
SUMMARY OF ACCURACY SCORES ON TRAINING DATASET

Model	**AUC ROC**	**MCC**	**Precision**	**Recall**	**F-Score**
Random Forest	0.705	0.148	0.53	0.71	0.51
Decision Tree	0.717	0.149	0.53	0.72	0.51
Gradient Boosting	0.691	0.22	0.56	0.69	0.59
Ada Boosting	0.629	0.187	0.57	0.63	0.59
XGBoost	**0.906**	**0.808**	**0.9**	**0.91**	**0.9**

TABLE II
ACCURACY SCORES ON 20% TEST DATA

Model	AUC ROC	MCC
Random Forest	0.801	0.613
Decision Tree	0.986	0.972
Gradient Boosting	0.917	0.837
AdaBoost	0.903	0.809
XGBoost	**0.997**	**0.994**

The evaluation metrics also provide additional insights into the performance of the models. For example, the XGBoost model achieved the highest accuracy of all the models, indicating that it correctly classified most test sets. In addition, the XGBoost model also has the highest F1 score, meaning a good balance between precision and recall. On the other hand, the other models, such as Gradieng and Ada Boosting, struggled to classify the fault class, reflected in their lower F1 scores. The MCC scores were also generally low for all models except XGBoost, indicating that the models had trouble with the imbalanced nature of the dataset. However, the AUC-ROC scores for Random Forest and Decision Tree were relatively high, suggesting they could distinguish between the positive and negative classes reasonably well. The results indicated that the XGBoost model is the most effective for this classification task.

Using the best accuracy model, XGBoost, to classify the additional unlabeled dataset, we identified 60,028 out of 1,183,748 samples as faulty. It means that the fault rate in the unlabeled dataset is approximately 5.07%. This information can help identify potential issues in the manufacturing process and improve the quality control procedures. However, the accuracy of the classification results on the unlabeled dataset may vary depending on the data's quality and representativeness and the model's performance on unseen data.

V. Conclusion

This paper presented a novel approach to predict malfunctions in manufacturing processes using the Bosch manufacturing dataset. The dataset is large and complex, containing many features with varying data types.

In this study, we addressed the challenge of handling a large number of features in the Bosch manufacturing dataset. The feature extraction process was a crucial step in the predictive modeling pipeline. We performed an iterative feature selection process to identify the most relevant features for predicting malfunctions in the manufacturing process. Additionally, time-dependent features were added to the dataset, improving the predictions' accuracy. The feature selection process was carried out carefully to ensure the selected features were relevant for the prediction task while avoiding overfitting the training data. The selected features were then used to train and evaluate various machine learning models. Handling the unbalanced dataset was another key factor in achieving high accuracy scores, and this was accomplished by performing both downsampling on the majority class and oversampling on the minority class. The study results showed that XGBoost outperformed the other models in terms of accuracy scores, including AUC ROC, MCC, and F1-score.

The proposed approach of supervised malfunction prediction using machine learning models and feature engineering can have significant implications in the manufacturing industry. Manufacturers can proactively prevent downtime, optimize maintenance schedules, and minimize production losses by accurately predicting malfunctions. As a result, it can improve manufacturing operations' efficiency and productivity, leading to cost savings and increased profitability. Moreover, the approach can also help identify patterns and insights in the data that can be used for process optimization and improvement.

In future work, we aim to extend our study to include categorical features. While we made significant progress in feature extraction and handling unbalanced data, the categorical features remain an important part of the dataset that needs further investigation. Therefore, we plan to explore various techniques for feature extraction on categorical data and evaluate their impact on the model's overall accuracy.

Acknowledgment

This work was partially financed by the Faculty of Computer Science and Engineering at the Ss. Cyril and Methodius University, Skopje, Macedonia. This work is also partially funded by FCT/MEC through national funds and, when applicable, co-funded by the FEDER-PT2020 partnership agreement under the project **UIDB/50008/2020**.

References

[1] M. Ghobakhloo, "Industry 4.0, digitization, and opportunities for sustainability," *Journal of cleaner production*, vol. 252, p. 119869, 2020.

[2] B. Natesha and R. M. R. Guddeti, "Fog-based intelligent machine malfunction monitoring system for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 7923–7932, 2021.

[3] K. Wang and Y. Wang, "How ai affects the future predictive maintenance: a primer of deep learning," in *Advanced Manufacturing and Automation VII 7*. Springer, 2018, pp. 1–9.

[4] P. Poór, J. Basl, and D. Zenisek, "Predictive maintenance 4.0 as next evolution step in industrial maintenance development," in *2019 International Research Conference on Smart Computing and Systems Engineering (SCSE)*, 2019, pp. 245–253.

[5] Z. M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, and B. Safaei, "Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0," *Sustainability*, vol. 12, no. 19, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/19/8211

[6] Y. Ren, "Optimizing Predictive Maintenance With Machine Learning for Reliability Improvement," *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 7, no. 3, 05 2021, 030801. [Online]. Available: https://doi.org/10.1115/1.4049525

[7] E. Zdravevski, P. Lameski, A. Kulakov, S. Filiposka, D. Trajanov, and B. Jakimovski, "Parallel computation of information gain using hadoop and mapreduce," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2015, pp. 181–192.

[8] E. Zdravevski, P. Lameski, A. Kulakov, B. Jakimovski, S. Filiposka, and D. Trajanov, "Feature ranking based on information gain for large classification problems with mapreduce," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 2. IEEE, 2015, pp. 186–191.

[9] E. Zdravevski, P. Lameski, C. Apanowicz, and D. Slezak, "From big data to business analytics: The case study of churn prediction," *Applied Soft Computing*, vol. 90, p. 106164, 2020.

[10] M. Grzegorowski, E. Zdravevski, A. Janusz, P. Lameski, C. Apanowicz, and D. Slezak, "Cost optimization for big data workloads based on dynamic scheduling and cluster-size tuning," *Big Data Research*, vol. 25, p. 100203, 2021.

[11] P. F. Orrù, A. Zoccheddu, L. Sassu, C. Mattia, R. Cozza, and S. Arena, "Machine learning approach using mlp and svm algorithms for the fault prediction of a centrifugal pump in the oil and gas industry," *Sustainability*, vol. 12, no. 11, p. 4776, 2020.

[12] K. Khalil, O. Eldash, A. Kumar, and M. Bayoumi, "Machine learning-based approach for hardware faults prediction," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 11, pp. 3880–3892, 2020.

[13] W. Rhmann, B. Pandey, G. Ansari, and D. K. Pandey, "Software fault prediction based on change metrics using hybrid algorithms: An empirical study," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 4, pp. 419–424, 2020.

[14] M. H. Chegeni, M. K. Sharbatdar, R. Mahjoub, and M. Raftari, "New supervised learning classifiers for structural damage diagnosis using time series features from a new feature extraction technique," *Earthquake Engineering and Engineering Vibration*, vol. 21, no. 1, pp. 169–191, 2022.

[15] A. Mangal and N. Kumar, "Using big data to enhance the bosch production line performance: A kaggle challenge," in *2016 IEEE international conference on big data (big data)*. IEEE, 2016, pp. 2029–2035.

[16] E. Zdravevski, P. Lameski, V. Trajkovik, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo, and N. Garcia, "Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering," *IEEE Access*, vol. 5, pp. 5262–5280, 2017.

[17] S. Tyagi and S. Mittal, "Sampling approaches for imbalanced data classification problem in machine learning," in *Proceedings of ICRIC 2019: Recent Innovations in Computing*. Springer, 2020, pp. 209–221.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 297–302

DOI: 10.15439/2023F1627
ISSN 2300-5963 ACSIS, Vol. 35

Reranking for a Polish Medical Search Engine

Jakub Pokrywka, Krzysztof Jassem, Piotr Wierzchoń
Adam Mickiewicz University
Faculty of Mathematics and Computer Science,
Email: {firstname.lastname}@amu.edu.pl

Piotr Badylak, Grzegorz Kurzyp
WN PWN,
Email: {firstname.lastname}@pwn.pl

***Abstract*—Healthcare professionals are often overworked, which may impair their efficacy. Text search engines may facilitate their work. However, before making health decisions, it is important for a medical professional to consult verified sources rather than unknown web pages. In this work, we present our approach for creating a text search engine based on verified resources in the Polish language, dedicated to medical workers. This consists of collecting and comprehensively analyzing texts annotated by medical professionals and evaluating various neural reranking models. During the annotation process, we differentiate between an abstract information need and a search query. Our study shows that even within a group of trained medical specialists there is extensive disagreement on the relevance of a document to the information need. We prove that available multilingual rerankers trained in the zero-shot setup are effective for the Polish language in searches initiated by both natural language expressions and keyword search queries.**

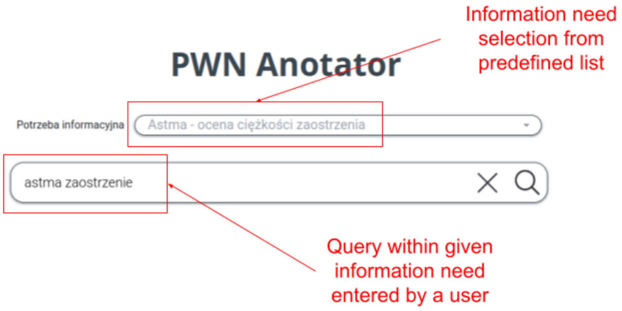
I. Introduction

WHEN seeking content in a domain-specific text, a medical professional is faced with the dilemma of whether to consult a work published by a verified source or to query the Internet. Often, verified documents are published only in print, and so browsing them is time-consuming. On the other hand, a lot of Internet content is created by non-professionals and is not error-free, thus finding accurate data is difficult. This is especially true in the case of non-English online resources. However, querying Google or Wikipedia is tempting when one has to act under time constraints, for example, during a medical appointment. Considering the workload of healthcare workers [1], this statement holds even more significance. To address this issue, medical publishers attempt to provide online access to their domain-specific resources.

This paper describes the results of a project aimed at creating an intuitive search engine encompassing 852 books on medicine published in the Polish language. The tool is designed to find a book passage (usually a paragraph) relevant to a question posed in natural language and to present it to the user.

We present our novel approach to data annotation, which distinguishes between information needs and term queries. Our annotation data analysis shows extensive disagreement between trained specialists regarding document relevance. We evaluate several rerankers for the domain-specific task in the Polish language, for which currently only zero-shot rerankers are available. Our experiments prove the superiority of such reranking models to a strong BM25-based baseline. It is found that rerankers trained on vast multilingual data in a zero-shot setup perform better than a language-specific model fine-tuned to minor domain reranking data.

The rest of the paper is organized as follows. Section II concerns related work in biomedical and medical natural language processing, especially information retrieval. In Section III, we explain our task as a reranking problem, differentiating it from a full retrieval setup, and provide an overview of our search engine configuration. In Section IV we describe a typical use case for our system, which determines our annotation process presented in Section V. In Section VI we report statistics and the conclusion of the collected dataset and present our dataset preparation steps. Then, in Section VII the reranking task setup is described, which leads to Section VIII, where reranking models are presented, and Section IX, where their results are reported. In sections X and XI the possible future work and conclusions of this paper are presented.


PWN Anotator

Potrzeba informacyjna: Astma - ocena ciężkości zaostrzenia

astma zaostrzenie

Information need selection from predefined list

Query within given information need entered by a user

Fig. 1. Information Need selection for annotators and sample query.

II. Related work

Recent findings in Natural Language Processing, particularly Large Language Models (LLMs), have significantly increased the level of language understanding not only in general-oriented tasks, but also in biomedical and medical tasks [2], [3], [4], [5], [6]. In [7] the authors present a benchmark for a question-answering task in the medical domain, and show that the answers provided by LLMs are in agreement with expert knowledge in 93% of cases. Another medical benchmark is introduced in [8], where the authors conclude that pretraining language models from scratch results in gains over fine-tuned general-domain language models. However, a comprehensive survey on biomedical question-answering [9] shows the immaturity of such systems in a real-life scenario. All the above-mentioned models and benchmarks presented are in English, and no such corpora and models exist for the Polish language, which differentiates this work from the others.

To make a binding decision on a medical case, a human expert (for example, a doctor) prefers to rely on verified medical knowledge sources, rather than one (even precise) answer generated by a language model. This is mainly due to the phenomenon of artificial hallucination [10], [11], [12]. Relevant information may be found in a digital resource by a ranking function (e.g. BM25), optionally modified by a reranker. One existing benchmark for the reranking task [13] is available for the English language. [14] reports on the machine translation of the MS MARCO dataset [15] into multiple languages. The authors claim that their reranking models perform well even in non-English languages when fine-tuned in a zero-shot manner. Healthcare decision-making based on a search engine are examined in [16], [17], and some medical search models and datasets are proposed in [18], [19], [20].

III. Search engine setup

According to [13], models based on reranking are superior to full retrieval models. Moreover, it is easier to perform automatic evaluation on reranking models than full retrieval models, because such evaluation avoids cases when the model retrieves a document unseen by any human annotator. For these reasons, we decided to formulate our task in a reranking setup.

In order to meet commercial expectations, we needed to craft as strong baseline as possible. We started with the SOLR engine, equipped with the Polish Morfologik [21] lemmatizer. We handcrafted the scoring function, awarding full n-gram matches higher scores than word matches. Moreover, we used carefully adjusted weights to ensure case sensitivity, as this is crucial for the recognition of medical abbreviations (AED, DIC, etc.).

IV. Medical search case scenario

To mirror user needs we fabricated case scenarios, namely real-world situations that may cause an Information Need (IN) on the part of the system user. A case scenario consists of an event description, initial conditions, and the Information Need, represented in two forms: a natural language expression and a term query. We define an IN as an abstract term: the knowledge that a user wants to acquire from the system.

An example scenario is shown here:

- Event description: *A 30-year-old female patient presents to a PCP (Primary Care Provider) in a small town. She has severe sore throat and a high temperature.*
- Initial conditions:
  - The doctor measured the patient’s temperature (38.5 °C).
  - The doctor confirmed characteristic symptoms of tonsillitis: distended and reddened mucous membrane of the tonsils and palate.
  - The patient reported that she is breastfeeding.
- Natural language description of the IN: *I want to learn how to treat tonsillitis in a breastfeeding woman.*
- Term query: *tonsillitis in a breastfeeding woman - treatment*

TABLE I
Statistics for annotators

	mean	stdev
Information Needs annotated	39	43
total queries used	264	230
total passages annotated	12,068	10,662
time for an Information Need	67 min	55 min
time for a query	6 min	2 min
time for an annotation	8 sec	3 sec
relevant/all annotations ratio	0.29	0.17

V. Annotation process

We hired 21 medical workers (doctors, paramedics, and medical students) for consultation on system requirements and for the annotation process. Initially, they were asked to propose some INs that they may encounter in their work. Additionally, to collect other potential INs, we used the website https://konsylium24.pl/, which is a Polish web forum for medical staff. The website verifies whether users are listed in Polish doctors’ registers.

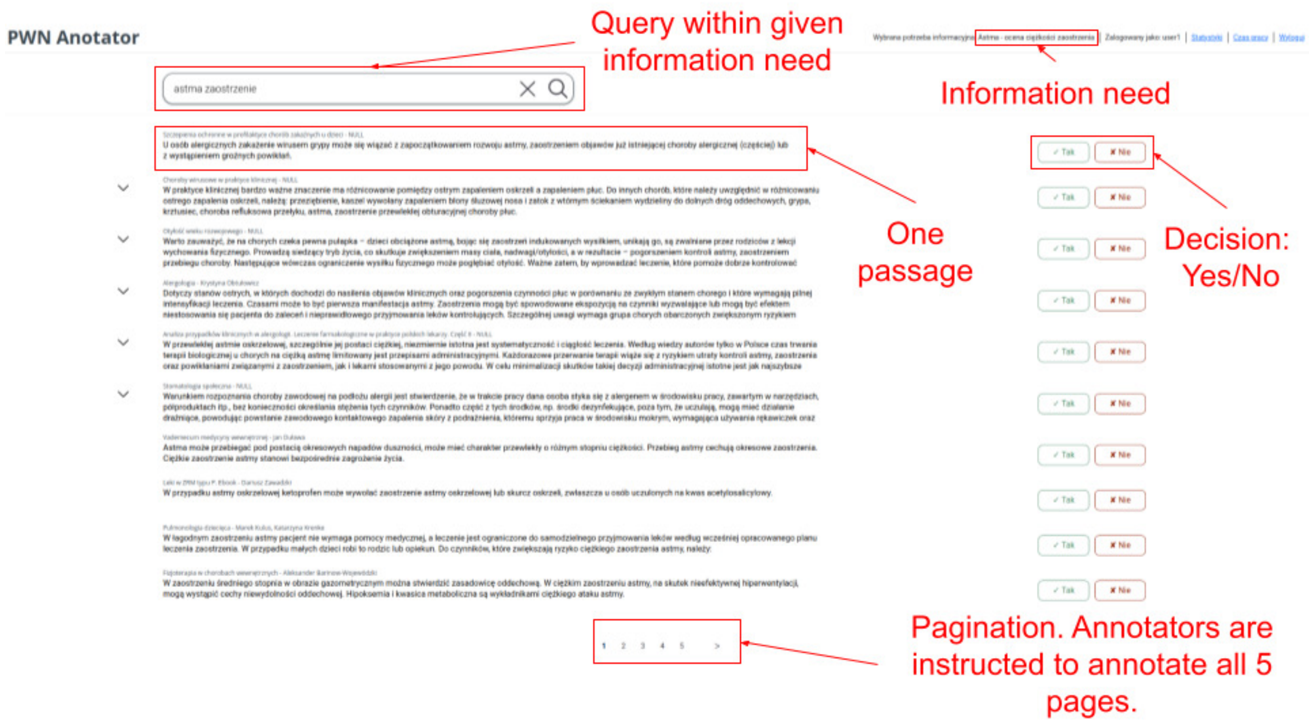
Once the set of INs had been established, we started the annotation process. After logging in, an annotator chooses an IN which he/she feels familiar with. The selection window is presented in Figure 1.

The annotator inputs a number of **queries** for each IN to a SOLR-based search engine, so that for one IN there are always multiple queries. The user may input any words that may help them find a relevant document (synonyms, hyperonyms, etc.).

Example queries for the above-mentioned IN may be: *how to treat tonsillitis in a breastfeeding woman?; tonsilitis breastfeeding treatment; breastfeeding medicines tonsilits; breastfeed woman amigdalitis;* etc.

The annotators were advised not to exceed 20 queries for an IN, and to stop when further enquiry was unlikely to return new relevant passages. The annotation platform returned a maximum of 5 pages, with 10 passages per page, for a query, as in Figure 2. The annotators were asked to read all returned passages and to tag them as relevant/irrelevant to the IN only, regardless of the input search query. If the same passage was returned again within an IN in response to a different query, the annotator would tag it once more. In total, the annotators spent over 478 hours actively tagging the passages. Statistics on their work are given in Table I.

The aim of the procedure was to acquire a more accurate dataset for training and evaluation than a simple query–passage relevancy dataset, which would be limited by the top documents returned by SOLR for one query. The dataset should help the reranker learn semantic structures such as synonyms and hyperonyms.


PWN Anotator

Wybrana potrzeba informacyjna: Astma - ocena ciężkości zaostrzenia | Zalogowany jako: user1 | Statystyki | Czas pracy | Wyloguj

astma zaostrzenie

Szczepienia ochronne w profilaktyce chorób zakaźnych u dzieci - NULL
U osób alergicznych zakażenie wirusem grypy może się wiązać z zapoczątkowaniem rozwoju astmy, zaostrzeniem objawów już istniejącej choroby alergicznej (częściej) lub z wystąpieniem groźnych powikłań.
✓ Tak ✗ Nie

Choroby wirusowe w praktyce klinicznej - NULL
W praktyce klinicznej bardzo ważne znaczenie ma różnicowanie pomiędzy ostrym zapaleniem oskrzeli a zapaleniem płuc. Do innych chorób, które należy uwzględnić w różnicowaniu ostrego zapalenia oskrzeli, należą: przeziębienie, kaszel wywołany zapaleniem błony śluzowej nosa i zatok z wtórnym ściekaniem wydzieliny do dolnych dróg oddechowych, grypa, krztusiec, choroba refluksowa przełyku, astma, zaostrzenie przewlekłej obturacyjnej choroby płuc.
✓ Tak ✗ Nie

Otyłość wieku rozwojowego - NULL
Warto zauważyć, że na chorych czeka pewna pułapka − dzieci obciążone astmą, bojąc się zaostrzeń indukowanych wysiłkiem, unikają go, są zwalniane przez rodziców z lekcji wychowania fizycznego. Prowadzą siedzący tryb życia, co skutkuje zwiększeniem masy ciała, nadwagi/otyłości, a w rezultacie − pogorszeniem kontroli astmy, zaostrzeniem przebiegu choroby. Następujące wówczas ograniczenie wysiłku fizycznego może pogłębiać otyłość. Ważne zatem, by wprowadzać leczenie, które pomoże dobrze kontrolować
✓ Tak ✗ Nie

Alergologia - Krystyna Obtułowicz
Dotyczy stanów ostrych, w których dochodzi do nasilenia objawów klinicznych oraz pogorszenia czynności płuc w porównaniu ze zwykłym stanem chorego i które wymagają pilnej intensyfikacji leczenia. Czasami może to być pierwsza manifestacja astmy. Zaostrzenia mogą być spowodowane ekspozycją na czynniki wyzwalające lub mogą być efektem niestosowania się pacjenta do zaleceń i nieprawidłowego przyjmowania leków kontrolujących. Szczególnej uwagi wymaga grupa chorych obarczonych zwiększonym ryzykiem
✓ Tak ✗ Nie

Analiza przypadków klinicznych w alergologii. Leczenie farmakologiczne w praktyce polskich lekarzy. Część II - NULL
W przewlekłej astmie oskrzelowej, szczególnie jej postaci ciężkiej, niezmiernie istotna jest systematyczność i ciągłość leczenia. Według wiedzy autorów tylko w Polsce czas trwania terapii biologicznej u chorych na ciężką astmę limitowany jest przepisami administracyjnymi. Każdorazowe przerwanie terapii wiąże się z ryzykiem utraty kontroli astmy, zaostrzenia oraz powikłaniami związanymi z zaostrzeniem, jak i lekami stosowanymi z jego powodu. W celu minimalizacji skutków takiej decyzji administracyjnej istotne jest jak najszybsze
✓ Tak ✗ Nie

Stomatologia społeczna - NULL
Warunkiem rozpoznania choroby zawodowej na podłożu alergii jest stwierdzenie, że w trakcie pracy dana osoba styka się z alergenem w środowisku pracy, zawartym w narzędziach, półproduktach itp., bez konieczności określania stężenia tych czynników. Ponadto część z tych środków, np. środki dezynfekujące, poza tym, że uczulają, mogą mieć działanie drażniące, powodując powstanie zawodowego kontaktowego zapalenia skóry z podrażnienia, któremu sprzyja praca w środowisku mokrym, wymagająca używania rękawiczek oraz
✓ Tak ✗ Nie

Vademecum medycyny wewnętrznej - Jan Duława
Astma może przebiegać pod postacią okresowych napadów duszności, może mieć charakter przewlekły o różnym stopniu ciężkości. Przebieg astmy cechują okresowe zaostrzenia. Ciężkie zaostrzenie astmy stanowi bezpośrednie zagrożenie życia.
✓ Tak ✗ Nie

Leki w ZRM typu P. Ebook - Dariusz Zawadzki
W przypadku astmy oskrzelowej ketoprofen może wywołać zaostrzenie astmy oskrzelowej lub skurcz oskrzeli, zwłaszcza u osób uczulonych na kwas acetylosalicylowy.
✓ Tak ✗ Nie

Pulmonologia dziecięca - Marek Kulus, Katarzyna Krenke
W łagodnym zaostrzeniu astmy pacjent nie wymaga pomocy medycznej, a leczenie jest ograniczone do samodzielnego przyjmowania leków według wcześniej opracowanego planu leczenia zaostrzenia. W przypadku małych dzieci robi to rodzic lub opiekun. Do czynników, które zwiększają ryzyko ciężkiego zaostrzenia astmy, należy:
✓ Tak ✗ Nie

Fizjoterapia w chorobach wewnętrznych - Aleksander Barinow-Wojewódzki
W zaostrzeniu średniego stopnia w obrazie gazometrycznym można stwierdzić zasadowicę oddechową. W ciężkim zaostrzeniu astmy, na skutek nieefektywnej hiperwentylacji, mogą wystąpić cechy niewydolności oddechowej. Hipoksemia i kwasica metaboliczna są wykładnikami ciężkiego ataku astmy.
✓ Tak ✗ Nie

1 2 3 4 5 >

Query within given information need
Information need
One passage
Decision: Yes/No
Pagination. Annotators are instructed to annotate all 5 pages.

Fig. 2. Passages annotation view for a given Information Need.

VI. Dataset preparation

We rejected INs with fewer than 100 annotations. The dataset consists of 231 INs, each of them represented by a natural language description and one best-fitted term query. In total, we obtained 230,808 triplets of the form (IN, passage, relevancy annotation). Of these, 50,608 decisions were positive and 180,200 were negative, which means that about 22% of annotations were marked as relevant. A passage could be annotated more than once with different tags within the same IN, for example, if two or more annotators worked with the same IN. An analysis of mismatched annotations is presented in Table III. The results show weak agreement on IN–passage relevance between annotators. The mixed opinion percentages range from 15.6% (for two annotations) to above 50% (seven annotations and more). Carelessness on the part of the annotators is probably not the main reason for the disagreement, as we monitored their activity in the annotation platform.

VII. Task setup

We carried out experiments with Information Needs being represented firstly by a term query and then by a natural language expression.

For each IN, we queried the backbone SOLR-based search engine. The returned documents (no more than 500 for each IN) formed an input sample for the reranking model. The proposed model was expected to return the same set of documents sorted in order of decreasing relevance.

TABLE II
Information Need statistics. Annotations concern relevance for (IN, passage) pairs. There may be multiple annotations for one pair.

items per IN	minimum	maximum	mean	median
queries	1	109	21	15
returned passages	50	2279	439	325
annotations	100	4491	999	714
decision: relevant	1	1868	219	119
decision: irrelevant	23	4396	780	562

The golden truth relevance of a document is binary. The document is regarded as relevant to an IN if it was tagged as positive by at least 50% of annotators. NDCG@10 and NDCG@50 are used as evaluation metrics. NDCG metric is well defined in [22].

VIII. Models

For a baseline, we used the SOLR-based model described in section III.

HerBERT [23] (huggingface: allegro/herbert-base-cased) is a Polish-language model which achieves good results in Polish language understanding tasks [24]. We fine-tuned it on our training dataset. Unfortunately, there are no mass Polish corpora for reranking to use along with our data.

There exist some multilingual neural cross-encoder rerankers running on Polish texts. They use an mT5 [25] or mMiniLM [14] backbone, also trained on Polish texts. These models are further fine-tuned for document reranking on multilingual MS MARCO datasets using one or more

TABLE III

Statistics on annotations for (Information Need, passage) pairs. One pair may be annotated by multiple annotators, **k** represents how many annotators annotated given (Information Need, passage) pair. Column **annotiations** equals to **k*pairs with k annotations**. Column **all annotations relevant** is simply a number of annotations in which all the annotators agree that a given pair is relevant. Column ≥ **0.5 annotations relevant** stands for a number of pairs in which at least half of the annotators agree that a pair is relevant.

k	pairs with k annotations	annotations	annotations %	all annotations relevant	all annotations irrelevant	annotations mixed	all annotations relevant %	all annotations irrelevant %	annotations mixed %	≥ 0.5 annotations relevant	≥0.5 annotations relevant %
1	70301	70301	30.5 %	10572	59729	0	15.0 %	85.0 %	0.0 %	10572	15.0%
2	15189	30378	13.2 %	1850	10970	2369	12.2 %	72.2 %	15.6 %	4219	27.8%
3	5452	16356	7.1 %	540	3478	1434	9.9 %	63.8 %	26.3 %	1112	20.4%
4	3607	14428	6.3 %	275	2055	1277	7.6 %	57.0 %	35.4 %	1009	28.0%
5	2134	10670	4.6 %	136	1069	929	6.4 %	50.1 %	43.5 %	438	20.5%
6	1510	9060	3.9 %	86	752	672	5.7 %	49.8 %	44.5 %	330	21.9%
7	1024	7168	3.1 %	43	458	523	4.2 %	44.7 %	51.1 %	222	21.7%
8	863	6904	3.0 %	36	377	450	4.2 %	43.7 %	52.1 %	197	22.8%
9	583	5247	2.3 %	30	255	298	5.1 %	43.7 %	51.1 %	141	24.2%
10	589	5890	2.6 %	34	224	331	5.8 %	38.0 %	56.2 %	151	25.6%
>10	3057	54406	23.6 %	117	910	2030	3.8 %	29.8 %	66.4 %	818	26.8%
total	104309	230808	100%	13719	80277	10313	13.2 %	77.0 %	9.9 %	19209	18.4%

languages other than English (but not including Polish). The authors of [14] proved that these models learn to rerank documents in this zero-shot setup. We used mT5-based rerankers (huggingface: unicamp-dl/mt5-base-mmarco-v2, unicamp-dl/mt5-3B-mmarco-en-pt, unicamp-dl/mt5-13b-mmarco-100k) and mMiniLM-based rerankers (huggingface: cross-encoder/mmarco-mMiniLMv2-L12-H384-v1), which vary in terms of number of parameters and inference time. We used these rerankers in two setups: without fine-tuning (no-ft) and with additional fine-tuning to our training data (ft). We did not fine-tune the mT5-based rerankers because of the long inference time, which meant that they would not be useful as production models. We also tested several multilingual bi-encoders, among which mpnet (huggingface: paraphrase-multilingual-net-base-v2) [26] performed best. All fine-tuned models were trained separately on term queries and natural language queries.

IX. Results

The results are given in Table IV. Almost all cross-encoder rerankers achieve better results than the SOLR baseline. Only HerBERT performs worse, probably due to its being trained with only 100 samples of INs, in contrast to other transformer models that were trained previously on the multilingual MS MARCO dataset containing millions of samples. Cross-encoder rerankers based on mMiniLM are the fastest as regards inference time and achieve results that are much better than the baselines, but not as good as those of the larger models, especially the reranker based on mT5 13B. Further fine-tuning of the reranker based on mMiniLM on our 100 samples dataset improves its quality on natural language queries, but not on term queries. All of the cross-encoder models produce better results when trained on term queries than when trained on natural language queries. This also holds for HerBERT, although that model did not see multilingual MS MARCO or another reranking dataset with short queries. For the bi-encoder mpnet the opposite is true, possibly because of the similarity of the natural language sentences in the corpus on which it was trained.

In our opinion, the ft mMARCO MiniLM appears to be the best model for production applications. Its inference time is satisfactory and increases the NDCG@10 from 34.30 to 43.76 in term queries, and from 28.18 to 40.52 in natural language queries. The NDCG@50 gains are not less resounding-respectively from 32.72 to 34.80 and from 25.16 to 32.53. However, in terms of business terms, we value NDCG@10 over NDCG@50, since we expect a user to be more likely to browse only the top ten search results than 50.

X. Future work

The next step is to perform an automatic translation of the MS MARCO dataset into the Polish language and to fine-tune a Polish or multilingual model. It would be beneficial to test models that have also been pre-trained on Polish medical text corpora. Another suggestion is to replace the SOLR search system with a fast bi-encoder network or late interaction transformer [27] in order to enrich the reranker input with passages using synonyms in the medical domain, which are difficult to create manually. After releasing the product for commercial use, we will collect real users' logs for the model training dataset, and run A/B tests.

TABLE IV

MODELS' RESULTS ON THE TEST DATASET. FINE-TUNED MODELS ARE TRAINED SEPARATELY ON TERM QUERIES AND NATURAL LANGUAGE QUERIES. THE INFERENCE TIME IS AVERAGED FOR ONE NATURAL LANGUAGE IN QUERY WITH UP TO 500 DOCUMENTS FOR BATCH SIZE 30 AND THE NVIDIA A100 80GB MODEL CARD. FOR BI-ENCODERS, DOCUMENT ENCODING IS NOT INCLUDED IN THE INFERENCE TIME, AS IT MAY BE DONE OFFLINE. THE ABBREVIATION FT INDICATES FINE-TUNING ON OUR TRAINING DATASET, AND NO-FT INDICATES NO FINE-TUNING.

	term query		natural language query			
method	NDCG@10	NDCG50	NDCG@10	NDCG50	inference time [s]	params
random baseline	6.19	7.12	4.58	5.99	-	-
SOLR	34.40	32.72	28.18	25.16	-	-
no-ft mmpnet bi-encoder	26.02	23.46	29.30	24.38	0.05	278M
ft HerBERT base	30.86	28.44	14.36	13.83	2.08	124M
no-ft mMARCO MiniLM	43.41	34.69	35.64	28.55	0.96	118M
ft mMARCO MiniLM	43.76	34.80	40.52	32.53	0.96	118M
no-ft mT5 base	41.17	33.81	36.94	29.70	3.90	582M
no-ft mT5 3B	44.78	38.02	43.13	34.11	27.60	3742M
no-ft mT5 13B	45.45	39.97	44.87	36.25	93.47	12921M

XI. CONCLUSIONS

In this paper, we have described the process of collecting datasets for a search engine for healthcare professionals. We placed emphasis on cooperation with specialized end users. We built models for queries formulated either in natural language or by means of keywords. We distinguished between an information need and a query that serves to satisfy such a need. We fine-tuned and evaluated several rerankers, which turned out to perform better than the baselines. In our experiments, searching with term queries yielded slightly better results than the use of natural language queries. Moreover, we observed a considerable lack of consent in annotations between qualified medical workers.

Our work is based on Polish medical texts, for which no mass reranker corpora or reranker models are available, except for those fine-tuned in a zero-shot manner. We have shown that the described setup is sufficient for creating a production-ready reranker for Polish medical texts and that zero-shot trained multilingual reranker models perform better than rerankers trained on a language-specific model fine-tuned on only a small number of INs.

REFERENCES

[1] I. Portoghese, M. Galletta, R. C. Coppola, G. Finco, and M. Campagna, "Burnout and workload among health care workers: the moderating role of job control," *Safety and Health at Work*, vol. 5, no. 3, pp. 152–157, 2014.

[2] P. Lewis, M. Ott, J. Du, and V. Stoyanov, "Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art," in *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, (Online), pp. 146–157, Association for Computational Linguistics, Nov. 2020.

[3] H.-C. Shin, Y. Zhang, E. Bakhturina, R. Puri, M. Patwary, M. Shoeybi, and R. Mani, "BioMegatron: Larger biomedical domain language model," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 4700–4706, Association for Computational Linguistics, Nov. 2020.

[4] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, pp. 1234–1240, 09 2019.

[5] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," 2020.

[6] R. Luo, L. Sun, Y. Xia, T. Qin, S. Zhang, H. Poon, and T.-Y. Liu, "Biogpt: generative pre-trained transformer for biomedical text generation and mining," *Briefings in Bioinformatics*, vol. 23, no. 6, 2022.

[7] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, N. Scharli, A. Chowdhery, P. Mansfield, B. A. y. Arcas, D. Webster, G. S. Corrado, Y. Matias, K. Chou, J. Gottweis, N. Tomasev, Y. Liu, A. Rajkomar, J. Barral, C. Semturs, A. Karthikesalingam, and V. Natarajan, "Large language models encode clinical knowledge," 2022.

[8] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Trans. Comput. Healthcare*, vol. 3, oct 2021.

[9] Q. Jin, Z. Yuan, G. Xiong, Q. Yu, H. Ying, C. Tan, M. Chen, S. Huang, X. Liu, and S. Yu, "Biomedical question answering: A survey of approaches and challenges," *ACM Comput. Surv.*, vol. 55, jan 2022.

[10] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, nov 2022. Just Accepted.

[11] Y. Xiao and W. Y. Wang, "On hallucination and predictive uncertainty in conditional language generation," *arXiv preprint arXiv:2103.15025*, 2021.

[12] N. Dziri, S. Milton, M. Yu, O. Zaiane, and S. Reddy, "On the origin of hallucinations in conversational models: Is it the datasets or the models?," *arXiv preprint arXiv:2204.07931*, 2022.

[13] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models," 2021.

[14] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira, "mmarco: A multilingual version of the ms marco passage ranking dataset," 2021.

[15] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, *et al.*, "Ms marco: A human generated machine reading comprehension dataset," *arXiv preprint arXiv:1611.09268*, 2016.

[16] A. Bondarenko, E. Shirshakova, M. Driker, M. Hagen, and P. Braslavski, "Misbeliefs and biases in health-related searches," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, (New York, NY, USA), p. 2894–2899, Association for Computing Machinery, 2021.

[17] D. Cohen, K. Du, B. Mitra, L. Mercurio, N. Rekabsaz, and C. Eickhoff, "Inconsistent ranking assumptions in medical search and their downstream consequences," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, (New York, NY, USA), p. 2572–2577, Association for Computing Machinery, 2022.

[18] N. Rekabsaz, O. Lesota, M. Schedl, J. Brassey, and C. Eickhoff, "Tripclick: The log files of a large health web search engine," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, (New York, NY, USA), p. 2507–2513, Association for Computing Machinery, 2021.

[19] J. Jimmy, G. Zuccon, J. Palotti, L. Goeuriot, and L. Kelly, "Overview of the clef 2018 consumer health search task," *International Conference of the Cross-Language Evaluation Forum for European Languages*, vol. 2125, 2018.

[20] K. Roberts, D. Demner-Fushman, E. M. Voorhees, W. R. Hersh, S. Bedrick, A. J. Lazar, S. Pant, and F. Meric-Bernstam, “Overview of the trec 2019 precision medicine track,” in *Proceedings of the Text Retrieval Conference (TREC)*, vol. 1250, NIH Public Access, 2019.

[21] M. Miłkowski and P. IFiS, “Morfologik,” *Web document: http://morfologik. blogspot. com*, 2007.

[22] Y. Wang, L. Wang, Y. Li, D. He, and T.-Y. Liu, “A theoretical analysis of ndcg type ranking measures,” in *Conference on learning theory*, pp. 25–54, PMLR, 2013.

[23] R. Mroczkowski, P. Rybak, A. Wróblewska, and I. Gawlik, “HerBERT: Efficiently pretrained transformer-based language model for Polish,” in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, (Kiyv, Ukraine), pp. 1–10, Association for Computational Linguistics, Apr. 2021.

[24] P. Rybak, R. Mroczkowski, J. Tracz, and I. Gawlik, “Klej: Comprehensive benchmark for polish language understanding,” *arXiv preprint arXiv:2005.00630*, 2020.

[25] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A massively multilingual pre-trained text-to-text transformer,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 483–498, Association for Computational Linguistics, June 2021.

[26] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2019.

[27] O. Khattab and M. Zaharia, “Colbert: Efficient and effective passage search via contextualized late interaction over bert,” in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 39–48, 2020.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 303–308
DOI: 10.15439/2023F3409
ISSN 2300-5963 ACSIS, Vol. 35

Developing Field Theory in Mizar

Christoph Schwarzweller
Institute of Informatics, Faculty of Mathematics, Physics and Informatics,
University of Gdańsk,
Wita Stwosza 57, 80-952 Gdańsk, Poland
christoph.schwarzweller@inf.ug.edu.pl

***Abstract*—As part of our ongoing project to prove Artin's solution of Hilbert's 17th problem in Mizar we are formalizing a great deal of basic field and Galois theory. In this paper we report on our formalization so far: we present basic mathematical structures and our Mizar definitions enriched with some main results. We also discuss some of our design decisions as well as subtleties – in particular connected with Mizar types.**

I. Introduction

INTERACTIVE theorem proving aims at developing systems to be used to formalize, that is both formulate and prove, mathematical theorems and theories in an accurate and comfortable way. The ultimate dream is a system containing all mathematical knowledge in which also mathematicians develop and prove new theorems. To come at least a little closer to this goal much effort has been spent building large repositories of computer-verified theorems such as the Coq library [7], the Isabelle2017 library [16], and the Mizar Mathematical Library [18]. A number of important mathematical theorems has been proven to illustrate the capability of interactive theorem proving, the most prominent examples being the proof of Kepler's conjecture in HOL Light [14], the Feit-Thompson theorem in Coq, and the Jordan curve theorem in Mizar.

Mizar [2], [12] is one of the pioneering systems for formalizing mathematics, after 50 years Mizar's proof checker still is actively developed and its library maintained and extended. One of the latest achievements in Mizar is the proof of the MRDP theorem solving Hilbert's 10th problem in the negative [20]. Another challenging problem is Hilbert's 17th problem: Given a multivariate polynomial that takes only non-negative values over the reals, can it be represented as a sum of squares of rational functions? Artin's positive solution [1] is a highlight in abstract algebra, introduced what today is known as formally real fields and initiated the development of real algebra.

Soon after starting the formalization of formally real fields it became clear that much more field theory is necessary than expected: not only field extensions and algebraically closed fields, but also basic Galois theory. Therefore we decided to formalize what usually appears in a one-semester graduate course on higher algebra [21], [9]. The main results of our formalization so far are

1) existence and uniqueness of splitting fields
2) existence and uniqueness of algebraic closures
3) simple extensions: characterization by intermediate fields, finite field extensions of characteristic 0 are simple
4) normal extensions: characterization by minimal polynomials, splitting fields, and fixing monomorphisms, counter example $\mathbb{Q}(\sqrt[3]{2})$
5) separable extensions: finite field extensions of characteristic 0 are separable, counter example $X^p - a$ for characteristic p, finite fields are perfect
6) formally and maximal formally real fields: formally real fields are exactly the ordered fields, sums of squares are exactly the total positive elements, real closed fields are maximal formally real

The complete formalization with entire proofs can be found in the Mizar Mathematical Library in the article series `FIELD_xx` and `REALALG_xx`.

To prove that maximal formally real fields are real closed we will formalize the fundamental theorem of Galois theory stating that for a (finite) Galois extension E of F the intermediate fields of E and F are in a one-to-one correspondence with the subgroups of E's Galois group. Note that a finite field extension E over F is Galois if and only if E is both separable and normal over F, so that for F with characteristic 0 a finite Galois extension E is also simple.

Related Work Formalizations of both field and Galois theory have been performed in different proof assistants: In Coq Galois theory has been developed to prove the Abel-Ruffini theorem [4] and also real closed fields can be found in [6]. Lean provides the theory up to the fundamental theorem of Galois theory [5]. General field theory also has been formalized in Isabelle - in particular the existence of algebraic closures of fields has been proved [8].

II. The Mizar System

Mizar is the name for both the proof checker and the formal language in which definitions and proofs are written. Mizar has often been described in the literature, for example in [19], [13], [10], [12] and [3]. We therefore here give only a very rough description of Mizar.

Mizar's logical basis is classical first-order logic, extended with so-called schemes. Schemes introduce free second-order variables enabling the definition of induction schemes among others. In addition, Mizar objects are typed, the types forming a hierarchy with the fundamental type `set`. The user can introduce new (sub)types describing mathematical objects such as groups, fields, vector spaces, or polynomials over rings or fields. The development of the Mizar Mathematical Library relies on Tarski-Grothendieck set theory – a variant of Zermelo-Fraenkel set theory using Tarski's axiom about arbitrarily large, strongly inaccessible cardinals which can be used to prove the axiom of choice. Mizar proofs are written in natural deduction style. The rules of the calculus are connected with corresponding (English) natural language phrases so that the Mizar language is close to the one used in mathematical textbooks, see [11] for an introduction to the Mizar language.

To define (algebraic) domains Mizar provides so-called structure modes fixing the domain's sets of elements and operations. So, for example[1]

```
definition
struct (addLoopStr,multLoopStr_0) doubleLoopStr
 (# carrier -> set,
    addF, multF -> BinOp of the carrier,
    OneF, ZeroF -> Element of the carrier #);
end;
```

defines the necessary backbone of rings and fields. Note that `doubleLoopStr` inherits from both `addLoopStr` and `multLoopStr_0`, that is it joins the operations of additive and multiplicative groups. Properties such as commutativity or the existence of inverse elements are described by attribute definitions for appropriate structures such as

```
definition
let L be addLoopStr;
attr L is right_zeroed means
  for a being Element of L holds a + 0.L = a;
end;
```

Here for elements `a` and `b` of (the `carrier`) of `R` `a+b` is a shortcut for `(the addF of R).(a,b)`. The type `Field` then is defined as a `doubleLoopStr` with the appropriate collection of attributes:

```
definition
mode Field is
  Abelian add-associative right_zeroed
  right_complementable associative commutative
  well-unital almost_left_invertible
  distributive non empty doubleLoopStr;
end;
```

As a consequence a Mizar object of type `Field` obtains all properties described by the defining attributes. We note here, that Mizar types have to be non-empty, so that each mode definition requires an existence proof.

Concrete algebraic domains are built by instantiation of structures. The field of rational numbers $\mathbb{Q}$, for example, is given by the set `RAT` of rational numbers and binary operations `addrat` and `multrat` defining addition and multiplication for elements of `RAT`. These are then glued together by the following

```
definition
func F_Rat -> Field equals
  doubleLoopStr(#RAT,addrat,multrat,1,0#);
end;
```

Note, that using the set `RAT` in defining the field `F_Rat` gives a particular representation of the rational numbers $\mathbb{Q}$ to be used when arguing about the rational numbers using the field `F_Rat`. Of course there are other fields, that is fields with a different set of elements, isomorphic to $\mathbb{Q}$. In fact any field of characteristic 0 contains a subfield isomorphic to $\mathbb{Q}$, so that every field of characteristic 0 can be considered as a field extension of $\mathbb{Q}$.

III. Field Extensions and Field Adjunctions

If F is a subfield of E then E is called a (field) extension of F. Note that this definition in particular means that the elements of F are a subset of the elements of E. Subfields (and subrings) already have been defined in Mizar, so we get

```
definition
let R,S be Ring;
attr S is R-extending means
  R is Subring of S;
end;

definition
let F be Field;
mode FieldExtension of F is F-extending Field;
end;
```

Note that in the definition instead of postulating that F is a subfield of E we demand that a ring R is a subring of another ring S. In this way our definition gets more flexible. For example, this allows to show that $\mathbb{Q}$ extends $\mathbb{Z}$. For fields, however, our definition is equivalent to the one from the literature given above, as stated by the following

```
theorem
for F,E being Field
holds E is FieldExtension of F iff
      F is Subfield of E;
```

There is an alternative equivalent definition stating that F embeds into E. In human mathematics it's obvious to switch between these two – ignoring usually the embedding. We decided to use the first option as it makes it easier to consider polynomials of F as polynomials of E (and also makes it more straightforward to define F-fixing morphisms needed later): In Mizar a polynomial of E must have coefficients of type E. Thus the second option would require to take care of the embedding φ: not p, but $\varphi(p)$ then is a polynomial of E.

However, even though an element $a \in F$ can naturally be considered as an element of E, this has to be made explicit in a typed system like Mizar: for a being an element of F and b an element of E the term $a+b$ is not defined as a and b must have the same type. This type can be element of E or element of F, if b turns out to be in F. In Mizar type casts are realized with the help of the `reconsider`-statement for changing types of objects, or one defines a functor for changing types, usually denoted by `@`. In both cases the result is independent of the field, that is for $a, b \in F$ we get

[1] Throughout the paper Mizar code is written in verbatim style

```
a + b = @(a,E) + @(b,E);

reconsider a1 = a, b1 = b as Element of E;
a + b = a1 + b1;
```

Both versions now allow to shift from one field to another. Note that this also works in towers of fields.

For $T \subseteq E$ the adjunction of F with T is the smallest extension of F containing T. Thus both $F(T)$ is an extension of F and E is an extension of $F(T)$. To "attach" both types to $F(T)$ we defined the type of `FAdj(F,T)` as subfield of E:

```
definition
let F be Field, E be FieldExtension of F;
let T be Subset of E;
func FAdj(F,T) -> Subfield of E means ...;
end;
```

Then the type of the field `E` can be easily changed into `FieldExtension of FAdj(F,T)`, if necessary. Because Mizar's typing mechanism allows to enrich types with further attributes the type of `FAdj(F,T)` can be "extended" with `F-extending`, hence then is `FieldExtension of F`. Note that this typing is necessary to prove $F(T_1 \cup T2) = F(T_1)(T_2)$ for $T_1, T_2 \subseteq E$, because then E must have type `FieldExtension of F(T`$_1$`)` on the right-hand side – in contrast to `FieldExtension of F` on the left-hand-side.

IV. Splitting Fields

A splitting field of a polynomial $p \in F[X]$ is an extension E in which p splits into linear factors and is generated by p's roots, e.g. $E = F(\alpha_1, \ldots \alpha_n)$ where the α_i are the roots of p – or equivalently a smallest field extension of F in which p splits:

```
definition
let F be Field;
let p be non constant Polynomial of F;
mode SplittingField of p
                        -> FieldExtension of F means
  p splits_in it &
  for E being FieldExtension of F
  st p splits_in E & E is Subfield of it
  holds E == it;
end;
```

Note again that in Mizar a mode definition requires an existence (but no uniqueness) proof, because the introduced type – here `Splittingfield of p` – is not allowed to be empty. Our proof follows [21] and does not use algebraic closures: Iterating Kronecker's construction [23] ensures that there exists an extension of F in which p splits, so one easily shows that there is a smallest one – of course this then is the extension of F generated by p's roots. Consequently, that a splitting field of p is generated by the roots of p now follows as a theorem.

```
theorem
for F being Field
for p being non constant Polynomial of F
for E being SplittingField of p
holds E == FAdj(F,Roots(E,p));
```

To prove uniqueness of splitting fields we introduced the notion of being isomorphic over a field F, e.g. there is an isomorphism that fixes the elements of F. Note that such an isomorphism also fixes polynomials $p \in F[X]$. We then lifted isomorphisms from $F_1 \longrightarrow F_2$ to $F_1(\{a\}) \longrightarrow F_2(\{b\})$ where a and b are algebraic elements of F_1 and F_2 respectively. Because splitting fields are generated by roots of a polynomial, hence by algebraic elements, then follows

```
theorem
for F being Field
for p being non constant Polynomial of F
for E1,E2 being SplittingField of p
holds E1,E2 are_isomorphic_over F;
```

so a splitting field of a non-constant polynomial is unique up to isomorphism.

V. Algebraic Closures

An algebraic closure A of F is an extension of F which is both algebraic closed and algebraic over F, that is every non-constant polynomial of A has a root and every element $a \in A$ is the root of a non-zero polynomial of F.

Our proof follows Artin's classical one as presented by Lang in [17]: Kronecker's construction is applied to each polynomial $p \in F[X]\backslash F$ simultaneously to get an extension E of F in which every non-constant polynomial $p \in F[X]$ has a root in E. For that we need the polynomial ring $F[X_1, X_2, ...]$ with infinitely many variables, one for each polynomial $p \in F[X]\backslash F$. The sought-after field extension E then is (isomorphic to) $F[X_1, X_2, ...]/I$, where I is a maximal ideal generated by all non-constant polynomials $p \in F[X]$. Note that to show that I exists Zorn's lemma is necessary.

Iterating this construction gives an infinite sequence of fields, whose union defines an extension A of F, in which every non-constant polynomial $p \in A[X]$ has a root. The field of algebraic elements of A then is an algebraic closure of F. With this existence proof we can define

```
definition
let F be Field;
mode AlgebraicClosure of F
                      -> FieldExtension of F means
  it is F-algebraic &
  it is algebraic-closed;
end;
```

To prove uniqueness of algebraic closures again the technique of lifting morphisms is applied: a monomorphism $F \longrightarrow A$, where A is an algebraic closure of F can be extended to a monomorphism $E \longrightarrow A$, where E is any algebraic extension of F. In case that E is algebraically closed this monomorphism is an isomorphism.

```
theorem
for F being Field
for A1,A2 being AlgebraicClosure of F
holds A1,A2 are_isomorphic_over F;
```

Note that the existence of the lifted monomorphism again relies on Zorn's lemma.

VI. Simple Extensions

An extension E of a field F is simple, if E is generated over F by a single element $a \in E$, e.g. $E = F(\{a\})$. The element a then is a primitive element.

```
definition
let F be Field, E be FieldExtension of F;
attr E is F-simple means
  ex a being Element of E st E == FAdj(F,{a});
end;
```

For infinite fields F we proved that a finite extension E of F is simple if and only if the number of intermediate fields between E und F is finite. In Mizar the intermediate fields of given fields E and F can be defined as a functor giving the appropriate set: because the elements of such a field must be a subset of the elements of E, one can pick up the subsets of the elements of E which constitute a field using Mizar's `replacement`-scheme.

```
theorem
for F being infinite Field
for E being F-finite FieldExtension of F
holds E is F-simple iff
        IntermediateFields(E,F) is finite;
```

The theorem holds for finite fields also. The proof, however, follows easily from group theory, in particular every finite extension of a finite field is simple.

For fields with characteristic zero we also proved that a linear combination of a and b generates $F(a, b)$ – in fact in doing so we already showed that in fields with characteristic 0 irreducible polynomials are separable.

```
theorem
for F being 0-characteristic Field
for E being FieldExtension of F
for a,b being F-algebraic Element of E
ex x being Element of F
st FAdj(F,{a,b}) = FAdj(F,{a+@(x,E)*b});
```

Note that to take the element x from F we again have to shift x into E using the functor `@`.

VII. Normal Extensions

An extension E of F is normal, if every polynomial over F that has a root in E – or equivalently every minimal polynomial – already splits in E. There is a number of equivalent characterizations of (finite) normal extensions (usually shown in a ring proof), for example, that normal extensions are given by splitting fields of polynomial $p \in F[X]$:

```
theorem
for F being Field,
    E being F-finite FieldExtension of F
holds E is F-normal iff
   ex p being non constant Polynomial of F
   st E is SplittingField of p;
```

Note that one direction of this theorem can be be automated by enriching the type `SplittingField of p` with the attribute `F-normal`. This is done using Mizar's cluster mechanism:

```
registration
let F be Field;
let p be non constant Polynomial of F;
cluster -> F-normal for SplittingField of p;
end;
```

Then the type `SplittingField of p` is extended to `F-normal FieldExtension of F` instead of only `FieldExtension of F`, so all theorems about normal extensions can be automatically applied.

Another important characterization deals with fixing morphisms. It states that for (finite) normal extensions E an F-fixing monomorphism $h : E \longrightarrow K$ into a larger field K actually maps to E only, and therefore is an isomorphism. Note here, that an extension E of F is finite if and only if there exist (algebraic) elements $a_1, \ldots a_n \in E$ such that $E = F(\{a_1, \ldots a_n\})$.

```
theorem
for F being Field
for E being F-finite FieldExtension of F
holds E is F-normal iff
   for K being FieldExtension of E
   for h being F-fixing Monomorphism of E,K
   holds h is Automorphism of E;
```

The proof turned out to be technical because one needs to show $h(E) = h(F(\{a_1, \ldots, a_n\})) \subseteq F(\{h(a_1), \ldots, h(a_n)\})$. In human mathematics this is almost obvious just because every element $a \in F(\{a_1, \ldots, a_n\})$ is given by $p(a_1, \ldots, a_n)$ for some polynomial $p \in F[X_1, \ldots X_n]$. The formal proof in Mizar is by induction on the degree of multivariate polynomials and hence needs to reduce the degree of a multivariate polynomial in order to apply the induction hypothesis.

VIII. Separable Extensions

A polynomial $p \in F[X]$ is separable, if p has no multiple roots in the (any) splitting field of p. This is equivalent to p being coprime with its formal derivative. An algebraic extension E of F is separable, if for all $a \in E$ the minimal polynomial μ_a is separable.

```
definition
let F be Field
let p be non constant
     Element of the carrier of Polynom-Ring F;
attr p is separable means
  for a being
       Element of the SplittingField of p
  st a is_a_root_of p,(the SplittingField of p)
  holds multiplicity(p,a) = 1;
end;
```

Note the use of the `the`-operator in the following definition which nicely puts into mind the fact that a splitting field is unique up to isomorphism. This, unfortunately, is not expressed by the definition as `the` just takes an arbitrary element of the non-empty type `SplittingField of p`. All the obvious properties about polynomials over isomorphic fields nevertheless have to be proved. In particular the fact that separability indeed is independent of the splitting field is established not before the following

```
theorem
for F being Field,
    p being non constant Polynomial of F
holds p is separable iff
   ex E being FieldExtension of F
   st p splits_in E &
      for a being Element of E
                 holds multiplicity(p,a) <= 1;
```

In fields with characteristic 0 every irreducible polynomial p is separable (such fields are called perfect), because p must be square-free to be relatively prime with its formal derivation. In fields with prime characteristic p, however, the polynomial $X^p - a$ is reducible only if a has a p-th root and then equals $(X - a)^p$. In the other case $X^p - a$ is irreducible and because $\sqrt[p]{a}$ is a p-fold root of $X^p - a = (X - a)^p$ in its splitting field we get

```
theorem
for p being Prime
for F being p-characteristic Field
for a being Element of F
st not a in F|^p
holds X^(p,a) is irreducible inseparable;
```

where `F|^p` denotes the subfield F^p of all p-th roots in F. Indeed, $F^p = F$ if and only all irreducible polynomials of F are separable, which we applied to finally prove that every finite field is perfect. On the other hand the field $F_p(X)$ of rational functions over a field F_p with characteristic $p \neq 0$ is not perfect.

IX. Formally Real Fields

Finally we return to our original motivation of formalizing formally real fields and Artin's solution of Hilbert's 17th problem. A field F is formally real, if -1 is no sum of squares. In this – and only this – case F can be ordered. Here, orders are usually defined as positive cones, the set of positive elements [22]. Note that formally real fields have characteristic 0. A first main result from [1] we proved states, that the elements of F that can be described as sums of squares are exactly the total positive ones:

```
theorem
for F being formally_real Field
for a being Element of F
holds a in Sums_of_squares_of F iff
      for P being Ordering of F holds a in P;
```

So, to solve Hilbert's 17th problem it's crucial to identify the total positive elements of the real numbers. In general fields allow for different orderings. Maximal formally real fields F – in the sense that there is no proper extension of F which again is formally real – however, have only one ordering, the set of squares `SQ F`:

```
definition
let F be Field;
attr F is maximal_formally_real means
  F is formally_real &
  for E being F-algebraic FieldExtension of F
  st E is formally_real holds E == F;
end;
```

```
theorem
for F being maximal_formally_real Field holds
SQ F is Ordering of F &
for P being Ordering of F holds P = SQ F;
```

We also proved that in maximal formally real fields every polynomial of odd degree has a root and that the field of real numbers is maximal formally real. Maximal formally real fields F then are characterized as real closed fields: a field F is real closed if the splitting field of the polynomial $X^2 + 1 \in F[X]$ – e.g. the field $F(i)$ – is an algebraic closure of F.

```
definition
let F be Field;
attr F is real_closed means
  not -1.F in SQ F &
  the SplittingField of X^2+(1.F)
                       is algebraic-closed;
end;
```

Note that this definition does not make use of orderings. So far we proved that real closed fields are maximal formally real. Maximality easily follows from the fact that -1 is a square in $F(i)$, hence the main part here is to show that real closed fields F are formally real, e.g. that the squares of F form an ordering. This requires to show that for $a, b \in F$ we have that $a^2 + b^2$ again is a square. This is shown by considering the polynomial $p = (X^2 - a)^2 + b^2$ in $F[X]$. p has no roots in F, but is reducible, because $F(i)$ is algebraic closed. So we get $p = p_1 \cdot p_2$ for irreducible quadratic polynomials p_1, p_2. Note that p splits in $F(i)$ by assumption, so the roots of p are $\pm\sqrt{a \pm b \cdot i}$ giving

```
theorem
for F being real_closed Field
for a,b being non zero Element of F
for p being Polynomial of F
  st p = Subst(X^2+b^2,X^2-a)
for i being a_Root of X^2+(1.F)
for ai,bi,w1,w2 being Element of
                the SplittingField of X^2+(1.F)
  st ai = a & bi = b &
     w1^2 = ai + i * bi & w2^2 = ai - i * bi
holds Roots(the SplittingField of X^2+(1.F),p)
    = { w1, -w1, w2, -w2 };
```

Of course also p_1 splits in $F(i)$, so $p_1 = (X - \alpha) \cdot (X - \beta) \in F(i)[X]$ must take two of these roots. But then $\alpha \cdot \beta =: v \in F$ and also $\alpha \cdot \beta = \sqrt{a^2 + b^2}$, so $a^2 + b^2 = (\alpha \cdot \beta)^2 = v^2$.

The other direction – that maximal formally real fields are real closed – will be proved by showing that for fields F, in which both the set of squares is an ordering and every polynomial of odd degree has a root, the extension $F(i)$ is algebraic closed. This is done by starting with a splitting field E of an arbitrary non-constant polynomial p. Then E is a Galois extension of F, because it's finite, normal and separable. Applying the fundamental theorem of Galois theory (and Sylow's theorems about finite groups) one can show that in fact $E = F(i)$, so p splits in $F(i)$ [22]. To formalize this we first need to further develop Galois theory in Mizar.

X. Conclusion and Further Work

We have presented the beginnings of formalizing field and Galois theory in Mizar. Three main lessons of our Mizar formalization so far we consider worth mentioning: Mathematical types such as rings, fields, vector spaces, topological spaces, and so forth are usually considered helpful in proof assistants as they automate applying theorems. However when it comes to field extensions where objects such as elements, subsets, and polynomials are shifted between different fields, it's often necessary to explicitly cast types in order to apply definitions or theorems. So, for example, that for a polynomial $p \in F[X]$ the degree of p is the same when considering p as a polynomial in an extension E of F is not obvious for Mizar: the type `Polynomial of F` has to be changed into `Polynomial of E`, which is expressed by the following

```
theorem
for F being Field, E being FieldExtension of F
for p being Polynomial of F,
    q being Polynomial of E
st p = q holds deg p = deg q;
```

Secondly, dealing with polynomials tends to cause much more work than expected: many properties considered obvious by human mathematics require a formalization resulting in a number of technical lemmas. For example, for $p_1, p_2 \in F[X]$ and $a \in F$ because of $(p1+p2)(a) = p_1(a)+p_2(a)$ obviously follows

$$\sum_{i=1}^{n} p_i(a) \;=\; (\sum_{i=1}^{n} p_i)(a)$$

by a straightforward induction. To prove this in Mizar the n polynomials have to be put together in a finite sequence `f`, so that $p = \sum_{i=1}^{n} p_i =$ `Sum f`. The second finite sequence `g` contains $p_i(a)$ for $i = 1, \ldots n$, hence `Sum g` $= \sum_{i=1}^{n} p_i(a)$ in the following

```
theorem
for F being Field
for f being FinSequence of Polynom-Ring F
for p being Polynomial of F st p = Sum f
for a being Element of F,
    g being FinSequence of F
st len g = len f &
   for i being Element of dom f,
   q being Polynomial of F
     st q = f.i holds g.i = eval(q,a)
holds eval(p,a) = Sum g;
```

Thirdly, the omnipresent "uniqueness up to isomorphism" also increases the formalization's length: each property carrying over to isomorphic fields has to be explicitly stated and proved. Of course, for example, because two splitting fields E_1 and E_2 of a polynomial $p \in F[X]$ are isomorphic the multiplicity of a root of p in E_1 and E_2 is the same. This, however, has to be stated and proved as a theorem. Another example concerns ordered fields: it's obvious that a field isomorphic to an ordered field is also ordered, but again this has to be explicitly proved.

The next step of our formalization will be the combination of normal and separable extensions to establish Galois extensions and their corresponding Galois groups. Group theory in Mizar is well developed, in particular Sylow's theorems can be found in the Mizar Mathematical Library. Galois theory will enable to further extend the formalization of real algebra. In particular the fundamental theorem of Galois theory will allow to conclude the proof that maximal formally real fields are real closed as a main step towards Artin's solution of Hilbert's 17th problem.

References

[1] E. Artin, *Über die Zerlegung definiter Funktionen in Quadrate*; Abh. Math. Sem. Univ. Hamburg 5(1), pp. 100–115, 1927.

[2] G. Bancerek et.al., *Mizar: State-of-the-art and Beyond.* in: M. Kerber et.al. (eds.), Proceedings of the 2015 International Conference on Intelligent Computer Mathematics, Lecture Notes in Computer Science 9150, 261–279, 2015. http://dx.doi.org/10.1007/978-3-319-20615-8_17

[3] G. Bancerek, C. Bylinski, A. Grabowski, A. Kornilowicz, R. Matuszewski, A. Naumowic, and K. Pak, *The Role of the Mizar Mathematical Library for Interactive Proof Development in Mizar*; Journal of Automated Reasoning, vol. 61(1-4), pp. 9–32, 2018.

[4] S. Bernard, C. Cohen, and A. Mahboubi and P. Strub, *Unsolvability of the Quintic Formalized in Dependent Type Theory*, available at https://hal.inria.fr/hal-03136002, 2021.

[5] T. Browning and P. Lutz, *Formalizing Galois Theory*; Experimental Mathematics, vol. 31(2), pp. 413–424 2022.

[6] C. Cohen, *Construction of Real Algebraic Numbers in Coq*, in: ITP - 3rd International Conference on Interactive Theorem Proving, pp. 67–82, 2012.

[7] *The Coq Proof Assistant.* available at www.coq.inria.fr.

[8] P.E. de Vilhena and L.C. Paulson, *Algebraically Closed Fields in Isabelle/HOL* in: Automated Reasoning. IJCAR 2020, Lecture Notes in Computer Science 12167, pp. 57–64, 2020.

[9] A. Gathmann, *Einführung in die Algebra*; Lecture notes, University of Kaiserslautern, Germany, 2010.

[10] A. Grabowski, R. Coghetto *Topological structures as a tool for formal modelling of rough sets*; Position Papers of the 2017 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 12, pp. 11-18, 2017.

[11] A. Grabowski, A. Korniłowicz, and A. Naumowicz, *Mizar in a Nutshell.* Journal of Formalized Reasoning 3(2), 153–245, 2010. https://doi.org/10.6092/issn.1972-5787/1980

[12] A. Grabowski, A. Korniłowicz, and C. Schwarzweller, *On Algebraic Hierarchies in Mathematical Repository of Mizar.* in: Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki (eds). ACSIS, Vol. 8, 363–371, 2016. http://dx.doi.org/10.15439/2016F520

[13] A. Grabowski, A. Korniłowicz, and A. Naumowicz, *Four Decades of Mizar.* Journal of Automated Reasoning, vol 55(3), 191–198, 2015. http://dx.doi.org/10.1007/s10817-015-9345-1

[14] J. Harrison, The HOL Light Theorem Prover. available at www.cl.cam.ac.uk/~jrh13/hol-light.

[15] *The HOL Interactive Theorem Prover.* available at hol-theorem-prover.org.

[16] *Isabelle.* available at isabelle.in.tum.de.

[17] S. Lang, *Algebra, 3rd edition*, Springer Verlag, 2002.

[18] *Mizar Home Page.* available at www.mizar.org.

[19] A. Naumowicz and A. Korniłowicz, *A Brief Overview of Mizar.* in: Theorem Proving in Higher Order Logics 2009, S. Berghofer, T. Nipkow, C. Urban, M. Wenzel (eds.), Lecture Notes in Computer Science, 5674, 67–72, *Springer Verlag*, 2009.

[20] K. Pak, *Formalization of the MRDP-Theorem in the Mizar System*, Formalized Mathematics, vol. 27(2), pp. 209–222, 2019.

[21] K. Radbruch, *Algebra I*; Lecture notes, University of Kaiserslautern, Germany, 1990.

[22] K. Radbruch, *Geordnete Körper*; Lecture notes, University of Kaiserslautern, Germany, 1991.

[23] C. Schwarzweller *Representation Matters: An Unexpected Property of Polynomial Rings and its Consequences for Formalizing Abstract Field Theory*; in: M. Ganzha, L. Maciaszek, M. Paprzycki (eds.), Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, ACSIS, vol. 15, pp. 67-72, 2018.

DOI: 10.15439/2023F1377
ISSN 2300-5963 ACSIS, Vol. 35

Clusterization methods for multi-variant e-commerce interfaces

Adam Wasilewski
0000-0002-1653-5005
Wroclaw University of Science and Technology
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Fast White Cat S.A., Poland
Email: adam.wasilewski@pwr.edu.pl

***Abstract*—E-commerce is a very popular method that lets consumers to purchase goods and services. The ability to purchase items online has increased the need for effective recommendation systems. Such recommendations usually relete to products in which the customer may be interested. However, there are wider opportunities to tailor e-commerce to individual customer needs and behaviour. In this paper, the architecture of the e-commerce platform (named AIM^2), which allows providing a dedicated interface to selected user groups, is discussed. A key component of the platform is the module responsible for dividing customers into groups, using selected clustering methods. Each of the implemented methods can be parameterised to adapt the customer segmentation to a given e-commerce business owner's requirements. This article describes the results of an analysis of the impact of selected methods and parameters on clustering results. Moreover, it identifies key metrics that should be considered when selecting clustering conditions during the implementation of the platform. Finally, the main results of the pilot implementation of AIM^2 are presented to assess the effectiveness of the multi-variant user interface.**

I. Introduction

ONE major drawback of existing e-commerce systems is that they display little ability to take into account differences in the users' knowledge, style, and preferences. Meanwhile, users of the systems are different, and the interfaces served to them could be different. The concept of AUI (Adaptive User Interfaces) is increasingly frequent in implementation in the modern IT systems, but this trend is less visible in e-commerce. AIM^2 platform, described in the paper, is an example of the practical implementation of the AUI concept in e-commerce systems. Its aim is to serve dedicated e-commerce user interfaces based on user groups defined by clustering using AI methods, monitoring the results and optimizing the solution.

Personalized web-based system user interfaces can be defined as systems that can automatically adjust their presentation, content, and structure based on the user's characteristics, needs, or preferences [12]. Such solutions can improve the usability and effectiveness of the interface by its adaptation to the user's behaviour. It may also reduce the user's cognitive load, which in turn reduces the user's tiredness and the number of committed errors. Better UX may increase the user's satisfaction and motivation. Design and implementation of AUI may be a complex task, requiring usage of interdisciplinary solutions [15]. Such system may be based on multi-agent infrastructure [3] but nowadays, great opportunities are offered by AI methods as well.

Among the various existing approaches to creating self-adaptive user interfaces is the use of AI-based clustering to divide customers into groups and serve these groups with a dedicated interface. AI clusterization methods leverage advanced algorithms to discover hidden patterns, structures, or relationships within datasets. Information about users' activities, events, purchases and other factors that should potentially affect the interface can be used to group e-commerce users. If designated user groups are served with an interface tailored to their behaviour, an increase in the e-commerce performance can be expected, which should ultimately have a positive impact on profitability in business terms.

In Section 2 previous works on using clusterizaton methods in e-commerce are described. Section 3 briefly shows the architecture of AIM^2 platform and research methodology. Experiments related to clusterization methods and their parameters are detailed and discussed in Section 4. Section 5 concludes the work.

II. Related Work

Clusterization methods are an effective way to group similar items together and provide personalized recommendations. However, the choice of method depends on the specific requirements and limitations of the e-commerce interface. Commonly used clusterization methods are:

- Hierarchical methods, which create a hierarchical structure of clusters by iteratively merging or splitting clusters, e.g. agglomerative clustering and divisive clustering;
- Partitioning methods, which divide the dataset into a predetermined number of clusters, e.g. K-means, K-medoids, and Fuzzy C-means clustering;
- Density-based methods, which focus on identifying regions of high-density data points and separating them from sparse regions, e.g. DBSCAN (Density-Based Spatial Clustering of Applications with Noise);

Project "Self-adaptation of the online store interface for the customer requirements and behaviour" co-funded by the National Centre for Research and Development under the Sub-Action 1.1.1 of the Operational Programme Intelligent Development 2014-2020.

- Spectral, which utilizes the concept of spectral graph theory, which relates the eigenvectors and eigenvalues of a similarity matrix to the underlying structure of the data;
- Model-based methods, which assume that the data points are generated from a statistical model or distribution, e.g. Gaussian Mixture Models (GMM) and Expectation-Maximization (EM) algorithm.

Almost all of the methods mentioned were verified for their application for e-commerce solutions.

Hierarchical clustering in e-commerce applications is discussed in [18]. It is worth noting that the standard algorithm for hierarchical agglomerative clustering (HAC) can be slow even for medium data sets [1]. Research related to the record linkage system for e-commerce products was also conducted using this approach [9]. HAC algorithm used for customer segmentation was described in [10] and the findings could also be used to segment e-commerce customers.

K-means algorithm can be used for segmenting e-commerce customers to obtain groups of customers with different characteristics [5]. In the clustering customer purchase data can be taken into account [17]. According to researchers K-means clustering is quite efficient algorithm. However, its computational difficulty is influenced by the size of the dataset, the number of clusters, and the initialization of the cluster centroids. K-medoids (uses the medoid instead of the mean) was used for e-commerce customer segmentation [21]. Classification method of e-commerce user behavior based on Fuzzy C-Means Clustering was proposed to improve the clustering analysis effect of e-commerce user behavior in [20].

DBSCAN [8] was used to process uneven density data based on information from e-commerces [19]. DBSCAN algorithm seems to be more advantageous when compared to other algorithms because doesn't need one to specify the number of clusters within the knowledge a priori [13].

Some researchers have also compared the effectiveness of different clustering algorithms used in e-commerce solutions. A comparison of the performance of the K-means and DBSCAN (Density-based spatial clustering of applications with noise) algorithms in e-commerce applications indicated slightly better results obtained by the second method [2]. In another study, an accuracy comparison indicated an advantage for DBSCAN over K-means [7]. DBSCAN seems to correspond more to human intuitions of clustering, rather than distance from a central clustering point (e.g. K-means) [13].

In some e-commerce applications spectral clustering can be an effective clustering method [6]. There is also the possibility of using spectral clustering in conjunction with K-means clustering [22].

There are relatively few publications on the application of the GMM method [23] to e-commerce-related analysis. Some possibilities for applying this method in product recommendation are indicated [11]. According to [16] the K-means method has lower computational requirements, and could potentially yield clustering results similar to those of the GMM method.

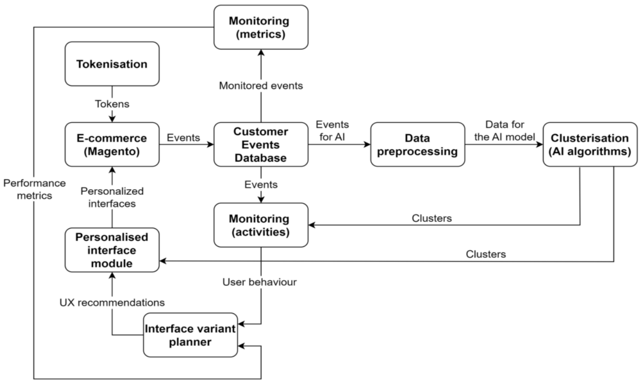


Fig. 1. AIM^2 Business Architecture

III. AIM^2 ARCHITECTURE

The AIM^2 business architecture contains the core modules of the system and key integration interfaces (Fig. 1). It includes:

- Tokenization - responsible for ensuring the anonymity of the data collected by the platform;
- E-commerce - Adobe Magento-based e-shop with implementation of PWA (Progressive Web App) technology;
- Customer Events Database - e-commerce customer behaviour data storage;
- Preprocessing - the module to prepare data to analyse and to decrease the time required to generate clusters;
- Clusterisation - analyses the information collected about the behaviour of e-commerce customers and to divide them into groups characterised by similar shop use;
- Monitoring - identifies user patterns that can be used to design variants of dedicated interfaces and verifies the performance of the interface variants;
- Interface management - supports variants definitions shown to selected users;
- Self-adaptation algorithm - automatically implement micro changes to the interface and accept or reject them depending on the impact on e-commerce performance metrics.

The interface adaptation process starts with initial clustering of e-commerce customers, using learning data from a possible long period. One of the clusterization effects is a set of customer groups that are heterogeneous based on certain characteristics. The AIM^2 platform has four clustering methods implemented: agglomerative clustering, the K-means method, DBSCAN and spectral clustering. Some of them can be applied with different parameter values, e.g.:

- model (type=string, default='kmeans') algorithm used - k-means ('kmeans'), agglomerative clustering ('agg'), DBSCAN ('dbscan') and spectral ('spectral')
- pca (type=float, default=0.999) - the amount of data variance retained is greater than the percentage specified in the parameter

- init (type=string, default='k-means++') - method for initialising cluster centres for the k-means algorithm:
  - 'k-means++' samples the dataset and counts k-means on it [4],
  - 'random' randomises the centres.
- max_iter (type=integer, default=300) - maximum number of iterations in the k-means algorithm

The choice of clustering module parameter settings affects the allocation of customers into clusters and, consequently, the interface variants that will be provided to them. For this reason, it is important to properly tune this module and match the requirements of a particular e-commerce. The research conducted was aimed at verifying the outflow of different values of clustering parameters on its effects. The quality of clustering can be considered in two aspects - the objective one, resulting from classical methods of cluster evaluation (as: Silhouette score, entropy, Calinski-Harabasz score and Davies-Bouldin score) and contextual, resulting from the requirements for clusters intended to be the basis for providing different variants of the e-commerce customer interface. During the research, experiments were carried out on a dataset covering a period of 4 months (548.922 e-commerce user sessions) to verify the impact of the choice of clustering method and the parameters (pca, init, max_iter) on the distribution of customers in the clusters. From a business point of view, clusters should be of similar size, since it makes most sense to prepare a dedicated interface variant. There should not be too many clusters (groups of customers), because the creation of an interface variant is a non-zero cost, so from a business point of view, it is preferable to use methods that generate clusters with a size of no less than X% of the population (X can be treated as a parameter and vary depending on the specific e-commerce, for the purposes of the study X=5 was assumed).

IV. Research results

The research was carried out in two stages. In the first part, clustering was performed with all four methods available on the AIM^2 platform (Tab. I).

The following were analysed: clustering time, number of clusters, size of the largest and smallest cluster. From the point of view of the adopted business requirements, the clustering time (without pre-processing) should be as short as possible, the number of clusters not exceeding 10 and the size of the smallest cluster not less than 5% of the total number of customers (91.819 e-commerce users were clustered in the experiment).

K-means and DBSCAN clustering were found to be significantly more time-efficient than the other methods. Nevertheless, the duration of the longest clustering (using the spectral method) would also be acceptable, as it would be feasible in the time allowed for cyclic updating of the cluster composition in AIM^2.

The number of clusters obtained, in particular as many as 282 clusters generated using the DBSCAN method, is noteworthy. This is well beyond the upper limit of the number of clusters taken as a business requirement. The lack of predictability of the number of clusters and the inability to reduce the number of clusters make the DBSCAN method practically useless from the point of view of the self-adaptation mechanism of the e-commerce interface implemented in AIM^2. The other three methods had the number of clusters set to 10. This number, from a business point of view, seems to be the maximum number of interface variants that should be served to e-commerce customers.

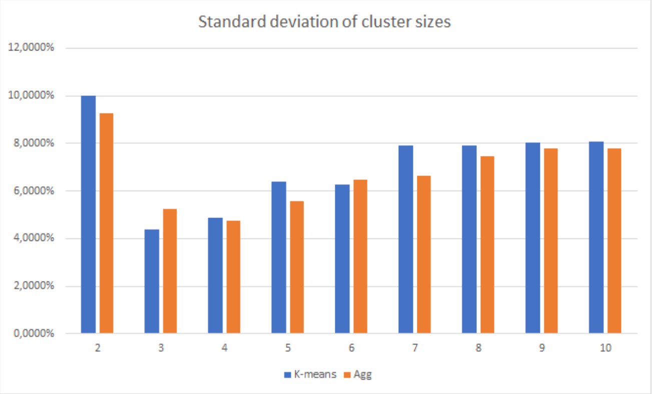


Fig. 2. AIM^2 Standard deviation of cluster sizes

The next characteristic analysed was the size of the largest and smallest cluster. In order to prepare a reasonable dedicated interface variant, the number of customers to whom it will be delivered should be sufficiently large. In this aspect, the DBSCAN method was again found to be useless, as the smallest clusters contained only one client. It turned out that the problem with the number of clusters also occurred with the spectral method. In this case, the largest cluster covered more than 93% of all customers, which calls into question the sense of preparing interface variants for the remaining clusters.

In summary, the results obtained from the experiment allowed to select two of the most promising approaches - agglomerative clustering and K-means method. A detailed analysis was carried out for them, taking into account the different clustering parameters. Further research looked at the impact of clustering parameters on the results obtained. The primary parameter influencing the results was the clustering method, with the additional ones being the fixed number of clusters, pca, init and max_iter.

Firstly, it was checked how the cluster size changes, depending on a fixed number of groups (from 2 to 10), with constant values for the other parameters (pca=0.999, init=k++, max_iter=300).

For the data set analysed, it turned out that the smallest standard deviation of cluster counts was calculated in both methods for 3–4 clusters (Fig. 2). Additionally, it was noted that the agglomerative clustering method yielded a lower standard deviation of cluster sizes in most cases. This means that clusters with less variation can be expected. From the point of

TABLE I
CLUSTERING EFFECTS OF THE METHODS AVAILABLE IN AIM^2

Clustering method	**Clustering time [mins]**	**Number of clusters**	**Largest cluster**	**Smallest cluster**
K-means	4	10* (fixed)	29,7389%	3.4895%
Agglomerative	122	10* (fixed)	29,7302%	2.9438%
DBSCAN	15	282	29.7302%	1 customer
Spectral	188	10* (fixed)	93.6353%	0.1993%

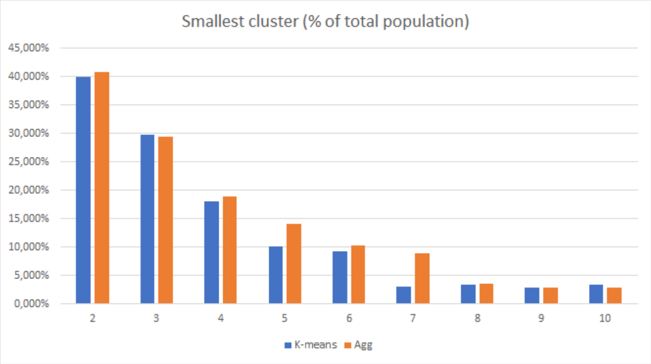


Fig. 3. Smallest clusters

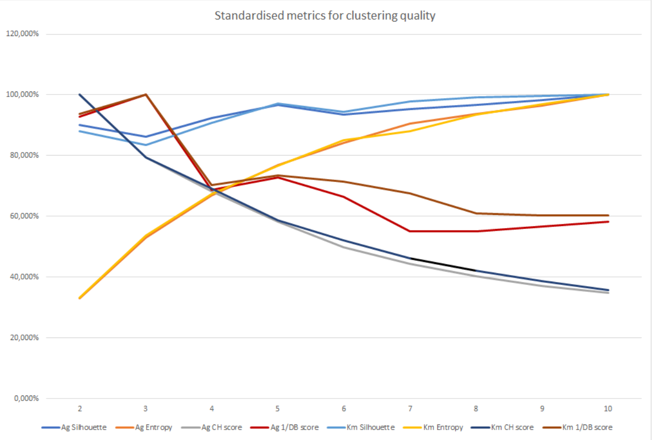


Fig. 4. Standardised metrics for clustering quality

view of variants of the e-commerce interface, agglomerative clustering therefore appears to be slightly better in this aspect. Similar conclusions can be drawn from an analysis of the size of the smallest clusters (Fig. 3). Assuming a cost-efficiency threshold for the development of an interface variant of 5% of the customer population size, it appears that results for more than 6 clusters with the K-means method and for more than 7 clusters with agglomerative clustering should be rejected. The difference does not seem large, but can be significant when choosing the optimal clustering parameters for the delivery of a dedicated e-commerce interface.

When selecting clustering parameters due to business requirements, typical clustering quality indicators cannot be overlooked. Cluster quality analysis for agglomerative clustering and the K-means method, with different numbers of clusters, was carried out for the indices: Silhouette score, entropy, Calinski-Harabasz (CH) score and Davies-Bouldin (DB) score. Given that the first three indicators should be as high as possible and DB as low as possible, it was assumed for simplicity that the analysis would include DB' metric:

$$DB' = \frac{1}{DB} \tag{1}$$

Under this assumption, the goal of clustering optimization is to maximize the value of all quality indicators. For the purpose of the analysis, the values of the indicators were standardized, assuming that the highest value is 100% (Fig. 4).

The results show that there is a very high correlation between the values of quality indicators for both clustering methods. In addition, it can be seen that the values of two indicators increase as the number of clusters increases, and the values of two indicators decrease. The intersection point for the three indicators is a value of 4 clusters. This value can be considered the minimum number of clusters worth generating for a multi-variant e-commerce interface.

The study showed that the other clustering parameters available in AIM^2 affect the clustering results to a lesser extent.

Increasing the value of max_iter (even 10 times) had no effect on the resulting clusters, so the default value (300) is sufficient.

On the other hand, the option of using the pca parameter to reduce cluster generation time with agglomerative clustering may be interesting. After setting pca=0.95, agglomerative clustering took 13 minutes, almost 10 times faster than calculations with pca=0.99. At the same time, the composition of the clusters has practically not changed, so the reduction in the quality of the input data for clustering should be acceptable.

Slightly more influential is the decision to initialize cluster centers. In the case of random selection of cluster centers, the clusters differ from the selection of the 'k-means++' option. However, the changes applied only to individual clusters, and the standard deviation of cluster sizes was larger each time for the random selection of cluster centers.

Considering the results obtained, for the data set used in the experiment, it could be recommended to use agglomerative clustering with 4 clusters [Fig. 5, 6], due to the smallest variance in cluster size. The number of clusters could be increased (up to a maximum of 7 clusters) if business considerations required it. If cluster calculation time needs to be reduced, the pca parameter could be further changed to 0.95.

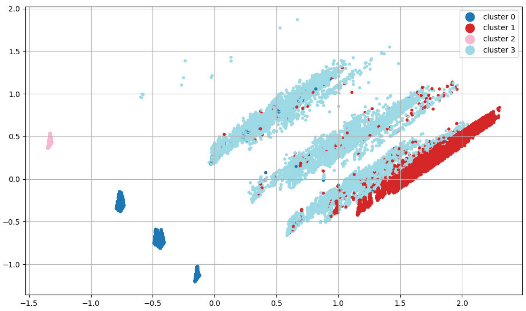


Fig. 5. Scatter plot along the two most significant dimensions produced by the PCA decomposition of the initial dataset

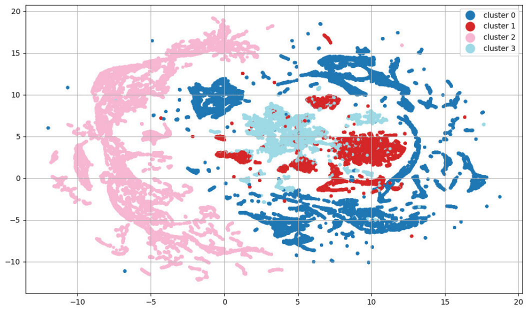


Fig. 6. Scatter plot along the dimensions of low-dimensional representation produced by the UMAP algorithm[14]

V. Conclusion

In this paper, different clusterization methods for multi-variant e-commerce interfaces were reviewed and their performance was compared. Summarizing the results of the study, it can be concluded that after taking into account business requirements and clustering quality metrics, agglomerative clustering for 4-7 clusters or K-means method for 4-6 clusters can be selected for clustering e-commerce customers. In addition, when choosing agglomerative clustering, it is possible to reduce the value of the pca parameter, in order to speed up calculations. The research has identified the most promising clustering methods that can be used to provide specific groups of e-commerce customers with dedicated user interface variants. The results obtained will be used to further develop and validate the AIM^2 platform in future implementations.

Acknowledgment

The datasets used during the study are available from the author upon reasonable request.

References

[1] J. Ah-Pine, ”An Efficient and Effective Generic Agglomerative Hierarchical Clustering Approach”. *Journal of Machine Learning Research* vol 19(1), 2018, pp. 1615-1658.

[2] F. Andriyani, Y. Puspitarani ”Performance Comparison of K-Means and DBScan Algorithms for Text Clustering Product Reviews”, *SinkrOn* vol. 7(3), 2022, pp. 944-949.

[3] L. Ardissono, A. Goy, G. Petrone, M. Segnan ”A multi-agent infrastructure for developing personalized web-based systems” *ACM Transactions on Internet Technology* vol. 5(1), 2005, pp. 47-69.

[4] D. Arthur, S. Vassilvitskii ”k-means++: the advantages of careful seeding”, *SODA ’07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027-1035.

[5] R.P. Chatterjee, K. Deb, S. Banerjee, A. Das, R. Bag ”Web Mining Using K-Means Clustering and Latest Substring Association Rule for E-Commerce”, *Journal of Mechanics of Continua and Mathematical Sciences* vol. 14(6), 2019, pp. 28-44.

[6] X. Chen, W. Sun, B. Wang, Z. Li, X. Wang, Y. Ye “Spectral Clustering of Customer Transaction Data With a Two-Level Subspace Weighting Method” *IEEE Transactions on Cybernetics* vol. 49, 2019, pp. 3230.

[7] Darwin, R. Purba, M.F. Pasha “Search Query Clustering Comparation On E-Commerce Using K-Means And Adaptive DBSCAN” *3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT)*, 2020, pp. 207-211.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu ”A density-based algorithm for discovering clusters in large spatial databases with noise”, *KDD’96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226-231.

[9] F. Gözükara, S.A. Özel “An Incremental Hierarchical Clustering Based System For Record Linkage In E-Commerce Domain” *The Computer Journal* vol. 66(3), 2021, pp. 581-602.

[10] P.D.Hung, N.T.T. Lien, N.D. Ngoc “Customer Segmentation Using Hierarchical Agglomerative Clustering” *ICISS ’19: Proceedings of the 2nd International Conference on Information Science and Systems*, 2019, pp. 33-37.

[11] P. Jiang, Y. Zhu, Y. Zhang, Q. Yuan “Life-stage Prediction for Product Recommendation in E-commerce” *KDD ’15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1879-1888.

[12] A. Kobsa, “Personalized hypermedia and international privacy” *Communications of the ACM* vol. 45, 2002, pp. 64-67.

[13] R.V.S. Kumar, S.S. Rao, P. Srinivasrao “An Efficient Clustering Approach using DBSCAN” *Helix* vol. 8(3), 2018, pp. 3399-2305.

[14] L. McInnes, J. Healy, J. Melville “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction” *ArXiv e-prints*, 2020, https://arxiv.org/pdf/1802.03426

[15] M. Montaner, B. López, J. L. de la Rosa “A Taxonomy of Recommender Agents on the Internet” *Artificial Intelligence Review* vol. 19, 2003, pp. 285-330.

[16] E. Patel, D.S. Kushwaha “Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model” *Artificial Procedia Computer Science* vol. 171, 2020, pp. 158-167.

[17] K. Tabianan, S. Velu, V. Ravi ”K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data”, *Sustainability* vol. 14(12), 2022, pp. 1-15.

[18] E. Triandini, F.A. Hermawati, I.K.P. Suniantara ”Hierarchical Clustering for Functionalities E-Commerce Adoption”, *Jurnal Ilmiah KURSOR* vol. 10(3), 2020, pp. 111-118.

[19] Y. Yang, J. Jiang, H. Wang ”Application of E-Commerce Sites Evaluation Based on Factor Analysis and Improved DBSCAN Algorithm”, *International Conference on Management of e-Commerce and e-Government*, 2018, pp. 33-38.

[20] L. Wang, Y. Jing ”Collocating Recommendation Method for E-Commerce Based on Fuzzy C-Means Clustering Algorithm”, *Journal of Mathematics* vol. 2022, 2022, pp. 1-11.

[21] Z. Wu, L. Jin, J. Zhao, L. Jing, L. Chen ”Research on Segmenting E-Commerce Customer through an Improved K-Medoids Clustering Algorithm”, *Computational Intelligence and Neuroscience* vol. 2022, 2022, pp. 1-10.

[22] B. Zhang, L. Wang, Y. Li ”Precision Marketing Method of E-Commerce Platform Based on Clustering Algorithm”, *Complexity* vol. 2021, 2021, pp. 1-10.

[23] Y. Zhang, M. Li, S. Wang, S. Dai, L. Luo, E. Zhu, H. Xu, X, Zhu, C. Yao, H. Zhou ”Gaussian Mixture Model Clustering with Incomplete Data”, *ACM Transactions on Multimedia Computing, Communications, and Applications* vol. 17(1s), 2021, pp. 1-14.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 315–318

DOI: 10.15439/2023F4101
ISSN 2300-5963 ACSIS, Vol. 35

QMAK: Interacting with Machine Learning Models and Visualizing Classification Process

Arkadiusz Wojna, Katarzyna Jachim, Łukasz Kosson, Łukasz Kowalski, Damian Mański, Michał Mański, Krzysztof Mroczek, Krzysztof Niemkiewicz, Robert Piszczatowski, Maciej Próchniak, Tomasz Romańczuk, Piotr Skibiński, Marcin Staszczyk, Michał Szostakiewicz, Leszek Tur, Damian Wójcik, Maciej Zuchniak
University of Warsaw
Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warsaw, Poland
Email: wojna@mimuw.edu.pl

***Abstract*—In various classification problems beside high accuracy data analysts expect often understanding and certain insight into the process of classification. To help them understand why a trained model selects a particular decision, how confident it is in the assigned decision, and to enable interactive improvement of trained models we present QMAK. The tool visualizes not only classification models but also the processes classifying individual objects. Five classical machine learning models and their classification process are visualized with QMAK: neural network, decision tree, k-nearest neighbors, classifier based on principal component analysis (PCA) and rough set based classifier. QMAK provides also exemplary functions enabling users to modify trained models interactively.**

***Index Terms*—explainable machine learning, classification visualization, interactive classifier.**

I. Introduction

As THE field of machine learning matured, beside a predicted value data analysts started to expect some insight into how a trained model made a decision. They expect explanation why decision was selected and how much the model is sure of its decision. Understanding why classification models classify incorrectly some cases can help both authors and analysts improve their models. Yet further step it is to provide the ability of modifying a trained model in an interactive process.

There are many visualization tools dedicated to machine learning but most of them use various charts, sometimes quite advanced, to visualize data, training metrics and classification results, for example Neptune [1]. Many others can visualize classification models based on the graph structure like Graphviz [2] and TensorBoard [3], or on the tree structure like dtreeviz [4], or both, for example Weka [5]. However, only the models having one of these two structures can be visualized with those tools. The tools visualizing classification models reflecting their specific structure are often most advanced and most detailed in how they present a model, but they are usually dedicated to a single model type, for example Netron [6] and NN-SVG [7] that visualize neural networks.

QMAK is a visualization and interactive platform gathering different classification models. It provides a framework not only for visualization and interaction with models but also for visualization of the classification process. It allows users to compare both the structure of different models and how they differ in the classification processes provided with the same object to be classified. QMAK implements visualization of popular machine learning models and gives examples of how users may interact with models to improve their classification accuracy. The platform can be used also as a didactic tool during machine learning courses.

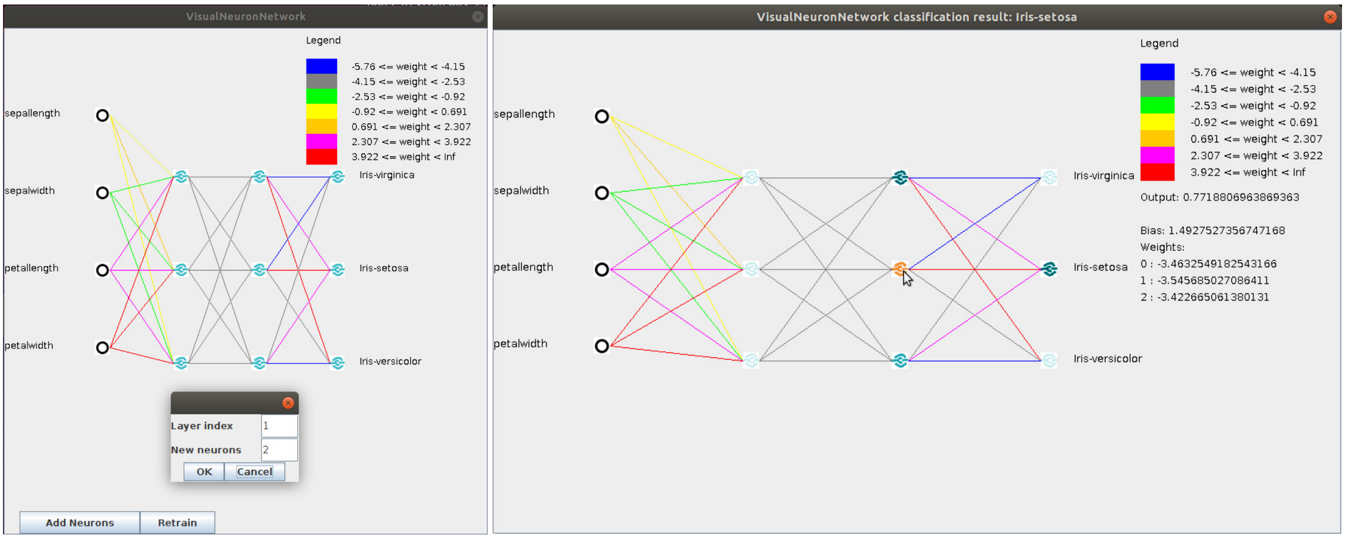
II. System Overview

QMAK is a graphical tool providing the following features: visualization of data, classifiers and single object classification, interactive classifier modification by a user, classification of test data with presentation of misclassified objects, and experiments comparing classification accuracy of different classifiers using different types of tests. It is an open source software issued under the GNU General Public License. The tool and its demo are available at http://rseslib.mimuw.edu.pl/qmak.

QMAK uses Rseslib library [8], [9] as the source of classification models. The version 3.3.0 provides visualization of five classifiers: neural network, decision tree, k-nearest neighbors, classifier based on principal component analysis (PCA) and rough set based rule classifier. Users can implement new classifiers and their visualization and add them easily to QMAK.

Neural network in QMAK is trained with the classical backpropagation algorithm and sigmoid activation functions [10]. Visualization of a neural network presents the neurons and the connections between them (see Figure 1). The neurons from the last layer correspond to decisions. The color of a connection represents its weight as it is defined in the legend. A user can select a neuron to display the exact weights of its input connections and its bias. They can also modify a trained network by adding new neurons in hidden layers and retraining the network. Visualization of classification presents also the strength of the output signal from each neuron with intensity of its color, and the exact value of the output signal after clicking on a selected node.

The decision tree in QMAK is implementation of the well-known C4.5 algorithm [11]. It is visualized by presenting the structure of the tree (see Figure 2). After selection of a node the decision distribution of the training objects entering that


VisualNeuronNetwork
Legend
🟦 -5.76 <= weight < -4.15
⬜ -4.15 <= weight < -2.53
🟩 -2.53 <= weight < -0.92
🟨 -0.92 <= weight < 0.691
🟧 0.691 <= weight < 2.307
🟪 2.307 <= weight < 3.922
🟥 3.922 <= weight < Inf
sepallength
sepalwidth
petallength
petalwidth
Iris-virginica
Iris-setosa
Iris-versicolor
OK
Cancel
Add Neurons
Retrain
VisualNeuronNetwork classification result: Iris-setosa

Layer index 1

New neurons 2

```
Output: 0.7718806963869363

Bias: 1.4927527356747168
Weights:
0 : -3.4632549182543166
1 : -3.545685027086411
2 : -3.422665061380131
```

Figure 1. Visualization of neural network (left) and its classification process (right)

node is displayed, and the branching condition for an internal node or the assigned decision for a leaf node. A user can cut off the subtree of any internal node and convert it to a leaf. Visualization of classification presents a decision tree with the path from the root to a leaf node highlighted in green corresponding to a classified object.

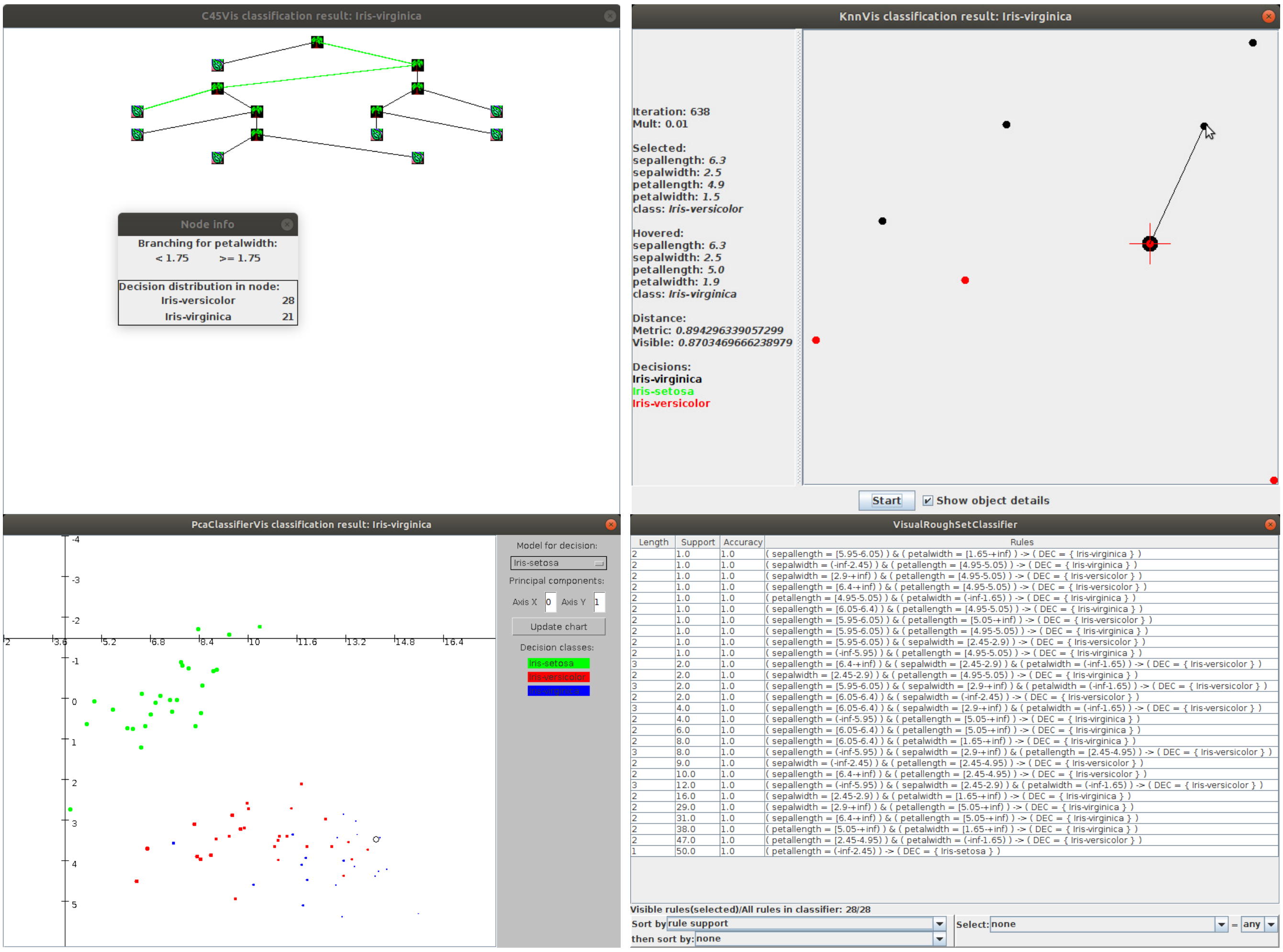
K-nearest neighbours classifier [12] provides distance measures working for data with both numerical and categorical attributes and optimized with attribute weighting. It optimizes automatically also the number k of nearest neighbours and applies weights in voting by nearest neighbours. Visualization of k-NN classifier projects all training objects onto the two-dimensional area of the window, marking the objects of each decision class with a different color. The process of searching for placement of the objects that most faithfully reflects the true distances between them in the induced metric, is displayed live, and can be stopped at any time. A user can select one object and hover the cursor over another one to display the attribute values of both objects and the true distance between them. Visualization of classification by k-NN classifier projects only the classified object and its k nearest neighbors onto the window area, also searching for placement most faithfully reflecting the true distances between them. As in the model visualization, the neighbors from different decision classes have different colors, and a user can display their attribute values and the true distances between them.

The visualized search for the best placement of objects in the two-dimensional area of the QMAK window for k-NN classifier uses an algorithm that combines simulation of spring-line attraction and repulsion with simulated annealing. Each object is assigned random initial coordinates, sampled uniformly from the unit square. The coordinates are then refined in an iterative process. In each iteration two distances are computed for each pair of objects: the true distance in the induced metric and the Euclidean distance between the current coordinates. The difference between these two distances is later applied as the multiplication factor to the vector between the current coordinates of the objects to obtain the force vector. The correction vector for each object is computed as the sum of all force vectors for that object. At the end of each iteration, the correction vectors are multiplied by a scaling factor, reduced to a maximum length, and applied to the current coordinates. The scaling factor decreases exponentially with each epoch to ensure long-term pseudostability of otherwise chaotic N-body problem, while reduction to maximum length reduces instabilities in early iterations.

PCA classifier finds a separate model of principal components for each decision class using Oja-RLS rule [13]. Its visualization projects all training objects onto the plane spanned by a selected pair of principal components of the model for a selected decision class. The objects of each decision class are marked with a different color. The objects closer to the plane are represented by larger dots, the ones more distant from the plane are represented by smaller dots. A user can switch between different decision classes and different pairs of principal components. Visualization of classification by PCA classifier marks additionally the position of the classified object on the presented plane.

Rough set classifier uses the algorithms computing discernibility matrix, reducts and rules generated from reducts [14], [15], [8]. Its visualization presents the decision rules with their length, support and accuracy. A user can filter and sort the rules by attribute occurrence, attribute values, rule length, support or accuracy. Visualization of classification shows only the rules matching the classified object enabling the same filtering and sorting criteria as visualization of the classifier.

Beside visualization of classification QMAK integrates also other features. Users can run experiments comparing the accuracy of different classification models, or the accuracy of the same classification model for different parameter settings,


C45Vis classification result: Iris-virginica
KnnVis classification result: Iris-virginica
PcaClassifierVis classification result: Iris-virginica
VisualRoughSetClassifier
2
3.6
5.2
6.8
8.4
10
11.6
13.2
14.8
16.4
-4
-3
-2
-1
0
1
2
3
4
5
Start
☑ Show object details
Update chart

Node info

Branching for petalwidth:
< 1.75 >= 1.75

Decision distribution in node:
Iris-versicolor 28
Iris-virginica 21

Iteration: 638
Mult: 0.01

Selected:
sepallength: *6.3*
sepalwidth: *2.5*
petallength: *4.9*
petalwidth: *1.5*
class: *Iris-versicolor*

Hovered:
sepallength: *6.3*
sepalwidth: *2.5*
petallength: *5.0*
petalwidth: *1.9*
class: *Iris-virginica*

Distance:
Metric: *0.894296339057299*
Visible: *0.8703469666238979*

Decisions:
Iris-virginica
Iris-setosa
Iris-versicolor

Model for decision: Iris-setosa
Principal components:
Axis X 0 Axis Y 1

Decision classes:
🟩 Iris-setosa
🟥 Iris-versicolor
🟦 Iris-virginica

Length	Support	Accuracy	Rules
2	1.0	1.0	(sepallength = [5.95-6.05)) & (petalwidth = [1.65-+inf)) -> (DEC = { Iris-virginica })
2	1.0	1.0	(sepalwidth = (-inf-2.45)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-virginica })
2	1.0	1.0	(sepalwidth = [2.9-+inf)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-versicolor })
2	1.0	1.0	(sepallength = [6.4-+inf)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-versicolor })
2	1.0	1.0	(petallength = [4.95-5.05)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-virginica })
2	1.0	1.0	(sepallength = [6.05-6.4)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-virginica })
2	1.0	1.0	(sepallength = [5.95-6.05)) & (petallength = [5.05-+inf)) -> (DEC = { Iris-versicolor })
2	1.0	1.0	(sepallength = [5.95-6.05)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-virginica })
2	1.0	1.0	(sepallength = [5.95-6.05)) & (sepalwidth = [2.45-2.9)) -> (DEC = { Iris-versicolor })
2	1.0	1.0	(sepallength = (-inf-5.95)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-virginica })
3	2.0	1.0	(sepallength = [6.4-+inf)) & (sepalwidth = [2.45-2.9)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-versicolor })
2	2.0	1.0	(sepalwidth = [2.45-2.9)) & (petallength = [4.95-5.05)) -> (DEC = { Iris-virginica })
3	2.0	1.0	(sepallength = [5.95-6.05)) & (sepalwidth = [2.9-+inf)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-versicolor })
2	2.0	1.0	(sepallength = [6.05-6.4)) & (sepalwidth = (-inf-2.45)) -> (DEC = { Iris-versicolor })
3	4.0	1.0	(sepallength = [6.05-6.4)) & (sepalwidth = [2.9-+inf)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-versicolor })
2	4.0	1.0	(sepallength = (-inf-5.95)) & (petallength = [5.05-+inf)) -> (DEC = { Iris-virginica })
2	6.0	1.0	(sepallength = [6.05-6.4)) & (petallength = [5.05-+inf)) -> (DEC = { Iris-virginica })
2	8.0	1.0	(sepallength = [6.05-6.4)) & (petalwidth = [1.65-+inf)) -> (DEC = { Iris-virginica })
3	8.0	1.0	(sepallength = (-inf-5.95)) & (sepalwidth = [2.9-+inf)) & (petallength = [2.45-4.95)) -> (DEC = { Iris-versicolor })
2	9.0	1.0	(sepalwidth = (-inf-2.45)) & (petallength = [2.45-4.95)) -> (DEC = { Iris-versicolor })
2	10.0	1.0	(sepallength = [6.4-+inf)) & (petallength = [2.45-4.95)) -> (DEC = { Iris-versicolor })
3	12.0	1.0	(sepallength = (-inf-5.95)) & (sepalwidth = [2.45-2.9)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-versicolor })
2	16.0	1.0	(sepalwidth = [2.45-2.9)) & (petalwidth = [1.65-+inf)) -> (DEC = { Iris-virginica })
2	29.0	1.0	(sepalwidth = [2.9-+inf)) & (petallength = [5.05-+inf)) -> (DEC = { Iris-virginica })
2	31.0	1.0	(sepallength = [6.4-+inf)) & (petallength = [5.05-+inf)) -> (DEC = { Iris-virginica })
2	38.0	1.0	(petallength = [5.05-+inf)) & (petalwidth = [1.65-+inf)) -> (DEC = { Iris-virginica })
2	47.0	1.0	(petallength = [2.45-4.95)) & (petalwidth = (-inf-1.65)) -> (DEC = { Iris-versicolor })
1	50.0	1.0	(petallength = (-inf-2.45)) -> (DEC = { Iris-setosa })

Visible rules(selected)/All rules in classifier: 28/28
Sort by rule support
then sort by: none
Select: none = any

Figure 2. Decision tree (upper left), k-nearest neighbors classifier (upper right), PCA classifier (lower left) and rough set classifier (lower right) visualized in QMAK

for example, using cross-validation or multiple random split and test. In those experiment all five visualized classification models are available for testing as well as other non-visualized classifiers: Support Vector Machine, AQ15, Naive Bayes, RIONIDA, and others.

The tool provides also three kinds of data graphs presenting different types of correlations between categorical and numerical attributes.

III. System Usage

QMAK can help users understand why a particular decision was selected. For example, the classification path for an object in a decision tree shows the attributes and the conditions on these attributes determining the decision. K-nearest neighbors model shows the training objects identified as the most similar to the classified object and used to vote for the predicted decision. Rough set classifier shows the decision rules matching the classified object.

Using QMAK users can also find out how confident a classifier is in the assigned decision. In neural network it is indicated by the difference between the strength of the output signal from the winning neuron and the strength of the signals from other neurons. In k-nearest neighbors model one can compare the number and the placement of the nearest neighbors with the winning decision with the number and the placement of the neighbors with other decisions.

Users can also find out which element of an already trained classifier needs to be improved. QMAK highlights misclassified objects in red, and for each misclassified object a user can command the classifier to visualize its classification. In some cases, the user can later modify the classifier interactively. QMAK demonstrates a few examples of such interaction.

Similar weights of the connections between two layers in a neural network like between the two hidden layers in Figure 1, may indicate the first of those two hidden layers was given too few neurons and the second of the two layers was unable to learn the desired functions on its neurons. A user can either train a new network with a different structure or add new neurons to hidden layers of the existing network and retrain it.

In a decison tree if a branch misclassifies many test objects that branch can be the result of overfitting to a training set. In QMAK a user can prune such a branch by turning the inner node from which the branch comes out into a leaf node.

In k-nearest neighbors model one can check whether changing the number of neighbors selected to vote or changing the voting method fixes classification of misclassified objects. The number of neighbors and the voting method can be changed without retraining an already trained classifier. Moreover, if many test objects are misclassified by a k-NN classifier because of a small subset of training objects, a user may consider removal of such objects from the training set.

IV. System Extensibility

The platform is designed to make addition of both new classification models and their visualization as simple as possible. It is intended to allow users implement visualization of their classifiers in such way that they do not need to have any knowledge of how a tool presenting visualization, in particular QMAK, is implemented. To achive that, a simple interface with two methods is defined.

The first method implements visualization of the structure of a classification model:

```
void
draw(JPanel canvas)
```

The second method implements visaualization of the process classifying a single test object:

```
void
drawClassify(JPanel canvas, DoubleData obj)
```

In both methods the author of a classifier uses the provided `canvas` object to draw whatever is best to visualize the classifier.

Many visualization tools provide graph-based framework for implementing visualization. But many classification models, for example: k-nearest neighbors, rule-based classifiers, support vector machine, do not fit such framework. In QMAK the author gets a general object-frame in which they can draw any graphical representation of a classifier or classification process. Decision trees and neural networks have graph-based visualization, k-NN and PCA classifiers each presents a certain type of projection of selected objects represented as points onto two-dimensional area, and rough set classifier present text description of the components of the model. Moreover, visualization can be animated like in case of k-nearest neighbors classifier.

After implementation of the two interface methods in the source code of a classifier a user can easily add it to QMAK using menu or by adding an entry in the configuration file. It does not require any change in QMAK itself.

V. Summary

The paper presents QMAK, an open platform for visualization of classification models and classification process.

Two main features that distinguish QMAK from other machine learning visualizers are that it integrates visualization of five different classification models in one tool and that all five models visualize also the very process of classifying a single object. This unique value of QMAK can help data analysts interpret their data and the models built from data in a way that has hitherto been unavailable. The platform can also be an important alternative as a didactic tool for teaching of machine learning.

References

[1] “Neptune: metadata store for machine learning operations,” accessed: 2023-05-15. [Online]. Available: https://neptune.ai

[2] “Graphviz: open source graph visualization software,” accessed: 2023-05-15. [Online]. Available: https://graphviz.org

[3] “Tensorboard: Tensorflow’s visualization toolkit,” accessed: 2023-05-15. [Online]. Available: https://www.tensorflow.org/tensorboard

[4] “dtreeviz: python library for decision tree visualization and model interpretation,” accessed: 2023-05-15. [Online]. Available: https://github.com/parrt/dtreeviz

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witen, “The weka data mining software: An update,” *SIGKDD Explorations*, vol. 11, no. 1, pp. 10–18, 2009. doi: 10.1145/1656274.1656278

[6] “Netron: visualizer for neural network, deep learning, and machine learning models,” accessed: 2023-05-15. [Online]. Available: https://github.com/lutzroeder/netron

[7] A. LeNail, “Nn-svg: Publication-ready neural network architecture schematics,” *Journal of Open Source Software*, vol. 4, no. 33, p. 747, 2019. doi: 10.21105/joss.00747

[8] A. Wojna and R. Latkowski, “Rseslib 3: Library of rough set and machine learning methods with extensible architecture,” *LNCS Transactions on Rough Sets XXI*, vol. 10810, pp. 301–323, 2019. doi: 10.1007/978-3-662-58768-3_7

[9] A. Wojna, R. Latkowski, and Ł. Kowalski, *RSESLIB: User Guide*, accessed: 2023-05-15. [Online]. Available: http://rseslib.mimuw.edu.pl/rseslib.pdf

[10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson, 2020.

[11] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[12] A. Wojna, “Analogy-based reasoning in classifier construction (phd thesis),” *LNCS Transactions on Rough Sets IV*, vol. 3700, pp. 277–374, 2005. doi: 10.1007/11574798_11

[13] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: John Wiley & Sons, Inc., 1996.

[14] Z. Pawlak, *Rough Sets - Theoretical Aspects of Reasoning about Data*. Dordrecht: Kluwer Academic Publishers, 1991.

[15] A. Skowron and C. Rauszer, “The discernibility matrices and functions in information systems,” in *Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory*, R. Slowinski, Ed. Dordrecht: Kluwer Academic Publishers, 1992, pp. 331–362.

Thematic Tracks

PARTS 4 and 5 of FedCSIS 2023 Proceedings contain contributions originating from Thematic Tracks. Let us present each one of them.

I. Advances in Programming Languages

Programming languages are programmers' most basic tools. With appropriate programming languages one can drastically reduce the cost of building new applications, as well as maintaining existing ones. In the last decades, there have been many advances in programming languages technology, in traditional programming paradigms such as functional, logic, and object-oriented programming, as well as the development of new paradigms, such as aspect-oriented programming. The main driving force was, and will be, to better express programmers' ideas. Therefore, research in programming languages is an endless activity and the core of computer science. New language features, new programming paradigms, and better compile-time and run-time mechanisms can be foreseen in the future. Here, the potential future role of AI models in programming should also be taken into account. In this context, the aim of this Thematic Track was to provide a forum for exchange of ideas and experience in topics concerned with programming languages and systems.

Thematic Track organizers:

+ Janousek, Jan, Czech Technical University, Prague, Czech Republic
+ Luković, Ivan, University of Belgrade, Serbia
+ Mernik, Marjan, University of Maribor, Slovenia
+ Rangel Henriques, Pedro, Universidade do Minho, Portugal
+ Slivnik, Boštjan, University of Ljubljana, Slovenia
+ Varanda Pereira, Maria Joao, Instituto Politecnico de Braganca, Portugal

II. Artificial Intelligence in Agriculture

AI is increasingly used in agriculture, to address multiple issues, from plant disease detection to weeding automation, soil status monitoring, crop prediction, irrigation management, and decreased use of resources, for improving product quality and process productivity. AI can, in fact, provide highly positive effects on precision agriculture by optimizing, automating and forecasting multiple aspects of farming and revolutionizing the sector, providing helpful information and driving decisions using multiple sources of data and different sensors. Moreover, in the climate change era, AI can improve sustainability by optimizing the use of resources, such as in the case of water and soil management. This Thematic Track welcomed contributions concerning all aspects of interdisciplinary research and applications related to AI in agriculture.

Thematic Track organizers:

+ Charvat, Karel, Czech Center for Science and Society, Czech Republic
+ Martinelli, Massimo, National Research Council of Italy, Italy
+ Moroni, Davide, National Research Council of Italy, Italy
+ Procházka, Ales, University of Chemistry and Technology & Czech Technical University CIIRC, Czech Republic

III. Artificial Intelligence in Digital Humanities, Computational Social Sciences and Economics Research

This Thematic Track was dedicated to the computational study of social sciences, economics and humanities, including all subjects like, for example, education, labor market, history, religious studies, theology, cultural heritage, and informative predictions for decision-making and behavioral-science perspectives. Besides new discoveries, it was dedicated to the reflections about their growth within the field of computer science and it emphasized the interdisciplinary exchange and dissemination with a clear focus on computational and AI-based methods. Since there is a clear methodological overlap between the considered domains of social sciences, economics and humanities, and often similar algorithms and AI approaches are considered for them, this track should be seen as a place for discussing a “joint toolbox” as a support for scholars from these fields with human and context-aware agents. It included also research related to trustworthy data infrastructure housing both quantitative and qualitative data.

Thematic Track organizers:

+ Cooper, Anthony-Paul, Durham University, Durham, United Kingdom and University of Turku, Finland
+ Dörpinghaus, Jens, BIBB and University of Koblenz, Germany
+ Helmrich, Robert, BIBB and University of Bonn, Germany
+ Speckesser, Stefan, Brighton University, United Kingdom

IV. Challenges for Natural Language Processing

This Thematic Track consisted of contributions related to all aspects of NLP. Of particular interest were works addressing NLP tools, multimodal problems, cross-lingual learning and processing of natural languages. Moreover, this track was also hosting the NLP-related competitions, results of which are presented in Part 6 of these proceedings. This Thematic Track was co-organized by the Multi-task, Multilingual, Multi-modal Language Generation COST Action (CA18231).

Thematic Track organizers:

+ Kobyliński, Łukasz, Institute of Computer Science, Polish Academy of Sciences, Poland
+ Kubis, Marek, Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poland

V. Complex Networks: Theory and Application

In the nature and the world around us, one can observe many network structures that interconnect various elements such as cells, people, urban centers, network devices, companies, manufacturing machines, etc. Moreover, it is easy to notice that most of them evolve over time. The analysis of such systems from the complex networks point of view allows for better understanding of the processes within them, which can be used to optimize their structure, improve their management methods, detect failures, improve their operating efficiency and plan their development and evolution. The main goal of this Thematic Track was to exchange knowledge and experience between specialists from different areas who, in their research and design work, use theories and solutions characteristic for complex systems.

This Thematic Track was organized within framework of the project financed by the Minister of Education and Science of the Republic of Poland within the “Regional Initiative of Excellence” program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

Thematic Track organizers:

+ Bolanowski, Marek, Rzeszów University of Technology, Poland
+ Kondratenko, Yuriy, Petro Mohyla Black Sea National University, Ukraine
+ Paszkiewicz, Andrzej, Rzeszów University of Technology, Poland

VI. Computational Optimization

Many real world problems, arising in engineering, economics, medicine and other domains, can be formulated as optimization tasks. These problems are frequently characterized by non-convex, non-differentiable, discontinuous, noisy or dynamic objective functions and constraints, which ask for adequate computational methods. The aim of this Thematic Track was to stimulate the communication between researchers working on different fields of optimization and practitioners who need reliable and efficient computational optimization methods. Contributions related to both theoretical and practical aspects of optimization methods were represented.

Thematic Track organizers:

+ Fidanova, Stefka, Bulgarian Academy of Sciences, Bulgaria
+ Mucherino, Antonio, IRISA, University of Rennes, France
+ Zaharie, Daniela, West University of Timisoara, Romania

VII. Computer Aspects of Numerical Algorithms

Numerical algorithms are widely used by scientists engaged in various areas. There is a special need of highly efficient and easy-to-use scalable tools for solving large scale problems. This Thematic Track was devoted to numerical algorithms, with the particular attention focused on the latest scientific trends in this area and on problems related to implementation of libraries of efficient numerical algorithms. The main goal of this track was to facilitate meeting of researchers and exchange of their experiences.

Thematic Track organizers:

+ Bylina, Beata, Maria Curie-Skłodowska University, Poland
+ Bylina, Jarosław, Maria Curie-Skłodowska University, Poland
+ Cyganek, Bogusław, AGH University of Science and Technology, Poland
+ Lirkov, Ivan, Bulgarian Academy of Sciences, Bulgaria
+ Stpiczyński, Przemysław, Maria Curie-Skłodowska University, Poland

VIII. Cyber-Physical Systems Software Engineering

This Thematic Track has its roots in the IEEE Software Engineering Workshop (SEW), which is the oldest Software Engineering event in the world, dating back to 1969. It was originally run as the NASA Software Engineering Workshop and focused on software engineering issues relevant to NASA and the space industry. After the 25th edition, it became the NASA/IEEE Software Engineering Workshop and expanded its remit to address many more areas of software engineering, with emphasis on practical issues, industrial experience and case studies in addition to traditional technical papers. Since its 31st edition, it has been sponsored by IEEE and has continued to broaden its areas of interest. Now it is an integral part of the FedCSIS conference series, as one of important Thematic Tracks.

One extremely hot new area are Cyber-Physical Systems (CPS), which encompass the investigation of approaches related to the development and use of modern software systems interfacing with real world and controlling their surroundings. CPS are physical and engineering systems closely integrated with their typically networked environment. Modern airplanes, automobiles, or medical devices are practically networks of computers. Sensors, robots, and intelligent devices are abundant. Human life depends on them. CPS systems transform how people interact with the physical world just like the Internet transformed how people interact with one another. Accordingly, this Thematic Track brought together researchers with interest in software engineering, both with CPS and broader focus. Moreover, it provided a forum for reporting on past experiences, for describing new and emerging results and approaches, and for exchanging ideas on best practice and future directions.

Thematic Track organizers:

+ Bowen, Jonathan, Museophile Ltd., United Kingdom
+ Hinchey, Mike, Irish Software Engineering Research Centre, Ireland
+ Szmuc, Tomasz, AGH University of Science and Technology, Poland
+ Zalewski, Janusz, Florida Gulf Coast University, United States

IX. Cyber Security, Privacy, and Trust

Nowadays, information security is a backbone for protecting both user data and electronic transactions. Protecting communications and data infrastructures of an increasingly inter-connected world have become vital. Security has also emerged as an important scientific discipline whose many multifaceted complexities require synergy of computer science, engineering, and information systems communities. Information security has some well-founded technical research directions which encompass access level (user authentication and authorization), protocol security, software security, and data cryptography. Moreover, some other emerging topics related to organizational security aspects have appeared beyond the long-standing research directions. In this context, this Thematic Track focused on the diversity of the cyber information security developments and deployments in order to highlight the most recent challenges and report the most recent researches. It was designed as an umbrella for all cyber security technical aspects, user privacy techniques, and trust. In addition, it went beyond the technicalities and covered some emerging topics like social and organizational security research directions.

Thematic Track organizers:

+ Białas, Andrzej, Institute of Innovative Technologies EMAG, Poland
+ Masud, Mohammad, United Arab Emirates University, United Arab Emirates

X. Data Science in Health, Ecology and Commerce

This Thematic Track was a forum for exchange of ideas concerning all forms of data analysis, data economics, information systems and data based research, focusing on the interaction of those three fields. Here, data-driven solutions can be generated by understanding complex real-world (health-related) problems, critical thinking and analytics to derive knowledge from (Big) data. The past years have shown a forthcoming interest on innovative data technology and analytics solutions that link and utilize large amounts of data across individual digital ecosystems. Here, scenarios, in the field of health, smart cities or agriculture, merge data from various IoT devices, social media or applications and demonstrate the great potential for gaining new insights, supporting decisions, or providing smarter services. Together with inexpensive sensors and computing power they provide foundation of a world that bases its decisions on data. However, this is only the beginning of the journey, and the pertinent methods and technologies, and the potential application fields, as well as the impact on society and economy, have to be explored. This endeavor needs the knowledge of researchers from different fields applying diverse perspectives and using different methodological directions to find a way to grasp and fully understand the power and opportunities of data science. Bringing together researchers and practitioners of pertinent fields was one of focal points of this Thematic Track.

Thematic Track organizers:

+ Bumberger, Jan, Helmholtz-Centre for Environmental Research – UFZ, Germany
+ Franczyk, Bogdan, University of Leipzig, Germany
+ Häckl, Dennis, University of Leipzig, Germany and WIG2 Institute for Health Economics and Health Service Research, Germany
+ Militzer-Horstmann, Carsta, WIG2 Institute for Health Economics and Health Service Research, Germany
+ Reinhold, Olaf, University of Leipzig / Social CRM Research Center, Germany

XI. Distributed Edge AI – Risks and Challenges

The advent of end and edge nodes, with increased computational and storage capabilities, makes it possible to perform a large range of computations, with special focus on AI-based tasks on privacy-sensitive data, locally at the edge. This makes it possible to rely on a backend cloud only for communicating results of non-private data, for example by aggregation, or performing computations requiring the combination and further processing of results from different edge devices, or historical data sources, as it is done in the federated learning approaches. In addition, this increase in computation capabilities enables also neighboring edge devices to perform tasks collaboratively, leveraging each other's computational resources, before offloading data and computations to the cloud backend.

This Thematic Track focused on AI and ML techniques related to edge computing systems, and security and privacy approaches in view of data sharing, in order to enable smart and sustainable planning and operation of resource constrained IoT ecosystems, and edge computing applications. It was organized within the scope of the European ITEA3 MIRAI R&D project.

Thematic Track organizers:

+ Hristoskova, Anna, Sirris, Belgium
+ Klein, Sarah, Sirris, Belgium

XII. Information Systems Management

This Thematic Track facilitated a forum for exchange of ideas, for practitioners and theorists working in the broad area of information systems management in organizations. It focused on three complimentary directions: management of information systems in an organization, uses of information systems to empower managers, and information systems for sustainable development. Here, the interest encompassed all aspects of planning, organizing, resourcing, coordinating, controlling and leading the management function to ensure a smooth operation of information systems in organizations. Moreover, the contributions discussing the uses of intelligence systems and information technology to automate or otherwise facilitate the management function were included. Researches on the influence of intelligence systems on sustainability were welcomed as well.

Thematic Track organizers:

+ Bicevska, Zane, University of Latvia, Latvia
+ Chmielarz, Witold, University of Warsaw, Poland
+ Duan, Yanqing, University of Bedfordshire, United Kingdom

+ Leyh, Christian, University of Applied Sciences, Germany

XIII. Internet of Things – Enablers, Challenges and Applications

The Internet of Things (IoT) is a technology which is rapidly emerging around the world. IoT applications include: smart city initiatives, wearable devices aimed to real-time health monitoring, smart homes and buildings, smart vehicles, environment monitoring, intelligent border protection, logistics support. IoT is a paradigm that assumes a pervasive presence in the environment of many smart things, including sensors, actuators, embedded systems and other similar devices. Widespread connectivity, getting cheaper smart devices and a great demand for data, testify to that the IoT will continue to grow by leaps and bounds. The business models of various industries are being redesigned on basis of the IoT paradigm. This Thematic Track focused on the IoT challenges in networking and information management, security and privacy, logistics, situation awareness, and medical care. It was organized within the scope of the European ASSIST-IoT and aerOS projects.

Thematic Track organizers:

+ Chudzikiewicz, Jan, Military University of Technology, Poland

+ Zieliński, Zbigniew, Military University of Technology, Poland

XIV. Knowledge Acquisition and Management

Knowledge management is a large multidisciplinary field having its roots in management and AI. Activity of an extended organization should be supported by an organized and optimized flow of knowledge to effectively help all participants in their work. The aim of this Thematic Track was to discuss approaches, techniques and tools in the knowledge acquisition and other knowledge management areas with a focus on contribution of AI for improvement of human-machine intelligence and face the challenges of this century. It shared information and experiences, as well as delved into current trends of methodological, technological and implementation aspects of knowledge management processes.

Thematic Track organizers:

+ Berka, Petr, Prague University of Economics and Business, Czech Republic

+ Hauke, Krzysztof, Wrocław University of Economics, Poland

+ Owoc, Mieczyslaw, Wrocław University of Economics, Poland

+ Pondel, Maciej, Wrocław University of Economics, Poland

XV. Meta Environment for Citizens, Business and Entertainment

This Thematic Track was focused on the development of Web 3.0, in particular in relation to the use of virtual reality (VR) and augmented reality (AR), on top of emerging technologies such as blockchain and widely understood IoT. The genesis of those ideas and components came from the video game industry, but also, its strong presence in business, medicine, banking, government, etc. All that, placed on top of IoT and other solicited concepts, as well as blockchain technologies, allowed discussion of new ideas related to the amalgamation of the technology aspects. The goals of the track were to expand understanding of current composition of different technologies cross business areas, including those that indicate the initial or minor presence in business and its opportunity to strengthen its presence, discover new areas of development of discussed subjects, as well as redesign well known applications with the new approach and concepts. Moreover, researches and observations related to the use of technology in business, medicine or science in a way that increases efficiency as well as ensuring a qualitative leap were welcome.

Thematic Track organizers:

+ Szumski, Oskar, University of Warsaw, Poland

+ Tan, Qing, Athabasca University, Canada

XVI. Multimedia Applications and Processing

Multimedia, computer vision, graphics, and machine learning have become ubiquitous in modern information systems, creating new challenges for detection, recognition, indexing, access, search, retrieval, automated understanding, and processing, resulting in many applications based on image and signal processing, machine learning and various multimedia technologies. Recent advances in pervasive computers, networks, telecommunications, and information technology, along with the proliferation of multimedia mobile devices, have stimulated the rapid development of intelligent applications. These key technologies, using virtual reality, augmented reality, and computational intelligence, are creating a multimedia revolution that significantly impacts a broad spectrum of consumer, business, healthcare, educational and governmental domains. Advancements in AI resulted in the rapid growth of both methods and applications of machine learning approaches in computer vision, image processing, and analysis. Further advances in parallel computing, in the first decade of the 21st century, combined with development of deep neural networks, became a real game-changer in machine vision. This Thematic Track covered a range of AI-based theories, methods, algorithms, technologies, and systems for diversified and heterogeneous digital multimedia, imaging, computer graphics and machine learning areas. Moreover, it provided an opportunity for researchers and professionals to discuss present and future challenges and potential collaboration for future progress in these fields.

Thematic Track organizers:

+ Iwanowski, Marcin, Warsaw University of Technology, Poland

+ Kwaśnicka, Halina, Wrocław University of Science and Technology, Poland

- Śluzek, Andrzej, Khalifa University, United Arab Emirates
- Stanescu, Liana, University of Craiova, Romania

XVII. Practical Aspects of and Solutions for Software Engineering

This Thematic Track was a follow-up to the KKIO series of software engineering conferences, organized, since 1999, under the auspices of the Polish Information Processing Society. The track was focused on emerging challenges and solutions for software engineering industry. A particular interest was related to validation and demonstration of practical applications of the proposed approaches. The track gathered contributions concerned with practical aspects of software engineering, relevant to the IT industry (including challenges and needs), research ideas and solutions aimed at addressing such aspects, and became a place to establish the cooperation between scientific and industrial partners.

Thematic Track organizers:

- Jarzębowicz, Aleksander, Gdańsk University of Technology, Poland
- Przybyłek, Adam, Gdańsk University of Technology, Poland
- Staroń, Mirosław, University of Gothenburg, Sweden

XVIII. Recent Advances in Information Technology

The aim of this Thematic Track was to provide a platform for exchange of ideas between early-stage researchers, in computer science and intelligence systems, Ph.D. students in particular. Furthermore, it provided all participants an opportunity to get feedback on their studies from experienced members of the AI and IT research communities. In fact, this track – as in previous years – played the role of so-called Doctoral Symposium, a well-established tradition of the FedCSIS conference series. Here, special care was taken to provide support and mentoring for young researchers and to facilitate frutiful discussion and exchange of ideas.

Thematic Track organizers:

- Dedinec, Aleksandra, Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, North Macedonia
- Gil, David, Computer Technology Department, University of Alicante, Spain
- Kowalski, Piotr, Systems Research Institute, Polish Academy of Sciences and AGH University of Science and Technology, Poland
- Łukasik, Szymon, Systems Research Institute, Polish Academy of Sciences and AGH University of Science and Technology, Poland

XIX. Rough Sets: Theory and Applications

This Thematic Track discussed research related to the state-of-the-art and future perspectives of rough sets, considered from both a theoretical standpoint and real-world applications. Rough set theory is a versatile mathematical framework that has proven successful in AI, knowledge representation, approximate reasoning, data mining, machine learning, and pattern recognition, among other areas. The track was devoted to all the mentioned areas, with an additional emphasis on problems of modeling AI processes using rough set-based techniques. The track provided an opportunity for interdisciplinary exchange and collaboration among scientists from diverse backgrounds, including mathematics, computer science, statistics, physics, engineering, and social sciences. Moreover, it allowed staying up-to-date with the state-of-the-art in rough set theory and its applications, and to discuss future research directions and opportunities. Last but not least, the track was synchronized with Rough Set School which provided a more fundamental background for the theory and applications of rough sets.

Thematic Track organizers:

- Artiemjew, Piotr, University of Warmia and Mazury in Olsztyn, Poland
- Chelly Dagdia, Zaineb, UVSQ, Paris-Saclay, France
- Mani, A., Indian Statistical Institute, India

XX. Scalable Computing

The world of large-scale computing continuously evolves. The most recent addition to the mix comes from numerous data streams that materialize from exploding number of cheap sensors installed "everywhere", on the one hand, and ability to capture and study events with systematically increasing granularity, on the other. To address the needs for scaling computational and storage infrastructures, concepts like: edge, fog and dew computing emerged. Novel issues, involved in "pushing computing away from the center", did not replace open questions that existed in the context of grid and cloud computing. Rather, they added new dimensions of complexity and resulted in the need of addressing scalability across more and more complex ecosystems consisting of individual sensors and micro-computers (e.g. Raspberry PI based systems) as well as supercomputers available within the Cloud (e.g. Cray computers and/or machines for large ML model training, facilitated – for instance – within the MS Azure Cloud). Addressing research questions that arise in individual "parts" as well as across the ecosystem viewed from a holistic perspective, with scalability as the main focus, was the goal of this Thematic Track.

Thematic Track organizers:

- Gepner, Paweł, Warsaw University of Technology, Poland
- Gusev, Marjan, University Ss. Cyril and Methodius, Macedonia
- Petcu, Dana, West University of Timisoara, Romania
- Ristov, Sashko, University of Innsbruck, Austria
- Stencel, Krzysztof, University of Warsaw, Poland

DOI: 10.15439/2023F8595
ISSN 2300-5963 ACSIS, Vol. 35

Psychological Safety, Leadership and Non-Technical Debt in Large-Scale Agile Software Development

Muhammad Ovais Ahmad
Dept. of Computer Science
Karlstad University
Karlstad, Sweden
Email: ovais.ahmad@kau.se

***Abstract*—Psychological safety has been hypothesised as an important antecedent of the success of agile software development (ASD) teams. However, there is a lack of investigation on psychological safety in large-scale agile (LSA) software development teams. This study explored the antecedents and effects of psychological safety on LSA teams. We conducted semi-structured interviews with software professionals working on LSA project in a Scandinavian technology company. The results suggest that building a psychologically safe environment is a multi-dimensional factor that requires proactive leadership approach, open communication and constructive feedback. The focus should be on designing teams for learning, remuneration safety, and a well-prepared onboarding process for new team members. A psychologically safe environment contributes to effective teamwork, work satisfaction, and promotion of learning. Absence of such an environment leads to brain drain, highlighting the consequences of neglecting this essential aspect of organisational culture. Future research directions are proposed in this paper.**

***Index Terms*—Psychological safety, leadership, non-technical debt, agile, large-scale, software development.**

I. Introduction

AGILE methods help software companies improve the quality of their products while maximising customer value. It also helps to have an efficient response to defects, improved communication, and effectiveness of coordination [2]–[5]. Nonetheless, it presents various management challenges, some of which originate from inadequacies in the ongoing and closely-knit communication necessary for the effectiveness of Agile methods. According to Boehm and Turner [35], agile brings various challenges such as development process conflicts, business process conflicts, ratings), and people conflicts. The foundation of the Agile approach rests on collaborative relationships and the interconnectedness among team members. It is of utmost importance that any questions teammates may have about the possible ramifications of expressing their opinions – whether it pertains to identifying gaps in others' work or struggles within their own tasks – do not hinder the overall performance. To enhance productivity in software development, it is important to understand the factors that influence individual and team performance. Additionally, it is important the team members feel safe and to "*offer ideas, admit mistakes, ask for help, or provide feedback in hierarchies*" [34].

Psychological safety is an important factor for teams working in agile environments and performing knowledge-intensive software tasks [1]–[3]. Psychological safety is "*a shared belief held by members of a team that the team is safe for interpersonal risk-taking*" [6].

ASD methods have been designed for small-scale projects, but their potential positive outcomes have made them attractive to LSA software development projects. Kalenda *et al.* [5] and Dikert *et al.* [4] reported LSA success factors (i.e., management support, executive sponsorship and teamwork support) and challenges (i.e., difficulty in implementing agile methods, coordination challenges in a multi-team environment, mid-level managers' unclear role in ASD, too much pressure and workload, and lack of knowledge, coaching and training [4, 5]. Another concept that significantly affects software development is "non-technical debt" (NTD). NTD covers non-technical or social aspects of software development [2]. Several factors contribute to social, process and people's debts in software engineering (i.e., lack of knowledge; lack of communication, collaboration and coordination; inadequate management decision; low developer morale; lack of psychological safety, etc.) [2], [4], [5].

To be successful in an agile environment and be able to handle the aforementioned challenges, teams must engage in more open communication and close collaborative relationships among their members. To do so, psychological safety is an important condition of the agile team environment. Psychological safety has been extensively studied in social science [3], [7–9] and has played an important role in organisational research, as reported in the Google Aristotle project [3], [10]. Psychological safety has positive effects on team performance, job satisfaction and team reflexivity . In the context of ASD, limited research has been conducted on psychological safety [33], specifically in LSA projects. To fill this knowledge gap, we need a holistic understanding of what it takes to work effectively in LSA teams. Thus, in the present study, we seek to answer this research question:

RQ: What are the antecedents and effects of psychological safety on LSA teams?

To answer the RQ, we report the qualitative findings from a survey of eight software professionals working on an LSA project in a Scandinavian technology company.

In Section II, we explain the LSA and psychological safety concepts. The research method that we used is presented in Section III, followed by the results in Section IV. The study's limitations and threats to validity are reported in Section V. We end this paper with the discussion and conclusion in Section VI.

II. Background

ASD is a set of iterative and incremental methods captured in the Agile Manifesto. The latter focuses on team interaction, working software, customers' requirements and promptness to change [12]. Such methods are used in both small-scale and large-scale ASD projects [4], [5]. There is a growing body of research on scaling ASD. Dikert [4] listed a range of LSA definitions and concluded that *large-scale denote software development organizations with 50 or more people or at least six teams*. There is a wide range of frameworks such as LeSS, SAFe, DAD, Spotify, Nexus and Scrum-at-Scale.

LSA teams face various challenges (i.e., managing complexities and interdependencies, diverse teams, roles and personalities, sub-optimal processes, conflicting agendas between teams, and complex and ambiguous goals) [4], [5], [13], which lead to NTD (i.e., social, process and people's debt) [8], [14]. The causes of social debt are gender biases; lack of communication and collaboration; power distance; organisational silos and lack of kindness [2]. Process debt mostly occurs when organisations ignore process competence development, process divergence and uncontrolled external dependencies [2]. People's debt is caused mostly by priggish members, demotivation of non-senior members, inadequate management decision and lack of psychosocial safety [2]. Most of these issues are either people-oriented or environment-related concerns. Enhancing psychological safety has a moderating effect on communication deficiencies and collaboration issues, whereas the intensity of task-related collaboration exhibits both promoting and mitigating effects [2].

There is a positive correlation between managers' openness and transformational leadership, on one hand, and psychological safety, on the other hand [15]. Leader's inclusiveness important. It encompasses verbal and behavioural actions of leaders aimed at signalling an invitation for open comments and feedback that are respected and valued, plays a pivotal role in cultivating an environment characterized by psychological safety [36]. Nonetheless, the process of feedback should always be approached with a mindful consideration of the potential conflicts it might elicit. It is important to recognize that not all forms of conflict are disadvantageous. Task-related conflicts, emerging from differing viewpoints concerning a specific task, might not yield as many adverse effects as relationship conflicts, which stem from interpersonal frictions, like harbouring negative sentiments towards an individual [37]. Further, psychological safety is important for successful collaboration, open communication, knowledge and information sharing, and learning from failures and performance [6], [16–19]. Inter-team coordination has no positive relation to team performance, but psychological safety has a significantly high positive correlation to team performance [7]. Social agile practices (e.g., daily scrums, retrospectives or pair programming) positively influence psychological safety, transparency, communication and ultimately, productivity [20].

To create a psychologically safe environment, it is vital to establish collective responsibility for team performance [21], [22]. Safdar *et al.* [11] quantitatively investigated knowledge sourcing in new product development teams through a psychological safety lens. Their study's results show that psychological safety plays a significant role in a software engineer's knowledge source selection. A software engineer who feels a high level of psychological safety is inclined to consult team members, whereas a software engineer with a low level of psychological safety tends to choose external sources [11]. In the ASD context, to institutionalise psychological safety, individuals, teams and the leadership should combine their efforts to implement strategies for no-blame, open and collective decision-making in the team and proactively supporting a psychologically safe environment [33].

Thorgren and Caiman [38] investigated the role of psychological safety in implementing agile methods across cultures. Their results show that psychological safety is essential for the successful implementation of agile methods in cross-cultural teams. Further, their investigation indicated that by cultivating psychological safety within a team, the possible conflicts and tensions that may arise from the intersection of agile practices, values, and the work environment culture can be reduced [38].

Hennel and Rosenkranz [20] conducted three case studies in two large insurance companies and one software development company. The goal was to investigate the effects of psychological safety and agile practices on team performance. Their results suggest that social agile practices (e.g., daily stand-ups, retrospectives, and Sprint planning) influence psychological safety, transparency, communication, and ultimately productivity [20].

III. Research settings

The grounded theory (GT) approach involves a set of steps for data collection, analysis, formulation of theory parameters and reporting [23]. The GT helps researchers to identify common patterns across interview transcripts by constantly comparing data at different levels of abstraction [24]. This approach does not rely on a preconceived hypothesis rather aims to uncover the interviewees concerns in the process. In this study, the focus was on the experience in LSA development project, working environment challenges and strategies in real-world settings.

We conducted semi-structured interviews with seven agile practitioners from a Scandinavian technology company. The selected case company is a partner in the NODLA project and uses various agile methods in their large-scale projects and product development. NODLA project aiming to investigate large-scale ASD and non-technical debt, funds by the Knowledge Foundation in Sweden.

TABLE I. INTERVIEWEES BACKGROUND

ID	Interviewee's title	Development Experience
EL1	Developer – Integration specialist	1 year
EL2	Project Manager	2 years
EL3	Scrum Master	7 years
EL4	Developer – with multiple roles	20 years
EL5	Manager	10 years
EL6	Scrum master	5 years
EL7	Business analyst – with multiple roles	3 years

The interviews were recorded via the Zoom application, each lasting 1–2 hours. These interviewees performed different roles and some have multiple responsibilities. Table 1 shows that interviewees were from diverse roles, including software developers, project managers, Scrum masters, and business analyst.

Out of seven participants, four participants have more than five years of experience, whereas remaining three interviewees have 1 - 3 years of working experience. All of the study participants had practical agile methods experiences. The interview questions were based on four broad areas: professional background; agile way of working in their team; communication, collaboration and knowledge-sharing practices; and teamwork environment. Each transcript was meticulously analysed by means of line-by-line reading to identify key points. Each identified point was recorded as an open code, which went through an iterative process of comparison throughout the analysis. Such techniques help researchers to check and compare a new code against the previously identified ones [23].

IV. RESULTS

The results of this study are presented in four sub-sections. First, the significance of leaders' behaviour as a key antecedent of fostering psychological safety is emphasised. Second, the importance of leaders' formation of teams with a focus on learning is highlighted. Third, the need to cultivate trusting and respectful interpersonal relation-ships, both inside and outside ASD teams, is noted. Lastly, the findings reveal that a lack of psychological safety within a company contributes to brain drain, highlighting the con-sequences of neglecting this essential aspect of organisational culture. The

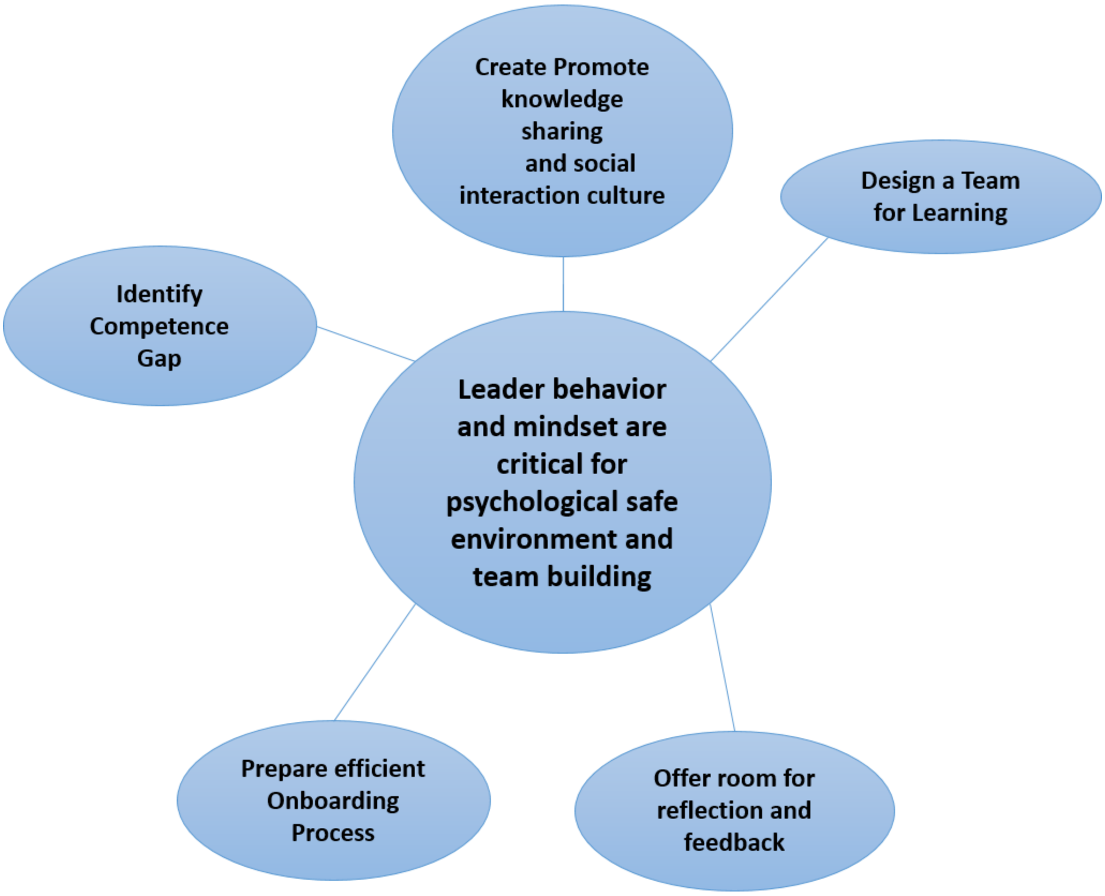

Create Promote knowledge sharing and social interaction culture
Design a Team for Learning
Identify Competence Gap
Leader behavior and mindset are critical for psychological safe environment and team building
Prepare efficient Onboarding Process
Offer room for reflection and feedback

Fig 1. Leadership role to enable psychological safety environment for LSA Software Development

Fig. 1 shows leadership role to enable psychological safety environment, whereas Fig. 2 shows its effects in LSA software development projects context.

A. Leader behaviour

The leaders' role is critical for preparing a good work environment where all team members feel comfortable in expressing their thoughts and ideas without fear of criticism or retribution. An interviewee said that paying attention to both customer and employee satisfaction should be the focus:

"*Our top focus is to be a great place to work and to have happy employees* and *at the same time, be a customer-centric organisation. That's where the agility mindset comes into play because if we have happy customers, then we normally deliver good value to our other stakeholders*" (EL5).

Establishing a psychologically safe environment is important as it gives individuals the confidence to speak up if they are unhappy about something.

"*I haven't felt at least that people are afraid to speak their minds because people do speak their minds if they are unhappy with something*" (EL2).

In such a safe environment, team members can engage in enjoyable activities outside of work, which can foster stronger relationships and promote team cohesion. An interviewee expressed the scenario as follows:

"*We do a lot of things together, like* [when we're] *off works and stuff. So, it's fun*" (EL1).

However, establishing such a safe environment comes with its challenges. An interviewee highlighted the hidden fear of openly admitting their team's mistakes or underperformance of activities:

"*It's very helpful to reflect and always keep in mind what we can do better and change next time. It is a problem to tell the truth if* [there is] *something that you think did not go as well as you wanted to. Maybe I can be a little bit scared to tell the whole truth. I think that could be a problem*" (EL1).

B. Trusting and respectful interpersonal relationships

When the work environment is respectful and trustworthy, the results are always positive despite strict time constraints or deadlines. This is very important in software development as it is a knowledge-intensive activity with many discussions around customers' demands. A psychologically safe environment enables difficult conversations without the need to tiptoe around the truth. In the case company, the team members were supportive and helpful towards one another, and they often worked collaboratively:

"*If I have a problem, I'm never alone. If I do not act alone, I need to ask someone. If I do that, they will always help me. We do a lot of funny things together. I do not think*

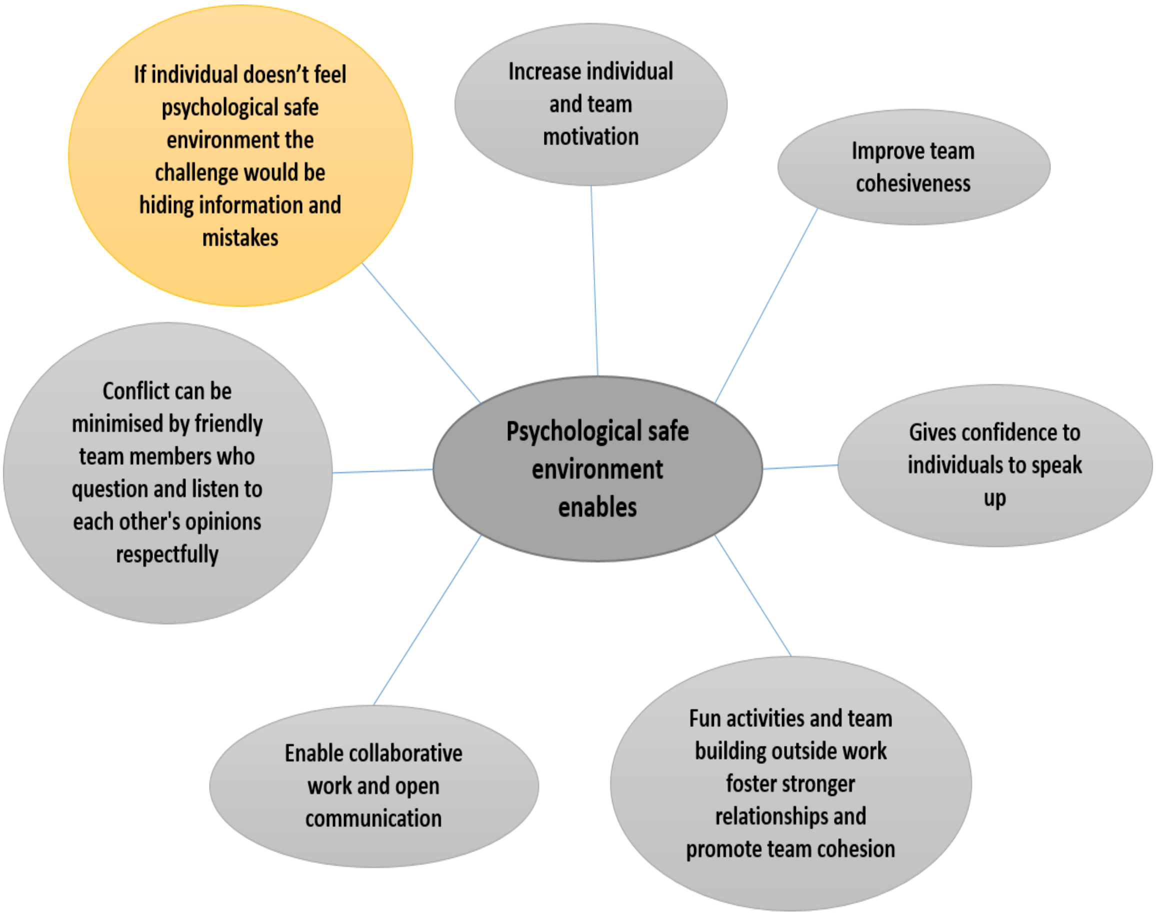

If individual doesn't feel psychological safe environment the challenge would be hiding information and mistakes
Increase individual and team motivation
Improve team cohesiveness
Conflict can be minimised by friendly team members who question and listen to each other's opinions respectfully
Psychological safe environment enables
Gives confidence to individuals to speak up
Enable collaborative work and open communication
Fun activities and team building outside work foster stronger relationships and promote team cohesion

Fig 2. Psychological safety environment and its effects for LSA software development

that we have had any conflicts. It is a really good team, and we are really supporting and lifting up each other. So, I think that everyone can see that our team is very friendly" (EL4).

It can also be a sign of team cohesiveness when everyone works towards the same goal. It is noteworthy that being friendly does not mean that professional activities are overlooked. Listening to each team member's opinions is essential to avoid conflicts, as an interviewee explained:

"*...friendly team is something that we do not have to be too much friends with. We question each other and listen to each other's opinions. I think that one of the reasons why we do not have this conflict. We are very friendly and we don't want to fire our voices if that can lead to a conflict*" (EL4).

Furthermore, it is important that meetings and discussions at the workplace should not be only work-related. It should also offer an opportunity for team members to connect and engage with one another on a social level.

"*Suppose there are problems, how people tell if they're satisfied and if they're ill or something. So it's just like a social meeting as well, not just work*" (EL3).

C. Brain drain

Brain drain occurs when employees feel that the work conditions are too demanding or feel a lack of intrinsic motivation or stimulation from their work and a lack of remuneration safety. Attraction and retention of skilled employees require competitive remuneration packages, a safe environment and a collaborative corporate culture. It is reasonable to assume that when employees feel safe psychologically and remuneratively, they are more likely to stay with the company. When a good employee leaves, it has an immediate effect on the team members and their work. An interviewee highlighted such skills gap:

"*He was a really good programmer, and we miss him a bit*" (EL6).

"*To be brutally honest, I know that some people left last year because they thought that they weren't getting paid enough*" (EL2).

However, employees' resignation from the company is not just due to the salary; it is a multidimensional factor, for example, caused by the nature of work, the place of work and some personal reasons. A senior team member expressed his observation:

"*One of the reasons that I've heard is that it's about the salary. That is important, and* [the income] *can differ if we live here* [a Scandinavian country's capital] *permanently as well. [The reasons for quitting] can depend on a lot of things. Some of my closest colleagues and I discuss. Sometimes, it feels like* [employees] *leave* [after just a short time]. *They understand things differently than what has actually happened to them*" (EL4).

When developers have no internal drive or interest in their work, it can be challenging to retain these less satisfied employees. Intrinsic motivation comes from activities that an individual finds enjoyable or stimulating, even without external rewards. Multiple interviewees highlighted such lack of exciting work, for example:

"*A risk or a factor that people want to live. It is not fun to have too much* [work] *to do and not fun* [activities] *to do; actually* [it is] *always tricky*" (EL1).

"*The most common reason is that they get another job offer on something they really want to work with. It has nothing to do with our company. It's more like they're going to work with something they appreciate more*" (EL7).

According to another interviewee, individuals leave their jobs because they feel that the work conditions are unsatisfactory and their work is too demanding.

"*They think that the conditions are not good enough. [They] are addicted to having projects; maybe like some months, they have much to do. And then for half a year, it's hard to get projects that they're used to, so they don't have much to do, and they feel under-stimulated and like, 'I want a new job so I can have more tasks to do'*" (EL1).

D. Designing a team for learning

Organisations need to develop the idea of designing teams for learning, which involves multiple factors such as an efficient onboarding process, creating a culture of continuous learning and knowledge sharing, reflection and feedback and so on. The starting point for designing a team for learning is to identify the competence development gap and then a good onboarding process for new employees. A point of caution in the onboarding process is whom to involve and when to be involved. When the process includes only senior members of the company, it becomes stressful for them. An interviewee expressed this situation as follows:

"*We have found that in the management team with the overall responsibility for competence development, that kind of role is sort of missing at the moment. We were working with scale agile. We had to fill in who would be responsible for each role, and we saw that overall competence development was lacking. We have an operation manager who is leading operations but maybe not clearly responsible for competence development*" (EL5).

"*Onboarding is a big issue that we have to work with and maybe not just talk about it. We have to take care of the people who are here and remain with a nice spirit. I think that it is important because if we get bigger and bigger, it puts a lot of pressure on senior consultants, and we have to be careful of them*" (EL3).

Software development is a knowledge-intensive activity, where knowledge sharing is an important element. An interviewee expressed positive experiences about their safe environment for sharing knowledge:

"*We have quite recently started knowledge-sharing sessions. One person in our team is responsible for administering these meetings and setting up the agenda. So, we are starting to work with it as it is in our spring planning to have knowledge sharing and talk about how we can do things better. It's an initiative from me or my colleagues who are the value-stream managers*" (EL3).

Along with the work practices, another aim was to establish good enterprise social media platforms (e.g., Slack, Microsoft Teams, etc.) for internal communication and social interaction within the company.

V. Limitations and threats to validity

We collected the data from interviews with software professionals from a Scandinavian software company. All the codes and concepts were directly obtained from the interviews. Our findings are sufficiently grounded in the substantive data [23] but cannot be generalised on a large scale due to the limited number of participants. Therefore, caution should be taken when applying these results to other software companies. The inherent limitation of the GT is that it is only based on a particular investigative context [25].

James and Busher [26] highlighted the risk regarding the authenticity of the participants in digital interviews. We were confident that all the participants were interviewed with the permission of the company representative and with a signed NDA. In this way, such risks are mitigated in this study. The GT approach used in this study involved subjective interpretation of the data. The findings and the emerging concepts presented in this study are based on the researchers' interpretation of the data, which may differ from other researchers' interpretations. Despite these limitations, this study's findings offer valuable insights into psychological safety, leadership and NTD in LSA development. The concepts are sufficiently supported with quotations from the participants' interviews, and the findings are discussed in detail and characterised by some existing concepts.

VI. Discussion and conclusion

In this study, we explored the antecedents and effects of psychological safety on LSA teams. The results suggest that building a psychologically safe environment is a multi-dimensional factor that requires a proactive leadership approach, a competent team design that focuses on learning, open communication and feedback, remuneration safety, and a well-prepared onboarding process for new team members. These factors contribute to effective teamwork, work satisfaction and learning, as well as promote a safe and collaborative learning environment.

Our results show that a psychologically safe environment can be enabled through enjoyable activities outside of work that also foster stronger relationships and promote team cohesion that is less prone to conflicts. In a software development project, team cohesion magnifies the impact of psychological safety on knowledge sharing [27]. Psychological safety directly contributes to effective [28]. The lack of psychological safety contributes to social and people's debts [8], [14], whereas a high level of psychological safety has significant positive correlations to LSA team performance [7] and the success of process innovations [17], mitigating effects of the lack of both communication and collaboration [8].

To create a more psychologically safe environment, leaders and the management should show appreciation for employees [19] and provide opportunities for their involvement in projects so that they can learn from their mistakes and failures. Detert and Burris [39] revealed a positive correlation between the managers' openness and transformational leadership with psychosocial safety. The management needs to identify competence gaps and design teams for learning. It is also evident that providing room for reflection and open feedback is important; otherwise, individuals or teams will hide their real troublesome situations. High-quality interpersonal relationships among the team members enhance their psychological security, leading to positive and effective learning [16, 32] and sharing behaviours [29]. Trust and organisational support are enablers of psychological safety at work [16]. Our results show that trusting and respectful interpersonal relationships in LSA teams help avoid conflicts and prepare a breeding ground for a safe and collaborative learning environment. Dreesen et al [14] and Ahmad et al [2] reported that lack of psychological safety in software development contributes to social debt.

A key manifestation of high-quality relationships is relational coordination, along with shared goals, shared knowledge and mutual respect [30], [31]. Relational coordination is "*a mutually reinforcing process of interaction between communication and relationships carried out for the purpose of task integration*" [30](p. 301). This is more important in software development as this knowledge-intensive activity requires creativity in solving a particular problem and completing a task. This recommendation would enable double-loop learning instead of a single loop in teams.

Agile teams strive for continuous improvement through recurrent feedback and introspection [40]. Psychological safety cultivates an environment wherein team constituents are enabled to provide and accept valuable feedback and facilitate learning [34]. While multiple studies in the realm of social sciences demonstrate that PS cultivates learning-oriented actions like soliciting feedback, experimentation, and deliberation of mistakes [6], [9], its pivotal role in LSA has not been recognised. The nature of ASD entails intricate knowledge that is collectively shared among multifaceted teams and evolves swiftly.

Edmondson highlighted psychosocial safety importance and its impact on learning within team and performance [6]. It is also important to design teams for learning and have a well-prepared onboarding process for new employees. It is vital to know whom to involve and when to involve senior members of a team or a project because it builds pressure on senior consultants and creates the need to take care of existing teams while remaining focused on company growth. It is interesting to note that despite the psychologically safe environment, the company noticed a brain drain due to the lack of both remuneration safety and intrinsic motivation and an office location far from the employees' families and friends. These findings highlight the importance of providing remuneration safety and addressing the factors that may lead to employee turnover. More research is needed to explore these issues in greater depth and develop strategies for solving them.

Future research should prioritise investigating the relations among organisational culture, employee diversity and psychological safety, as well as their impacts on team building, job satisfaction and performance. This includes addressing the reasons why knowledgeable team members leave the company. Another important area of study involves identifying competence gaps and designing teams that prioritise learning, including effective methods for assessing individual and team competencies, strategies for fostering continuous learning, proactive leadership and the significance of providing opportunities for reflection and open feedback. Additionally, exploring the influence of social activities on team cohesion and conflict prevention can provide insights into fostering strong relationships. It is also essential to investigate factors beyond psychological safety, such as remote work, intrinsic motivation and office location proximity to family and friends, in order to mitigate their impacts, prevent brain drain and enhance employee satisfaction and loyalty while creating inclusive and supportive environments for LSA teams and projects.

In conclusion, this study offers the above-mentioned recommendations. Overall, a psychologically safe environment can foster increased confidence, collaboration, communication, motivation and job satisfaction of individuals and teams. Finally, exploring both task and social cohesion regarding learning and performance in LSA teams and projects would be of interest. We posit that future research has the potential to delve into the impacts and advantages of psychological safety on the intricacies of agile teams and their effectiveness to develop and deliver software products. In light of this, we intend to expand upon the present study by examining the influence of psychological safety on software quality, individual and team learning.

Acknowledgment

This work was supported by the KKS foundation, Sweden through the NODLA project and Helge Ax:son Johnsons Stiftelse.

References

[1] S. Beecham, N. Baddoo, T. Hall *et al.*, "Motivation in software engineering: A systematic literature review," *Information and Software Technology*, vol. 50, no. 9–10, pp. 860–878, 2008. https://doi.org/10.1016/j.infsof.2007.09.004.

[2] M. O. Ahmad and T. Gustavsson, "The Pandora's box of social, process, and people debts in software engineering," *Journal of Software: Evolution and Process*, 2022. doi.org/10.1002/smr.2516.

[3] P. Lenberg and R. Feldt, "Psychological safety and norm clarity in software engineering teams," Workshop on Cooperative and Human Aspects of Software Engineering. pp. 79–86. 2018. DOI: 10.1145/3195836.3195847.

[4] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87–108. 2016. https://doi.org/10.1016/j.jss.2016.06.013

[5] M. Kalenda, P. Hyna, and B. Rossi, "Scaling agile in large organizations: Practices, challenges, and success factors," *Journal of Software: Evolution and Process*, vol. 30, no. 10, pp. e1954, 2018. https://doi.org/10.1002/smr.1954

[6] A. Edmondson, "Psychological safety and learning behavior in work teams," *Administrative Science Quarterly*, vol. 44, no. 2, pp. 350–383, 1999. https://doi.org/10.2307/2666999.

[7] T. Gustavsson, "Team performance in large-scale agile software development," in *Advances in Information Systems Development: Crossing Boundaries Between Development and Operations in Information Systems*, Springer, 2022, pp. 237–254.

[8] M. O. Ahmad, I. Ahmad, and F. Qayum, "Early career software developers and work preferences in software engineering*," Journal of Software: Evolution and Process*, e2513, 2022. https://doi.org/10.1002/smr.2513

[9] A. Newman, R. Donohue, and N. Eva, "Psychological safety: A systematic review of the literature," *Human Resource Management Review*, vol. 27, no. 3, pp. 521–535, 2017. https://doi.org/10.1016/j.hrmr.2017.01.001

[10] L. Delizonna, "High-performing teams need psychological safety: Here's how to create it." https://hbr.org/2017/08/high-performing-teams-need-psychological-safety-heres-how-to-create-it

[11] U. Safdar, Y. F. Badir, and B. Afsar, "Who can I ask? How psychological safety affects knowledge sourcing among new product development team members," *Journal of High Technology Management Research*, vol. 28, no. 1, pp. 79–92, 2017. https://doi.org/10.1016/j.hitech.2017.04.006

[12] Agile-Manifesto, "Manifesto for agile software development." https://agilemanifesto.org/

[13] H. Edison, X. Wang, and K. Conboy, "Comparing methods for large-scale agile software development: A systematic literature review," *IEEE Transactions on Software Engineering*, 2021. DOI: 10.1109/TSE.2021.3069039

[14] T. Dreesen, P. Hennel, C. Rosenkranz *et al.*, "The second vice is lying, the first is running into debt." *Antecedents and Mitigating Practices of Social Debt: An Exploratory Study in Distributed Software Development Teams*, HICSS. 2021. DOI: 10.24251/HICSS.2021.818

[15] J. R. Detert, and E. R. Burris, "Leadership behavior and employee voice: Is the door really open?," *Academy of Management Journal*, vol. 50, no. 4, pp. 869–884, 2007. https://doi.org/10.5465/amj.2007.26279183

[16] A. C. Edmondson, R. M. Kramer, and K. S. Cook, "Psychological safety, trust, and learning in organizations: A group-level lens," *Trust and Distrust in Organizations: Dilemmas and Approaches*, vol. 12, no. 2004, pp. 239–272, 2004.

[17] M. Baer and M. Frese, "Innovation is not enough: Climates for initiative and psychological safety, process innovations, and firm performance," *Journal of Organizational Behavior*, vol. 24, no. 1, pp. 45–68, 2003. https://doi.org/10.1002/job.179.

[18] A. Carmeli and J. H. Gittell, "High-quality relationships, psychological safety, and learning from failures in work organizations," *Journal of Organizational Behavior*, vol. 30, no. 6, pp. 709–729, Aug. 2009. https://www.jstor.org/stable/41683863

[19] I. M. Nembhard and A. C. Edmondson, "Making it safe: The effects of leader inclusiveness and professional status on psychological safety and improvement efforts in health care teams," *Journal of Organizational Behavior*, vol. 27, no. 7, pp. 941–966, 2006. doi.org/10.1002/job.413

[20] P. Hennel and C. Rosenkranz, "Investigating the "socio" in socio-technical development: The case for psychological safety in agile information systems development," *Project Management Journal*, vol. 52, no. 1, pp. 11–30, 2021. DOI: 10.1177/8756972820933057.

[21] M. A. Valentine and A. C. Edmondson, "Team scaffolds: How mesolevel structures enable role-based coordination in temporary groups," *Organization Science*, vol. 26, no. 2, pp. 405–422, 2015. https://doi.org/10.1287/orsc.2014.0947

[22] O. A. O'Neill, "Workplace expression of emotions and escalation of commitment," *Journal of Applied Social Psychology*, vol. 39, no. 10, pp. 2396–2424, 2009. doi.org/10.1111/j.1559-1816.2009.00531.x.

[23] B. G. Glaser, *Basics of Grounded Theory Analysis: Emergence vs Forcing*. Sociology Press, 1992.

[24] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research: A critical review and guidelines," pp. 120–131. DOI: 10.1145/2884781.2884833.

[25] S. Adolph and P. Kruchten, "Summary for scrutinizing agile practices or shoot-out at process corral!," pp. 1031–1032. 2008. https://doi.org/10.1145/1370175.1370232.

[26] N. James and H. Busher, "Credibility, authenticity and voice: Dilemmas in online interviewing," *Qualitative Rresearch*, vol. 6, no. 3, pp. 403–420, 2006.

[27] A. K. Kakar, "How do team cohesion and psychological safety impact knowledge sharing in software development projects?," *Knowledge and Process Management*, vol. 25, no. 4, pp. 258–267, 2018. DOI: 10.1002/kpm.1584.

[28] C. Verwijs and D. Russo, "The double-edged sword of diversity: How diversity, conflict, and psychological safety impact agile software teams," arXiv preprint arXiv:2301.12954, 2023.

[29] A. Carmeli, D. Brueller, and J. E. Dutton, “Learning behaviours in the workplace: The role of high-quality interpersonal relationships and psychological safety,” *Systems Research and Behavioral Science*, vol. 26, no. 1, pp. 81–98, 2009. https://doi.org/10.1002/sres.932.

[30] J. H. Gittell, “A theory of relational coordination,” in *Positive Organizational Scholarship: Foundations of a New Discipline*, K. S. Cameron, J. E. Dutton, and R. E. Quinn, Eds. San Francisco: Berrett-Koehler Publishers, 2003, pp. 279–295.

[31] J. H. Gittell, “Relational coordination: Coordinating work through relationships of shared goals, shared knowledge and mutual respect,” in *Relational Perspectives in Organizational Studies: A Research Companion*, pp. 74–94. 2006. DOI: 10.1002/9781118785317.weom110025.

[32] W. A. Kahn, “Psychological conditions of personal engagement and disengagement at work,” *Academy of Management Journal*, vol. 33, no. 4, pp. 692–724, 1990. https://doi.org/10.2307/256287.

[33] A. Alami, M. Zahedi, and O. Krancher, “Antecedents of psychological safety in agile software development teams,” *Information and Software Technology*, 107267, 2023. doi.org/10.1016/j.infsof.2023.107267.

[34] A.C. Edmondson, and Lei , Z. Psychological safety: The history, renaissance, and future of an interpersonal construct . Annual Review of Organizational Psychology and Organizational Behavior 1(1), pp: 23–43. 2014. https://doi.org/10.1146/annurev-orgpsych-031413-091305

[35] B. Boehm, R. Turner. Management challenges to implementing Agile processes in traditional development organizations. IEEE Software. 2005. 22(5): 30–39. https://doi.org/10.1109/MS.2005.129

[36] I.M. Nembhard, and A.C. Edmondson. Making it safe: The effects of leader inclusiveness and professional status on psychological safety and improvement efforts in health care teams. Journal of organizational behavior 27 (7): 941 – 966. 2006. https://doi.org/10.1002/job.413

[37] K.A. Jehn, and E.A, Mannix. The dynamic nature of conflict: A longitudinal study of intragroup conflict and group performance. Academy of Management Journal 44 (2): 238 – 251. 2001. https://doi.org/10.2307/3069453

[38] S. Thorgren, and E, Caiman. The role of psychological safety in implementing agile methods across cultures. Research-Technology Management, 62(2), 31-39. 2019. https://doi.org/10.1080/08956308.2019.1563436

[39] Detert, J. R., & Burris, E. R.. Leadership behavior and employee voice: Is the door really open?. Academy of management journal, 50(4), 869-884. 2007. https://doi.org/10.5465/amj.2007.26279183

[40] Alami, A., Krancher, O., & Paasivaara, M. The journey to technical excellence in agile software development. Information and Software Technology, 150, 106959. 2022. https://doi.org/10.1016/j.infsof.2022.106959.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 335–346

DOI: 10.15439/2023F6950
ISSN 2300-5963 ACSIS, Vol. 35

Comparative Analysis of Word Embedding and Machine Learning Techniques for Classification of Software Developer Communications on Gitter

Tumu Akshar[1]
Department of Computer Science & Information Systems
BITS Pilani Hyderabad Campus
f20200003@hyderabad.bits-pilani.ac.in

Lov Kumar[2]
Department of Computer Engineering
National Institute of Technology, Kurukshetra
lovkumar@nitkkr.ac.in

Yogita[3]
Department of Computer Engineering
National Institute of Technology, Kurukshetra
yogita@nitkkr.ac.in

Lalita Bhanu Murthy[4]
Department of Computer Science & Information Systems
BITS Pilani Hyderabad Campus
bhanu@hyderabad.bits-pilani.ac.in

***Abstract*—In recent times, software developers widely use instant messaging and collaboration platforms, as these platforms aid them in exploring new technologies, raising different development-related issues, and seeking solutions from their peers virtually. Gitter is one such platform that has a heavy userbase. It generates a tremendous volume of data, analysis of which is helpful to gain insights about trends in open-source software development and the developers' inclination toward various technologies. Analyzing these trends helps these platforms better cater to the needs of the developers, in turn increasing the usage of these platforms and promoting collaborations between more developers. The classification techniques can be deployed for this purpose. The selection of an apt word embedding for a given dataset of text messages plays a vital role in determining the performance of classification techniques. In the present work, the comparative analysis of nine-word embeddings in combination with seventeen classification techniques with onevsone and onevsrest has been performed on the GitterCom dataset for categorizing text messages into one of the pre-determined classes based on their purpose. Further, two feature selection methods have been applied. The SMOTE technique has been used for handling data imbalance. It resulted in a total of 1836 classification pipelines for analysis. The objective is to analyze their performances to recommend efficient pipelines for the classification task at hand. The experimental results show that word2vect, GLOVE with 300 vector size, and GLOVE with 100 vector size are three top-performing word embeddings having performance values taken across different classification techniques. The models trained using ANOVA features performed similarly to those models trained using all features. Finally, using the SMOTE technique helps models to get a better prediction ability.**

***Index Terms*—Functional Requirements, Non-Functional Requirements, Deep Learning, Data Imbalance Methods, Feature Selection, Classification Techniques, Word Embedding.**

I. Introduction

THE development of modern complex software systems requires a lot of meticulous planning and large teams of software developers and designers. The members of these teams are often geographically distributed across various locations and rely on online communication and collaboration modes. Many open-source projects and software development teams have shifted towards platforms such as Gitter and Slack due to the features and the support they offer for collaboration between software developers worldwide.Gitter has revolutionized team communications and project coordination, especially for distributed software development teams, by providing a user-friendly way of managing and organizing conversations. Gitter also provides public access to user-generated data, and the historic data is accessible indefinitely through chat room logs. Classifying the purpose of the messages in the developer communications on such platforms assists in better organization of messages into categories, making them easy to be retrieved by the users as per their requirements and deriving various insights into the general trends in open-source software development. This classification also helps understand the major reasons why people use these platforms for. Then, the platforms can be updated to fulfill the requirements of the users, which attracts more people to use these platforms, promoting better collaborations, meaning both the organizations running the platforms and the developers benefit from this. Manual classification of this data is not feasible due to the sheer volume of the data available. So there is a need to deploy Machine Learning (ML) techniques to automate the process and minimize the errors in classification.

The objective of the present work is to perform a comparative analysis of the classification pipelines developed using nine different word embeddings, two feature selection techniques, and thirty-four classifiers. We believe this analysis will help establish a strong foundation for future researchers to select the techniques and pipelines resulting in the best predictive ability of the developed models to identify the purpose of messages in developer communications. For this purpose, GitterCom dataset, which contains around 10,000 messages from various channels on Gitter has been adopted.

Each message in it has three labels viz. Purpose, Category, and Sub-Category, which the dataset creators curated manually. The process of creation of classification pipelines started by applying nine different word-embedding techniques to vectorize the data into a numeric form for further analysis. These word-embedding techniques generate an abundance of features, not all of which are influential in the classification task. Hence, two feature selection techniques, namely One-Way ANOVA and PCA are employed to synthesize or extract the most relevant and influential features to optimize the performance of the models. Then to test the predictive ability of the word-embedding and feature-selection techniques, we have used seventeen different variants of classification techniques with one-vs-one and one-vs-rest approaches. The classification techniques are used by considering various ML and Neural Network (NN) classification algorithms and ensemble learning methods in association with them. The predictive ability of these classifiers was validated using the 5-fold cross-validation method. The SMOTE data sampling technique is also applied to combat data imbalance, leading to incorrect and unreliable results. Finally, the performance of the developed models has been compared using evaluation metrics such as Accuracy, Sensitivity, Specificity, and Geometric Mean, which are extracted using the Box-plot Visualization technique and the Friedman Test.

II. Related Work

The proposed research is based on classifying and analyzing the purpose or rationale behind developer communications on modern instant messaging and communication tools. The developers use it for project coordination or technological discussions. This research also performs a comparative analysis of various NLP models that best analyze the communications on such platforms. In this work, we have used the 'GitterCom' dataset introduced by Parra et al. [1], a manually created dataset of 10,000 messages from developer discussions on the Gitter Platform.

The related work is divided into two sub-sections. The first sub-section focuses on the earlier works that have studied developer communications on instant messaging platforms such as Slack or Gitter. The second sub-section is dedicated to the analysis of messages on other platforms such as Q&A platforms (Eg: Stack Overflow), Social Media Platforms (Eg: Twitter), etc. In both sections, we perform a critical comparison of the previous endeavors with our research approach.

A. Instant Messaging and Communication Tools for Software Developers

Earlier research works performed an empirical study of the properties of the communications on Gitter and Slack. While some of these works focused on topics discussed and problem resolution, others introduced certain classes to classify the purpose or rationale of the messages in such communications. The survey confirmed that various classification techniques were used to identify the topic of discussion or the purpose and rationale of the messages. Ehsan et al.[2] performed an empirical study of developer discussions in Gitter to understand the nature of developer discussions, the type of questions developers ask, and the proposed solutions. The authors proposed an approach for the automatic identification of discussion threads in developer chatrooms using hierarchical clustering algorithms and other heuristics. They identified four patterns of responses based on the response length and complexity. The authors also created a taxonomy of resolution types and discussed topics in the chatrooms. The topics discussed were classified into five types - Installation and configuration, Debugging and troubleshooting, Feature requests and enhancements, Code review and feedback, and General discussion.

Sahar et al. work focused on the analysis of how developers discuss issue reports in Gitter chat rooms related to open-source systems [3]. The authors proposed an approach involving clustering algorithms to identify issue report discussions automatically. These discussions were classified into four types based on the number of messages, participants, and duration of the discussion. In our work, we employ various word embedding techniques to assess which technique captures these key patterns and strategies the best, aiding the classification performance. The above two works broadly focus on identifying the topics of discussion and defining a classification nomenclature based on it. Our work focuses on a possible next step of their research, which is to build and compare Machine Learning pipelines that can automatically classify present and incoming messages based on the considered classification nomenclature.

A study with a research methodology similar to ours was done by Parra et al. [4]; their objective was built upon their previously published paper (Parra et al.[1]) in which they introduced the GitterCom dataset for the first time. The three categories defined to classify each message in the developer discussions based on its purpose in this paper (team-wide, personal benefits, community support) were derived from the work of Lin et al. for the Slack platform [5]. The authors also analyzed the performances of nine supervised machine learning and deep learning algorithms, such as SVM, Decision Trees, AdaBoost, LSTM, etc., along with certain data sampling techniques to classify the purpose of a given message as one of the pre-defined categories. A key difference between our work and theirs is the analysis of the benefits of various word embedding techniques in better capturing the contexts and patterns of the messages which aid the classification. We also explore the possible effects of feature selection techniques since they help reduce the complexity of the models while retaining the key features of the data.

The analysis of the topics and discussion and purpose of messages was also undertaken for Stack Overflow. Lin et al. performed an empirical study to understand the purposes for which developers use Slack [5]. Upon surveying 53 software developers, the authors found that the developers self-reported using Slack for various purposes. They broadly classified these purposes into personal benefits, community support, and team-wide purposes. Another such empirical study was undertaken by Stray et al. [6]. The authors explored the use of Slack for communication and project coordination in virtual agile development teams as they studied the communications of a group of 30 software developers at a large software development organization. They found that the messages could be broadly classified into one of the following purposes: general information/coordination, general discussions, problem-focused communication, technical communication, and socializing.

Alkadhi et al. performed an exploratory study to investigate the presence of rationales in chat messages during software development [7]. They collected and analyzed chat messages from three open-source software development teams on GitHub comprising students working on capstone projects. Based on this analysis, the authors defined a set of categories for the types of rationale found in the messages - Design, Functionality, Implementation, and Maintenance. Their findings also indicate the usefulness of classification algorithms such as SVM and Naïve Bayes and the data sampling techniques for automatically identifying and classifying messages based on the rational information they contain. Our work, however, focused on recommending the best techniques and pipelines for building such models to extract the best performance out of them rather than just checking for feasibility. In subsequent work, Alkadhi et al. presented a new approach called REACT (Rationale Extraction from Chat Transcripts) [8]. They considered five rationale elements for their work - issues, alternatives, pro-arguments, con-arguments, and decisions. The REACT approach enabled the developers to explicitly record the rationale in messages on the Slack platform through manual annotation.

B. Other Communication Tools for Software Developers

Various other platforms facilitate software development communication. The concept of such tools is based on Question & Answer forums like Stack Overflow and social media platforms like Twitter. Software developers spread out across various geographic locations relied majorly on such platforms for communication and coordination before the advent of the modern instant messaging tools dedicated to facilitating such discussions. Q&A platforms such as Stack Overflow encourage efficient problem-solving, community support, and knowledge sharing among its users across the globe. Social Media sites like Twitter also help in this regard by providing real-time updates, networking opportunities, and crowd-sourced solutions through hashtags and mentions to reach out to relevant experts for assistance.

The current tags provided for questions on Stack Overflow are based on the technologies used or being referred to in the question. Classifying these questions based on the purpose or context will serve the users better since it makes the process of finding relevant posts quicker. Beyer et al. performed such work where they obtained a taxonomy of seven question categories: API change, API usage, Conceptual, Discrepancy, Learning, Errors, and Review, and they manually curated a dataset consisting of 500 posts classified into these categories [9]. The authors then developed classification models using the Random Forest and SVM algorithms along with data sampling techniques and performed a comparative analysis with 82 different configurations regarding the preprocessing and representation of the input text data to analyze which configuration captured the context and intricacies of the data better. In place of such configurations, we explored word embeddings for this reason since they offer better generalizability to out-of-vocabulary words as they provide continuous representations of words by considering semantic equivalence, something which pre-processing techniques might struggle with.

Venigalla et al. undertook a similar study using the Latent Dirichlet Allocation Algorithm to model six questions topics based on their purpose [10]. The authors also came up with names for those topics based on the taxonomy used in the literature. They also used various Machine Learning algorithms, such as Linear SVC, Logistic Regression, Multinomial Naive Bayes, Random Forest, etc., to develop classification models for the questions on Stack Overflow.

Guzman et al. performed an analysis of the tweets on Twitter related to software applications[11]. The intention was to obtain tweets that could be useful for developers by using classification algorithms. The purpose of these tweets includes improvement suggestions, user needs, bug reports, feature requests, etc. The authors introduced ALERTme, an approach to automatically classify, group, and rank such tweets. This approach relied on classification algorithms such as Multinomial Naive Bayes and word-embedding techniques such as TF-IDF for classification.

We preferred Gitter over such platforms because Gitter's data has been largely untapped for such analysis, and Gitter provides more intricate and essential details of software development pipelines and requirements as developers use it regularly to discuss such details.

III. STUDY DESIGN

This section presents the details regarding various design setting used for this research.

A. Experimental Dataset

We plan to use the GitterCom dataset for this experiment, the first manually labeled dataset of Gitter instant message histories in open-source systems. It contains 10,000 messages collected from 10 open-source software development Gitter communities (1,000 messages per community). Each message in this dataset was manually labeled to capture the purpose of the communication expressed by the message. Each record in the dataset consists of the following information: (i) The channel of communication (ii) the Message ID (iii) The date and time when the message was posted (iv) The author of the message (v) The message in plain text (vi) Purpose of the message (vii) The subclass of the purpose it belongs to - category (viii) The subclass of the category it belongs to i.e., the sub-category.

In this experiment, we intend to predict the Purpose label for any given message. There are three possible classes (labels) of purpose:

- **Personal Benefits** - Includes messages posted to satisfy the developer's personal needs.
- **Team-wide activities** - Includes messages related to the discussion among the members of a team working on software development activities related to the system they are developing.
- **Community Support** - Includes messages posted for general discussions among developers from communities or special-interest groups who intend to explore new tools and frameworks or brainstorm ideas.

B. Word Embeddings

The textual data of the dataset is to be represented in numeric vector form for further analysis. To achieve this, nine different word embedding techniques, including Continuous Bag of Words (CBOW), Skip-Gram (SKG), Global Vectors for Word Representation (GloVe) (with 50 dimensions, 100 dimensions, and 300 dimensions), Word2vec, fastText (FST), Generative Pre-trained Transformer (GPT), and Generative Pre-trained Transformer-2 (GPT-2), will be applied on the dataset. These techniques help represent the textual data as a vector in an n-dimensional space. All the stopwords, spaces, and other bad symbols will be removed from the textual data before applying the word embedding techniques. The generated vectorial representations will then be used to develop models to determine a message's purpose. [12].

C. Feature Selection Techniques

Since vectorial representations are used as inputs to develop the classification models, and each of the vectorial representations contains a lot of columns (features), some of which may not be as influential as the others, optimization of the data and selection of the influential features becomes crucial to improve the performance of the models. To extract the important features, we plan to use two different Feature Selection Techniques - One-Way Analysis of Variance (One-Way ANOVA) and Principal Component Analysis (PCA) to discard irrelevant features and obtain the set of relevant and influential features.

D. Training of Models from Imbalanced Data Set

Upon analysis of the dataset, if we observe the problem of class imbalance in our dataset, i.e., the number of data samples in each class is different, we resort to the data sampling techniques to enhance the performance of the developed models. We propose to use the Minority Oversampling Technique (SMOTE) on the dataset to balance the data in such a scenario.

E. Classification Techniques

We propose to develop thirty-four different classifiers to perform a comparative analysis of the developed models. For that, classification algorithms such as Multinomial Naive Bayes (MNB), Bernoulli's Naive Bayes (BNB), Gaussian Naive Bayes (GNB), Decision Tree Classifier, Logistic Regression, K-Nearest Neighbours Classifier, Random Forest Classifier, Extra Trees Classifier, Multi-Layer Perceptron with Limited-memory BFGS, Stochastic Gradient Descent, and Adam Optimizers will be utilized along with ensemble modeling techniques such as Bagging with KNN, Multinomial Naive Bayes, Logistic Regression and Decision Tree Classifiers, Ada Boost Classifier and Gradient Boosting Classifier. Each classification technique mentioned above will be implemented using one-vs-Rest and One-vs-One Multi-class classification strategies, thus giving a total of thirty-four classifiers. These classification strategies are defined below:

- **One-vs-One classification**: The model trains on pair-wise comparisons between each possible combination of classes. The final prediction is made by aggregating the votes from all binary classifiers.
- **One-vs-Rest classification**: The model trains a binary classifier for each class to distinguish between that class and the rest of the classes combined. The item is assigned to the class having the maximum probability of having that item.

IV. Research Methodology

In this work, we started with pre-processing the dataset by removing unnecessary punctuation, spaces, and stop-words. We also had to manually delete some records that consisted of messages with empty or unrecognizable symbols. After obtaining the pre-processed dataset, We applied nine different word embedding techniques to extract numeric feature vectors from the messages on the Gitter platform. We considered these features as inputs to develop models for predicting the purpose label of a given message. After applying the word-embedding techniques, we employed the One-Way ANOVA, and PCA to obtain the best combination of relevant features. The One-Way ANOVA test helps find features with equal variance between groups, which means these features don't impact the response much and hence can be discarded. PCA considerably reduces the number of features while trying to retain a significant portion of the variance in the original dataset.

To make the computation of the models simpler and feasible, we sampled 3000 rows randomly from the dataset. Then, we observed the presence of a class imbalance in the sampled rows as the no. of messages with the 'Team-wide' purpose label was much higher than other labels. To combat this issue, we employed the Synthetic Minority Over-sampling (SMOTE) technique, which creates synthetic data samples based on the original data. The models obtained using the above techniques were trained using 34 different classifiers developed using various classification algorithms and ensemble methods and implemented using both One vs. One and One vs. Rest Classification strategies. The predictive ability of these classifiers was validated using the 5-fold cross-validation method. The performance of the developed models was compared using evaluation metrics such as Accuracy, Sensitivity, Specificity, and Geometric Mean, which were extracted using the Box-plot Visualization technique and the Friedman Test.

The detailed overview of the proposed framework is shown in Figure 1. The initial glance of the figure suggests that the

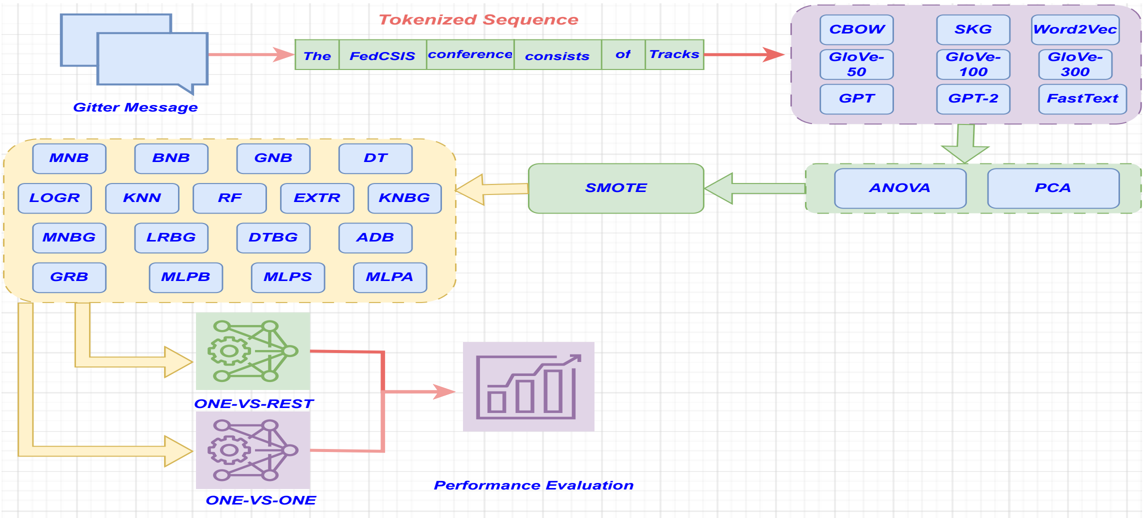


Fig. 1: Framework of proposed work

proposed framework is a multi-step process involving feature extraction from text data using word embedding techniques, removal of irrelevant features, handling class imbalance using SMOTE and development of prediction model using seventeen classification algorithms implemented using two multi-class classification strategies. First, the messages in the GitterCom dataset is pre-processed by removing unnecessary characters and stopwords. Then as shown in Figure 1, these messages are tokenized and nine different word embedding techniques are applied to find the numeric vector representations of these messages. Then the feature selection techniques of One-Way ANOVA and PCA are employed to discard the non-influential features of the numeric data. The next step involves applying the SMOTE technique to handle the presence of class imbalance in the dataset.Then, the prediction models are developed using seventeen classification algorithms, implemented using two multi-class classification strategies – One vs One and One vs Rest. The performance of the models developed using these two strategies is then evaluated using various evaluation metrics.

V. Empirical Results and Analysis

In this work, we applied nine different word embedding techniques, two feature selection techniques, one class balancing technique, and thirty-four different classifiers to analyze and classify the purpose of the messages posted on the Gitter platform. Thus, a total of 1836 [1 dataset * 9 word-embedding techniques * (2 sets generated using feature selection techniques + 1 set of original data) * (1 set generated using a class balancing technique + 1 original dataset) * 34 classifiers] distinct predictive models were built. The predictive ability of the developed models was evaluated using the Accuracy, Sensitivity, Specificity, and Geometric Mean (G-Mean) metrics. These models were validated using the 5-fold cross-validation method.

Accuracy is the most commonly used metric for evaluating the performance of a classifier. It is a measure of the proportion of the correctly classified instances out of the total number of available instances. While it is a useful metric, it can give misleading results in the presence of a class imbalance in the data. Hence, we also incorporate the Sensitivity and Specificity metrics to deal with this issue. While sensitivity is the measure of the proportion of the actual positive cases correctly identified by the classifier, Specificity is the measure of the proportion of the actual negative cases correctly identified by the classifier. Although these two metrics give better predictions than Accuracy in the presence of imbalanced data, there are some instances where these metrics fail, such as when the classifier always predicts the majority class; the models would get high sensitivity but low Specificity. To deal with such cases, we also consider the Geometric Mean metric, which is the geometric mean of both Sensitivity and Specificity.

Tables I and II report the accuracies obtained for the various models developed by applying different classifiers to original data and sampled data on different sets of features. By analyzing the information in the table, the following inferences can be derived that the model with the highest accuracy is obtained by applying the Random Forest Classifier using the One vs Rest classification strategy on the data obtained from the Word2Vec word embedding technique and employing the SMOTE technique to address the class imbalance problem. Similarly, Table I reports the Geometric Mean (G-Mean) values obtained for the various models developed by applying different classifiers to original data and sampled data on different sets of features. By analyzing the information in the table, the following inferences can be derived that the model with the highest accuracy is obtained by applying the Random Forest Classifier using the One vs Rest classification strategy on the data obtained from the Word2Vec word embedding technique and employing the SMOTE technique to address the class

TABLE I: Accuracy and G-Mean

Accuracy																		
	ORG_DATA									**SMOTE_DATA**								
Embedding	**MNB**	**DTC**	**LRC**	**MNBG**	**LRBG**	**DTBG**	**RF**	**ADB**	**MLPB**	**MNB**	**DTC**	**LRC**	**MNBG**	**LRBG**	**DTBG**	**RF**	**ADB**	**MLPB**
One-Vs-One: AF																		
CBOW	96.97	90.10	96.97	96.97	96.97	96.87	96.77	96.93	96.43	60.86	93.21	69.41	60.55	67.68	97.01	98.04	80.92	52.07
SKG	96.77	90.67	96.97	96.97	96.97	96.97	96.93	96.80	95.70	63.07	93.69	80.72	62.79	76.06	97.83	98.54	84.06	98.72
GLOVE50	96.97	90.47	96.97	96.97	96.97	97.00	96.93	96.90	95.40	65.44	92.38	77.69	64.70	71.43	98.22	98.83	80.81	32.38
GLOVE100	96.97	89.83	96.97	96.97	96.97	96.93	96.93	96.73	95.70	64.65	91.99	88.01	64.83	81.41	98.68	98.99	82.31	87.65
GLOVE300	96.90	90.93	97.00	96.97	96.97	96.97	96.97	96.80	95.20	71.07	93.35	95.66	70.98	92.88	98.88	99.56	88.04	97.17
W2V	96.97	90.60	96.93	96.97	96.97	96.93	96.93	96.73	95.80	72.63	94.13	96.51	72.52	94.50	99.37	99.70	88.77	63.78
FST	96.67	90.27	96.97	96.97	96.97	96.87	97.00	96.67	95.27	61.98	94.00	76.82	60.62	73.79	97.79	98.58	84.04	91.84
GPT	92.93	95.43	96.97	94.60	96.97	96.83	96.53	96.90	96.97	46.11	94.86	81.24	46.01	75.42	96.10	96.24	79.34	95.36
GPT2	95.63	95.90	96.93	95.77	96.97	96.87	96.60	96.83	96.97	55.53	94.17	90.52	56.10	89.50	96.31	96.53	83.87	94.04
One-Vs-One: ANOVA																		
CBOW	96.97	90.67	96.97	96.97	96.97	96.77	96.83	96.93	96.93	60.77	92.96	69.51	60.32	67.61	97.00	97.88	80.18	91.93
SKG	96.93	91.13	96.97	96.97	96.97	96.93	96.90	96.93	96.07	63.56	93.92	80.07	62.91	75.67	98.01	98.52	84.24	95.89
GLOVE50	96.97	90.40	96.97	96.97	96.97	96.93	96.97	96.83	96.10	65.53	92.37	76.93	61.97	71.31	98.17	98.92	79.08	32.55
GLOVE100	96.97	90.03	96.97	96.97	96.97	96.93	96.93	96.83	95.33	64.67	91.98	86.67	65.36	80.76	98.40	99.16	80.84	96.31
GLOVE300	96.97	92.10	96.97	96.97	96.97	96.93	96.93	96.83	95.67	70.95	93.51	95.37	70.69	92.24	98.97	99.34	87.90	98.03
W2V	96.97	90.97	96.97	96.97	96.97	96.83	96.83	96.70	96.97	72.40	94.02	96.15	72.05	93.64	99.30	99.62	88.89	32.24
FST	96.77	90.60	96.97	96.97	96.97	96.83	96.80	96.60	95.87	61.71	93.62	76.83	61.36	72.77	97.88	98.53	82.30	98.25
GPT	93.17	95.43	96.97	94.67	96.97	96.87	96.50	96.93	96.97	47.90	94.47	79.03	47.12	73.83	96.07	96.46	79.16	49.48
GPT2	95.67	95.70	96.97	95.77	96.97	96.90	96.70	96.97	96.60	55.67	94.29	90.28	55.56	89.47	96.21	96.29	84.18	94.80
One-Vs-Rest: AF																		
CBOW	96.97	93.77	96.97	96.97	96.97	96.70	96.77	96.93	95.30	63.68	94.98	68.81	62.68	67.02	96.96	97.90	80.75	31.91
SKG	96.73	93.13	96.97	96.97	96.97	96.93	96.90	96.77	95.43	63.76	95.06	79.58	63.29	74.99	97.82	98.59	84.51	77.33
GLOVE50	96.97	93.40	96.97	96.97	96.97	96.97	96.97	96.60	94.67	65.89	94.36	79.13	64.64	72.90	97.97	98.68	80.66	32.38
GLOVE100	96.97	92.87	96.97	96.97	96.97	96.97	96.93	96.70	92.93	66.88	95.36	87.40	67.09	81.29	98.38	99.13	81.22	97.54
GLOVE300	96.47	93.13	97.00	96.97	96.97	96.93	96.93	96.40	94.50	73.92	95.39	96.07	73.89	93.22	98.73	99.43	86.51	46.02
W2V	96.97	93.27	96.93	96.97	96.97	96.97	96.93	96.47	93.73	74.60	96.38	96.92	74.55	95.10	99.06	99.53	89.19	52.53
FST	96.57	93.33	96.97	96.97	96.97	96.87	96.83	96.60	94.80	63.08	95.28	77.36	62.82	73.85	97.74	98.42	82.98	52.65
GPT	89.33	96.03	96.97	94.47	96.97	96.90	96.57	96.87	96.97	48.38	94.96	81.15	47.52	76.91	95.76	96.22	79.66	93.97
GPT2	95.50	96.23	96.90	95.77	96.97	96.93	96.73	96.83	96.97	56.14	95.27	90.60	55.60	89.68	96.04	96.39	84.36	95.94
One-Vs-Rest: ANOVA																		
CBOW	96.97	93.37	96.97	96.97	96.97	96.87	96.73	96.70	96.97	63.74	95.08	68.96	63.22	67.11	96.57	97.67	81.17	88.24
SKG	96.70	93.57	96.97	96.97	96.97	96.97	96.83	96.70	95.33	63.70	95.44	78.72	62.39	74.55	97.82	98.40	83.37	32.51
GLOVE50	96.97	93.73	96.97	96.97	96.97	96.93	96.93	96.57	95.27	65.67	94.76	77.80	64.76	71.62	98.04	98.87	80.12	32.55
GLOVE100	96.97	93.57	96.97	96.97	96.97	96.93	96.93	96.77	94.57	67.16	95.46	86.64	67.40	80.27	98.38	98.89	80.61	97.23
GLOVE300	96.40	93.57	96.97	96.97	96.97	96.87	96.93	96.27	94.13	73.35	95.86	95.61	72.95	92.49	98.79	99.43	86.97	58.38
W2V	96.97	93.67	96.93	96.97	96.97	96.97	96.93	96.37	93.57	74.36	96.00	96.52	74.18	94.34	99.13	99.63	88.75	32.24
FST	96.67	93.43	96.97	96.97	96.97	96.83	96.90	96.43	94.30	62.97	95.51	77.32	62.77	73.24	97.90	98.66	82.40	89.85
GPT	89.43	96.07	96.97	94.27	96.97	96.83	96.50	96.93	96.97	46.30	94.81	80.17	46.68	75.68	95.82	96.16	78.66	40.98
GPT2	95.50	96.30	96.97	95.77	96.97	96.70	96.77	96.87	96.33	54.97	94.90	90.50	55.30	89.60	95.89	96.28	84.67	91.68
G-Mean																		
One-Vs-One: AF																		
CBOW	0.17	0.42	0.17	0.17	0.17	0.20	0.17	0.17	0.17	0.70	0.95	0.77	0.70	0.75	0.98	0.99	0.86	0.63
SKG	0.17	0.38	0.17	0.17	0.17	0.20	0.22	0.20	0.33	0.72	0.95	0.85	0.71	0.82	0.98	0.99	0.88	0.99
GLOVE50	0.17	0.31	0.17	0.17	0.17	0.20	0.17	0.20	0.26	0.74	0.94	0.83	0.73	0.78	0.99	0.99	0.85	0.46
GLOVE100	0.17	0.32	0.17	0.17	0.17	0.17	0.17	0.22	0.28	0.73	0.94	0.91	0.73	0.86	0.99	0.99	0.87	0.91
GLOVE300	0.17	0.32	0.20	0.17	0.17	0.17	0.20	0.20	0.35	0.78	0.95	0.97	0.78	0.95	0.99	1.00	0.91	0.98
W2V	0.17	0.32	0.17	0.17	0.17	0.17	0.17	0.20	0.28	0.79	0.96	0.97	0.79	0.96	1.00	1.00	0.92	0.72
FST	0.22	0.35	0.17	0.17	0.17	0.20	0.22	0.22	0.32	0.71	0.95	0.82	0.70	0.80	0.98	0.99	0.88	0.94
GPT	0.43	0.24	0.17	0.42	0.17	0.17	0.20	0.17	0.17	0.58	0.96	0.86	0.58	0.81	0.97	0.97	0.84	0.97
GPT2	0.43	0.26	0.17	0.43	0.17	0.17	0.25	0.17	0.17	0.66	0.96	0.93	0.66	0.92	0.97	0.97	0.88	0.96
One-Vs-One: ANOVA																		
CBOW	0.17	0.40	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.70	0.95	0.77	0.70	0.75	0.98	0.98	0.85	0.94
SKG	0.17	0.46	0.17	0.17	0.17	0.17	0.22	0.20	0.20	0.72	0.95	0.85	0.72	0.82	0.99	0.99	0.88	0.97
GLOVE50	0.17	0.32	0.17	0.17	0.17	0.17	0.20	0.17	0.17	0.74	0.94	0.82	0.71	0.78	0.99	0.99	0.84	0.46
GLOVE100	0.17	0.29	0.17	0.17	0.17	0.17	0.17	0.20	0.24	0.73	0.94	0.90	0.74	0.85	0.99	0.99	0.85	0.97
GLOVE300	0.17	0.37	0.17	0.17	0.17	0.17	0.17	0.22	0.30	0.78	0.95	0.97	0.78	0.94	0.99	1.00	0.91	0.99
W2V	0.17	0.34	0.17	0.17	0.17	0.17	0.17	0.22	0.17	0.79	0.96	0.97	0.79	0.95	0.99	1.00	0.92	0.46
FST	0.27	0.43	0.17	0.17	0.17	0.20	0.20	0.28	0.33	0.71	0.95	0.82	0.70	0.79	0.98	0.99	0.87	0.99
GPT	0.43	0.22	0.17	0.42	0.17	0.17	0.22	0.17	0.17	0.60	0.96	0.84	0.59	0.80	0.97	0.97	0.84	0.61
GPT2	0.43	0.24	0.17	0.43	0.17	0.17	0.20	0.17	0.17	0.66	0.96	0.93	0.66	0.92	0.97	0.97	0.88	0.96
One-Vs-Rest: AF																		
CBOW	0.17	0.36	0.17	0.17	0.17	0.17	0.22	0.17	0.20	0.72	0.96	0.76	0.71	0.75	0.98	0.98	0.85	0.46
SKG	0.25	0.34	0.17	0.17	0.17	0.22	0.22	0.22	0.32	0.72	0.96	0.85	0.72	0.81	0.98	0.99	0.88	0.83
GLOVE50	0.17	0.19	0.17	0.17	0.17	0.20	0.20	0.25	0.31	0.74	0.96	0.84	0.73	0.79	0.98	0.99	0.85	0.46
GLOVE100	0.17	0.22	0.17	0.17	0.17	0.20	0.17	0.22	0.29	0.75	0.97	0.90	0.75	0.86	0.99	0.99	0.86	0.98
GLOVE300	0.27	0.30	0.20	0.17	0.17	0.17	0.20	0.24	0.34	0.80	0.97	0.97	0.80	0.95	0.99	1.00	0.90	0.58
W2V	0.17	0.26	0.17	0.17	0.17	0.17	0.17	0.27	0.40	0.81	0.97	0.98	0.81	0.96	0.99	1.00	0.92	0.63
FST	0.27	0.28	0.17	0.17	0.17	0.17	0.25	0.22	0.35	0.72	0.96	0.83	0.72	0.80	0.98	0.99	0.87	0.63
GPT	0.44	0.20	0.17	0.42	0.17	0.17	0.17	0.17	0.17	0.60	0.96	0.86	0.59	0.82	0.97	0.97	0.85	0.95
GPT2	0.43	0.24	0.17	0.43	0.17	0.17	0.25	0.17	0.17	0.66	0.96	0.93	0.66	0.92	0.97	0.97	0.88	0.97
One-Vs-Rest: ANOVA																		
CBOW	0.17	0.30	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.72	0.96	0.76	0.72	0.75	0.97	0.98	0.86	0.91
SKG	0.27	0.37	0.17	0.17	0.17	0.22	0.20	0.28	0.33	0.72	0.97	0.84	0.71	0.81	0.98	0.99	0.87	0.46
GLOVE50	0.17	0.26	0.17	0.17	0.17	0.17	0.17	0.20	0.24	0.74	0.96	0.83	0.73	0.78	0.99	0.99	0.85	0.46
GLOVE100	0.17	0.28	0.17	0.17	0.17	0.17	0.17	0.20	0.28	0.75	0.97	0.90	0.75	0.85	0.99	0.99	0.85	0.98
GLOVE300	0.22	0.24	0.17	0.17	0.17	0.17	0.17	0.30	0.34	0.80	0.97	0.97	0.79	0.94	0.99	1.00	0.90	0.68
W2V	0.17	0.22	0.17	0.17	0.17	0.17	0.17	0.28	0.40	0.81	0.97	0.97	0.80	0.96	0.99	1.00	0.92	0.46
FST	0.28	0.30	0.17	0.17	0.17	0.20	0.22	0.30	0.34	0.72	0.97	0.83	0.71	0.80	0.98	0.99	0.87	0.92
GPT	0.47	0.17	0.17	0.42	0.17	0.17	0.20	0.17	0.17	0.58	0.96	0.85	0.59	0.82	0.97	0.97	0.84	0.54
GPT2	0.43	0.17	0.17	0.43	0.17	0.17	0.17	0.17	0.20	0.65	0.96	0.93	0.66	0.92	0.97	0.97	0.88	0.94

imbalance problem. This model gives the highest classification accuracy, sensitivity, and specificity, as mentioned in Table II.

A. COMPARATIVE ANALYSIS

In this section, we examine and compare the efficacy of the models created through a diverse set of word-embedding

TABLE II: Sensitivity and Specificity

Sensitivity																		
	ORG_DATA									SMOTE_DATA								
Embedding	**MNB**	**DTC**	**LRC**	**MNBG**	**LRBG**	**DTBG**	**RF**	**ADB**	**MLPB**	**MNB**	**DTC**	**LRC**	**MNBG**	**LRBG**	**DTBG**	**RF**	**ADB**	**MLPB**
One-Vs-One: AF																		
CBOW	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.20	0.03	0.03	0.03	0.04	0.03	0.03	0.03
SKG	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.16	0.03	0.03	0.03	0.04	0.05	0.04	0.12
GLOVE50	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.10	0.03	0.03	0.03	0.04	0.03	0.04	0.07
GLOVE100	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.11	0.03	0.03	0.03	0.03	0.03	0.05	0.08
GLOVE300	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.11	0.04	0.03	0.03	0.03	0.04	0.04	0.13
W2V	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.11	0.03	0.03	0.03	0.03	0.03	0.04	0.08
FST	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.05	0.14	0.03	0.03	0.03	0.04	0.05	0.05	0.10
GPT	0.93	0.95	0.97	0.95	0.97	0.97	0.97	0.97	0.97	0.20	0.06	0.03	0.19	0.03	0.03	0.04	0.03	0.03
GPT2	0.96	0.96	0.97	0.96	0.97	0.97	0.97	0.97	0.97	0.19	0.07	0.03	0.19	0.03	0.03	0.06	0.03	0.03
One-Vs-One: ANOVA																		
CBOW	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.03	0.18	0.03	0.03	0.03	0.03	0.03	0.03	0.03
SKG	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.23	0.03	0.03	0.03	0.03	0.05	0.04	0.04
GLOVE50	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.11	0.03	0.03	0.03	0.03	0.04	0.03	0.03
GLOVE100	0.97	0.90	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.09	0.03	0.03	0.03	0.03	0.03	0.04	0.06
GLOVE300	0.97	0.92	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.03	0.15	0.03	0.03	0.03	0.03	0.03	0.05	0.09
W2V	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.03	0.13	0.03	0.03	0.03	0.03	0.03	0.05	0.03
FST	0.97	0.91	0.97	0.97	0.97	0.97	0.97	0.97	0.96	0.07	0.20	0.03	0.03	0.03	0.04	0.04	0.08	0.12
GPT	0.93	0.95	0.97	0.95	0.97	0.97	0.97	0.97	0.97	0.20	0.05	0.03	0.19	0.03	0.03	0.05	0.03	0.03
GPT2	0.96	0.96	0.97	0.96	0.97	0.97	0.97	0.97	0.97	0.19	0.06	0.03	0.19	0.03	0.03	0.04	0.03	0.03
One-Vs-Rest: AF																		
CBOW	0.61	0.93	0.69	0.61	0.68	0.97	0.98	0.81	0.52	0.80	0.97	0.85	0.80	0.84	0.99	0.99	0.90	0.76
SKG	0.63	0.94	0.81	0.63	0.76	0.98	0.99	0.84	0.99	0.82	0.97	0.90	0.81	0.88	0.99	0.99	0.92	0.99
GLOVE50	0.65	0.92	0.78	0.65	0.71	0.98	0.99	0.81	0.32	0.83	0.96	0.89	0.82	0.86	0.99	0.99	0.90	0.66
GLOVE100	0.65	0.92	0.88	0.65	0.81	0.99	0.99	0.82	0.88	0.82	0.96	0.94	0.82	0.91	0.99	0.99	0.91	0.94
GLOVE300	0.71	0.93	0.96	0.71	0.93	0.99	1.00	0.88	0.97	0.86	0.97	0.98	0.85	0.96	0.99	1.00	0.94	0.99
W2V	0.73	0.94	0.97	0.73	0.94	0.99	1.00	0.89	0.64	0.86	0.97	0.98	0.86	0.97	1.00	1.00	0.94	0.82
FST	0.62	0.94	0.77	0.61	0.74	0.98	0.99	0.84	0.92	0.81	0.97	0.88	0.80	0.87	0.99	0.99	0.92	0.96
GPT	0.46	0.95	0.81	0.46	0.75	0.96	0.96	0.79	0.95	0.73	0.97	0.91	0.73	0.88	0.98	0.98	0.90	0.98
GPT2	0.56	0.94	0.91	0.56	0.90	0.96	0.97	0.84	0.94	0.78	0.97	0.95	0.78	0.95	0.98	0.98	0.92	0.97
One-Vs-Rest: ANOVA																		
CBOW	0.61	0.93	0.70	0.60	0.68	0.97	0.98	0.80	0.92	0.80	0.96	0.85	0.80	0.84	0.98	0.99	0.90	0.96
SKG	0.64	0.94	0.80	0.63	0.76	0.98	0.99	0.84	0.96	0.82	0.97	0.90	0.81	0.88	0.99	0.99	0.92	0.98
GLOVE50	0.66	0.92	0.77	0.62	0.71	0.98	0.99	0.79	0.33	0.83	0.96	0.88	0.81	0.86	0.99	0.99	0.90	0.66
GLOVE100	0.65	0.92	0.87	0.65	0.81	0.98	0.99	0.81	0.96	0.82	0.96	0.93	0.83	0.90	0.99	1.00	0.90	0.98
GLOVE300	0.71	0.94	0.95	0.71	0.92	0.99	0.99	0.88	0.98	0.85	0.97	0.98	0.85	0.96	0.99	1.00	0.94	0.99
W2V	0.72	0.94	0.96	0.72	0.94	0.99	1.00	0.89	0.32	0.86	0.97	0.98	0.86	0.97	1.00	1.00	0.94	0.66
FST	0.62	0.94	0.77	0.61	0.73	0.98	0.99	0.82	0.98	0.81	0.97	0.88	0.81	0.86	0.99	0.99	0.91	0.99
GPT	0.48	0.94	0.79	0.47	0.74	0.96	0.96	0.79	0.49	0.74	0.97	0.90	0.74	0.87	0.98	0.98	0.90	0.75
GPT2	0.56	0.94	0.90	0.56	0.89	0.96	0.96	0.84	0.95	0.78	0.97	0.95	0.78	0.95	0.98	0.98	0.92	0.97
Specificity																		
One-Vs-One: AF																		
CBOW	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.14	0.03	0.03	0.03	0.03	0.05	0.03	0.04
SKG	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.06	0.13	0.03	0.03	0.03	0.05	0.05	0.05	0.10
GLOVE50	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.04	0.03	0.03	0.03	0.04	0.04	0.06	0.10
GLOVE100	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.97	0.93	0.03	0.05	0.03	0.03	0.03	0.04	0.03	0.05	0.09
GLOVE300	0.96	0.93	0.97	0.97	0.97	0.97	0.97	0.96	0.95	0.07	0.09	0.04	0.03	0.03	0.03	0.04	0.06	0.13
W2V	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.96	0.94	0.03	0.07	0.03	0.03	0.03	0.03	0.03	0.07	0.17
FST	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.07	0.08	0.03	0.03	0.03	0.03	0.06	0.05	0.13
GPT	0.89	0.96	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.22	0.04	0.03	0.19	0.03	0.03	0.03	0.03	0.03
GPT2	0.96	0.96	0.97	0.96	0.97	0.97	0.97	0.97	0.97	0.19	0.06	0.03	0.19	0.03	0.03	0.06	0.03	0.03
One-Vs-One: ANOVA																		
CBOW	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.03	0.09	0.03	0.03	0.03	0.03	0.03	0.03	0.03
SKG	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.07	0.15	0.03	0.03	0.03	0.05	0.04	0.08	0.12
GLOVE50	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.07	0.03	0.03	0.03	0.03	0.03	0.04	0.06
GLOVE100	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.03	0.08	0.03	0.03	0.03	0.03	0.03	0.04	0.08
GLOVE300	0.96	0.94	0.97	0.97	0.97	0.97	0.97	0.96	0.94	0.05	0.06	0.03	0.03	0.03	0.03	0.03	0.09	0.13
W2V	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.96	0.94	0.03	0.05	0.03	0.03	0.03	0.03	0.03	0.08	0.17
FST	0.97	0.93	0.97	0.97	0.97	0.97	0.97	0.96	0.94	0.08	0.09	0.03	0.03	0.03	0.04	0.05	0.09	0.13
GPT	0.89	0.96	0.97	0.94	0.97	0.97	0.97	0.97	0.97	0.24	0.03	0.03	0.19	0.03	0.03	0.04	0.03	0.03
GPT2	0.96	0.96	0.97	0.96	0.97	0.97	0.97	0.97	0.96	0.19	0.03	0.03	0.19	0.03	0.03	0.03	0.03	0.04
One-Vs-Rest: AF																		
CBOW	0.64	0.95	0.69	0.63	0.67	0.97	0.98	0.81	0.32	0.82	0.97	0.84	0.81	0.84	0.98	0.99	0.90	0.66
SKG	0.64	0.95	0.80	0.63	0.75	0.98	0.99	0.85	0.77	0.82	0.98	0.90	0.82	0.87	0.99	0.99	0.92	0.89
GLOVE50	0.66	0.94	0.79	0.65	0.73	0.98	0.99	0.81	0.32	0.83	0.97	0.90	0.82	0.86	0.99	0.99	0.90	0.66
GLOVE100	0.67	0.95	0.87	0.67	0.81	0.98	0.99	0.81	0.98	0.83	0.98	0.94	0.84	0.91	0.99	1.00	0.91	0.99
GLOVE300	0.74	0.95	0.96	0.74	0.93	0.99	0.99	0.87	0.46	0.87	0.98	0.98	0.87	0.97	0.99	1.00	0.93	0.73
W2V	0.75	0.96	0.97	0.75	0.95	0.99	1.00	0.89	0.53	0.87	0.98	0.98	0.87	0.98	1.00	1.00	0.95	0.76
FST	0.63	0.95	0.77	0.63	0.74	0.98	0.98	0.83	0.53	0.82	0.98	0.89	0.81	0.87	0.99	0.99	0.91	0.76
GPT	0.48	0.95	0.81	0.48	0.77	0.96	0.96	0.80	0.94	0.74	0.97	0.91	0.74	0.88	0.98	0.98	0.90	0.97
GPT2	0.56	0.95	0.91	0.56	0.90	0.96	0.96	0.84	0.96	0.78	0.98	0.95	0.78	0.95	0.98	0.98	0.92	0.98
One-Vs-Rest: ANOVA																		
CBOW	0.64	0.95	0.69	0.63	0.67	0.97	0.98	0.81	0.88	0.82	0.98	0.84	0.82	0.84	0.98	0.99	0.91	0.94
SKG	0.64	0.95	0.79	0.62	0.75	0.98	0.98	0.83	0.33	0.82	0.98	0.89	0.81	0.87	0.99	0.99	0.92	0.66
GLOVE50	0.66	0.95	0.78	0.65	0.72	0.98	0.99	0.80	0.33	0.83	0.97	0.89	0.82	0.86	0.99	0.99	0.90	0.66
GLOVE100	0.67	0.95	0.87	0.67	0.80	0.98	0.99	0.81	0.97	0.84	0.98	0.93	0.84	0.90	0.99	0.99	0.90	0.99
GLOVE300	0.73	0.96	0.96	0.73	0.92	0.99	0.99	0.87	0.58	0.87	0.98	0.98	0.86	0.96	0.99	1.00	0.93	0.79
W2V	0.74	0.96	0.97	0.74	0.94	0.99	1.00	0.89	0.32	0.87	0.98	0.98	0.87	0.97	1.00	1.00	0.94	0.66
FST	0.63	0.96	0.77	0.63	0.73	0.98	0.99	0.82	0.90	0.81	0.98	0.89	0.81	0.87	0.99	0.99	0.91	0.95
GPT	0.46	0.95	0.80	0.47	0.76	0.96	0.96	0.79	0.41	0.73	0.97	0.90	0.73	0.88	0.98	0.98	0.89	0.70
GPT2	0.55	0.95	0.91	0.55	0.90	0.96	0.96	0.85	0.92	0.77	0.97	0.95	0.78	0.95	0.98	0.98	0.92	0.96

techniques, classification algorithms, feature selection, and data sampling techniques. To evaluate the models devised for classifying the purpose of the messages in Gitter communications, we have employed evaluation metrics for statistical analysis, box plots for visual representation, and the Friedman test to identify the significant differences among the models.

The Friedman test is a non-parametric test used to determine if there is a significant difference in the average ranks of multiple related samples. In our research, this test was used to test the validity of the following hypothesis:

- **Null Hypothesis** - *The predictive abilities of the models developed using the combination of various word-embedding techniques, classifiers, feature selection, and data sampling techniques are similar.*
- **Alternate Hypothesis** - *The models' predictive abilities developed using various word-embedding techniques, classifiers, feature selection, and data sampling techniques significantly differ.*

B. Word-Embedding Techniques

The text messages present in the dataset were converted to a numeric vector representation to develop classification models. The nine-word embedding techniques employed for this purpose are Continuous Bag of Words (CBOW), Skip-Gram (SKG), Global Vectors for Word Representation (GloVe) with 50 dimensions (GLOVE50), 100 dimensions (GLOVE100), and 300 dimensions (GLOVE300), Word2Vec (W2V), fastText (FST), Generative Pre-trained Transformer (GPT), and Generative Pre-trained Transformer-2 (GPT2).

Comparison of Word-embedding techniques using Box Plots: Figure 2 provides a graphic representation of the values of the evaluation metrics - Accuracy, Sensitivity, Specificity, and G-Mean of different word embedding techniques applied in the form of Box-plots. While we can observe that the average G-Mean is highest for models based on the GPT-2 word-embedding technique with a value of 0.54, followed by GPT with 0.49, the highest maximum G-Mean is for the models developing using Word2Vec word embeddings with a value of 0.9978, followed by GloVe 300d with a value of 0.9972. The Q3 G-Mean is highest for models with GPT-2, followed by GloVe 300d. Comparing the accuracies of the models provides more stable results as models built using Word2Vec and GloVe 300d emerge as the best-performing models in terms of maximum, Q3, and median accuracy values, while models using GPT and GPT-2 lag behind in this analysis. Since the box plots for the G-Mean values don't paint a clear picture, we rely on the Friedman Rank Test to draw inferences.

Comparison of Word-embedding techniques using the Friedman Test:
The study also employs the Friedman test to evaluate the performance of the models developed using various word embedding techniques. This test aims to test the validity of the null hypothesis, which states that "the different word embedding techniques do not significantly impact the performance of the classification models developed." The test was carried out at a significance level of 0.05 and with nine degrees of freedom. Table III displays the Friedman mean ranks of the G-Mean metric obtained for the various word-embedding algorithms. A lower mean rank indicates a better-performing technique. GloVe 300d has the lowest mean rank of 3.77, followed by Word2Vec, which has a mean rank of 4.05. CBoW has the highest mean rank of 6.2, followed by GPT, which has a mean rank of 6.01. From these observations, it can be inferred that models developed using GloVe and Word2Vec have the highest predictive ability, while the models that use CBoW or GPT suffer the most. Another inference that can be drawn is that models developed with word embeddings with generic pre-trained vectors perform superior for the Gitter-Com dataset compared to those developed with other word-embedding techniques, even those that are domain-specific to software systems.

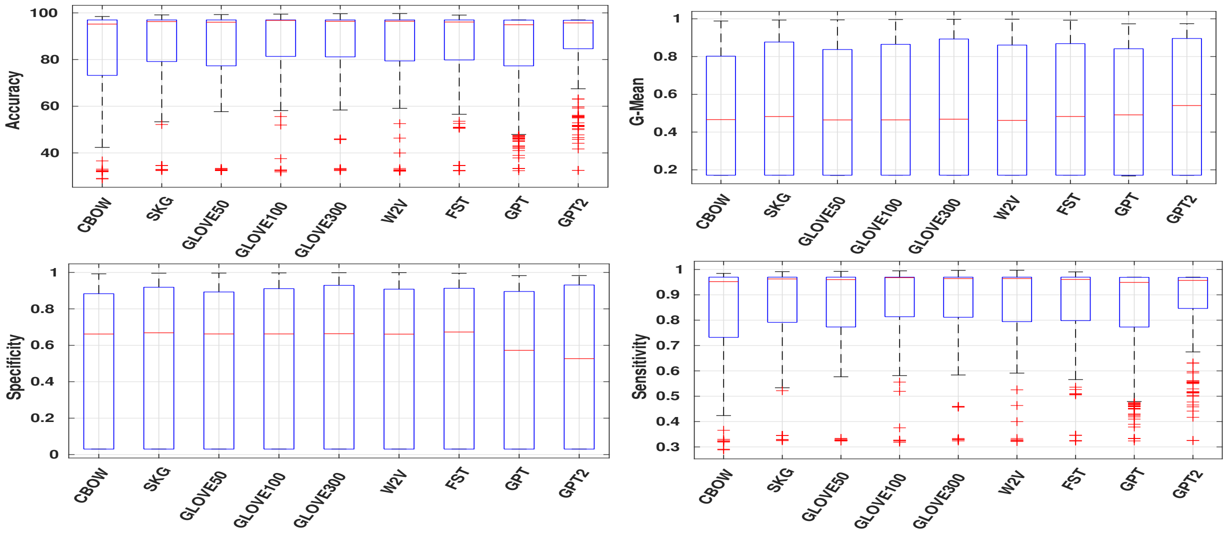


Fig. 2: Performance of Nine Word Embedding

TABLE III: Friedman: Nine Word Embedding

	Accuracy	G-Mean	Specificity	Sensitivity
CBOW	6.11	6.2	6.06	6.11
SKG	4.72	4.25	4.25	4.72
GLOVE50	5.22	5.6	5.63	5.22
GLOVE100	4.57	4.96	4.98	4.57
GLOVE300	3.89	3.77	3.83	3.89
W2V	3.74	4.05	4.12	3.74
FST	5.31	4.71	4.65	5.31
GPT	6.04	6.01	6.01	6.04
GPT2	5.39	5.45	5.47	5.39

C. Feature Selection Techniques

In this work, we have employed two different feature selection techniques - the One-way analysis of variance test and Principal Component Analysis. These techniques were applied to reduce the complexity of the models by retaining or generating the relevant features and discarding the rest. This gave us three different sets of features to compare, the original feature set and the feature seats generated using the feature selection techniques.

Comparison of the Different Sets of Features using Box Plots:

Figure 3 provides a graphic representation of the values of the evaluation metrics - Accuracy, Sensitivity, Specificity, and G-Mean of the models trained using selected sets of features and all features in the form of Box-plots. From the figure, we can draw the inference that the models developed by considering all the features have a marginally better predictive ability for the GitterCom dataset. The average G-Mean of the models developed using the original data is 0.483, with a maximum G-Mean of 0.998 and a Q3 G-Mean of 0.879. The mean G-Mean of the models that use the ANOVA test (0.471) and PCA (0.461) is also quite close to the mean G-Mean of the original data, with ANOVA performing better than PCA for the GitterCom dataset. This is also backed up by the box-plots of accuracies of the models where the models built with all the features slightly out-performs the ANOVA test-based models, which in turn perform much better than models built by considering PCA.

Comparison of the Different Sets of Features using the Friedman Test:

This study also considers the Friedman test to compare the performance of the models using the set of all features or the sets of features generated by the feature selection techniques. This test aims to test the validity of the null hypothesis, which states that "there is no significant difference in the performance of models trained on the data with different sets of features." Table IV displays the Friedman mean ranks of the G-Mean metric obtained for models trained on the original and selected set of features. The models trained using the original feature set have the lowest mean rank of 1.76. The models trained using the feature set generated from the One-way ANOVA test have a slightly higher mean rank of 1.87, and the models trained using PCA have the highest mean rank of 2.37. This analysis indicates that introducing feature selection techniques depreciates the model's performance and that the model works best with all the original features. However, the One-Way ANOVA test models only perform slightly worse than the models with all the features since the mean G-Mean and the mean Friedman ranks are quite close. Hence, if certain word-embedding techniques generate a large number of features for the GitterCom dataset, the One-Way ANOVA test can be considered for feature selection.

TABLE IV: Friedman: Feature Selection Techniques

	Accuracy	G-Mean	Specificity	Sensitivity
AFV	1.86	1.76	1.75	1.86
ANOVA	1.94	1.87	1.88	1.94
PCA	2.2	2.37	2.37	2.2

D. Class Balancing Technique

The Synthetic Minority Oversampling Technique (SMOTE) was used to rectify the class imbalance problem of the dataset, as this technique synthesized data points for the minority classes of the “personal benefits” and “community support” classes.

Comparison of original data and SMOTE synthesized data using Box Plots:

Figure 4 provides a graphic representation of the values of the evaluation metrics - Accuracy, Sensitivity, Specificity, and G-Mean in the form of Box-plots for the models developed using original data and the balanced data obtained from the SMOTE technique. From the figure, we can draw the inference that the models developed by applying the SMOTE technique have a much better predictive ability for the GitterCom dataset compared to the models relying on the original data. The average G-Mean of the models developed using SMOTE is 0.857, with a maximum G-Mean of 0.998 and a Q3 G-Mean of 0.956, which indicates that 25% of the models created using SMOTE have a G-Mean greater than 0.956. Since the

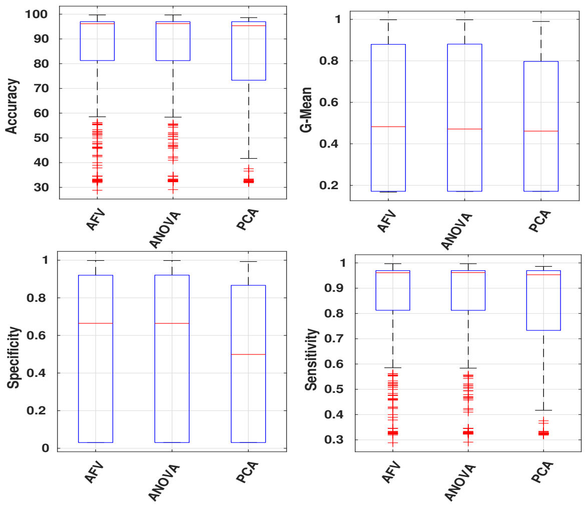


Fig. 3: Performance of Feature Selection Techniques

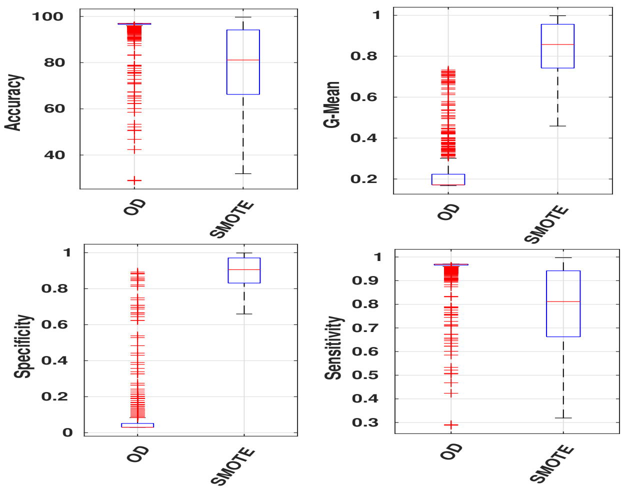


Fig. 4: Performance Class Balancing Technique

original data is highly class-imbalanced, accuracy is not a reliable metric for this analysis since it projects the models with original data to perform better than models with the SMOTE synthesized data.

Comparison of original data and SMOTE synthesized data using the Friedman Test:
This study also employs the Friedman test to compare the performance of the models developed using the original imbalanced dataset and the SMOTE-synthesized dataset. This test aims to test the validity of the null hypothesis, which states that "there is no significant difference in the performance of models trained with dataset having balanced or imbalanced classes, and the algorithm used to balance classes has no significant impact on their performance." Table V displays the Friedman mean ranks of the G-Mean metric obtained for models trained on the original and the SMOTE-balanced dataset. While the latter models have a mean rank of 1, the former models have a mean rank of 2. This is in accordance with the inference obtained from the box plots, thus declaring the use of the SMOTE data sampling technique to improve the performance of the models trained on the GitterCom dataset. The mean ranks of Accuracy provide a different picture where the models trained using the original data have a mean rank of 1.2 which is lower than the mean rank of the models trained using the SMOTE balanced dataset (1.8). This shows the unreliability of the accuracy metric in the presence of class-imbalanced data, thus establishing the G-Mean as the dependable metric in such circumstances.

TABLE V: Friedman: Class Balancing Technique

	Accuracy	G-Mean	Specificity	Sensitivity
OD	1.2	2	1.99	1.2
SMOTE	1.8	1	1.01	1.8

E. Classification Techniques

In this work, we have used seventeen different classification algorithms such as Multinomial Naive Bayes (MNB), Bernoulli's Naive Bayes (BNB), Gaussian Naive Bayes (GNB), Decision Tree (DTC), Logistic Regression (LRC), K-Nearest Neighbours (KNN), KNN with Bagging (KNBG), Multinomial Naive Bayes with Bagging (MNBG), Logistic Regression with Bagging (LRBG), Decision Trees with Bagging (DTBG), Random Forest (RF), Extra Trees (EXTC), Ada Boost (DBG), Gradient Boosting (GRB), Multi-Layer Perceptron with Limited-Memory BFGS (MLPB), SGD (MLPS) and ADAM (MLPA).

Comparison of Classification Techniques using Box Plots:
Figure 5 provides a graphic representation of the values of the evaluation metrics - Accuracy, Sensitivity, Specificity, and G-Mean of different classification algorithms applied in the form of Box-plots. From the figure, we can observe that while tree-based classifiers generally perform better than the rest, the Naive Bayes-based classifiers lag behind in their performance. Random Forest and Extra Trees Classifiers have the highest maximum and Q3 G-Mean values, while Bernoulli and Multinomial Naive Bayes classifiers have the least values. The median G-Mean values paint a slightly different picture as Gaussian Naive Bayes has the highest value, followed by the Decision Tree classifier. The accuracy box plots suggest that Random Forest and Extra Trees classifiers are best performing, followed by Decision Trees with Bagging, while Gaussian Naive Bayes lags behind. Since we can't zero in on the best-performing classifier using the Box Plots, we rely on the Friedman Rank test.

Comparison of Classification techniques using the Friedman Test:
The study also employs the Friedman test to evaluate the performance of the models developed using various classification algorithms. This test aims to test the validity of the null hypothesis, which states that "the choice of the classification algorithms does not significantly impact the performance of the classification models developed." The test was carried out at a significance level of 0.05 and with seventeen degrees of freedom. Table VI displays the Friedman mean ranks of the G-Mean metric obtained for the various classification algorithms. A lower mean rank indicates a better-performing algorithm. We can observe that the Decision Tree classifier has the lowest mean rank of 4.53 among all classifiers, followed by the Extra Trees Classifier at 4.81 and the Random Forest Classifier at 5.46. Bernoulli's Naive Bayes Classifier has the highest rank of 14.13, followed by the Multinomial Naive Bayes classifier with Bagging, with a mean rank of 12.09. Thus, we can infer that the Decision Tree Classifier is the best-performing classification algorithm for the GitterCom dataset, followed by Extra Trees and Random Forest Classifiers, while Bernoulli's Naive Bayes classifier performs the worst for the dataset.

F. One-vs-One and One-vs-rest multi-class classification:

The above-mentioned classification algorithms were implemented using one-vs-one and one-vs-rest multi-class classifi-

TABLE VI: Friedman: Class Balancing Technique

	Accuracy	G-Mean	Specificity	Sensitivity
MNB	10.95	11.3	11.3	10.95
BNB	13.73	14.13	14.13	13.73
GNB	14.52	6.69	6.69	14.52
DTC	10.36	4.53	4.53	10.36
LRC	7.18	10.16	10.16	7.18
KNN	8.6	9.79	9.79	8.6
KNBG	6.23	9.25	9.25	6.23
MNBG	10.45	12.09	12.09	10.45
LRBG	7.87	10.89	10.89	7.87
DTBG	5.85	8.04	8.04	5.85
RF	5.75	5.46	5.46	5.75
EXTC	5.54	4.81	4.81	5.54
ADB	10.34	9.54	9.53	10.34
GRB	11.26	7.45	7.45	11.26
MLPB	10.19	8.95	8.95	10.19
MLPS	6.61	9.63	9.63	6.61
MLPA	7.57	10.31	10.31	7.57

cation strategies, thus generating thirty-four different classifiers for this analysis.

Comparison of the two classification strategies using Box Plots:

Figure 6 provides a graphic representation of the values of the evaluation metrics - Accuracy, Sensitivity, Specificity, and G-Mean of the two classification strategies applied in the form of Box-plots. From these figures, we can observe that the One-vs-Rest slightly out-performs the one-vs-one strategy as the models developed using the former have a maximum G-Mean of 0.9977 and a Q3 value of 0.86 as compared to its counterpart's values of 0.9974 and 0.85. The accuracy box plot also gives similar results as one-vs-rest has slightly higher maximum and median accuracies as compared to models using one-vs-one classification.

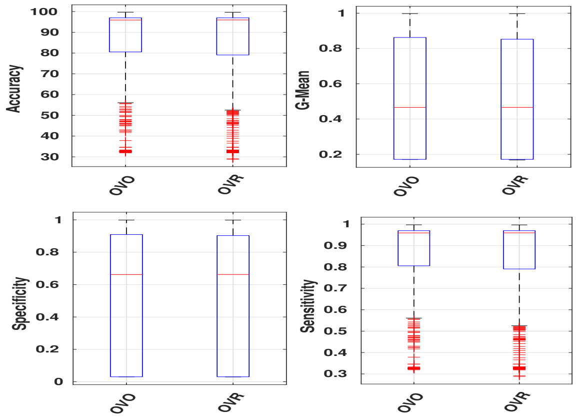


Fig. 6: Performance One-vs-One and One-vs-rest multi-class classification

Comparison of Classification strategies using the Friedman Test:

The study also employs the Friedman test to evaluate the performance of the models developed using the two classification strategies. This test aims to test the validity of the null

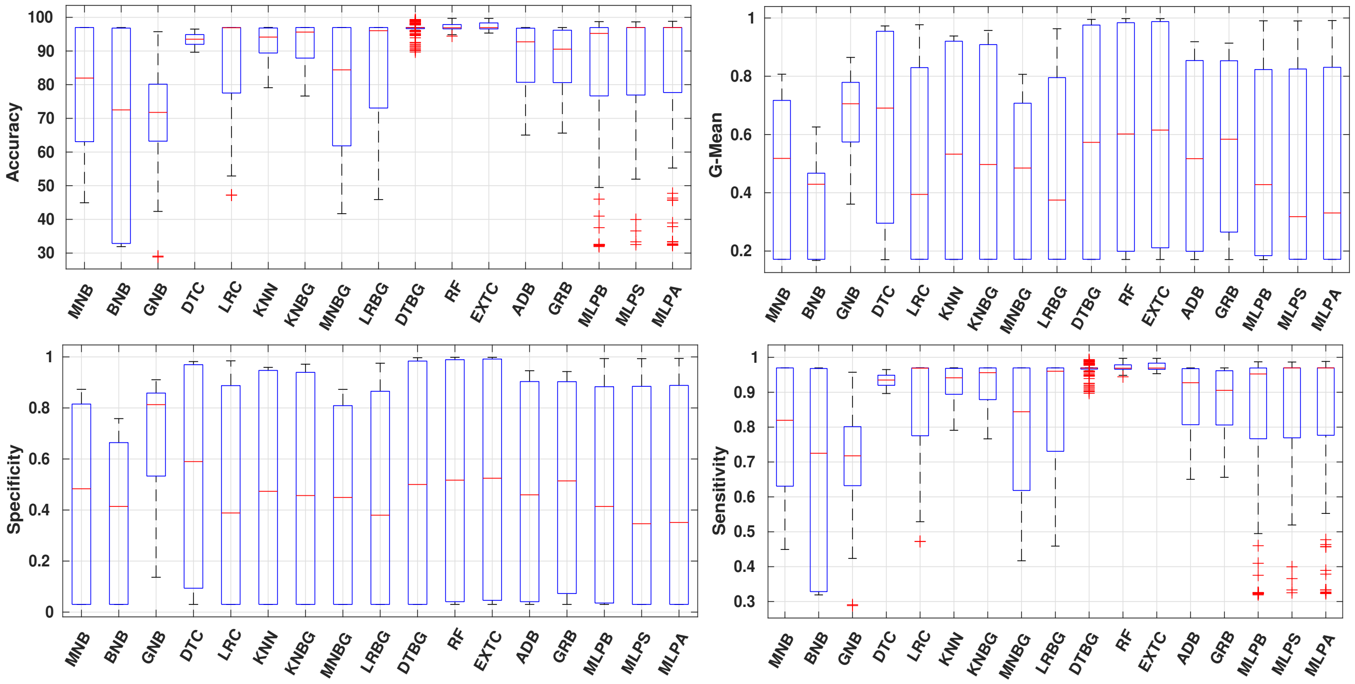


Fig. 5: Performance Classification Techniques

hypothesis, which states that "the choice of the classification strategy does not significantly impact the performance of the classification models developed." The test was carried out at a significance level of 0.05 and with two degrees of freedom. Table VII displays the Friedman mean ranks of the G-Mean metric obtained for the various classification algorithms. We can observe that the models using one-vs-one classification have a slightly lower Friedman mean Rank of 1.47 as compared to the one-vs-rest models' rank of 1.53. This trend is also followed by the mean ranks of accuracies of the two strategies. Thus, the Friedman Rank test concludes that the one-vs-one classification is slightly better than the one-vs-rest classification.

TABLE VII: Friedman: One-vs-One and One-vs-rest

	Accuracy	G-Mean	Specificity	Sensitivity
OVO	1.49	1.47	1.48	1.49
OVR	1.51	1.53	1.52	1.51

VI. CONCLUSION

Analyzing and classifying the purpose of the messages on software messaging and collaboration platforms such as Gitter provides various benefits, such as analyzing the present open-source development trends and understanding why developers prefer such platforms. Automated message classification can save time and labor and reduce misclassification errors. This paper aims to improve the classification accuracy of the purpose of the messages in the GitterCom dataset by finding the right combination of ML and NLP techniques for the best performance. Various word-embedding techniques, feature selection techniques, classification algorithms, and a data sampling technique were employed in this work. The comparative analysis helps analyze each technique's merits and demerits for analysis of the GitterCom dataset. The key conclusion obtained in this work are:

- Models trained using GloVe 300d and Word2Vec perform superior to models trained with other word-embedding techniques, even the ones that are domain-specific to software systems.
- The application of feature selection techniques slightly degrades the performance of the classifier models. However, the ANOVA test is still a viable alternative in case the computational complexity of the models needs to be improved.
- The class-balancing technique (SMOTE) improved the performance of the models by addressing the class imbalance problem.
- The tree-based classification algorithms outperformed others as the Decision Trees classifier emerged as the best, followed by Random Forest and Extra Trees classifiers.
- The one-vs-one multi-class classification strategy performed better than the one-vs-rest classification for the GitterCom dataset message purpose classification.

While our research focuses on the purpose of the messages, future research could look into insights that can be drawn from messages of each type of purpose. The team-wide purpose messages can be used to analyze collaboration patterns and geographical trends influencing the development projects. Similarly, the personal benefit messages can be used to analyze the commonly faced issues and the extent of help being provided to these users on the platform. Topics of discussion can be analyzed to identify the latest trends and popular technologies in software development. These insights help update the platforms to suit the needs of the developers better, attracting more users and promoting collaboration. Developing such models also helps users understand these trends, which they can incorporate into their own development practices, making them more efficient. Future research to draw insights from the data of such developer communication platforms can be made easier by adapting the techniques and pipelines shown to be more effective and hence recommended by the paper.

VII. ACKNOWLEDGEMENTS

This research is funded by TestAIng Solutions Pvt. Ltd.

REFERENCES

[1] E. Parra, A. Ellis, and S. Haiduc, "Gittercom: A dataset of open source developer communications in gitter," in *Proceedings of the 17th International Conference on Mining Software Repositories*, 2020, pp. 563–567.

[2] O. Ehsan, S. Hassan, M. E. Mezouar, and Y. Zou, "An empirical study of developer discussions in the gitter platform," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 30, no. 1, pp. 1–39, 2020.

[3] H. Sahar, A. Hindle, and C.-P. Bezemer, "How are issue reports discussed in gitter chat rooms?" *Journal of Systems and Software*, vol. 172, p. 110852, 2021.

[4] E. Parra, M. Alahmadi, A. Ellis, and S. Haiduc, "A comparative study and analysis of developer communications on slack and gitter," *Empirical Software Engineering*, vol. 27, no. 2, p. 40, 2022.

[5] B. Lin, A. Zagalsky, M.-A. Storey, and A. Serebrenik, "Why developers are slacking off: Understanding how software teams use slack," in *Proceedings of the 19th acm conference on computer supported cooperative work and social computing companion*, 2016, pp. 333–336.

[6] V. Stray, N. B. Moe, and M. Noroozi, "Slack me if you can! using enterprise social networking tools in virtual agile teams," in *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE, 2019, pp. 111–121.

[7] R. Alkadhi, T. Lata, E. Guzmany, and B. Bruegge, "Rationale in development chat messages: an exploratory study," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 2017, pp. 436–446.

[8] R. Alkadhi, J. O. Johanssen, E. Guzman, and B. Bruegge, "React: An approach for capturing rationale in chat messages," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2017, pp. 175–180.

[9] S. Beyer, C. Macho, M. Pinzger, and M. Di Penta, "Automatically classifying posts into question categories on stack overflow," in *Proceedings of the 26th Conference on Program Comprehension*, 2018, pp. 211–221.

[10] A. S. M. Venigalla, C. Lakkundi, and S. Chimalakonda, "Sotagger - towards classifying stack overflow posts through contextual tagging (s)," 07 2019, pp. 493–496.

[11] E. Guzman, M. Ibrahim, and M. Glinz, "A little bird told me: Mining tweets for requirements and software evolution," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, 2017, pp. 11–20.

[12] S. Tiun, U. Mokhtar, S. Bakar, and S. Saad, "Classification of functional and non-functional requirement in software requirement using word2vec and fast text," in *journal of Physics: conference series*, vol. 1529, no. 4. IOP Publishing, 2020, p. 042077.

DOI: 10.15439/2023F978
ISSN 2300-5963 ACSIS, Vol. 35

Dual-Path Image Reconstruction: Bridging Vision Transformer and Perceptual Compressive Sensing Networks

Zakaria Bairi*, Kadda Beghdad Bey*, Olfa Ben-Ahmed†, Abdennour Amamra* and Abbas Bradai†
*Ecole Militaire Polytechnique, Bordj El Behri, Algiers, Algeria
†XLIM Research Institute, UMR CNRS 7252
University of Poitiers, France

***Abstract*—Over the past few years, notable advancements have been made through the adoption of self-attention mechanisms and perceptual optimization, which have proven to be successful techniques in enhancing the overall quality of image reconstruction. Self-attention mechanisms in Vision Transformers have been widely used in neural networks to capture long-range dependencies in image data, while perceptual optimization has been shown to enhance the perceptual quality of reconstructed images. In this paper, we present a novel approach to image reconstruction by bridging the capabilities of Vision Transformer and Perceptual Compressive Sensing Networks. Specifically, we use a self-attention mechanism to capture the global context of the image and guide the sampling process, while optimizing the perceptual quality of the sampled image using a pre-trained perceptual loss function. Our experiments demonstrate that our proposed approach outperforms existing state-of-the-art methods in terms of reconstruction quality and achieves visually pleasing results. Overall, our work contributes to the development of efficient and effective techniques for image sampling and reconstruction, which have potential applications in a wide range of domains, including medical imaging and video processing.**

I. Introduction

COMPRESSIVE Sensing (CS) is an important technique in the field of signal processing and computer vision. CS is a technique for acquiring and processing signals at a lower rate than required by the Nyquist-Shannon sampling theorem. CS is used for image reconstruction, by reconstructing a high-quality image from a set of low-quality or incomplete observations. It has emerged as an alternative to traditional image compression techniques. The potential of CS and image reconstruction in computer vision lies in their ability to enable high-quality imaging and data acquisition with minimal resources. Indeed, CS can be used to reduce the amount of data required to capture an image, making it possible to store or transmit images more efficiently. It can also be used to reduce the amount of data required for image processing, enabling real-time processing of large datasets. Most CS approaches are CNN-based models, which represent a limitation by the receptive field of the convolution kernels and their non-ability to handle long-term dependencies.

Deep learning models have shown impressive success in various computer vision tasks, but they are not always efficient at processing large and complex images. One possible solution to this problem is to incorporate visual attention mechanisms into deep learning models, inspired by the way humans selectively process relevant information and filter out distractions. Attention mechanisms allow deep learning models to focus on important regions of an image and suppress irrelevant information, leading to improved accuracy and efficiency.

In addition to visual attention, which is a selective process that allows us to focus on important information in the environment while ignoring distractions, we can also find visual perception, which refers to the process of interpreting and making sense of visual information from the environment. This process involves multiple stages of processing, where the basic features of a stimulus (e.g., color, shape, motion) are detected and encoded then integrated into meaningful objects and scenes. Overall, visual attention and visual perception are both important processes in visual recognition. Visual attention allows us to selectively focus on important information in the environment, while visual perception allows us to interpret and make sense of visual information. These processes are intricately linked and work together to support our visual experiences.

In this paper, we propose a novel CS approach for image sampling and reconstruction. Our proposed model combines self-attention and perceptual information to selectively attend to different regions of an image at multiple levels of abstraction. We evaluate the effectiveness of our proposed model using experiments on benchmark datasets and demonstrate that it outperforms existing models in terms of reconstruction quality.

Hence, the main contributions of this paper are :

- We propose a framework based on a hybrid architecture that combines the self-attention mechanism provided by vision transformers for image long-range dependencies and global context modeling with the advantages of convolutional neural networks for optimal local feature extraction.
- We propose to add a transformer-based coding path, so the model coding is done in two paths, a CNN-based CSNet sampling path, and a transformer path. These two paths are linked by a fusion layer to merge the features and produce the vector that presents the input image.
- We use a perceptual optimization in the training process, to semantically guide the model to learn long-range

and local high-frequency details of visual and contextual features.

- Finally, we run extensive experiments to evaluate our approach in term of reconstruction quality and compare it to state-of-the-art methods on different image compression benchmarks.

The remainder of this paper is organized as follows. In section II, we present and discuss previous works on image-based CS reconstruction. In section III, we explain the proposed approach. In section IV, experimental results and comparisons with State-Of-The-Art (SOTA) methods are carried out. Finally, in section V, we summarize our findings and present some opportunities for future works.

II. Related Work

In this section, we present a CS image reconstruction literature review. We first discuss the existing deep learning-based CS methods. Then we review the recent development of vision transformers for image reconstruction.

A. Deep learning-based CS approaches

Compressing Sensing theory was first proposed in 2004 by David Donoho [1]. Deep learning-based CS approaches have been proposed to solve the CS reconstruction problem through the extraction (learning) of significant features from the input signal itself. Several reconstruction algorithms based on CNNs have been proposed to overcome the complexity of traditional methods. At the outset, Kulkarni et al. [2] developed a non-iterative reconstruction model using CNN (ReconNet). Based on iterative thresholding algorithms, Zhang et al. [3] proposed the convolutional ISTA-Net model for image recovery. Afterward, Shi et al. proposed a Scalable Convolutional Neural Network (SCSNet), and right after proposed a sampling reconstruction framework called CSNet [4], which replaces the sampling model with a convolutional layer. However, these methods have limitations due to their random sampling. To address this problem, Siwang Zhou et al. in [5] have proposed a Block-Based Image Compressive Sensing (BCS-Net), which uses block correlation for sampling. Nevertheless, the model training overlooked the semantic information of the image to draw the prior knowledge. Hence, in order to improve the reconstruction quality by considering the prior knowledge, Wenxue Cui et al. [6] have proposed a non-local CSNet (NL-CSNet) based on non-local self-similarity priors.

However, all the previous methods did not consider perceptual information, which is important for visual and semantic content reconstruction of the images. Recently, in [7], Bairi et al. proposed a perceptual-optimized CS framework that uses perceptual information for image reconstruction. The model is based on an auto-encoder, which is trained using perceptual optimization. Despite its power in the reconstruction of semantic information, this model still lacks high-frequency feature extraction.

B. Transformer-based image reconstruction

The first transformer was proposed by Vaswani et al. [8] for Natural Language Processing (NLP) tasks. In the latter, the long-range dependencies were given by multi-headed self-attention and feed-forward Multi Layer Perceptron (MLP) block. Among the best-known models dealing with this type of task are BERT [9] and GPT [10]. Based on the transformer force in NLP, transformers were recently integrated into the context of image processing. For classification tasks, the innovative work of Vision Transformer (ViT) [11] divides an image into 16 by 16 patches, to use the previous multi-headed self-attention and feed-forward MLP to build a classifier. In addition to the original ViT, transformer models, with different versions and architectures were proposed for several computer vision tasks namely for classification [12], [13], [14], [15], [16], [17], [8], for object detection [18], [19]and for image segmentation tasks [20], [21], [22].

Few works have investigated transformers for image reconstruction. Indeed, this task produces images as a final output, which is more difficult than high-level vision tasks such as classification, segmentation, and object detection, whose outputs are labels or areas. For transformer-based image reconstruction, Hanting et al. [23] proposed a pre-trained model called IPT that can be used for computer image reconstruction tasks. This approach suffers from the large number of parameters and image features are still extracted from CNN. A concurrent work [24] proposed a U-shaped transformer for image reconstruction, which is built upon the UNet architecture and based on the Swin's transformer block. However, these models, based solely on pure transformers, overlook local feature identification and low-frequency information. To preserve the advantages of both CNN-based networks for the local description and the transformer for long-range dependencies handling, Liang et al. [25] proposed a SwinIR model for the restoration of compressed or noisy images based on both Swin transformer blocks and CNNs which were designed for image classification in [15], this model showed better results than those obtained by IPT. Similarly, a transformer-based image reconstruction (TIC) method is developed in [26]. The latter uses a canonical architecture of the VAE variational autoencoder in the form of convolutional layers and Swin transform blocks to capture long and short-term dependencies of the input image. Test results on the Kodark dataset show the good performance of this approach. Dongjie et al. [27] extend the technique of self-attention in compressed sensing to overcome the limitations of convolution layers in modeling global features, by a CSformer model that combines the advantages of CNNs and transformers. The model contains a sampling module as a convolution layer and a reconstruction module in the form of two branches that integrate local and global-range dependencies. Nevertheless, these architectures need the integration of perceptual information, which helps the reconstruction of semantic details of the image.

III. Proposed approach

In this section, we present the proposed PCST-Net framework by using self-attention through vision transformer for better feature extraction and visual perception to make sense of these features. Fig.1 illustrates the proposed approach architecture. Indeed, it is based on CS sampling/reconstruction autoencoder which adopts an attention mechanism to capture long-range contextual information. The learning process is guided by the image's visual content information. The proposed approach involves two neural networks, an encoder, and a decoder. The encoder network compresses the input images by projecting them into a lower-dimensional space, while the decoder network restores the original image representation from the compressed representation. The network is trained in an end-to-end manner to minimize image reconstruction error, allowing it to find the optimal parameters that enable sampling and reconstruction for any input image.

A. Sampling network

The Sampling network (Encoder) is a combination of CNN and transformer models to take advantage of the spatial locality and self-attention mechanisms. The CNN model is inspired by PSCS-Net[7] and is laid out as three Convolution/MaxPooling blocks. In the original CS framework, encoded data are the result of sampling the input image. The latter are called encoded data as they correspond to the rows of the sampled image. In the context of deep learning, the encoded data is arranged rather like an ordinary 3D tensor like any CNN feature map. Theoretically, they still correspond to CS sampled vectors, just stacked in a 3D tensor. When we apply the sampling operator S_{CNN} on the input image x, we obtain y_1, which corresponds to the encoded data obtained by the CNN sampling Network.

$$S_{CNN}(x) = W_s^1 * x \tag{1}$$

In Eq.1, the network operates on 2D image patches with the convolution operator $(*)$ with the sampling Matrix W_s^1. Such an operation projects an input image $x \in R^{d_x}$ onto one of the encoded vectors $y_1 \in R^{d_y}$. The sampling matrix W_s^1 is a composition of convolutions and nonlinear activation functions f that allows for better features extraction. The obtained result y_1 can be written as:

$$y_1 = S_{CNN}(x) = f(W_3 * f(W_2 * f(W_1 * x + b_1) + b_2) + b_3) \tag{2}$$

Transformer-based encoder aims to capture long-range visual dependencies through the self-attention mechanism. It is composed of a projection layer and a transformer block which is the architecture of the ViT backbone [11]. An image projection is a lower-dimensional representation of the image. In other words, it is a dense vector representation of the image. First, the image is divided into P × P non-overlapping patches, then this feature projection layer projects the input patches having a size of (P x P x C) into a dimension of (1 x Pd) such that Pd is the projection dimension. The self-attention mechanism is an integral component of a transformer, which explicitly models the interactions between all entities in a sequence. For an input sequence of Np elements, self-attention captures the interaction between all Np entities and encodes each entity in terms of global contextual information. For this fair, three weight learning matrices are defined, *Queries* ($W^Q \in \mathbb{R}^{Pd*q}$), *Keys* ($W^K \in \mathbb{R}^{Pd*k}$), and *Values* ($W^V \in \mathbb{R}^{Pd*v}$). The input sequence X is projected onto these weight matrices to obtain:

$$\begin{aligned} Q &= XW^Q \\ K &= XW^k \\ V &= XW^V \end{aligned} \tag{3}$$

Self-attention is formulated by:

$$A = softmax(\frac{QK^T}{\sqrt{q}})V \tag{4}$$

Fig.2 shows the transformer block architecture which consists of two LN normalization layers, a multi-headed self-attention layer MSA and a MLP made up of two fully connected layers, the τ norm is inserted before MSA and MLP.

The multi-headed self-attention MSA comprises several blocks of self-attention, each block has its own set of learnable weight matrices *Query*, *key*, and *Value*. Multi-headed self-attention runs h times in parallel, such that h is the number of heads, then concatenated into a single matrix. This block takes a series of sequences I patches of size (Np x Pd) as input and globally calculates the self-attention between them. The whole process of this block can be formulated as follows:

$$\begin{aligned} F_t &= MSA(\tau(I)) + I \\ y_2 = S_{ViT}(x) &= MLP(\tau(F_t)) + F_t \end{aligned} \tag{5}$$

The transformer path is composed of four transformer blocks. Feature fusion aims to extract the most discriminating information and eliminate redundant information. The fusion function combines the global features of the transformer and the local features of the CNN by a fusion strategy, such as addition or average. The fusion of y_1 and y_2 is given by Eq.6.

$$y = Fusion(y_1, y_2) \tag{6}$$

Since the stems of the transformer and the CNN have different dimensions, we need to modify the characteristics of the transformer to match those of the CNN.

B. Reconstruction Network

The upsampling network (Decoder) is designed in [7] as a three-block de-convolutional network to learn the inverse convolution filters to reconstruct images. The decoder returns y to the input space by obtaining the feature representation in the image recovery process. The decoder represents a nonlinear mapping that is learned from measurements y to its original image x by training. The decoder is symmetric with the CNN sampling network and consists of three layers: the input layer and two hidden layers. The decoder function (Eq. 7) is used to

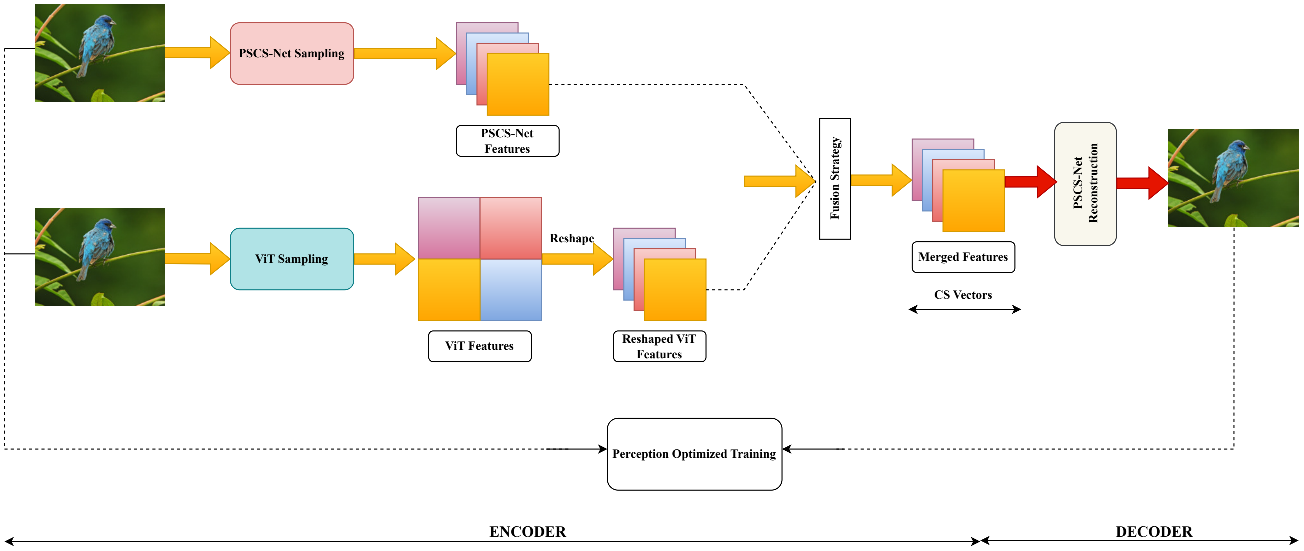


Fig. 1: Overview of the proposed image reconstruction-based framework: the model is trained using the combination of visual perception and self-attention.

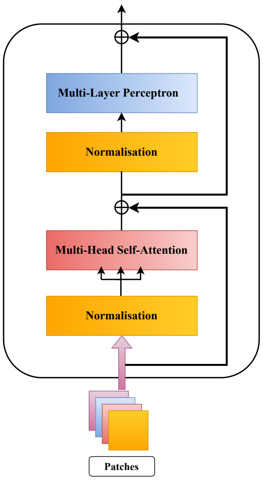


Fig. 2: Architecture of a transformer block

recover the reconstruction images $\widetilde{x}$ from measurement vector y.

$$\widetilde{x} = R(y) = f(W_6 * f(W_5 * f(W_4 * y + b_4) + b_5) + b_6) \quad (7)$$

C. Training of PCST-Net

To semantically guide our model to learn visual and contextual features, we use perceptual loss optimization in the training process as shown in [7]. The used perceptual loss measures the distance between images in high-level feature space using a pre-trained compressing sensing network [4] (CSNet). This model is originally trained on ImageNet dataset. The PCST-Net network is trained in an end-to-end fashion through the minimization of the global loss term expressed as:

$$\mathscr{L}_{total}(x, \widetilde{x}) = \alpha_1 L_p(x, \widetilde{x}) + \alpha_2 L_2(x, \widetilde{x}) + \alpha_3 L_s(W, b) \quad (8)$$

With : L_p in Eq.9 is the perceptual loss, L_s in Eq.10 is the sparsity loss, and L_2 in Eq.11 is the L_2 Norm between the original and reconstruted image. The three terms are weighted by α_1, α_2, and α_3, respectively.

$$L_p(x, \widetilde{x}) = MSE(\phi(x) - \phi(\widetilde{x})) \quad (9)$$

Where ϕ is the sampling operator of CSNet to compute the difference between the feature vector of the input image x and the predicted image $\widetilde{x}$.

$$\mathrm{L}_s(W_s^1, b) = 1/2\beta_1 \sum ||W_s^1||^2 + \beta_2 \sum_{j=1} KL(\rho||\rho_j) \quad (10)$$

The first term of L_s in Eq.10 limits the weight parameters W with L_2 norm as to penalize large weight. The second term is the sparsity regularizer. β_1 is the penalty term and KL is the Kullback-Leibler divergence for penalizing active code units. β_2 is the intensity of the sparsity, ρ is the sparse factor, and ρ_j represents the mean value of activation of the j^{th} neuron in each batch of the training set.

$$L_2(x, \widetilde{x}) = \sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (x(i,j) - \widetilde{x}(i,j))^2} \quad (11)$$

The L_2 Norm is used to profit from the qualities of pixel-wise loss functions.

The goal of training PCST-Net model is to minimize $\mathscr{L}_{total}$ as shown in Algorithm 1. First, parameters W_s^1, W^Q, W^K, and W^V are randomly initialized to serve the purpose of symmetry breaking. Then, encoded data y and the reconstruction images $\widetilde{x}$ are obtained through the encoder and decoder subnetworks, respectively.

Algorithm 1 PCST-Net training

```
Input:
Input original image $x$
Output:
Sampling Network weights $W_s^1$, $W^Q$, $W^K$, and $W^V$
Encoded data $y$
Reconstruction Network weights $W_r$
Instructions:
$W_s$, $W_r$ : Randomly initialize
for   epoch = 1 to number of epochs
$y_1 = S_{CNN}(x)$
$y_2 = S_{ViT}(x)$
$y = Fusion(y_1, y_2)$
$\widetilde{x} = R(y)$
Compute encoded image $y$
Compute perceptual loss $\mathscr{L}_{total}(x, \widetilde{x})$ (Eq.8)
Minimize final loss by gradient descent algorithm
Update $W_s^1$, $W^Q$, $W^K$, $W^V$, and $W_r$
end for
```

IV. Experiments and Results

In this section, we first introduce the dataset used for PCST-Net model training and the evaluation metrics. Second, we present the model settings for better training (Section IV-B). Next, in section IV-C, we conduct an experimental study on image compression benchmarks for model objective evaluation and compare the proposed approach with state-of-the-art methods. Finally, in Section IV-D, we evaluate the quality of PCST-Net image reconstruction with a subjective evaluation.

A. Datasets and evaluation metrics

PCST-Net is trained using a large-scale dataset which is COCO 2017 dataset [1]. 118k and 40k images have been used for training and validation respectively.

We evaluate our PCST-Net on different widely used benchmark datasets, such as Set5 [28], Set14 [29], and BSD100 [30].

To evaluate the model, two metrics are computed: Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM). PSNR measures image reconstruction quality, while SSIM, a perceptual metric, quantifies image degradation.

B. PCST-Net model settings and training

1) Model hyper-parameters selection: After an axial empirical study, the hyper-parameters of the model are set to 256 x 256 x 3, 8 x 8 x 3, and 8 x 8 x 3 for Image size, Patch size, and Block size, respectively. The perceptual loss function is optimized using the Adam optimizer with a batch size equal to 32 and a learning rate of 0.002 for 100 epochs.

2) Fusion strategy selection: Our method adopts an addition strategy to merge the features of different paths. To illustrate the effectiveness of this method, we construct a variant in which the features of the CNN and the transformer are averaged rather than summed.

Fig.3 shows the PSNR results of the two models on Set5, Set14, and BSD100. The feature addition fusion operation shows superior PSNR performance with different compression ratios. The feature averaging operation achieves a close performance when the compression ratios are lower than 10%, but above this compression ratio, the addition shows its efficiency against the average.

3) Path selection: **PCST-Net** is a **Dual Path** model that aims to combine the efficiency of convolution in extracting local features with the capability of the transformer in modeling global representations. To compare the advantages of the two branches-based approach, we created a **Single Path** model called **SPCST-Net**, which uses only the transformer path for compression. The results of the tests on three datasets (Set5, Set14, and BSD100) are presented in Fig.4.

Obtained results on Set5, Set14, and BSD100 datasets confirm that PCST-Net helps in recovering more details and semantic information of the images compared to PSCS-Net (based only on CNN) or SPCST-Net (based only on transformers).

C. Objective Evaluation

The results of the comparative study of PSNR and SSIM, between the different state-of-the-art reconstruction methods namely ISTA-Net+[3], CSNet+[4], NL-CSNet*[6], DPA-Net[31], CSFormer[27], PSCS-Net[7], and our PCST-Net, applied on the Set11, Set5, and BSD100 reconstruction datasets are shown in Table I-III, while varying the compression ratio between 0.1 and 0.5.

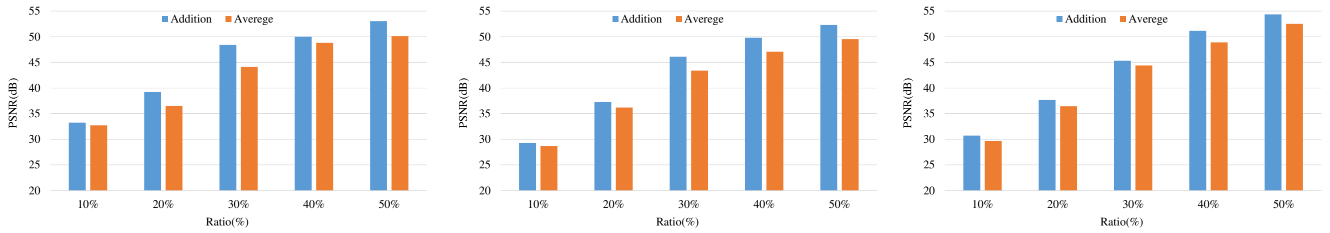
Our experimental results show that our approach achieves higher performance for image reconstruction compared to state-of-the-art algorithms.

TABLE I: Comparaison of PSNR(dB) and SSIM on Set5

Algorithm/Ratio	0.1	0.2	0.3	0.4	0.5
ISTA-Net+[3]	28.61 0.8315	33.12 0.9058	35.45 0.9408	36.94 0.9612	38.42 0.9804
CSNet+[4]	32.59 0.9062	36.05 0.9481	38.25 0.9644	40.11 0.9740	41.79 0.9803
NL-CSNet*[6]	33.84 0.9312	36.91 0.9589	38.86 0.9703	41.20 0.9895	43.15 0.9942
DPA-Net[31]	30.32 0.8713	-	36.17 0.9495	38.05 0.9632	39.57 0.9716
CSFormer[27]	34.20 0.9262	36.88 0.9514	39.74 0.9689	-	43.55 0.9845
PSCS-Net[7]	**33.75** **0.9422**	38.68 **0.9893**	47.10 **0.9946**	49.92 0.9950	52.27 0.9973
Ours	33.25 0.9387	**39.19** 0.9747	**48.40** 0.9899	**50.02** **0.9955**	**53.04** **0.9975**

The results obtained by PCST-Net on the different compression datasets benefited from the coupling between perception and self-attention to give the best PSNR and SSIM values compared to other state-of-the-art reconstruction methods.

[1] https://cocodataset.org/home



(a) (b) (c)

Fig. 3: PSNR(dB) histogram for each fusion strategy on Set5(a), Set14(b), and BSD100(c).

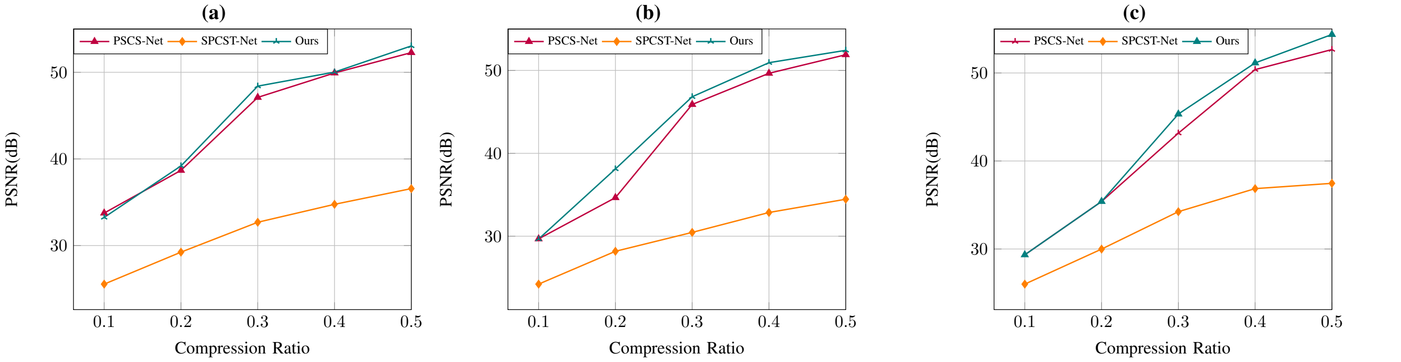


Fig. 4: Average PSNR(dB) for different Path-based methods on Set5(a), Set14(b), and BSD100(c).

TABLE II: Comparaison of PSNR(dB) and SSIM on Set14

Algorithm/Ratio	0.1	0.2	0.3	0.4	0.5
ISTA-Net+[3]	26.49 0.8010	30.79 0.8950	33.76 0.9345	36.03 0.9547	38.49 0.9127
CSNet+[4]	29.13 0.8169	32.15 0.8941	34.34 0.9297	36.16 0.9502	37.97 0.9754
NL-CSNet*[6]	30.16 0.8527	32.96 0.9150	34.88 0.9405	37.21 0.9752	40.17 0.9891
DPA-Net[31]	27.22 0.8401	31.51 0.9249	33.37 0.9395	35.91 0.9592	37.84 0.9701
CSFormer[27]	30.85 0.8515	34.02 0.9274	36.47 0.9459	-	40.41 0.9730
PSCS-Net[7]	**29.68** **0.8987**	34.65 0.9644	45.89 0.9920	49.65 0.9967	51.89 0.9981
Ours	29.31 0.8921	**37.25** **0.9720**	**46.11** **0.9929**	**49.81** **0.9967**	**52.30** **0.9985**

TABLE III: Comparaison of PSNR(dB) and SSIM on BSD100

Algorithm/Ratio	0.1	0.2	0.3	0.4	0.5
ISTA-Net+[3]	24.79 0.6726	27.64 0.7906	29.86 0.8580	31.70 0.9003	33.02 0.9513
CSNet+[4]	28.53 0.7834	31.05 0.8721	33.08 0.9171	34.91 0.9443	36.68 0.9618
NL-CSNet*[6]	28.61 0.8361	31.20 0.9141	33.30 0.9354	36.91 0.9627	39.94 0.9845
DPA-Net[31]	26.47 0.7388	29.87 0.8611	30.23 0.8894	32.70 0.9241	34.19 0.9488
CSFormer[27]	28.28 0.8078	31.62 0.9110	33.57 0.9399	-	38.01 0.9712
PSCS-Net[7]	29.34 0.8884	35.40 0.9632	43.16 0.9924	50.38 0.9969	52.66 0.9982
Ours	**30.71** **0.9044**	**37.72** **0.9680**	**45.33** **0.9932**	**51.14** **0.9971**	**54.37** **0.9989**

D. *Subjective Evaluation*

In this section, we describe the subjective evaluation to visualize the quality of reconstructed images. This qualitative assessment is done with the naked eye by noting the differences between images at a ratio of 0.25. We also provide PSNR and SSIM values for each image to highlight quantitative differences.

The visualization obtained by PCST-Net in Fig.5 shows again that the use of both perception and self-attention gives the best result compared to other reconstruction methods. Obtained results suggest that the combination of self-attention and perceptual optimization can provide a powerful tool for improving the quality of image reconstruction. The use of self-attention mechanisms to capture long-range dependencies in the image data can lead to better sampling performance, while the incorporation of perceptual optimization can enhance the perceptual quality of the reconstructed images.

V. Conclusion

In this paper, we proposed a novel approach for image sampling and reconstruction that combines Vision Transformer and perceptual optimization techniques. Our approach leverages the power of self-attention to capture the global context of the image and guide the sampling process while optimizing the perceptual quality of the sampled image using a perceptual loss function. We have demonstrated the effectiveness of our proposed approach through experiments on several benchmark datasets, and we have shown that it outperforms existing state-of-the-art methods in terms of reconstruction quality

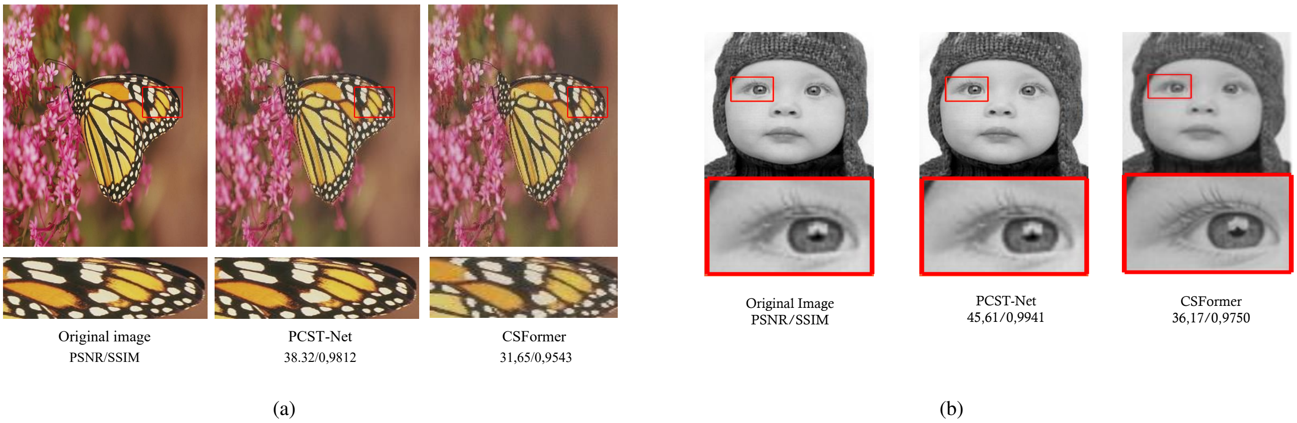


Fig. 5: Comparison of the visual quality of image reconstruction using a ratio of 0.2(a) and 0.4(b).

and visual fidelity. Our approach has potential applications in a wide range of domains, including medical imaging, video processing, and computer graphics. In conclusion, our work contributes to the development of efficient and effective techniques for image sampling and reconstruction, which are critical components in the field of multimedia processing. We believe that our proposed approach can serve as a foundation for future research in this area, and we hope that it will inspire further innovations in the field of computer vision.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 449–458.

[3] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1828–1837.

[4] W. Shi, F. Jiang, S. Liu, and D. Zhao, "Image compressed sensing using convolutional neural network," *IEEE Transactions on Image Processing*, vol. 29, pp. 375–388, 2019.

[5] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multi-channel deep networks for block-based image compressive sensing," *IEEE Transactions on Multimedia*, 2020.

[6] W. Cui, S. Liu, F. Jiang, and D. Zhao, "Image compressed sensing using non-local neural network," *IEEE Transactions on Multimedia*, 2021.

[7] Z. Bairi, O. Ben-Ahmed, A. Amamra, A. Bradai, and K. B. Bey, "Pscs-net: Perception optimized image reconstruction network for autonomous driving systems," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[8] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, "Scaling local self-attention for parameter efficient visual backbones," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 894–12 904.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," *Advances in neural information processing systems*, vol. 32, 2019.

[13] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International conference on machine learning*. PMLR, 2020, pp. 1691–1703.

[14] B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda, "Visual transformers: Token-based image representation and processing for computer vision," *arXiv preprint arXiv:2006.03677*, 2020.

[15] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 012–10 022.

[16] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021.

[17] Y. Liu, Y.-H. Wu, G. Sun, L. Zhang, A. Chhatkuli, and L. Van Gool, "Vision transformers with hierarchical attention," *arXiv preprint arXiv:2106.03180*, 2021.

[18] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[19] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, pp. 261–318, 2020.

[20] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6881–6890.

[21] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*. Springer, 2023, pp. 205–218.

[22] Y. Xie, J. Zhang, C. Shen, and Y. Xia, "Cotr: Efficiently bridging cnn and transformer for 3d medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*. Springer, 2021, pp. 171–180.

[23] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 299–12 310.

[24] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 683–17 693.

[25] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2021, pp. 1833–1844.

[26] M. Lu, P. Guo, H. Shi, C. Cao, and Z. Ma, "Transformer-based image compression," *arXiv preprint arXiv:2111.06707*, 2021.

[27] D. Ye, Z. Ni, H. Wang, J. Zhang, S. Wang, and S. Kwong, "Csformer: Bridging convolution and transformer for compressive sensing," *arXiv preprint arXiv:2112.15299*, 2021.

[28] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[29] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, 2010, pp. 711–730.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 416–423.

[31] Y. Sun, J. Chen, Q. Liu, B. Liu, and G. Guo, "Dual-path attention network for compressed sensing image reconstruction," *IEEE Transactions on Image Processing*, vol. 29, pp. 9482–9495, 2020.

DOI: 10.15439/2023F997
ISSN 2300-5963 ACSIS, Vol. 35

On combining image features and word embeddings for image captioning

Mateusz Bartosiewicz, Marcin Iwanowski, Martika Wiszniewska, Karolina Frączak, Paweł Leśnowolski
Institute of Control and Industrial Electronics, Warsaw University of Technology
ul. Koszykowa 75; 00-662 Warszawa POLAND
mateusz.bartosiewicz.dokt@pw.edu.pl; marcin.iwanowski@pw.edu.pl

***Abstract*—Image captioning is the task of generating semantically and grammatically correct caption for a given image. Captioning model usually has an encoder-decoder structure where encoded image is decoded as list of words being a consecutive elements of the descriptive sentence. In this work, we investigate how encoding of the input image and way of coding words affects the result of the training of the encoder-decoder captioning model. We performed experiments with image encoding using 10 all-purpose popular backbones and 2 types of word embeddings. We compared those models using most popular image captioning evaluation metrics. Our research shows that the model's performance highly depends on the optimal combination of the neural image feature extractor and language processing model. The outcome of our research are applicable in all the research works that lead to the developing the optimal encoder-decoder image captioning model.**

***Index Terms*—image captioning, neural image feature extractors, embedding models, LSTM**

I. Introduction

IMAGE captioning is a task of generating a verbal description of an image. It combines Natural Language Processing (NLP) and Computer Vision. Image captioning solutions are used in many application areas. They are adapted for content-based image retrieval or automated labeling of online images. Also, in the human-machine interaction field, they are used to assist visually impaired people in understanding the surrounding world or to search fast for photos on the internet.

We focus in this paper on the baseline captioning model [24] consisting of encoder and decoder. Encoder extracts a pair of image and text features in parallel. Text features encoder is responsible for the dense representation of each word in embedding space providing semantic context for each token. Image encoder uses convolutional neural network (CNN) backbone which extracts high-level image features. Decoder combines image and text features and generates the resulting image caption. It is based on the long-short term memory (LSTM) module [18], that generates the descriptive sentence word-by-word.

In this work, we improve the effectiveness of the baseline image captioning model by changing the encoding of the input data. We assume that different image features extractors, even pretrained on the same training set, provide with various high-level knowledge of the image content and similarly, different language processing models extract different semantics of captions.

During experiments, we investigated how different encoding of an image and text influence the captioning accuracy. We tested several backbone models based on pretrained CNN networks and embedding schemes as image and language inputs, respectively. It allowed us to investigate which pairs work best, hence finding the optimal combination of neural image feature extractor and language processing model.

As a result, we achieved 20 models trained on CNN networks: Xception, InceptionV3, Resnet152V2, Resnet50, VGG16, VGG19, DenseNet121, DenseNet201, MobileNet, MobileNetV2 along with Glove and FastText embeddings. For training and testing we used MSCOCO 2014 dataset and as the evaluation metrics: BLEU, METEOR, CIDEr, SPICE, ROGUE-L, WMD. Finally, thanks to the mentioned metrics, we assessed which pairs of image features and embeddings produce better results on the baseline image captioning model.

This paper is organized as follows. Section II describes how image captioning methods evolved from template-based techniques to deep neural architectures. Next, in section III, we describe how our base image captioning model is built and what neural image features extractors and language embedding models we use. The experimental procedure applied in our research is presented in Section IV. Section V have experimental analysis and finally, the final conclusions are found in Section VI.

II. Previous works

Image captioning methods combining text and visual data belong to the multi-modal machine-learning approaches [22], [40], [59]. Captioning models can be divided into traditional and deep-learning-based. Originally, traditional image captioning methods were based on hard-coded rules and human-made features. In [27], [29], [36], authors applied fixed templates with blank slots filled with various objects, descriptive tokens and situations extracted from images by the object detection systems. On the other hand, in [12], authors used already existing, predefined sentences. They created space of meaning from images features and compared images with sentences to find the most appropriate sentences for a photo. Despite semantic and grammatical correctness, captions from traditional methods differ often from the way a human described the image content.

Deep learning image captioning methods tries to overcome those limitations. In pioneering work [25] authors suggested that neural networks can interpret deep semantics of images and word embeddings. They proved that combined image features extracted by the convolutional neural networks (CNN) and word embeddings could hold semantic meaning. In [11], authors suggested passing image features and text features sequentially and individually to the language model. Inspired by the success in machine translation, [51] proposed using an encoder-decoder framework in image captioning, which has recently become dominant in the image captioning field.

Paper [24] by Karpathy et al. introduced architecture similar to human perception. Method generates novel descriptions over image regions, with R-CNN (Regional Convolutional Neural Networks) [13] for image feature extraction and recurrent neural network (RNN) to iteratively generate consecutive words of caption. Model using the multimodal embeddings space tries to find the parts of the sentence that best fits the image regions. Differently from other proposed methods ([9], [25], [51]), where a global image vector was used, Karpathy focused on image regions, and a separate caption described each region. Finally, a spatial map generates the target word for image regions. These image captioning approaches, focusing on generating captions for each region of an image, are called dense image captioning [23], [49], [54].

Encoder-decoder architecture [2], [15], [51], [55] considers the task of image captioning as the sequence-to-sequence problem. Encoder encodes the image to the fixed length vector using the image features extractor. Most widely used are CNN networks as VGG [14], [32], [45], ResNet [32], [34], [56] or Inception [10], [53]. Decoder, which in image captioning is represented by a language model, generates natural language descriptions to the output. Most popular approaches used RNN. However, due to the vanishing gradient problem that occurs in long sequence tasks, LSTM which is a variation of RNN achieved better results [18]. Most popular encoder-decoder approaches are the CNN-RNN [33], [41], [51] and GRU [8].

During the rapid development of image captioning methods, researchers also investigated other aspects of captions than just comparability to human judgment. Researchers focused on captions with a specific style. In [2], authors improved the descriptiveness of generated captions by combining CNN and LSTM. In [52], authors focused on captions for visually impaired people. Developed model tends to create captions that describe the surrounding environment.

III. Preliminaries

A. Image captioning model

Image captioning encoder-decoder model investigated in this study is depicted in Fig. 1. Encoder consists of two parts working, in the learning phase, simultaneously. One is for handling image features and another is for handling words in sequences. Firstly, image features are extracted using one of the image features extractors described in the next section. They are processed by a dense (fully connected layer) layer with ReLU (rectified linear unit) activation functions [37]. Its usage was motivated by promising results in very deep vision neural networks [17]. Compared with non-linear functions like sigmoid, ReLU is faster and harder to overfit. Dense layer is responsible for reducing the dimension of the image feature space (i.e. the length of the feature vector) to 256 to match the size of the word sequence prediction output.

In parallel, the text input (caption) is transformed into the sequence of indices of consecutive sentence words. Although the length of a caption varies, the length of vector of indices is constant and equal to 51, which is the maximum sentence lenght (i.e. number of words in the longest caption sentence). Such a vector is fed to the embedding layer. It encodes the semantic meaning of words represented by vectors in embedding space. We used pretrained Glove and FastText embeddings as two alternative ways of encoding the consecutive words of a descriptive sentence. Thanks to the embeddings layer, we reduced the text features size from the vocabulary size to the vectors of embeddings. Embedding vectors are passed through a long-short term memory (LSTM) model of size 256. After the LSTM layer, the outputs of language model and the image part of the image captioning model are added and finally forwarded to the decoder consisting of two dense layers.

Long-short term memory (LSTM) was designed for long-sequence problems and can predict next word in the sequence based on its predecessors. Each LSTM unit consists of three gates, that control and monitor the information flow in LSTM cells. Forgetting gate decides, which information from previous iteration will be stored in the cell state or is irrelevant and can be forgotten. In the input gate, the cell attempts to learn new information. It quantifies relevance of new input value of the cell and decides to process it or not. Output gate transfers the updated information from the current iteration to the next iteration. State of the cell also contains the information along with a timestamp.

Decoder processes an image feature vector and a sequence vector to predict captions. Following two dense layers processes, added language and image model to reduce the number of features to the vector of size equal to vocabulary size. Finally, the softmax layer generates the probability distribution of the next word in the sequence and selects the word with maximum probability. Previous words are converted to embeddings during training to develop the next word. Image feature vector is fed to the decoder. Goal of the training is to minimize loss function based on the error between target and predicted words.

Trained model predicts captions word-by-word, where the prediction of the next word is based on the previously generated one and image features. At each iteration, greedy search algorithm looks for the word in the dictionary with the highest probability of following words in the sequence. Process continues till the end of the caption is detected or the max length of the caption is achieved. Greedy search takes only tokens with the highest possibility of occurring in the final sequence based on previously generated tokens.

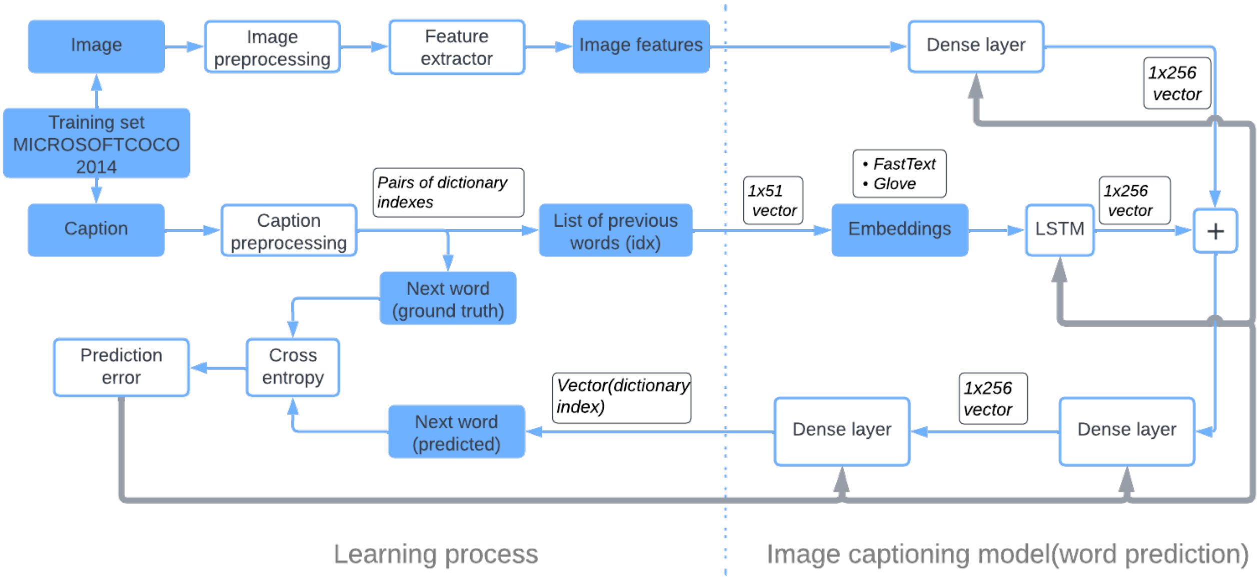


Fig. 1: Diagram of image captioning model training process.

B. Neural image feature extractors

Image features are essential in image captioning. In our experiments we used backbone CNN networks pretrained on a large number of images, the backbone networks. It makes possible to focus on the captioning model and restrict training to the remainder of the model.

The VGG [44] is a group of convolutional neural networks (CNNs) widely used for image classification tasks. Most popular variants are VGG16 and VGG19. VGG16 consists of 13 convolutional and 3 dense layers and was trained to recognize 1000 object classes referring to objects depicted on input 224x224x3 color images. By cutting out the dense layers, the backbone network that produces the image feature vector of length 4096 has been obtained. VGG19 has 3 more CNN layers than VGG16. Thanks to this, allows to learn richer representations of the data and achieves higher prediction results. On the other hand, VGG19 is more exposed to the vanishing gradient problem, than VGG16 and requires more computational power.

The Resnet [16] network was created to support many layers while preventing the phenomenon of vanishing gradient in deep neural networks. Most popular variants are Resnet18, Resnet50, and Resnet100, where the number represents a number of layers. Network architecture is built among two stages. In the beginning, the stack of skip connections is built. Those layers are omitted and the activation function from the previous layer is used. In the next stage, the network is learned again, layers are expanded and other parts of the network (residual blocks) learn deeper features of the image. Residual blocks are the heart of residual convolutional networks. They add skip connections to the network, which preserve essential elements of the picture till the end of the training, simultaneously allowing smooth gradient flow.

The Inception [47] model was created to deal with overfitting in very deep neural networks by going wider in layers rather than deeper. It is build among inception blocks that process input and repetitively passes the result to another inception block. Each block consists of four parallel layers 1x1, 3x3, 5x5, and max-pooling. 1x1 is to reduce dimension by channel-wise pooling. Thanks to that network can increase in depth without overfitting. Convolution is computed between each pixel and filter in the channel dimension to change the number of channels rather than the image size. 3x3 and 5x5 filters learn spatial features of the image in different scales and act similarly to human perception. Final max-pooling reduces the dimensions of the feature map. Most popular versions of the Inception network are Inception, InceptionV2 and InceptionV3.

The InceptionV3 [48] incorporated the best techniques to optimize and reduce the computational power needed for images features extraction in the network. It is a deeper network than InceptionV2 and Inception, but its effectiveness was not compromised. Also, use auxiliary classifiers that improve the convergence of very deep neural networks and combat the vanishing gradient problem. Factorized convolutions were used to reduce the number of parameters needed in the network and smaller asymmetric convolutions allowed to fasten computations.

The Xception [6] is a variation of an Inception [47] model that decouples cross-channel correlations and spatial correlations. Architecture is based on depthwise separable convolution layers and shortcuts between convolution blocks, as in Resnet. It consists of 36 convolutional layers divided into 14 modules. Each module is surrounded by residual connections, except the first and last module. It has a simple and modular architecture and achieved better results than VGG16, Resnet and InceptionV3 in classical classification challenges.

The backbone networks based on the three above ones, in contrast to the VGG16, produce the image feature vector of length 2048.

DenseNet [21] Network was created to overcome vanishing gradient problem in very long deep neural networks, by simplifying data flow between layers. Architecture is similar to Resnet, but thanks to the simple change in connection between layers, DenseNet allow to reuse parameters within network and produce models with high accuracy. Structure of DenseNet is based on stack of connectivity, transition and bottleneck layers, grouped in dense blocks. Every layer is connected, with every another layer in dense way. Dense block is main part of DenseNet and reduces the size of feature maps by lowering their dimensions. In each dense block dimensions of feature maps are constant, but number of filters change. Between each dense block, transition layer is placed to concatenate all previous inputs, hence reduce number of channels and number of parameters needed in the network. Also, between every layer bottleneck layer is placed to reduce number of inputs especially in far away layers. DenseNet also introduced growth rate parameter to regulate quantity of information added in each layer. Most popular implementations are DenseNet121, DenseNet201, where number denotes quantity of layers in the network.

MobileNet [20] is a small and efficient CNN Network especially designed for mobile computer vision tasks. It is built of layers of depthwise separable convolutions, composed of depth-wise and point-wise layers. MobileNet also introduced width multiplier and resolution multiplier hyperparameters. Width multiplier allows to decrease computational power needed during training, resolution multiplier decreases the resolution of the input image during training. Most popular versions of MobileNet are MobileNetV1 and MobileNetV2. In comparison with MobileNet, MobileNetV2 introduced inverted residual blocks and linear bottlenecks. Also, Relu activation function was replaced by Relu6 (ReLu with saturation at value 6). Thanks to that accuracy of the model significantly improved.

C. Word embedding models

Word embeddings are vector representations of tokens that are fed to a deep learning model. The most common embedding systems used for natural language processing and image captioning are Glove, Word2Vec and FastText.

One of the first word embedding techniques was one-hot encoding, where each token is encoded to the binary vector representation. Method is based on the dictionary created for all unique tokens in the corpus. A fixed-length binary vector with the size of a dictionary represents each word. Index of the word in the vector represents presence. If a word is present in with vector, just one value is one and others are 0. It is a straightforward technique that captures a wide variety of words but misses the semantic relation of words. Furthermore, fixed-length vectors are sparse, which is not computationally efficient.

Computationally efficient, Word2Vec [35] method simultaneously captures semantic relations between words. It is based on two techniques: CBOW (Continuous Bag of Words) allows the prediction of words from the context word list vector and the Continuous Skip-Gram model, a simple one-layer neural network that predicts context based on a given word.

FastText [4] comes from the Word2Vec model but analyzes words as n-grams. An algorithm is similar to the CBOW from Word2Vec but focuses on a hierarchical structure, representing a word in a dense form. Each n-gram is a vector and the whole phrase is a sum of those vectors. To achieve a word embeddings vector, training is similar to the CBOW.

Glove [39] word embeddings are based on unsupervised learning to capture words that occur together frequently. Thanks to the global and local statistics, it creates semantic relations in the whole corpus. Furthermore, it uses global matrix factorization to represent the word of lack of words in the document. It is also called the "count-based model" because Glove tries to learn how the words co-occur with other words in the corpus, allowing it to reflect the meaning of the words conditionally of the other words.

D. Text evaluation metrics

Image captioning is a task that belongs to both computer vision and natural language processing (NLP) domains. It must capture objects, the relations between them and the whole scene context to produce readable sentences in natural language. Due to the complexity of the image captioning results, the evaluation of the image captioning is still a complicated and comprehensive problem.

Evaluation metrics in image captioning measure the correlation of generated captions with human judgment. They estimate grammatical correctness, the complexity of the description and how generated caption generalizes the corresponding image. Evaluation metrics apply their own technique for computation and have distinct advantages. Standard evaluation metrics for image captioning are BLEU-1 to BLEU-4, METEOR, ROUGE-L, SPICE, and WMD [43]. They calculate word overlap between candidate and reference sentences and range it between 1-100. Higher values indicated better results.

BLEU (Bilingual Evaluation Understudy) [38] metric measures the correlation between predicted and human-made captions. It compares n-grams in predicted and reference sentences, where more common n-grams result in higher metric values. It is worth mentioning that metric exclusively count n-grams, locations of the n-grams in sentences are not considered. Metric also allows addition weights for specific n-grams to prioritize longer, common sequences of words. Usually, the 1 to 4-grams used when computing the metric – the respective variants are called BLEU-1 up to BLEU-4.

METEOR (Metric for Evaluation of Translation with Explicit Ordering) [3] measures the correlation between the predicted caption and human judgment. Compared with BLEU, parts of the sentence are analyzed, not the whole corpus. METEOR algorithm have two stages. At first, tokens from reference captions and candidates are compared. In the second stage final result is calculated. METEOR also analyzes and allows synonyms.

CIDEr (Consensus-based Image Description Evaluation) [50] metric calculates correspondence between candidate and reference captions. It is based on the TF-IDF metric, calculated for each n-gram. It is widely used for SCST [41] training, where the strategy is to optimize the model for a specific metric. It results in higher results during the testing phase compared with [41]. Furthermore, CIDEr optimization during training impacts on high scores in BLEU, METEOR and SPICE metrics.

ROGUE-L (Recall-Oriented Understudy for Gisting Evaluation) [30] is a set of metrics: Recall, F1 and precision. Algorithm finds the longest common sequence of tokens between predicted and reference captions. Sequences must be in the same order but not next to each other.

WMD (Word Mover's Distance) [26] is based on a machine learning model to count similarity between texts. Metric is distinguished from others because it measures common sense between texts. It does not investigate the occurrence of tokens. Instead, it measures the semantic length between sentences by counting the probability of the occurrence of synonyms.

All the above metrics are used in various NLP tasks. However, according to some investigations [7], they do not correlate with a human judgment, what makes them not adequate to measure the similarity of image captions [1]. Among the known metrics the one that correlates with the human judgment is SPICE (Semantic Propositional Image Caption Evaluation) [1]. This metric measures similarity between sentences, represented by a directed graph. SPICE algorithm at the beginning creates two directed graphs. First one is for all reference captions and the second is for the candidate sentence. Graphs elements can belong to three groups. First group is objects and activity performers, the second group consists of descriptive tokens (adjectives adverbs) and the last group represents relations between objects and links other groups of tokens on the graph. Based on this representation, sentences are compared.

IV. Experimental setup

A. Datasets

There are several datasets used for image captioning. They differ in the number of images and their size, also captions can vary in format and length. Most commonly used are Flickr8k [19], Flickr30k [58] and MSCOCO 2014 [5], [31]. All these sets consist of a number of images with associated captions, usually 5 per image.

Dataset Flickr30k includes 30k images and each photo has five captions. Training set consists of 29k images and 1k is for testing. Flickr8k is a subset of Flickr30k and contains 8k images, with five annotations for each picture. Each caption fully describes a scene and is entirely based on human judgment. In the test split, there are 7k images and the rest of the data is used for testing.

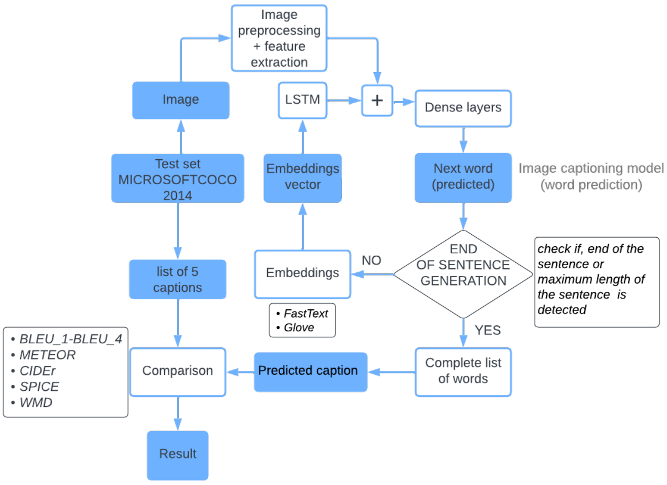


Image preprocessing + feature extraction
Image
LSTM
+
Dense layers
Test set MICROSOFTCOCO 2014
Embeddings vector
Next word (predicted)
Image captioning model (word prediction)
list of 5 captions
Embeddings
NO
END OF SENTENCE GENERATION
check if, end of the sentence or maximum length of the sentence is detected
• FastText
• Glove
YES
• BLEU_1-BLEU_4
• METEOR
• CIDEr
• SPICE
• WMD
Comparison
Predicted caption
Complete list of words
Result

Fig. 2: Diagram of experimental setup.

During our experiments, we used for the evaluation and training the MSCOCO dataset. It consists of more than 120k images from various everyday scenes. Five captions describe each photo in natural language. In the image captioning area, the most popular MSCOCO data partitioning for testing, validation and training purposes is Karpathy split [24], where there are 113k images in training, 5k in validation and 5k in test disjoint subsets.

B. Image preprocessing

Motivated by [24], and considering the variety of available pretrained object detection CNN models and language processing models, we conducted experiments to check how input data to the model can affect the learning process. The whole experimental process involves encoding images and text features simultaneously and generating a final sequence of tokens (caption) word by word during decoding.

Images from the dataset are resized and normalized before entering the image captioning model to be compatible with one of the CNN networks. For VGG16, VGG19, Resnet152V2, Resnet50, DenseNet121, DenseNet201, MobileNet, MobileNetV2 input shape is 224x224x3 and 299x299x3 for InceptionV3, Xception. As a result, we obtained features vectors with the following sizes, corresponding to the preprocessed input image: 4096-element vector for VGG16, VGG19; 2048 for InceptionV3, Xception and Resnet152V2; size 1024 for denseNet121; size 1920 for DenseNet201; 1000 elements for MobileNet; size 1280 for MobileNetV2. We used CNN models pretrained on the ImageNet [42] dataset, where the network's fully connected layers is removed since we do not need

[1] The authors of [7] propose their own metric, but due to much its much lesser (more that 10x) popularity comparing with SPICE, we decided to use that latter in the current study.

the probability distribution on 1000 image categories from ImageNet.

C. Text preprocessing

A separated preprocessing was performed for captions. At the beginning, all words were converted to lowercase, tokenized. We removed punctuations, hanging single-letter words and discarded rare words that occurred less than five times. As a result, we achieved the following vocabularies, also called dictionaries: Flickr8k, Flickr30k, and MSCOCO 2014, that will be used to create embedding matrixes from embedding vectors. Before being handled by LSTM Network, word sequences must be represented in word embeddings vectors. In our model, Glove and FastText have been used as embedding.

Preprocessed captions, consumed by the captioning model, are appended with *start* and *stop* tokens to mark the beginning and end of the sentence, respectively. In the next step, a vocabulary of all words occurring in the captions in the training set is prepared (along with *start* and *stop* tokens). As a result, a dictionary of all words in our corpus is produced to identify tokens by index explicitly. Each generated word is processed by embedding prior to its providing into the LSTM model input.

We adopted pretrained versions of FastText and Glove to extract the text features. We preprocessed sentences from the train and test dataset (described in the previous section) and finally achieved a vocabulary of size 7293. Each word is then embedded to a 200-element vector for Glove and 300-element vector for FastText word embedding space.

D. Training and testing

During training, the model processes combined 256-element vector of word embeddings and image feature vectors based on the CNN model for a given image. At each time step model predicts a word for the processed image and compares it with the ground truth word from the training set, which corresponds to the processed image. Predicted word and ground truth word (from the training set) are compared using the cross-entropy measure (see Fig. 1).

During the testing, image captioning model is fed by a preprocessed photo. In the beginning, at the 0-time step, there is no previously predicted word. Therefore, to denote the start of prediction, a start of sentence token *start* is used. Words are served as the embeddings, corresponding to the dictionary. Next, the image captioning model predicts words recursively until the sentence's end (marked by *stop* token) or the maximum length of the sentence has been reached and adds it to the word list. At each step, the chance of the occurrence of one word next to another is calculated using embeddings specific to the tested text features. Finally, a full caption for the tested image is generated and compared with ground-truth phrases for the tested image, using specific metrics.

E. Evaluation

We investigated the performance of each image encoder, with each text encoder mentioned previously, with BLEU-1 – BLEU-4, METEOR, ROGUE-L, WMD, CIDEr, and SPICE metrics. The complete process is repeated for other CNN architectures and embedding methods to achieve a comprehensive perspective of the performance of different CNN architectures along with different embedding methods. Backbone-embedding pairs tested during experiments are presented in Table. I. The complete process of evaluation is presented in Fig. 2.

For further analysis, we also examined word and bigrams occurences from a training set and predicted captions to determine why some captions are incorrectly generated and what are the collocations of a training set with the parts of the sentence that do not describe the real image content.

V. Results

Table I shows the results of image captioning metrics calculated for different image and text features extractors. We analyzed all models accordingly to the BLEU-1 – BLEU-4, METEOR, ROUGE-L, WMD, CIDEr, and SPICE metrics. Following the literature, to evaluate the performance we used most recent CIDEr and SPICE metrics, keeping the remainder for comparative purposes. For the same purposes we added four reference methods in last four rows of the table.

From the obtained results, we can see that model performance depends mostly on the CNN backbone used. Best results considering the CIDEr metric has been achieved for Xception backbone feature extractor, second place belong to DenseNet201. The spread between the highest (Xception with Glove, 78.13) and the lowest (VGG with Glove, 67.35) metrics value equals 10.78 points difference, which makes the model strongly dependent from the image backbone feature extractor. The evaluated quality of caption extractors is correlated with the accuracy of backbones. Practically for each metric, the order of models sorted by the metric value is similar to the order of backbones when sorted by accuracy both in top-1 and top-5 variants[2]. One cannot observe any remarkable superiority of one embedding model over another. For some metrics the Glove model performs better, while for the remainder – the FastText. In most cases, FastText embeddings achieve higher results than Glove for the same image features extractor. Which suggests that FastText adapts easier for different CNN models, than Glove. Long feature vectors does not imply higher performance. The longest feature vectors that are generated by VGG backbones does not imply higher values of measures. The winning models are using 2048 (Xception) and 1920 (DenseNet201) vectors. Average time of sequence generation is not correlated with the model complexity (no. of model params). Differences in execution time between models spreads from 874 to 1417 ms. The fastest is DenseNet201, which is also second best model.

Example correct captions obtained by the Xception + Glove pair are given in Table II, the respective images are shown

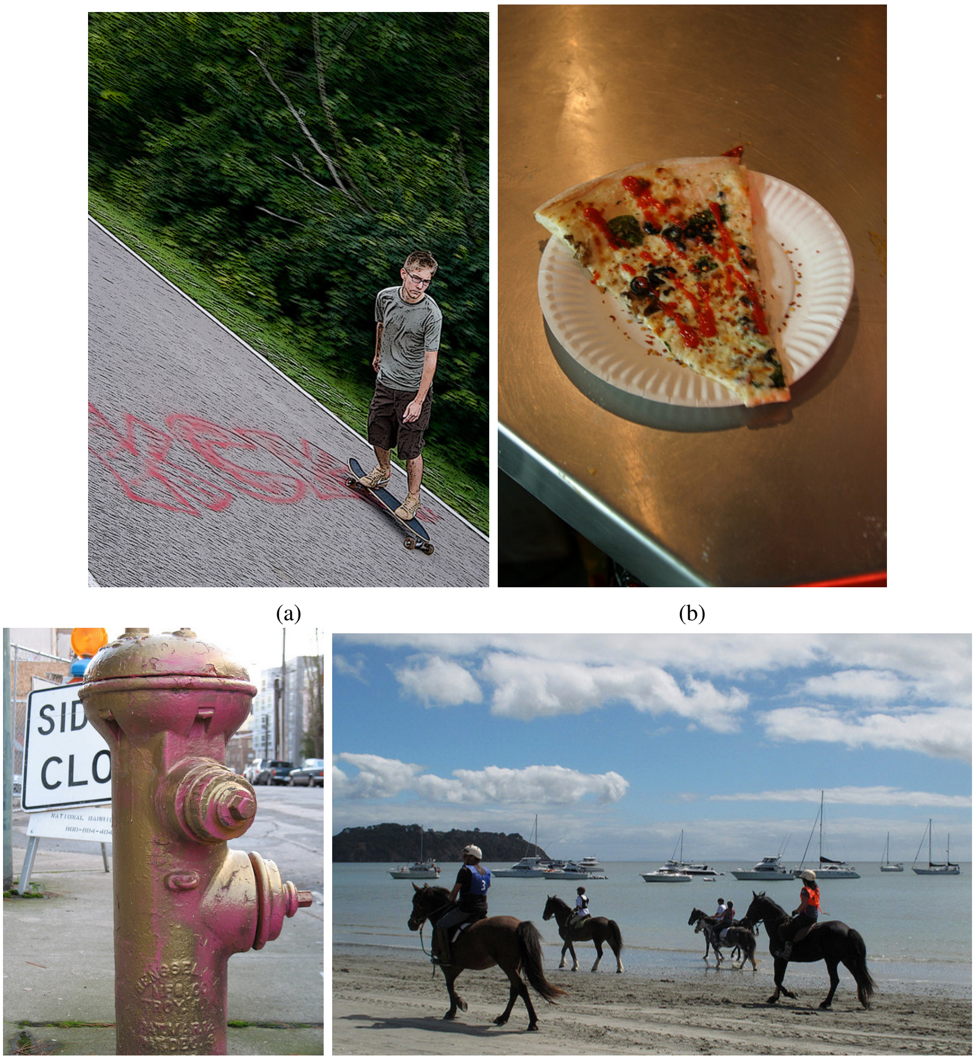
[2] Where top-n means that – in case of complete initial model of the backbone (i.e. model that contains both, the convolutional and fully-connected layers) the proper answer i.e. predicted class is among n-classes of highest output probability.

TABLE I: Evaluation results for MSCOCO 2014 test dataset (5000 images). Metrics' values are averaged over the whole test dataset. Higher results implies better image captioning performance.

Image features	No. of model parameters(mln)	Size of the input image features vector	top-1	top-5	Embeddings	Time of sentence generation (ms)	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROGUE-L	WMD	CIDEr	SPICE
Vgg16	14.71	4096	71.3	90.1	FastText	1043	64.47	45.73	31.54	21.86	20.86	47.01	47.43	67.76	13.81
					Glove	1088	64.25	45.62	31.63	22.09	20.78	46.99	47.35	67.35	13.64
Vgg19	20.02	4096	71.3	90.1	FastText	1045	65.42	46.89	32.72	22.93	21.61	48.09	48.40	71.79	14.46
					Glove	1023	64.10	45.83	31.86	22.34	21.11	47.24	47.71	69.62	13.93
Resnet152V2	58.33	2048	78	94.2	FastText	1209	65.28	46.78	32.47	22.61	21.30	47.58	48.04	70.07	14.16
					Glove	1417	64.91	46.78	32.57	22.86	21.38	47.80	48.05	70.77	14.08
Resnet50	23.59	2048	74.9	92.1	FastText	1058	65.97	47.82	33.79	24.02	21.88	48.39	48.79	74.47	14.71
					Glove	1234	65.33	47.26	33.26	23.44	21.65	48.28	48.46	73.12	14.43
InceptionV3	21.8	2048	77.9	93.7	FastText	961	66.15	47.87	33.57	23.63	21.92	48.41	48.92	75.04	14.83
					Glove	980	66.12	47.72	33.35	23.38	21.84	48.20	48.75	74.16	14.72
Xception	20.86	2048	79	94.5	FastText	1026	67.01	48.80	34.45	24.30	22.36	48.85	49.50	77.64	15.18
					Glove	1107	66.59	48.63	34.34	24.33	22.43	48.91	49.43	**78.13**	15.16
DenseNet121	7.04	1024	75	92.3	FastText	1180	65.39	47.09	32.89	23.09	21.60	47.99	48.31	72.36	14.25
					Glove	1234	65.03	47.02	32.96	23.26	21.64	47.87	48.38	71.94	14.13
DenseNet201	18.32	1920	77.3	93.6	FastText	874	66.59	48.73	34.57	24.55	22.25	49.01	49.20	76.74	14.83
					Glove	914	66.35	48.41	34.26	24.18	22.46	49.08	49.29	76.54	14.96
MobileNet	4.25	1000	70.4	89.5	FastText	976	65.02	46.93	32.85	23.02	21.65	47.98	48.15	71.24	14.31
					Glove	965	64.35	46.14	32.12	22.42	21.20	47.45	47.64	69.28	13.76
MobileNetV2	2.26	1280	71.3	90.1	FastText	1072	65.13	47.22	33.17	23.32	21.79	48.22	48.62	73.79	14.62
					Glove	1048	65.39	47.14	33.04	23.24	21.64	47.96	48.35	73.03	14.55
Karpathy [24]							62.50	45.00	32.10	23.00	19.50	-	-	66.00	-
Xu [57]							67.9	49.3	34.7	24.3	22.2	48.8	-	75.4	-
Sugano [46]							71.4	50.5	35.2	24.5	21.9	52.4	-	63.8	-
Lebret [28]							73	50	34	23	-	-	-	-	

TABLE II: Overview of four images with properly predicted captions (Xception image features extractor, Glove embeddings). along with the results of evaluation metrics for them.

Image	Fig. 3a	Fig. 3d	Fig. 3b	Fig. 3c
Ground truth captions	*A young man riding a skateboard down a street. *A man riding a skateboard down a road. *A man skateboards down a steep incline on an area painted with graffiti. *Man on a skateboard crossing over some graffiti *A man riding a skateboard down a hill.	*Horses walk along a beach while boats ride at their moorings offshore. *Some people riding horses on some sand and some boats and water *A group of people riding horses on a beach. *Some people are riding horses along a shoreline. *A group of people riding horses on top of a sandy beach.	*A slice of pizza on a paper plate. *A slice of pizza being served on a plate. *A slice of pizza sits on the paper plate *The metal table has a slice of pizza on a plate. *A slice of pizza is sitting on the top of a paper plate.	*A red and gold painted fire hydrant on the street *A fire hydrant on the side of the road *A multicolored fire hydrant that is on the sidewalk. *A fire hydrant on the side of a street. *A fire hydrant is standing on the sidewalk with two spouts.
Predicted caption	A man riding a skateboard down a street	A group of people riding horses on a beach	A slice of pizza on a plate	A fire hydrant on the side of the street
BLEU-1	100.00	100.00	70.00	100.00
BLEU-2	100.00	100.00	68.31	100.00
BLEU-3	100.00	100.00	66.32	94.99
BLEU-4	100.00	100.00	63.89	91.93
METEOR	41.35	14.28	15.11	28.97
CIDEr	482.15	419.35	411.58	479.30
ROGUE_L	93.13	100.00	79.37	88.89



(a) (b)

(c) (d)

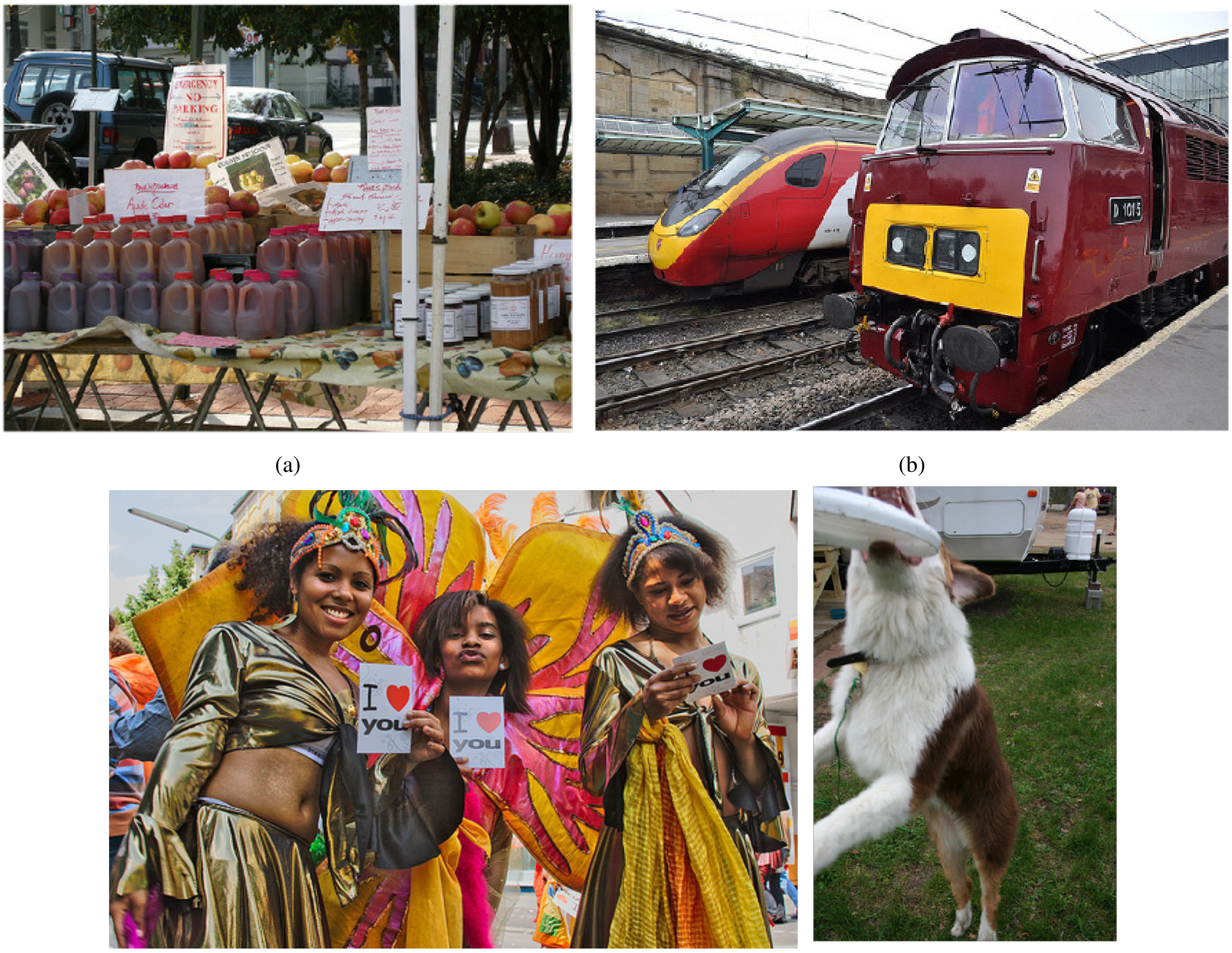
Fig. 3: Images with properly predicted captions (see Table II for details)

in Fig. 3. The table contains ground-truth 5 captions from the dataset metadata, captions obtained from the model and values of metrics. The generated captions sound good, are grammatically correct and consistent with the image content.

In contrast to the above, Table III presents inadequately predicted captions for four images obtained using different methods.

During this experiment, we checked that the resulting captions' wrong parts occur more often in the training set data. For Fig. 4d wrong part of the caption is the *with people standing*. Bigram *with people* occurs 1328 times, *people standing* 2740 times in training set. Those bigrams occur relatively often compared to other parts of the sentence. Also, for Fig. 4b bigrams that form *laying on a couch* occur very often in MSCOCO 2014 training dataset. Especially in the example Fig. 4c, bigrams "front of", "woman holding" are very common in the training dataset.

To explore deeply the possible reasons for incorrect captions, we investigated vocabulary of single words and bigrams used for training. The total size of vocabulary (the number of unique words) equals 26335 for 113350 images described using 5 alternative sentences each, which gives us 566747 captions. The similar numbers for the training set are the following: number of images 5000, of sentences 25000, of unique words: 7197 among which 503 words were used only in the captions in the test set (the remainder i.e. 6694 words are also present in the training set vocabulary). Considering the fact that each of investigated models is being learned on the training set, only words that are present in this vocabulary are used to predict ANY output sentence (correct or not). In case the captions in the test set, the number of words that was not present in the training set equals 503. This implies that, object, actions, situation, scene elements etc. that was described using these words, would never be produced properly (when testing,



(a) (b) (c) (d)

Fig. 4: Images with improperly predicted captions (see Table III for details)

the words in the test set vocabulary are obviously not used).

For further analysis, we tried to find why parts of the sentence are inadequate and how it is affected by training data. Regarding that, we examined how bigrams from predicted captions compare to those in the training set. We extracted bigrams from the MSCOCO 2014 training set with the number of their occurrences. Then we also extracted bigrams from predicted captions. As a result, we achieved a summary of bigrams in the training dataset and in a set of predicted captions, along with a number of their occurrences. The result for four example images are shown in Fig. III. Not surprisingly, the model, to construct captions, is using more frequent bigrams from the training set.

VI. Conclusions

In this paper, we analyzed how image features and word encoding affect the results of the encoder-decoder image captioning model. Our experiments proved that encoding input data plays in this area the primary role. During our research, we recognized that image captioning involves merging features from different modalities. Because of that, encoding of both image and features must cooperate, so finding the optimal pair for specific model architecture is crucial and we can significantly improve the results of the model predictions with that principle. The influence of the image feature extractor by the CNN backbone is crucial in this type of captioning model, it affects more the performance than the word embedding scheme. The Xception with Glove and DenseNet201 with FastText, according to our experiment are the best combinations of models' components.

The outcome of our research are applicable in all the research works that lead to the developing the optimal encoder-decoder image captioning model.

References

[1] P. Anderson, B. Fernando, M. Johnson, and S. Gould. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer, 2016.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

TABLE III: Overview of four images with incorrectly predicted captions. with the results of evaluation metrics for them. Table also consists bigrams of tokens from predicted caption. along with quantity of those bigrams in the training set.

Image	Fig. 4a		Fig. 4b		Fig. 4c		Fig. 4d	
Ground truth captions	*A pile of containers filled with lots of apple juice. *The stand is selling apples and apple cider outside. *Many various sized bottles of apple cider are on the table. *A bunch of apples and cider for sale on a table. *A farm stand selling apple cider and apples.		*Two red and yellow trains parked next to each other. *Two railroad trains with different front cars together *Two yellow and red trains parked on the tracks *Red and yellow trains sitting side by side to each other. *The front of modern commuter trains at the station		*Three women in bright colors and headdresses are holding love message cards *Three women in costume are holding papers that say "I love you" *Girls in bright costumes holding little signs that say I love you. *Three women in elaborate costumes hold up "I Love You" cards. *Three women in colorful costumes holding I love you signs.		*A brown and white dog with a Frisbee in his mouth . *A dog with his front paws off the ground holds a white Frisbee in his mouth in an RV campground . *A white and brown dog jumps up for a white Frisbee . *Dog catching Frisbee . *The brown and white dog is catching a Frisbee in his mouth .	
Model configuration	Resnet152V2 and FastText		Resnet152V2 and FastText		Xception and FastText		InceptionV3 and Glove	
Predicted caption	a market with a bunch of bananas and vegetables		a train is pulling into a station with people standing by it		a woman holding a umbrella in front of a crowd		a dog is laying on a couch with a frisbee	
	n-gram	**quantity**	**n-gram**	**quantity**	**n-gram**	**quantity**	**n-gram**	**quantity**
bigrams	**’and vegetables’**	**977**	’by it’	85	**’front of’**	**12517**	couch with’	449
	’bananas and’	**411**	’into station’	128	’holding umbrella’	63	’dog is’	1305
	’bunch of’	3724	’is pulling’	285	**’in front’**	**12363**	’is laying’	1294
	’market with’	140	**’people standing’**	**2740**	’of crowd’	199	**’laying on’**	**3539**
	’of bananas’	1027	’pulling into’	246	’umbrella in’	328	**’on couch’**	**1410**
	’with bunch’	386	’standing by’	1071	**’Woman holding’**	**1562**	’with frisbee’	853
			’station with’	292				
			’train is’	1310				
			’with people’	**1328**				
BLEU-1	44.44		33.33		19.99		54.29	
BLEU-2	0		28.87		0		46.73	
BLEU-3	0		22.83		0		29.12	
BLEU-4	0		0		0		0	
METEOR	2.24		4.52		10.5		1.65	
CIDEr	0.55		24.75		7.2		86.90	
ROGUE_L	39.29		33.33		0.1		58.65	

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*, 2005.

[4] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.

[6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[7] Y. Cui, G. Yang, A. Veit, X. Huang, and S. Belongie. Learning to evaluate image captioning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5804–5812, Los Alamitos, CA, USA, jun 2018. IEEE Computer Society.

[8] J. Delbrouck and S. Dupont. Bringing back simplicity and lightliness into neural image captioning. *CoRR*, abs/1810.06245, 2018.

[9] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.

[10] H. Dong, J. Zhang, D. McIlwraith, and Y. Guo. I2t2i: Learning text to image synthesis with textual data augmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, page 2015–2019. IEEE Press, 2017.

[11] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig. From captions to visual concepts and back. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[12] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 15–29, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[14] J. Gu, G. Wang, J. Cai, and T. Chen. An empirical study of language cnn for image captioning. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1231–1240, 2016.

[15] L. Guo, J. Liu, J. Tang, J. Li, W. Luo, and H. Lu. Aligning linguistic words and visual semantic units for image captioning. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, page 765–773, New York, NY, USA, 2019. Association for Computing Machinery.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] S. Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6:107–116, 04 1998.

[18] S. Hochreiter and J. Schmidhuber. Lstm long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[19] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 05 2013.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[22] A. Janusz, D. Kałuża, M. Matraszek, Łukasz Grad, M. Świechowski, and D. Ślęzak. Learning multimodal entity representations and their ensembles, with applications in a data-driven advisory framework for video game players. *Information Sciences*, 617:193–210, 2022.

[23] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4565–4574, 2016.

[24] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[25] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, number 2 in Proceedings of Machine Learning Research, pages 595–603, Bejing, China, 22–24 Jun 2014. PMLR.

[26] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *ICML*, 2015.

[27] P. Kuznetsova, V. Ordonez, A. Berg, T. Berg, and Y. Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 359–368, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[28] R. Lebret, P. O. Pinheiro, and R. Collobert. Phrase-based image captioning. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2085–2094. JMLR.org, 2015.

[29] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[30] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[32] S. Liu, L. Bai, Y. Hu, and H. Wang. Image captioning based on deep neural networks. *MATEC Web of Conferences*, 232:01052, 11 2018.

[33] J. Lu, C. Xiong, D. Parikh, and R. Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3242–3250, 2017.

[34] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *CoRR*, abs/1410.1090, 2014.

[35] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[36] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daumé. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, page 747–756, USA, 2012. Association for Computational Linguistics.

[37] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[39] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[40] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[41] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, 2017.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[43] A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A survey of evaluation metrics used for NLG systems. *CoRR*, abs/2008.12009, 2020.

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[45] R. Subash, R. Jebakumar, Y. Kamdar, and N. Bhatt. Automatic image captioning using convolution neural networks and lstm. *Journal of Physics: Conference Series*, 1362(1):012096, nov 2019.

[46] Y. Sugano and A. Bulling. Seeing with humans: Gaze-assisted neural image captioning. *ArXiv*, abs/1608.05203, 2016.

[47] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[48] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[49] M. Toshevska, F. Stojanovska, E. Zdravevski, P. Lameski, and S. Gievska. Exploration into deep learning text generation architectures for dense image captioning. In *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, pages 129–136, 2020.

[50] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015.

[51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, 2015.

[52] S.-S. Wang and R.-Y. Dong. Learning complex spatial relation model from spatial data. *Journal of Computers*, 30(6):123–136, 2019.

[53] Y. Xian and Y. Tian. Self-guiding multimodal lstm-when we do not have a perfect training dataset for image captioning. *IEEE Transactions on Image Processing*, PP, 09 2017.

[54] X. Xiao, L. Wang, K. Ding, S. Xiang, and C. Pan. Dense semantic embedding network for image captioning. *Pattern Recognition*, 90:285–296, 2019.

[55] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR.

[56] K. Xu, H. Wang, and P. Tang. Image captioning with deep lstm based on sequential residual. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 361–366, 2017.

[57] N. Xu, A. Liu, J. Liu, W. Nie, and Y. Su. Scene graph captioner: Image captioning based on structural visual representation. *J. Vis. Commun. Image Represent.*, 58:477–485, 2019.

[58] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[59] X. Zhang, S. He, X. Song, R. W. Lau, J. Jiao, and Q. Ye. Image captioning via semantic element embedding. *Neurocomputing*, 395:212–221, 2020.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 367–376

DOI: 10.15439/2023F7745
ISSN 2300-5963 ACSIS, Vol. 35

IoT and Edge Computing using virtualized low-resource integer Machine Learning with support for CNN, ANN, and Decision Trees

Stefan Bosse
0000-0002-8774-6141
University of Bremen
Dept. Mathematics and Computer
Science, Bremen, Germany
Email: sbosse@uni-bremen.de

***Abstract*—Data-driven models used for predictive classification and regression tasks are commonly computed using floating point arithmetic preserving accuracy by automatic scaling even in high non-linear functions. With respect to distributed sensor networks like the IoT, sensor data is acquired on low-resource embedded systems and delivered to data servers characterized by big data volumes. In specific use cases and domains, local predictive modelling on low-power devices is desired or required. But heterogeneity of host platforms and dynamic programming disables machine code deployment. This work addresses Tiny ML on very low-resource devices (microcontrollers, less than 32 kB RAM and ROM) by using a stack-based Tiny Virtual Machine providing core ML operations to implement Decision Trees (DT), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN). VM program code is always provided in textual format and compiled just-in-time to Bytecode to ensure portability, servicability, and mobility. Two damage diagnostics use-cases demonstrate the suitability of the VM approach, and even time consuming computational tasks do not compromise the overall responsiveness of the platform by using a real-time approach. This work addresses the underlying integer arithmetic operations required to implement efficient and fast computable ML models on microncontrollers.**

***Index Terms*—Tiny ML, Virtualization, Embedded Systems.**

I. Introduction

To ADDRESS ubiquitous computing, edge computing, and distributed sensor networks, driven by a significant increase in device density and sensor deployment toward smart and self-contained sensors, advanced and dependable data processing architectures are required. Tiny machine learning is a new and challenging field [1]. In order to calculate ML models, high precision floating point arithmetic is frequently used. Only integer arithmetic (8–32 bits) is offered by low-resource tiny embedded systems, therefore direct training using integer arithmetic [2] or model transformation and freezing [3] are required, ideally on the target device itself [4]. These issues are also addressed in our study. Ultra low-power devices place additional restrictions on the computation of deep learning (DL) models [5] and hardware designs are becoming more popular [6]. An example for such a tiny low-resource embedded system is shown in Fig. 1.

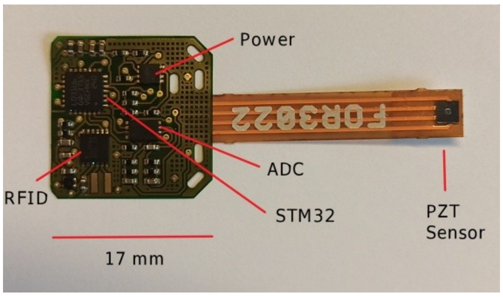


Fig. 1. An example of a highly integrated and miniaturized sensor node with a STM32 ARM Cortex microcontroller supplied entirely by an RFID energy harvester (source with courtesy: IMSAS, B. Lüssem, University of Bremen).

In this study, a real-time capable and extendable application-specific stack virtual machine (REXA-VM) with several distinctive and unique features is introduced and analyzed, specifically addressing ML computations. In contrast to common integer-based ML models using 8 bit scaled arithmetic [2], this VM supports 16 and 32 bit operations. The novelty of this work is the capability of a VM to process common ML models delivered in text format. The program text embeds model parameters as well the forward computation function for a specific already trained model. Virtualization of services and data processing in embedded devices play an important role in heterogeneous network environments [7].

Another problem involves non-continuous energy supply, such as that delivered to the sensor node from external sources utilizing RFID/NFC. This type of non-continuous energy supply introduces severe power restrictions limiting the set of usable microcontrollers (mainly without FPU) and necessitates real-time data processing to the appropriate degree. Running computationally expensive operations without jeopardizing IO event handling (i.e., the device's responsiveness) requires the VM's real-time capability, which is not covered in this work. It is anticipated that a REXA VM node receives remote communications over wireless tech-

This work was supported by DFG Grant 418311604

nology. Direct transmission of program text code to the VM for processing and compilation is possible.

II Arithmetic

Linear and moreover non-linear functions are commonly computed using at least 32 Bit Floating Point (FP) arithmetic. FP bases on an exponential representation (and approximation) of real numbers. Most ML models are linear or non-linear functions. Deep neuronal networks with non-linear activation functions can approximate highly non-linear models. The dynamic scaling of FP values and operations enables the computation of functions with high gradients and large value interval spans. Although, there is 16 Bit FP arithmetic, this reduced precision arithmetic commonly have no advantages over fixed point arithmetics (XP) except providing a higher dynamic range. The required dynamic range depends on the value range of the input and output variables as well as on latent (hidden) variables of intermediate functions, e.g., hidden nodes of an ANN. XP is often used in hardware implementations of ANNs [8] and rely on integer arithmetic that is the only available arithmetic on low-resource computers, e.g., the STM32 ARM Cortex M microcontroller series. XP arithmetic has the disadvantages of underflow and the requirement of software post-correction (multiplication) when used on microcontrollers, lowering the arithmetic's performance.

In contrast to common numerical approaches, in this work, XP values are replaced by in-advance dynamically scaled arithmetic (SA) using <value,scale> tuples. Scaling is only applied after an aggregating operation was performed, e.g., the computation of a vector product in an ANN layer, as illustrated in Fig. 2. This is relaxed by the fact that N Bit integer arithmetic (e.g., 32) is assumed at the core, but only M=N/2 Bit integer values (e.g., 16 Bits) are used to represent operands and result values. SA is used in this work to approximate complex (nested) functions. The scaling values are computed from data for a particular function, e.g., an ANN classifier. The relative approximation error increases with decreasing (real) values. A scaling factor can be shared by multiple values (e.g., vector elements), reducing memory requirements.

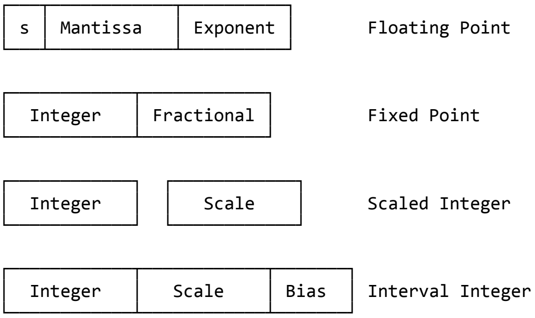


Fig. 2. Comparison of different arithmetic classes

A function F is transformed to a an integer approximation by:

1. Decomposing arithmetic expressions (and functions) in scalable arithmetic functions;
2. Annotating original expressions and functions with value intervals based on a representing test data set (input and output values of the composed function);
3. Calculating the scaling factors based on the interval annotations and pre-defined function value range annotations, e.g., a pre-defined sigmoid function.
4. Calculating the approximation error for the test data (eventually modifying the functional structure or changing scaling factors to reduce the overall function error).

The transformation process is not addressed in this work. Values of function variables (input, output, latent) in a specific data context and application can lie in a small interval, e.g., [0.11,0.12]. Pure scaling, e.g., with M=16 Bits), would use k=250000, but the entire integer range would be only [27500,30000], effectively reducing the resolution to 12 Bits with significantly increased approximation error. To increase the usable range for integer approximations of real numbers, a bias offset can be introduced, approximating a real number by a <bias,scale,value> tuple. But this kind of arithmetic would require post-corrections and dedicated arithmetics, and scaling factors and bias must be specified for each particular value (in contrast, to pure scaling), increasing memory storage.

A data-driven predictive model function is composed of vector operations and transfer functions. The approximation error in such a composed and chained functional system is accumulative. Using linear transfer functions the error is linear accumulative and show no exploding gradients. But using non-linear functions, e.g., based on logarithmic functions, the approximation error is non-linear with exploding gradients and underflows, at consist of approximation based on SA and approximation of non-linear functions itself, as discussed in Sec. IV.F.

III Virtual Machine

Details of the REXA VM architecture, features, capabilities and the compiler can be found in the technical paper [9]. In the following section the ML-relevant features are summarized only. The REXA-VM may be implemented in compact embedded systems with a microcontroller and as little as 8 KB of data RAM and 16 kB of code ROM. In large-scale and heterogeneous networks, virtualization and Machine Learning (ML) are essential for unified sensor and data processing [10]. A scriptable Tiny ML interface and signal analysis numbers utilizing 16-bit scaled arithmetic are two important features. This VM supports 16 and 32 bit operations natively, preventing frequent arithmetic overflow and underflow problems. In contrast to common integer-based ML models using 8 bit scaled arithmetic [2], this VM supports 16 and 32 bit operations.

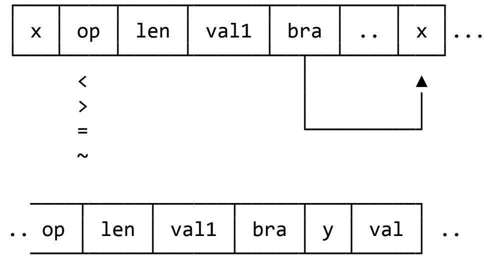
The REXA VM was designed especially for the deployment on low-resource microcontrollers with less than 64 kB RAM and low clock frequencies below 50 MHz. It utilizes a freely programmable ISA, but the ISA of the VM used in this work is closely related to the FORTH programming language [11]. The VM is a pure stack processor, i.e., most operations processing data via multiple stack memories with a zero-operand instruction format. The VM instruction loop processes Bytecode programs stored in a code segment (CS).

Each VM program consists of data and instructions stored in a code fragment in the CS. The main user program memory is the code segment of the VM (CS), which is organized in byte cells and has a static fixed size. An important feature of the CS is the direct embedding of program data besides code instructions. The Bytecode is compiled just-in-time by an integrated compiler. The VM and the compiler operate both incrementally, i.e., the processing time of each of them can be limited and scheduled, a primary feature required in single IO task programs with a main loop processing IO events and performing computations. Since the ISA of stack processors consists mostly of zero-operand instructions, it supports fine-grained compilation at the token level. The source text can be directly stored in the code segment (in-place) referenced by a code frame (or any other data buffer, alternatively). Most instruction words can be directly mapped to a consecutively numbered operation code.

IV ML Modells

A Decision Trees

Decision trees, as lightweight predictor models well suited for tiny embedded systems, can be efficiently stored in Linear Search Tables (LST), as introduced earlier for compiler parsing.



Def. 1. Format of a Linear Search Tree (LST) implementing a decision tree

Decision trees consist of nodes associated with input variables x_j or output variables y_k (and specific outcomes of a prediction). Directed edges connecting nodes are functional evaluations of a node variable.

There are three basic operations: Binary relation (</>), equality (=), and nearest value approximation (≈). The data format is shown in Def. 1. Each slide starts with the input variable to be evaluated (or target for output), the operation applied to choices, a field specifying the number of choices, and value-branch pairs. Decision tress can always approximated by integer arithmetic without error accumulation or exploding gradient (and underflow) issues. Therefore, the decision tree is here the gold standard for classifications problems and compared with ANN implementations.

B Artificial Neural Network (ANN)

An ANN consists of two parts:

1. The data, i.e., for parameter, input, and output variables;
2. The structure and functions processing the data.

For the sake of simplicity, fully connected networks are assumed, but any irregular network structure is a sub-set of a fully connected structure and can be used with the following operational architectures, too. In contrast to common ANN software frameworks, REXA VM provides only core vector operations, as discussed later on. The parameter data is embedded in a code frame by using the initialized *array* constructor. Both parameter and input/output data can be stored in the program code frame, shown in the next section.

ANN computations are decomposed in vector operations provided by the VM platform, discussed below. It can be shown that the complexity and memory requirement of this (textual data) approach is low even for complex network structures. Compiled code embedding data require typically less than 1 kBytes of RAM.

The principle structure of an ANN model and its forward computation using the vector operations discussed at next is shown below. There are initialized parameter arrays (weights, biases, and scaling factors) and latent variable arrays (neuron output).

```
array input N
array wghtsL1 { 1 2 3 .. }
array biasL1  { 1 2 .. }
array scaleL1 { 1 2 .. }
array outL1 N
..
: fwd
  .. vecmul
  .. vecadd
  .. vecmap
  ..
;
```

C Convolutional Neural Networks (CNN)

The structure of a CNN consists of different layers. A minimal basic layer architecture set consists of:

1. A convolutional layer applying a kernel filter mask to an input image (linear multiply-summation operation) producing a filtered output image;
2. A pooling layer extracting relevant features from images by applying special filters (e.g., a maximum value selection);
3. An ANN layer (commonly fully connected).

CNN computations are decomposed in vector operations provided by the VM platform, discussed below. The complexity and memory requirements is much higher than compared with ANN implementations. Especially the ANN layer is connected to all elements of the arrays of the pooling layer. Memory requirement is typically more than 4 kBytes, depending on the network structure, input dimension, and layer sizes. More details and evaluations can be found in the use-case sections.

The principle structure of a CNN model and its forward computation using the vector operations discussed at next is shown below (here the first convolution and the second pooling layer are merged to save storage space). There are initialized parameter arrays (kernel weights, biases, and scaling factors) and latent variable arrays (intermediate images, neuron outputs).

```
array input N
array kernL1p1 { 1 2 3 .. }
array kernL1p2 { 1 2 3 .. }
array kernL1p3 { 1 2 3 .. }
array cnvtmpL1 N
array poolL1p1 N
array poolL2p2 N
array poolL2p3 N
...
: fwd
  ( conv & pool )
  .. vecconv
  .. vecmap
  .. vecconv
  .. vecconv
  .. vecmap
  .. vecconv
  ..
;
```

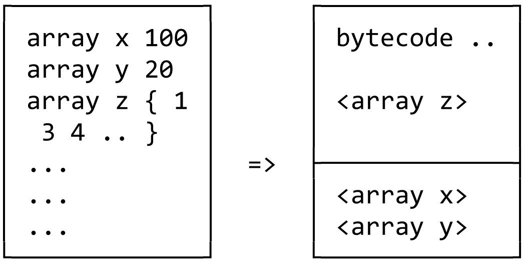
D ML Core Operations

ANN and CNN computations required efficient and generic vector operations crucial to implement ML on microcontrollers. The REXA VM provides a core set of vector operations that can be used for the computation of ANN and CNN models. Training using classical error back-propagation is currently not supported due to requirement of storing a suitable training and test data set.

All the basic operations you need to implement ANNs and perform forward activation computations are:

1. Element-wise vector operations (e.g., *vecmul*: *op1vec op2vec dstvec scalevec*);
2. Dot-product operation performing a sum of product data fusion (*vecprod*: *veca vecb scale* → number);
3. A folding operation for node layer computations (*vecfold*: *invec wgtvec outvec scalevec*);
4. A convolution operation for CNN computations (*vecconv*: *invec wgtvec outvec scale inwidth kwidth stride pad*). This function also serves as a pooling operation;
5. A mapping operation applying a function elementwise (*vecmap*: *srcvec dstvec func scalvec*);
6. A reduction operation applying a function to all elements returning an aggregate (*vecred*: *vec vecoff veclen op*); Supported functions are *min*, *max*, *sum*, and *average*;
7. A vector reshape operation shrinking or expanding a vector (*vecshape*: *srcvec dstvec scale*);
8. A generic scaling operations (*vecscale*: *srcvec dstvec scalevec*).

Vector operations commonly operate on arrays embedded in code frames, as shown in Def. 2. Scaling is typically applied after an aggregation operation (results of operation), e.g., after computing a sum of products (using 2N arithmetics), to avoid overflow. Some operations use one scaling factor for all elements, discussed in the following section.



```
array x 100
array y 20
array z { 1
 3 4 .. }
...
...
...
```

=>

```
bytecode ..

<array z>

<array x>
<array y>
```

```
<array>: [LEN:2][DATA:LEN*WORDSIZE]
```

Def. 2. Initialized arrays embedded in-place in code frames and non-initialized arrays stored at the end of the compiled code frame

E Vector Operations

The core set of vector operations provided by the REXA VM supporting integer arithmetic ANN computations is summarized in Tab. 1.

Vector Operation

`array <ident> <#cells>`
Allocates a data array at the end of the code segment

`array <ident> { v1 v2 .. }`
Allocates an initialized data array inside the code segment.

`vecload`
`( srcvec srcoff dstvec -- )`
Loads a data array into another array buffer. The source can be any external data provided by the IOS or internal embedded data.

Vector Operation

`vecscale`
`( srcvec dstvec scalevec -- )`
Scales the source data array with scaling factors from the scale array and stores the result in the destination array. Negative scaling values reduce, positive values expand the source data values.

`vecadd,vecmul`
`( op1vec op2vec dstvec scalevec -- )`
Adds two vectors element-wise with an optional result scaling (value 0 disables scaling). Both input and the destination vectors must have the same size. Constant down-scaling of all elements is provided by a negative scaling value (instead of vector reference).

`vecfold`
`( invec wgtvec outvec scalevec -- )`
Performs a folding operation *ivec* × *wgtvec* with a given filter. The weights vector *wgtvec* must have the size ||*invec*||*||*outvec*||.

`vecconv`
`( invec wgtvec outvec scale inwidth wgtwidth stride pad -- )`
Performs a two-dimensional kernel-based convolution operation *ivec* ⊗ *wgtvec*. The width of the input and kernel matrix (still a linear array) must be provided, the width of the output and the heights are computed automatically from the vector lengths. If *wgtwidth* is negative, a pooling operation is performed. The *wgtvec* argument provides then the height of the filter and the operation to be performed.

`vecmap`
`( srcvec dstvec func scalevec -- )`
Maps all elements from the source array onto the destination array using an external (IOS) or internal (user-defined word) function, e.g., the sigmoid function.

`vecred`
`( vec vecoff veclen op -- index value / valueL valueM )`
Reduces a vector to a scalar value. Supported operations are *min* (1) and *max* (2) returning position and value, *mean* (4) and *average* (8) returning a double word value.

TAB. 1. BASIC VECTOR ANN FUNCTIONS OPERATING ON EMBEDDED OR EXTERNAL ARRAY DATA (E.G., THE SAMPLE BUFFER)

Vector operations always operate on single data words (16 bit), but internally 32 bit arithmetic is used to avoid over- and underflows. To scale to signed 16 bit integer, some of the operations use a scale factor or scale factor vector (negative scale values reduce, positive expand the values by the scale factor) to avoid overflows or underflows in following computations, similar to scaled tensors in [4,12]. Vector operations can access arrays stored in code frames or provided externally by the host application (e.g., a signal buffer).

The *vecconv* operation can be used for convolutional and pooling layers (pooling is used if *wgtwidth* is negative and the *wgtvec* value contains the weight matrix height combined with the pooling function selector). The application of an activation function must be done separately using the *vecmap* operation, e.g., by applying a *sigmoid* function to all elements of a vector.

The computation of these operations are defined by the following formulas:

$$\begin{aligned}
&\mathrm{vecmul}\left(\vec{a},\vec{b}\right)=\left(a_1\cdot b_1,a_2\cdot b_2,..,a_n\cdot b_n\right)^T\\
&\mathrm{dotprod}\left(\vec{a},\vec{b}\right)=\sum_{i=1}^{n}a_i\cdot b_i\\
&\mathrm{fold}\left(\vec{a},\hat{c}\right)=\left(\sum_{i=1}^{n}a_i\cdot c_{i,1},\sum_{i=1}^{n}a_i\cdot c_{i,2},..,\sum_{i=1}^{n}a_i\cdot c_{i,n}\right)^T\\
&\mathrm{conv}\left(\vec{a},\vec{c}\right)=\sum_{i=1,+s}^{a_h}\ \sum_{j=1,+s}^{a_w}\ \sum_{k=1}^{c_h}\sum_{l=1}^{c_w}a_{i+k,j+l}\cdot c_{k,l}\\
&\mathrm{map}\left(\vec{a},f\right)=\left(f\left(a_1\right),f\left(a_2\right),..,f\left(a_n\right)\right)^T\\
&n=\left|\vec{a}\right|=\left|\vec{b}\right|,\left|c\right|=n\cdot m
\end{aligned}\tag{1}$$

F Activation Functions

There are different transfer (activation) that are used in ANN and CNN modells, mosr prominent examples are:

- Linear function (*linear*) without x- and y-limits
- Logistic or sigmoid function (*sigmoid*) with y-limit=[-1,1]
- Tangents hyperbsolic function (*tanh*) with y-limit=[-1,1]
- Rectifying linear unit (*relu*) with one-side open y-limit=[0,∞)

The *linear* and *relu* functions can be directly implemented with integer arithmetic without loss of accuracy (except due to integer discretizing). The highly non-linear *sigmoid* and *tanh* functions require an appropriate approximation by using a hybrid approach of the usage of a (compacted) look-up table (LUT) and interpolation. The *tanh* function can be neglected since it can be replaced in most cases by the *sigmoid* function without loss of generalization (of course, prior to training).

Trigonometric functions and functions composed of trigonometric functions are implemented with segmented linear and non-linear look-up tables. For example, the error of the discrete sigmoid function is always less than 1%, while only requiring 30 bytes of LUT space and less than 10 unit operations, as shown in Alg. 1. These software functions can be immediately implemented in hardware, too. The LUTs are computed with Alg. 2.

```
static ub1 sglut13[] = { <24 values> };
static ub1 sglut310[] = { <6 elements> };
// y scale 1:1000 [0,1], x scale 1:1000
sb2 fpsigmoid(sb2 x) {
  sb2 y;
  ub1 mirror=x<0?1:0;
  if (mirror) x=-x;
  if (x>=10000) return mirror?0:1000;
  if (x<=1000) {
    y = 500+(((x*231)/1000));
    return mirror?1000-y:y;
  } else if (x<3000) {
    ub2 i10 = ((fplog10((x/5)|0)/2))-65;
    y = ((sb2)sglut13[i10])+731;
    return mirror?1000-y:y;
  } else {
    ub2 i10 = ((fplog10((x/10)|0)/10))-14;
    y = ((sb2)sglut310[i10])+952;
    return mirror?1000-y:y;
  }
  return 0;
}
static ub1 log10lut[] = { <100 values> }
// x-scale is 1:10 and log10-scale is 1:100
sb2 fplog10(sb2 x) {
  sb2 shift=0;
  while (x>=100) { shift++; x/=10; };
  return shift*100+(sb2)log10lut[x-10];
}
```

Alg. 1. Range-segmented and LUT-based implementation of the sigmoid function with less than 1% approximation error (using approximated LUT-based log10 function)

The LUT tables can be computed as follows:

$$\text{log10lut} = \left\{ int\left(\log_{10}\left(\frac{i}{10} \right) 100 \right) : i \in \mathbb{I}, 0 \le i \le 99 \right\} \qquad (2)$$

The *fpsigmoid* function LUTs are computed iteratively using the *fplog10* function, described by the following pseudo code algorithm Alg. 2:

```
sglut13 := []
for x=1 to 2.95 step 0.05 do
  i10 := int(fplog10(int(x*1000/5))/2)-65
  if sglut13[i10] = undefined then
    sglut13[i10] := int(sigmoid(x)*1000)-731
  endif
done
sglut310 := []
for x=3 to 9.9 step 0.1 do
  i10 := int(fplog10(int(x*1000/10))/10)-14
  if sglut310[i10] = undefined then
    sglut310[i10] := int(sigmoid(x)*1000)-952
  endif
done
```

Alg. 2. Computation of the LUTs for the scaled integer sigmoid function

The accuracy (relative error) of the sigmoid approximation is plotted in Fig. 3 with an input and output scaling factor of 10000 (i.e., 1:10000). For $x > -3$ the error is below 5% and decreases to 1% in average. Only for $x < -3$ the relative error increases significantly due to the integer resolution.

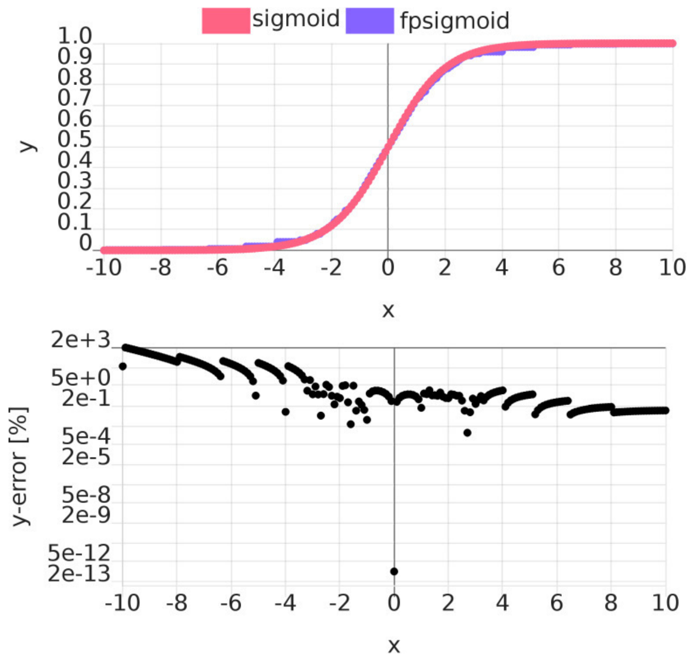


Fig. 3. Relative discretization error of scaled integer LUT-interpolated approximation of the *sigmoid* function

V Evaluation

Computation time results for ANN and CNN models are shown in Fig. 4 and 5. The code size required to store static and dynamic model parameters are shown in Fig. 6 and 6. Two different host platforms were tested: A generic i5 x86 clocked @2900 MHz (during test) and a STM32F103C8 microcontroller clocked @72 MHz with 20 kB RAM. All tests are processed by the operational same REXA VM. The computation time was normalized to the CPU clock frequency to enable comparison between different platforms. The REXA VM provided a code segment with 6k words capacity and a data stack with 256 words. The VM was compiled with GNU CC (gcc version 7), and the ARM-STM32 version was compiled with the Arduino software toolkit. With the configuration described above, 3 kB RAM remains for the VM program stack, which is sufficient. The REXA VM allocates memory only statically on the heap, there is no dynamic memory allocation during run-time.

The computational times were plotted against the number of neurons (ANN) and cells (CNN). The number of cells of a CNN is the sum of the static parameters and dynamic variables.

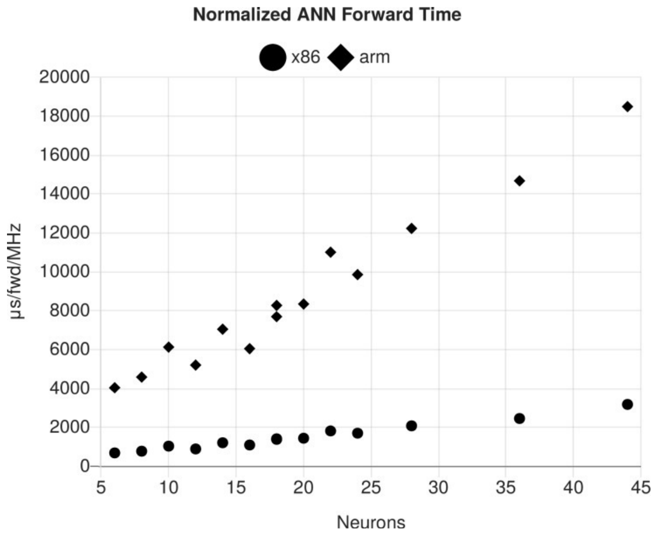


Fig. 4. Normalized computation times for ANNs of different size (with two, three, and four layers) and two different host platforms (Generic i5 x86 @2900 MHz and STM32F103C8 @72MHz) as a function of neurons. The computation time is approximately linear with the number of neurons (independent of network layer structure)

.

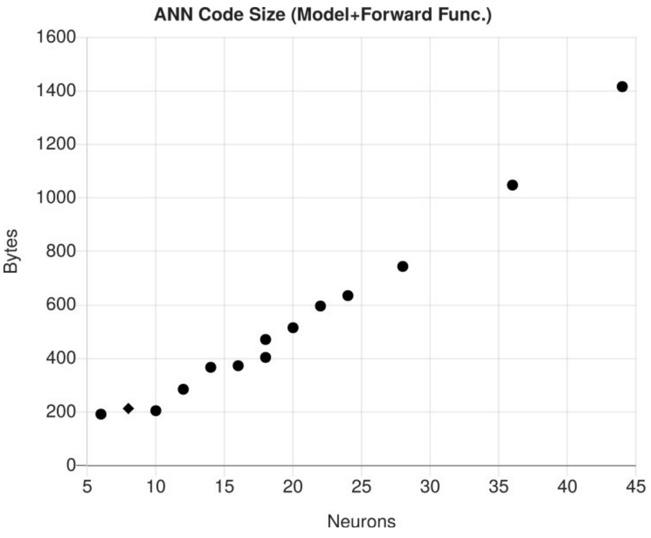


Fig. 6. Code size of ANN as a function of the number of neurons.

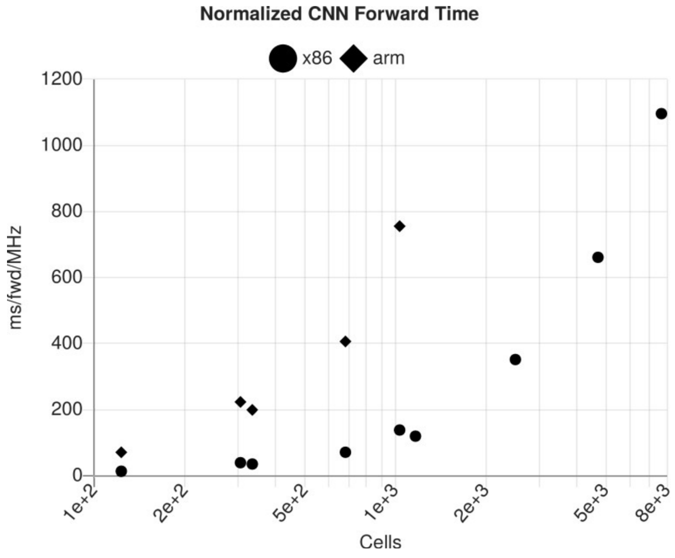


Fig. 5. Normalized computation times for CNNs of different size (with one and two convolution-pooling layer pairs) and two different host platforms (Generic i5 x86 @2900 MHz and STM32F103C8 @72MHz) as a function of cells (product of parameters and variables). The computation time grows about O(*n-log(n)*) with the number of cells *n*.

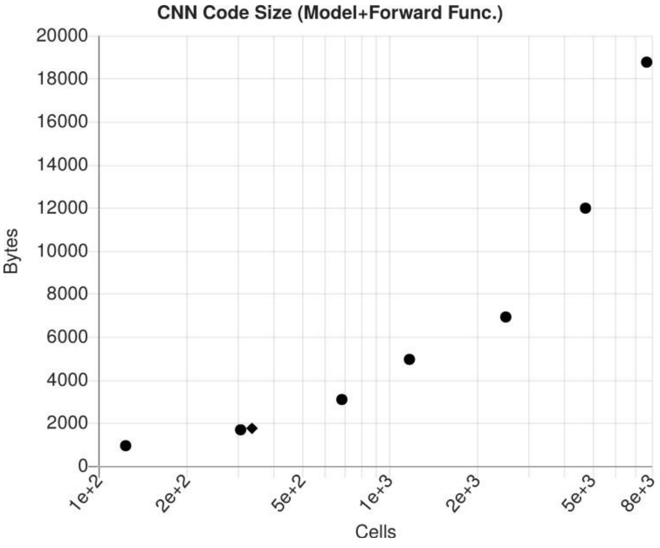


Fig. 7. Code size of CNN as a function of the number of cells.

The performance test shows the suitability of a low-resource microcontroller to store and compute small and medium sized ANN and even smaller CNN models. The forward (inference) computation time is always below one Second, typically about 10-100 ms with 16 MHz clock frequency. The required code space (including model data and code) is below 10 kBytes, typically about 1-2 kBytes. The ARM Cortex M processor under performs by a factor of 5 compared with a x86 processor, which is well known.

VI USE CASES

A Damage detection with an ANN

In this use-case, aggregated feature variables derived from time-dependent Ultrasonic signals (Guided Ultrasonic

Waves, GUW) from multi-path measurements were used to predict a damage in a composite materials and to estimate its location. Details can be found in [13]. The processing architecture is shown in Fig. 8.

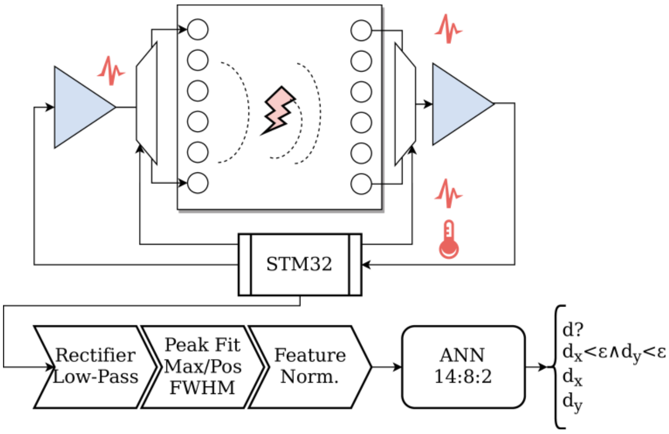


Fig. 8. Multi-path GUW measurement and data processing for damage detection (classification and location regression)

The feature variables were computed from the signal hull, mainly by analyzing the first main maximum (position and width). The hull signal was computed using (1) the analytical signal via the Hilbert transform (using FFT) and (2) by applying a rectifier and low-pass filter. Only the second method can be implemented on the STM32 microcontroller. Assuming six measuring paths and the two most significant feature variables normalized peak position and peak height, additionally using a measure temperature and the base frequency of the pitch signal, the feature vector consists of 14 variables in total. This scaled feature vector is the input for a simple ANN (three layers, one hidden, typical layer structure [14,8,2], sigmoid activation functions). The output of the ANN provided an estimation of the x- and y-coordinates of the damage location (or close to 0 if there is no damage detected). This is a hybrid classification and regression model. If only classification is required, one output neuron is sufficient.

The ANN with a [14,8,1] layer structure providing a binary damage classification was trained and transformed to the proposed integer numerics requiring about 1k Bytes code size, shown in Ex. 1.

```
( Layers: 14,8,2 #neurons:24 )
array input 14
( Layer I )
array wghtsI { 329 -499 ... 10 400 }
array biasI { -764 389 ... -907 -405 }
array scaleI { -3 9 ... 5 9 }
array actI 14
( Layer H1 )
array wghtsH1 { 622 -790 ... 708 248 }
array scaleH1 { 0 5 ... -4 7 }
array actH1 8
( Layer O )
array wghtsO { 869 939 ... 785 910 }
array biasO { 252 -565 }
array scaleO { 4 5 }
array output 2
( Input data is stored in input )
( Output data is stored in output )
: forward
  ( Layer I )
  input wghtsI actI scaleI vecmul
  actI biasI actI 0 vecadd
  actI actI $ sigmoid 0 vecmap
  ( Layer H1 )
  actI wghtsH1 actH1 scaleH1 vecfold
  actH1 biasH1 actH1 0 vecadd
  actH1 actH1 $ sigmoid 0 vecmap
  ( Layer O )
  actH1 wghtsO output scaleO vecfold
  output biasO output 0 vecadd
  output output $ sigmoid 0 vecmap
;
```

Ex. 1. REXA VM program for an ANN classifier for damage prediction from 14 aggregate feature variables and two output variables (parameter values are only for illustration)

The ANN requires only 620 Bytes in the CS memory of the REXA VM. The computation time (prediction) is about 1 ms/MHz (Intel x86 i5, i.e. 0.5μs @2900 MHz) and about 5 ms/MHz (STM32 ARM Cortex).

B Damage detection with a CNN

Similar to the previous use-case, single-path Ultrasonic time-dependent measuring signals are used to predict a damage in a composite material. In contrast to the previous example, no strong aggregate feature variables could be identified. Instead, a discrete wavelet transform using high- and low-pass filters are used to decompose the sensor signal into wavelet coefficients (first 5 levels were chosen). The output of the filters (detail and approximation) are decimated by a factor of two, retaining only the even samples, since each filter output contains half of the frequency content, but an equal amount of samples as the input signal. With increasing level the number of data elements decreases by a factor 2. To provide the output of multiple levels in matrix form, the higher levels are expanded. Here, we shrink the lower levels to the number of elements of the highest level (5). The original signal window contained about 2000 samples, finally providing only 50 data points for the fifth DWT decomposition layer. All DWT vectors are combined into a 50 × 5 elements matrix, treated as a two-dimensional spectogram image. The processing architecture is shown in Fig. 9.

A simple CNN was used to classify signals and predict damages. The CNN consists of one convolution layer with three filters (3 × 3 pixel), striding and padding set to two, output applied to a *relu* function, followed by one max-pooling layer (striding=2, padding=0). Finally, a softmax/fully connected two-neuron layer performs the classification (sigmoid activation function). The REXA VM program is shown in Ex. 2.

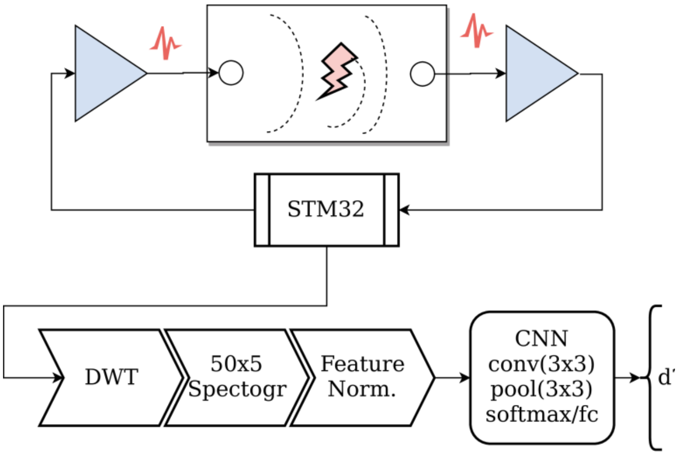


Fig. 9. Single-path GUW measurement and data processing using a CNN for damage detection

```
( Layers: conv,pool,fc )
array input 250
( Layer 1 conv )
array cK0L1 { -612 -692 ... 962 -467 }
array cK1L1 { -214 832 ... -644 -455 }
array cK2L1 { 764 -275 .. 978 600 }
array cSL1 { 817 390 572 }
array cOL1 104
( Layer 2 pool )
array pO0L2 12
array pO1L2 12
array pO2L2 12
( Layer 3 fc )
array fW0L3P0 { -468 -905 ... -632 518 }
array fW0L3P1 { -147 -932 ... -275 872 }
array fW0L3P2 { -327 -798 ... -61 -621 }
array fW1L3P0 { -126 894 ... -818 -870 }
array fW1L3P1 { 488 -408 ... 963 -887 }
array fW1L3P2 { -519 963 .. 895 -170 }
array fAL3 12
array fBL3 { -746 -776 }
array fSL3 { 1 3 }
array fOL3 2
array output 2
( Input data is stored in input )
( Output data is stored in output )
: forward
  ( Layer 1 conv )
  ( merged with Layer 2 pool )
  input cK0L1 cOL1 cSL1 0 cell+ @ 50 3 2 2 vec-
conv
  cOL1 cOL1 $ relu 0 vecmap
  cOL1 256 3 + pO0L2 0 26 -3 2 0 vecconv
  input cK1L1 cOL1 cSL1 1 cell+ @ 50 3 2 2 vec-
conv
  cOL1 cOL1 $ relu 0 vecmap
  cOL1 256 3 + pO1L2 0 26 -3 2 0 vecconv
  input cK2L1 cOL1 cSL1 2 cell+ @ 50 3 2 2 vec-
conv
  cOL1 cOL1 $ relu 0 vecmap
  cOL1 256 3 + pO2L2 0 26 -3 2 0 vecconv
  ( Layer 3 fc )
  pO0L2 fW0L3P0 fAL3 0 vecmul
  fAL3 0 12 8 vecreduce
  pO1L2 fW0L3P1 fAL3 0 vecmul
  fAL3 0 12 8 vecreduce
  pO2L2 fW0L3P2 fAL3 0 vecmul
  fAL3 0 12 8  vecreduce
  2+ 2+ fSL3 0 cell+ @ 2ext 2/ 2red sigmoid
  fOL3 0 cell+ !
  pO0L2 fW1L3P0 fAL3 0 vecmul
  fAL3 0 12 8 vecreduce
  pO1L2 fW1L3P1 fAL3 0 vecmul
  fAL3 0 12 8 vecreduce
  pO2L2 fW1L3P2 fAL3 0 vecmul
  fAL3 0 12 8 vecreduce
  2+ 2+ fSL3 1 cell+ @ 2ext 2/ 2red sigmoid
  fOL3 1 cell+ !
;
```

Ex. 2. REXA VM program for a CNN classifier for damage prediction from 50 × 5 feature variables (DWT spectogram) and two output variables (parameter values are only for illustration)

The CNN requires only 1500 Bytes in the CS memory of the REXA VM, fitting into a STM32F103C8 (20 kBytes RAM). The computation time (prediction) is about 30 ms/MHz (Intel x86 i5, i.e. 10μs @2900 MHz) and about 150 ms/MHz (STM32 ARM Cortex).

Due to the high integration level and the minimization of components the measuring data is characterized by noise (analog and digital sources), drift, and a superposition by environmental signals (main AC line, e.g.). Despite the data quality limitations, a damage prediction accuracy about 95% can be achieved by the CNN. Considering the low complexity of the CNN, the results showing the suitability of even simple data-driven classifier models processed directly by a material-integrated sensor node with a low-resource microcontroller.

VII Conclusion

The stack-based REXA VM was introduced targeting CPUs with integer arithmetic only and providing virtualization and a unique set of vector operations used to compute Artificial and Convolutional Neural Networks under high resource constrains. It could be shown that even with less than 20 kBytes of RAM memory (simple) CNNs can be computed. The VM has a built just-in-time text-to-Bytecode compiler. A ML model is provided on programming level with a mix of data and computational statements. The VM uses a shared code segment for program text and compiled Bytecode with embedded data without necessity to have a dynamic memory management (heap). The computational times for medium sized ANNs and small CNNs are about 1-300 ms/MHz, reasonable for sef-powered sensor networks. The source code of the REXA VM can be downloaded from github [14].

References

[1] Guo, S., Zhou, Q. , Machine Learning on Commodity Tiny Devices, Taylor & Francis, 2023

[2] Ray, P. P., A review on TinyML: State-of-the-art and prospects, Journal of King Saud University-Computer and Information Sciences, 2021, pp.1595-1623, https://doi.org/10.1016/j.jksuci.2021.11.019

[3] Wang, X., Magno, M. , Cavigelli, L., Benini, L., FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference at the Edge of the Internet of Thing, arXiv:1911.03314v3, 2022

[4] Banner, R. , Hubara, I., Hoffer, E., Soudry, D., Scalable Methods for 8-bit Training of Neural Networks, arXiv:1805.11046, 2018

[5] Alajlan, N. N., Ibrahim, D. M., TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications, micromechanics, vol. 13, no. 851, 2022, https://doi.org/10.3390/mi13060851

[6] Jain, V., Giraldo, S., Roose, J. D., Linyan, Mei, B. B., Verhelst, M. , TinyVers: A Tiny Versatile System-on-chip with State-Retentive eM-RAM for ML Inference at the Extreme Edge, arXiv:2301.03537, 2023,

[7] Heiser, G., The role of virtualization in embedded systems, In Proceedings of the 1st workshop on Isolation and integration in embedded systems, 11-16 April, 2008,
https://doi.org/10.48550/arXiv.2301.03537

[8] Zhang, L., Implementation of fixed-point neuron models with threshold, ramp and sigmoid activation functions, In IOP Conference Series: Materials Science and Engineering (Vol. 224, No. 1, p. 012054). IOP Publishing, 2017

[9] Bosse, S., Bornemann, S., Lüssum, B., Virtualization of Tiny Embedded Systems with a robust real-time capable and extensible Stack Virtual Machine REXAVM supporting Material-integrated Intelligent Systems and Tiny Machine Learning, arXiv:2302.09002 [cs.OS], 2023,
https://doi.org/10.48550/arXiv.2302.09002

[10] Bauer, M., IoT Virtualization with ML-based Information Extraction, in IEEE 7th World Forum on Internet of Things 2021, https://doi.org/10.1109/WF-IoT51360.2021.9595119

[11] Hayes, J. R. Fraeman, M. E., Williams, R. L. Zaremba, T., An architecture for the direct execution of the Forth programming language, ACM SIGARCH Computer Architecture News, 15(5), 1987, pp. 42-49. https://doi.org/10.1145/36177.36182

[12] Ghaffari, A.,. Tahaei, M. S., Tayaranian, M., Asgharian, Vahid, M., Nia, P., Is Integer Arithmetic Enough for Deep Learning Training?, Advances in Neural Information Processing Systems 35. 2022: 27402-27413.

[13] Bosse, S., Polle, C., Fast Temperature-Compensated Method for Damage Detection and Structural Health Monitoring with Guided Ultrasonic Waves and Embedded Systems, Eng. Proc. 2021, 10(1), 78; https://doi.org/10.3390/ecsa-8-11283

[14] https://github.com/bsLab/rexavm, REXA VM repository, on-line, accessed 31.7.2023

DOI: 10.15439/2023F6213
ISSN 2300-5963 ACSIS, Vol. 35

Impact of processor frequency scaling on performance and energy consumption for WZ factorization on multicore architecture

Beata Bylina, Jaroslaw Bylina, Monika Piekarz
0000-0002-1327-9747, 0000-0002-0319-2525, 0000-0002-3457-9335
Institute of Computer Science, Marie Curie-Sklodowska University
Pl. M. Curie-Skłodowskiej 5
Lublin, 20-031, Poland
Email: {beata.bylina, jaroslaw.bylina, monika.piekarz}@mail.umcs.pl

***Abstract*—With the growing demand for computing power, new multicore architectures have emerged to provide better performance. Reducing their energy consumption is one of the main challenges in achieving high performance computing. Current research trends develop new software and hardware techniques to achieve the best performance and energy compromise. In this work, we investigate the effect of processor frequency scaling using Dynamic Voltage Frequency Scaling on performance and energy consumption for the WZ factorization. This factorization is implemented both without optimization techniques and with strip mining. This technique involves transforming the program loop to improve program performance. Based on time and energy tests, we have shown that for the WZ factorization algorithm, regardless of the presence of manual optimization, it pays to reduce the frequency to save energy without losing performance. The conclusion can be extended to analogous algorithms — also having a high ratio of memory access to computational operations.**

***Index Terms*—processor frequency scaling, performance, energy, WZ factorization**

I. Introduction

MULTICORE architectures are now common in all computing environments, from portable handheld devices to HPC computing platforms and supercomputers. The advent of multicore architectures has increased application performance by allowing them to run at a higher level of parallelism. This opened a new era for High Performance Computing (HPC). Of course, the increase in efficiency is closely related to the increase in energy consumption. Computing energy is currently a serious problem with dire environmental and economic consequences, and its mitigation is extremely important.

Top500 [1] is a website that has been updating the top 500 supercomputers list for performance in the LINPACK [2] benchmark since 1993. GREEN500 [3] is a complementary list to part of the TOP500 list, which since 2007 has ranked the top 500 supercomputers in terms of energy efficiency. The discrepancy between the energy-saving supercomputers and the fastest supercomputers may be seen from their respective positions in both lists. The relationship between performance and energy consumption is unclear and depends on many factors.

Energy efficiency can be achieved on the following two levels: hardware [3] and software [4], [5], [6], [7], [8], [9], [10]. The first approach is based on innovation in computer hardware, represented by microarchitecture and advances in the design of integrated circuits. Based on the software, the second approach can be divided into two categories: optimization of energy consumption at the operating system level and optimization of energy consumption at the application level. The approach to optimizing power consumption at the operating system level focuses on minimizing the power consumption of the entire node through the use of techniques such as clock and power gating [11], dynamic voltage, and frequency scaling (DVFS) [12], [13]. In contrast, application-level optimization methods use application-level parameters and models to maximize the energy efficiency of the application.

Some papers have analyzed the energy impact of numerical linear algebra algorithms that do not change the code but only control software parameters (e.g. block size) accordingly. ATLAS (Automatically Tuned Linear Algebra Software) [14] is an example of the automatic tuning of a library to the architecture of the machine for reducing execution time. The focus of the paper [15] is to propose a method for tuning the ATLAS library, whereby it is possible to replace the execution time tuning process by tuning the energy consumption. In that work, the performance and the number of MFlops/J as well as the execution time and energy consumption were investigated for single and double precision for different array sizes.

In the article [16], different techniques related to algorithm transformation were used to reduce static and dynamic energy consumption on multicore machines with shared memory. In particular, one of the most popular matrix factorizations, namely the LU factorization, was considered. In the LU factorization code, the most energetically expensive instructions were extracted, and then, by performing a code transformation, an attempt was made to reduce their number. That work investigated the effect of the number of threads on dynamic and total energy consumption, performance, and also the number of

MFlops/W for different matrix sizes at a fixed number of threads.

The authors of [17], [18], present algorithms for solving systems of equations, trying to improve their performance, in particular in parallel. Improvement in performance was obtained by appropriate transformation of the underlying algorithm using looping tiling and appropriate data structures. There are currently not too many studies in the literature on the analysis of both efficiency and energy consumption in the context of loop transformations.

In our work, we study a numerical algorithm (the WZ factorization) in which loops are transformed. This algorithm concerns numerical linear algebra, particularly solving systems of equations on multi-core architectures using OpenMP in constant performance and energy consumption.

In this paper, we will focus on the combination of two approaches at the software level, namely the DVFS technique (at the operating system level) and the optimization of the program algorithm (at the application level) on one of the latest multicore architectures. The main idea is to capture the actual performance and energy consumption of a multi-threaded computing application when it is executed on a multicore computing platform by changing the clock frequency.

The LU factorization is a well-known factorization used to solve the linear systems. From the very nature, this factorization is sequential. Throughout recent years, parallel implementations of LU decomposition have been implemented on various modern computers. In particular, the researchers have taken into account the improvement of its parallel performance. The WZ factorization is designed straight into parallel computers of SIMD type according to Flynn classification and it seems to be a potentially attractive alternative to Gaussian elimination or Cholesky factorization for parallel computations, especially for SIMD computers. The advantage of the WZ factorization is that it simultaneously evaluates two columns or two rows instead of one column or one row as it happens with the LU factorization. The WZ factorization has a fewer number of iterations of the loop but more computations in each iteration in comparison with the LU factorization. For the WZ factorization, we can achieve higher parallelism. The algorithms with higher parallelism are desirable for multicore architectures.

In work [19], we presented the detailed implementation of multi-threaded WZ factorization with OpenMP [20] on multicore architecture using various nested loop transformation strategies to program optimization. In this work, we investigate the effect of CPU frequency scaling on performance and energy consumption for the WZ factorization in two selected versions, namely basic and strip-mining (optimized). We selected these versions for testing based on the results of the work [21]. Strip-mining is a loop transformation technique to improve memory performance. Additionally, we are examining the influence of the parameter value of the strip-mining — block size — for the WZ factorization in the strip-mining version on energy efficiency.

The rest of the paper is organized as follows. Section 2 introduces the DVFS technique. Section 3 describes the WZ factorization algorithm using OpenMP programming models in two versions. Section 4 presents the numerical experimental evaluation of the impact of processor frequency scaling on performance and energy consumption for WZ factorization on multicore architecture. Lastly, Section 5 concludes the paper.

II. DVFS — PROCESSOR FREQUENCY SCALING

Nowadays, computers incorporate various energy management techniques that support the reduction of energy consumption. Examples are Dynamic Voltage Frequency Scaling (DVFS) or, for example, the use of special instructions and specialized coprocessors. DVFS [12], [13] is a technique that reduces the clock frequency and voltage level of various computing node components (CPU, DRAM, etc.) at the cost of some performance degradation. Today, DVFS is widely supported by energy-saving and efficient processors supplied by different vendors under different names (eg SpeedStep for Intel [22] and PowerNow processors or Cool'n'Quiet for AMD processors [23]). DVFS can reduce the power consumption of a CMOS chip such as modern processors by reducing the frequency at which it operates as shown in the formula:

$$P = CfV^2 + P_{static}$$

where C is the capacitance of the transistor gates, f is the operating frequency, V is the supply voltage, and P_{static} is a static power which is mainly due to various leakage currents. The voltage required for stable operation is determined by the clock frequency of the circuit and may be reduced if the frequency is also reduced. This can result in a significant reduction in energy consumption due to the compound V^2 shown above.

However, dynamic power CfV^2 alone is not the total power of the system. Due to the static power consumption and the asymptotic execution time, it has been shown that the power consumption of the software shows a convex energy behavior, i.e. there is an optimal processor frequency at which energy consumption is minimized [24].

III. WZ FACTORIZATION

We present shortly the WZ factorization [25], [26]. We transform a square and nonsingular matrix $\mathbf{A}$ into a product of two matrices, namely $\mathbf{WZ}$. The matrix $\mathbf{W}$ is a matrix of the form of a butterfly with units on its main diagonal, the matrix $\mathbf{Z}$ is a matrix of the form of an hourglass. Both the matrices are complements of each other in the sense of the structure of non-trivial elements (one has non-trivial elements in places where the other has zeros/units — and vice versa). The forms of these matrices can be seen in Figure 1.

We chose this numerical algorithm here because it's quite complicated and difficult to optimize by the compiler. Figure 2 presents a parallel basic algorithm for the WZ factorization for an even size of the matrix (we only consider even sizes — without loss of generality).

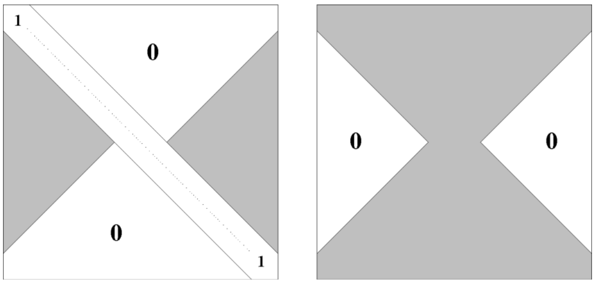


Fig. 1: The output of the WZ factorization — forms of the matrices **W** (left) and **Z** (right).

```
for(k = 0; k < n/2-1; k++) {
    // the following four lines are omitted
    // in the next versions of the algorithms
    // (thay are always the same)
    p = n-k-1;
    akk = a[k][k];    akp = a[k][p];
    apk = a[p][k];    app = a[p][p];
    detinv = 1 / (apk*akp - akk*app);
#pragma omp parallel for
    for(i = k+1; i < p; i++) {
        w[i][k] = (apk*a[i][p] - app*a[i][k])
                  * detinv;
        w[i][p] = (akp*a[i][k] - akk*a[i][p])
                  * detinv;
#pragma simd
        for(j = k+1; j < p; j++)
            a[i][j] = a[i][j]
                      - w[i][k]*a[k][j]
                      - w[i][p]*a[p][j];
    }
}
```

Fig. 2: The parallel basic algorithm for the WZ factorization — pseudocode.

Considering performance and energy consumption, it is important to have optimized algorithms and their implementation. A general technique for improving performance is to take full advantage of feature multicore architectures. A good example is the use in the code of loop optimization as the most common critical places are just the loops. One of the known loop optimization techniques is strip-mining. A loop in the process of strip-mining is divided into two loops, where the inner one has BLOCK_SIZE iterations and the outer one has n/BLOCK_SIZE iterations (n is the number of iterations in the original loop). The strip-mining alone can have some positive impact on the performance (by easing the automatic vectorization process).

In Figure 3, we present a parallel strip-mining algorithm for the WZ factorization with the parameter of this algorithm, namely n/BLOCK_SIZE. We use the compiler clause __assume which tells the compiler that a given condition is fulfilled — here, we declare that ii and jj are multiples of the BLOCK_SIZE.

The number of floating-point operations for the WZ factorization is:

$$\frac{2}{3}n^3 + O(n^2)$$

and number of memory access is:

$$\frac{7}{6}n^3 = O(n^2)$$

what gives the ratio of memory access to computations — $\frac{7}{4}$. This means that we need almost two memory accesses for one floating point operation to perform our algorithm.

```
for(k = 0; k < n/2-1; k++) {
    . . .
#pragma omp parallel for
    for(i = k+1; i < p; i++) {
        w[i][k] = (apk*a[i][p] - app*a[i][k])
                  * detinv;
        w[i][p] = (akp*a[i][k] - akk*a[i][p])
                  * detinv;
        start = RDTTNM(k+1, BLOCK_SIZE);
        for(jj = start; jj < p;
                        jj += BLOCK_SIZE) {
            __assume(jj % BLOCK_SIZE == 0);
#pragma simd
            for(j = jj; j < jj+BLOCK_SIZE;
                        ++j)
                a[i][j] = a[i][j]
                          - w[i][k]*a[k][j]
                          - w[i][p]*a[p][j];
        }
    }
}
```

Fig. 3: Parallel strip-mining in the basic algorithm — pseudocode.

IV. Numerical Experiments

A. *Methodology*

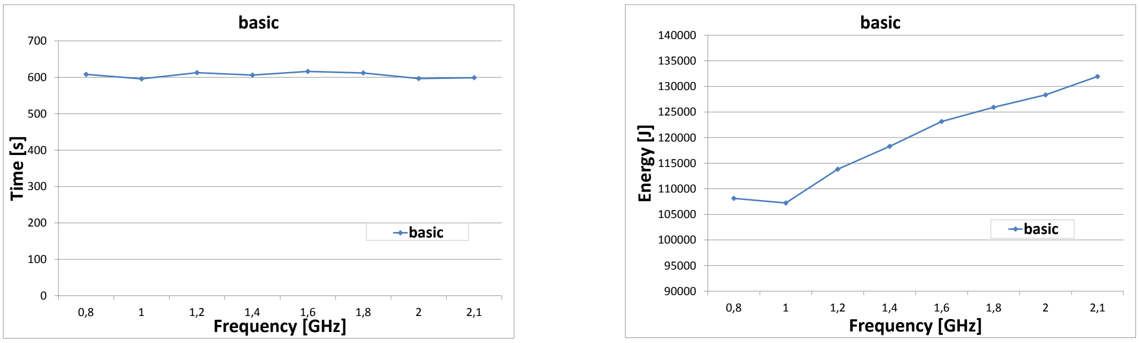
We test two types of versions of the WZ factorization algorithm: the basic algorithm and block algorithms with strip-mining. Our test dataset is a random square matrix with dimensions of $n \times n$ double-precision values, $n = 32768$. So our test dataset is 1073741824 cells, which is 8GB of data. All versions of the algorithm match the row-wise layout and are implemented in C++ with vectorization and parallelism. The following block sizes were checked during the tests: 64, 128, 256, 512.

Our experimental setup includes the following computing platform equipped with a multicore processor with the following parameters:

- processor: Intel(R) Xeon(R) Gold 5218R
- CPU @ 2.10GHz HT (2x20 cores)
- Cache: L1: 32KB, L2:1024KB, L3: 28MB

The following software was installed during tests:

- operating system: CentOS 7.5;
- kernel: Linux 3.10.0;



(a) Runtime

(b) Energy consumption

Fig. 4: Runtime and energy consumption of `basic` for data size 32768

TABLE I: Energy efficiency for `basic` (32768)

Frequency [GHz]	Time [s]	Total energy [J]	Performance [Gflops]	Energy efficiency [Gflops/J]
0.8	607.88	108132.066	38.59	0.217
1.0	**595.58**	**107218.43**	**39.38**	**0.219**
1.2	612.59	113827.77	38.29	0.206
1.4	605.92	118292.45	38.71	0.198
1.6	616.01	123146.44	38.08	0.190
1.8	611.79	125920.48	38.34	0.186
2.0	596.57	128343.30	39.32	0.183
2.1	598.88	131927.20	39.17	0.178

- icc compiler v. 2021.5.0 with the following compiler options:
  `-qoenmp -O3 -ipo -no-prec-div`
  `-fp-model fast=2`

We used the RAPL (Running Average Power Limit) interface [27] to measure the power and energy consumption of CPU-level components. We access RAPL energy meters via Machine-Specific Registers (MSR). Counters are 32-bit registers that indicate the amount of energy used since the processor was started, they are updated approximately once every 1 ms or 1000 Hz. Since its introduction, RAPL has been widely used in energy measurement and modeling. The results presented in the work [27] suggest that RAPL can be a very useful tool for measuring and monitoring energy consumption on multicore computers without the need to implement complicated power meters. The experience of the authors of the work [28] and our experience [21], [29] with RAPL confirms the results from the literature. RAPL is able to measure the energy consumption of a complex scientific application with acceptable accuracy and detail.

We carry out 5 iterations of each version of the algorithm for each tested frequency and then average the results to obtain a statistically correct result. As shown in [21], running HT for the tested versions of the WZ factorization algorithm has no benefit in speeding up the calculations so we run all versions without HT on 40 threads.

We made changes to the clock frequencies using CPU frequency scaling. The `intel_pstate` driver is used by default to control the performance of Intel processors on GNU/Linux systems. With this driver, we did not get a satisfactory effect of forcing the clock frequency, so we used the `acpi_cpufreq` driver, which by default follows the `conservative` governor. Governor `conservative` increases or decreases the clock frequency depending on the core load, selecting one of several available frequencies from the minimum to the maximum supported by the editor. For the Intel Xeon Gold 5218R processors we use, the permissible frequencies range from 0.8 GHz to 2.1 GHz. Using the `cpupower` program, we change the minimum and maximum values of the CPU frequency limit to a given level. The frequency setting was done for all cores of both installed processors with the commands:

```
cpupower frequency-set -d 1800000
cpupower frequency-set -u 1800000
```

for setting the minimum and maximum frequency limit values to 1.8 GHz.

We conducted our tests for frequencies ranging from 0.8 GHz to 2.1 GHz, making changes every 0.2 GHz, 0.8 GHz is the lowest frequency to which we could lower the clock.

B. The performance and energy consumption for `basic` WZ factorization algorithm

First, we measure the runtime of the `basic` version of the WZ factorization algorithm. The test results are presented in Figure 4.

In Figure 4 on the left diagram we can see that the algorithm runtime is similar regardless of the clock frequency. The difference between the shortest operating time (frequency 1.0

GHz) and the longest (frequency 1.6 GHz) is 3% (20 seconds). In Figure 4 on the right diagram we can see that the energy consumption increases with increasing clock frequency. For a higher frequency, we have a greater instantaneous power consumption, hence, with a similar runtime, we have a greater energy consumption. We have the lowest energy consumption for the frequency of 1.0 GHz. Lots of energy were consumed at the highest frequency. Lowering the clock frequency to 1.0 GHz saves about 19% of the energy consumed here.

In Table I, we have a summary of the runtime, total energy, performance, and energy efficiency for the `basic` algorithm. We can see that both the best performance and the energy efficiency of the algorithm will be achieved at the frequency of 1.0 GHz. Thus, there are benefits to lowering the clock frequency.

C. The performance and energy consumption for `basic-sm` versions of WZ factorization algorithm

Next, we test block versions of the WZ factorization algorithm with strip-mining (abbreviated as `sm`). We consider four block sizes: 64, 128, 256, 512 so we have the following versions: `basic-sm-64`, `basic-sm-128`, `basic-sm-256`, `basic-sm-512`. Our goal is to answer the question of whether the `sm` optimization will affect the performance and energy consumption and whether the additional clock frequency scaling for the `sm` version will also have a positive effect on the reduction of energy consumption without loss of performance.

The test results are presented in Figure 5. Here we see a similar situation as in the case of the `basic` version, the operating time of the tested versions of the algorithm slightly fluctuates for different frequencies (left diagram in Figure 5), while the energy consumption increases with increasing frequency (right diagram in Figure 5). We can also observe that the lowest energy consumption for each of the tested blocks was for the lowest tested frequency: 0.8 GHz. In addition, we can notice that, regardless of the frequency, block 64 was the weakest, both in terms of operating time and energy consumed. The best, however, are blocks 256 and 512.

In Table II and Table III, we have a summary of the runtime, total energy, performance and energy efficiency for the `basic-sm-256` and `basic-sm-512` versions of algorithm respectively.

In this case, other than for the `basic` version of the algorithm, the best performance and the best energy efficiency are not achieved for the same frequency. It is the same for both blocks, the best performance was obtained for the 1.4 GHz frequency and the best efficiency for 0.8 GHz. Here, too, we can infer that lowering the frequency (in the case of our algorithm to 0.8 GHz) results in less energy consumption. We save 21% of energy consumption without losing time for block 256 and save 19% of energy consumption lose about 2% (11 seconds) of time for block 512 compared to 2.1 GHz. If we look at high frequencies (2.0 and 2.1 GHz), we get slightly better results in terms of efficiency and energy efficiency for block 512 compared to block 256. However, if we want to reduce energy consumption without losing time, a bit better results are observed for block 256. The Figure 6 shows a juxtaposition of tests for the `basic` and `baisc-sm -256` versions. Lowering the clock to 0.8 GHz we can see that strip-mining will pay off regardless of the clock frequency. Meanwhile, if we use the clock frequency reduction mechanism, we can get 21% for `basic-sm-256` without wasting time, and for `basic` equal to 19% without wasting time, but for `basic-sm-256` it pays to lower the clock frequency to 0.8 GHz, and for `basic` it is enough to 1.0 GHz.

V. Conclusion

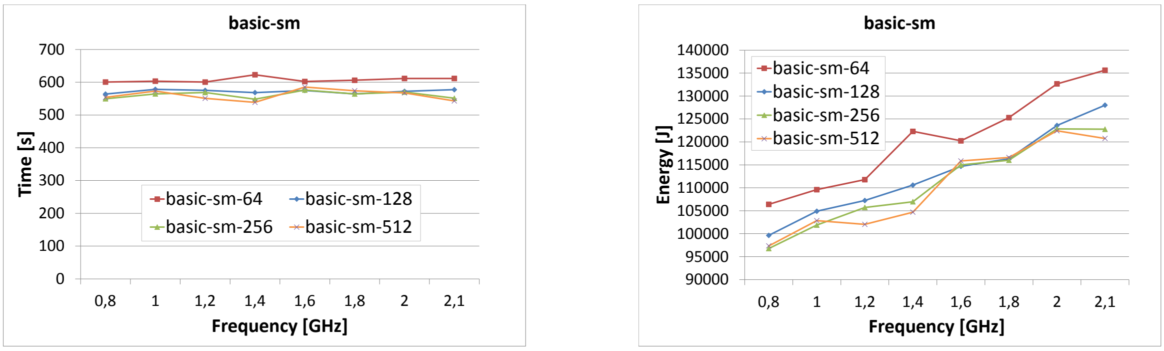
In this paper, we focused on the combination of two approaches, namely the DVFS technique and the optimization of the strip-mining loop. Our goal was to answer the question of whether traditional methods of strip-mining loop optimization in combination with the DVFS technique will reduce energy consumption without major losses on performance. Measurements were made on a 2nd Generation Intel Xeon Scalable Processors using the Intel RAPL interface.

There are a lot of memory references in our algorithm, so frequency scaling does not significantly affect performance, as our tests showed. The reduction of the frequency to the level of 0.8–1.0 GHz did not cause a decrease in performance in the case of the `basic` version, Table I. We improved memory access using strip-mining transformation, which resulted in a performance increase of about 9%. Although in the case of the `basic-sm` versions lowering frequency we lose a bit of performance (less than 2%, Table II), we can still lower the frequency while maintaining better performance than for the basic version.

Our tests also showed two facts. First, the first version of the `basic` will use more energy than the `basic-sm`, which means that the strip-mining transformation will pay off. Second, with this algorithm, the frequency scaling affects the energy consumption. By reducing the frequency to 0.8 GHz, we can reduce the power consumption for `basic-sm-256` by 21%, Table II.

The conclusion from our tests is that the highest frequency is not always the best in terms of time and energy consumption. For the WZ factorization algorithm, it pays to reduce the frequency to save energy without losing performance. The frequency that works best during experiments is the smallest that can be tested here, i.e. 0.8 GHz.

In the future, we plan to extend our tests by a wide one a range of architectures, including graphics cards. Moreover, we will evaluate the performance and energy consumption impact of various execution systems for OpenMP loop configurations and transformations for WZ and the three decomposition main kernels in dense linear algebra algorithms (Cholesky, LU and QR).



(a) Runtime

(b) Energy consumption

Fig. 5: Runtime and energy consumption of `basic-sm` for data size 32768

TABLE II: Energy efficiency for `basic-sm-256` (32768)

Frequency [GHz]	Time [s]	Total energy [J]	Performance [Gflops]	Energy efficiency [Gflops/J]
0.8	549.62	**96781.44**	42.68	**0.242**
1.0	564.49	101886.49	41.55	0.230
1.2	568.83	105726.79	41.24	0.222
1.4	**548.34**	106968.62	**42.78**	0.219
1.6	576.02	114994.40	40.72	0.204
1.8	564.11	116015.06	41.58	0.202
2.0	570.03	122833.96	41.15	0.191
2.1	551.03	122763.00	42.57	0.191

TABLE III: Energy efficiency for `basic-sm-512` (32768)

Frequency [GHz]	Time [s]	Total energy [J]	Performance [Gflops]	Energy efficiency [Gflops/J]
0.8	553.72	**97352.90**	42.36	**0.240**
1.0	573.04	102853.71	40.93	0.228
1.2	550.89	102039.32	42.58	0.230
1.4	**538.57**	104694.09	**43.55**	0.224
1.6	585.31	115855.60	40.07	0.202
1.8	574.01	116610.11	40.86	0.201
2.0	567.50	122428.81	41.33	0.192
2.1	542.94	120763.71	43.20	0.194

REFERENCES

[1] "Top500," https://www.top500.org/, 2022.

[2] J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, *LINPACK users' guide*. : SIAM, 1979.

[3] "Green500," https://www.top500.org/lists/green500/, 2022.

[4] J. V. Lima, I. Raïs, L. Lefevre, and T. Gautier, "Performance and energy analysis of openmp runtime systems with dense linear algebra algorithms," in *2017 International Symposium on Computer Architecture and High Performance Computing Workshops (SBAC-PADW)*, 2017, pp. 7–12.

[5] M. Mirka, G. Devic, F. Bruguier, G. Sassatelli, and A. Gamatié, "Automatic energy-efficiency monitoring of openmp workloads," in *2019 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, 2019, pp. 43–50.

[6] M. A. Shahneous Bari, M. M. Abid, A. Qawasmeh, and B. Chapman, "Performance and energy impact of openmp runtime configurations on power constrained systems," *Sustainable Computing: Informatics and Systems*, vol. 23, pp. 1–12, 2019.

[7] J. V. F. Lima, I. Raïs, L. Lefèvre, and T. Gautier, "Performance and energy analysis of OpenMP runtime systems with dense linear algebra algorithms," *The International Journal of High Performance Computing Applications*, vol. 33, no. 3, pp. 431–443, 2019.

[8] J. Dongarra, H. Ltaief, P. Luszczek, and V. M. Weaver, "Energy footprint of advanced dense numerical linear algebra using tile algorithms on multicore architectures," in *2012 Second International Conference on Cloud and Green Computing*, 2012, pp. 274–281.

[9] L. Szustak, R. Wyrzykowski, T. Olas, and V. Mele, "Correlation of performance optimizations and energy consumption for stencil-based application on Intel Xeon scalable processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2582–2593, 2020.

[10] T. Jakobs and G. Rünger, "Examining energy efficiency of vectorization techniques using a Gaussian elimination," in *2018 International Conference on High Performance Computing Simulation (HPCS)*, 2018, pp. 268–275.

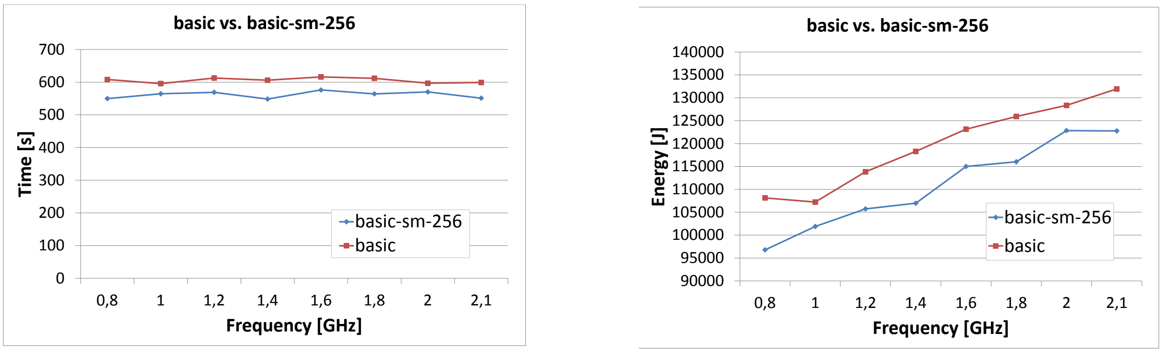
[11] A. Shahid, S. Arif, M. Qadri, and S. Munawar, "Power optimization using clock gating and power gating: A review," in *Innovative Research and Applications in Next-Generation High Performance Computing*, Q. F. Hassan, Ed. : IGI Global, 2016.

[12] M. Weiser, B. Welch, A. Demers, and S. Shenker, "Scheduling for reduced cpu energy," *1st OSDI*, pp. 13–23, 1994.

[13] W. A. and F. Bellosa, "Process cruise control - event-driven clock scaling for dynamic power management," *CASES*, 2002.

[14] R. C. Whaley, A. Petitet, and J. J. Dongarra, "Automated empirical optimizations of software and the ATLAS project," *Parallel Comput.*, vol. 27, no. 1-2, pp. 3–35, 2001. [Online]. Available: https://doi.org/10.1016/S0167-8191(00)00087-9

[15] T. Jakobs, J. Lang, G. Rünger, and P. Stocker, "Tuning linear algebra for energy efficiency on multicore machines by adapting the ATLAS library," *Future Gener. Comput. Syst.*, vol. 82, pp. 555–564, 2018.



(a) Runtime

(b) Energy consumption

Fig. 6: Runtime and energy consumption of `basic` and `basic-sm-256` for data size 32768

[Online]. Available: https://doi.org/10.1016/j.future.2017.03.009

[16] E. Garcia, J. Arteaga, R. S. Pavel, and G. R. Gao, "Optimizing the lu factorization for energy efficiency on a many-core architecture," in *International Workshop on Languages and Compilers for Parallel Computing*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:489258

[17] S. Donfack, J. Dongarra, M. Faverge, M. Gates, J. Kurzak, P. Luszczek, and I. Yamazaki, "A survey of recent developments in parallel implementations of Gaussian elimination," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 5, pp. 1292–1309, 2015.

[18] J. J. Dongarra, M. Faverge, H. Ltaief, and P. Luszczek, "Achieving numerical accuracy and high performance using recursive tile LU factorization," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 6, pp. 1408–1431, 2013.

[19] B. Bylina and J. Bylina, "Nested loop transformations on multi- and many-core computers with shared memory," in *Selected Topics in Applied Computer Science*, J. Bylina, Ed. Lublin: Maria Curie-Skłodowska University Press, 2021, pp. 167–186, http://stacs.matrix.umcs.pl/v01/stacs_v01.pdf.

[20] R. Chandra, L. Dagum, D. Kohr, D. Maydan, R. Menon, and J. McDonald, *Parallel Programming in OpenMP*. San Francisco: Morgan Kaufmann Publishers, 2001.

[21] J. Bylina, B. Bylina, and M. Piekarz, "Influence of loop transformations on performance and energy consumption of the multithreded wz factorization," *Preproceedings of the of the 17th Conference on Computer Science and Intelligence Systems*, pp. 479–488, 2022, https://annals-csis.org/proceedings/2022/pliks/251.pdf.

[22] E. Rotem, A. Mendelson, A. Naveh, and M. Moffie, "Analysis of the enhanced intel® speedstep® technology of the pentium® m processor," https://www.cs.virginia.edu/~skadron/tacs/rotem_slides.pdf, 2004.

[23] "Amd powernow! technology dynamically manages powerand performance," https://www.amd.com/system/files/TechDocs/24404a.pdf, 2000.

[24] K. De Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "The energy/frequency convexity rule:modeling and experimental validation onmobile devices," *PPAM'2013*, 2014.

[25] D. Evans and M. Hatzopoulos, "A parallel linear system solver," *International Journal of Computer Mathematics*, vol. 7, no. 3, pp. 227–238, 1979.

[26] P. Yalamov and D. Evans, "The wz matrix factorisation method," *Parallel Computing*, vol. 21, no. 7, pp. 1111–1120, 1995.

[27] K. Khan, M. Hirki, T. Niemi, J. Nurminen, and Z. Ou, "RAPL in action: Experiences in using RAPL for power measurements," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 3, 2018.

[28] L. Szustak, R. Wyrzykowski, T. Olas, and V. Mele, "Correlation of performance optimizations and energy consumption for stencil-based application on Intel Xeon scalable processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2582–2593, 2020.

[29] B. Bylina, J. Potiopa, M. Klisowski, and J. Bylina, "The impact of vectorization and parallelization of the slope algorithm on performance and energy efficiency on multi-core architecture," *Annals of Computer Science and Information Systems*, vol. 25, pp. 2283–290, 2021.

DOI: 10.15439/2023F2137
ISSN 2300-5963 ACSIS, Vol. 35

Simulating Large-Scale Topographic Terrain Features with Reservoirs and Flowing Water

Łukasz Błaszczyk, Michalina Mizura, Aleksander Płocharski, Joanna Porter-Sobieraj
0000-0002-9827-3562, 0009-0007-1683-6442, 0000-0002-7487-8153, 0000-0002-1411-475X
Warsaw University of Technology, Faculty of Mathematics and Information Science
ul. Koszykowa 75, 00-662 Warszawa, Poland
Email: {lukasz.blaszczyk, michalina.mizura.stud, aleksander.plocharski, joanna.porter}@pw.edu.pl

***Abstract*—The flow and accumulation of water are essential aspects when it comes to generating realistic terrains. In this article, we have set out to create a method for generating the distribution and levels of water in a virtual world. Our method fully simulates the water entering and exiting the system through rainfall, perspiration, and flowing out of the domain. Also, it simulates the phenomena of flow and accumulation in an iterative process. According to our observations, only allowing the user to influence the terrain and then simulating the placement of water bodies provides a natural result while preserving a high level of control.**

I. Introduction

THE rapid growth of the game and movie industries has generated a rise in demand for virtual worlds generation methods, especially when generating realistic natural environments. Modeling the environment by hand is, of course, still possible but only realistic when it comes to small parts of the vast virtual worlds that have become common. The workforce needed to create a whole extensive world by hand would be too substantial to consider these days. That is why the need for an automatic generation has drastically risen over the last decade to shorten this possibly tedious process.

The field in recent years has been moving in various directions depending on the use case being considered. Some algorithms focus purely on generating realistic natural environments from a humble set of parameters. While these methods usually provide very little control over the result, they are strongly based on geological knowledge (like tectonic plates) and climate data [1] and produce compelling results. These algorithms provide the foundation for most automatic world generation research. However, since the desired methods are supposed to replace modeling by hand, the methods at the forefront of interest usually provide much control over the result while still retaining a satisfying level of realism. The user may want to place mountain tops in certain specific places or keep a section of the environment flat. Unfortunately, the problem that arises seems to be that walking that thin line between control and realism is not so obvious and means that some natural processes might become neglected in the generation process.

We have observed that one of the environmental aspects that has a high chance of being treated like that is the placement and flow of water bodies in the result. While the methods usually aim for realistic placement of rivers and lakes, it is achieved mainly by some heuristic metrics rather than simulation loops. In most cases, the water bodies are just placed according to the positions specified by the user, and the environment is supposed to adjust independently to those requirements.

Since the flow and accumulation of water are essential aspects when it comes to keeping the veil of realism intact, we have set out to create a method for generating the distribution and levels of water in a virtual world. We have decided on a method that fully simulates the process of water entering and exiting the system (through rainfall, perspiration, and flowing out of the domain) and also simulates the phenomena of flow and accumulation in an iterative process. The method in itself does not offer much control over the result. However, the user can still use a terrain generation method which allows for controlling the shape of the terrain and place valleys and depressions in places where they want rivers and lakes to appear, and by applying our algorithm after that, the likelihood of meeting those water bodies placement expectations is very high. We have observed that only giving the user the ability to influence the terrain and then simulating the placement of water bodies naturally usually provides a very natural result while still preserving a high level of control based on our human intuition of where water is most likely to flow in a system of a given shape.

II. Related work

Terrain generation methods are usually grouped into two distinct categories based on their output. The first and more popular category is generating a heightfield representing the resulting terrain. In contrast, the second group of algorithms focuses on producing a three-dimensional output using various volumetric structures.

We first examine the former category, which, while having some apparent drawbacks – not being able to model things like caves and overhangs – usually provides the performance edge, which is invaluable for most iterative world creation pipelines [2]. One approach to the problem of 2D generation is constructing the terrain from primitive structures and defining their connections using a tree structure [3]. The primitives can be modeled by hand or generated from real-life data, for instance, using point cloud data from photogrammetry [4]. The method can yield compelling results, but the quality depends on the artist's skill in fitting those primitives together. While still cutting much modeling time, this approach cannot guarantee realistic results and will only provide partially automatic generation.

A significant subset of methods focuses on getting a more realistic result by employing geological knowledge. One of the aspects which can be used is the presence of tectonic plates since the process of forming mountains is influenced mainly by their movement and the placement of their borders. The position and shape of the plates can be generated from features that the user wants to appear in the final result, like the position of mountain tops and rivers [5]. The tectonic plate information can then be used to simulate the geological terrain folding process. Another example of using geological knowledge would be generating the resulting terrain using examples from the real world. Various distributions of geological parameters can be gathered for specific regions of the Earth, like the Himalayas or Norwegian fjords. Given some user input like positions and heights of mountain tops, the generation process can then try to recreate those distributions in the virtual environment [6]. This can produce convincing and realistic results, but since the output is majorly based on real-life data, it is limited to only resembling naturally occurring environments.

Rough sketches of the desired results can also guide the generation of terrains. The user could draw a 2D representation of the required scene using color-coded shapes or control points representing various topographical features like mountain ridges, plateaus, rivers, etc. [7], [8]. Unfortunately, this approach, like some of the methods mentioned above, suffers from very varied levels of realism in the resulting terrain based on the input data.

A good compromise between control and realism seems to be allowing the user to create maps of geological parameters that should be applied to the domain instead of specific topographical features that must appear in the resulting terrain. One such geological parameter is the tectonic uplift defining the growth rate of a particular point in the two-dimensional domain. The generation method can then simulate the growth of mountain ranges using the geological data provided while considering other factors like fluvial erosion [9].

Now we focus on the volumetric algorithms to briefly overview the rest of the generation methods spectrum. After adding dimension, a primitive-based modeling solution is still an excellent way to produce a convincing environment. It seems to be one of the most prominent methods for volumetric generation. One possible approach is basing the primitive structures on B-splines [10], which most artists are used to working with. The result becomes more flexible since it can model things like overhangs, allowing for smoother control but straying further from geological realism. The terrain could also be generated using a generalization of the previously mentioned primitive tree-based method [11]. Unfortunately, since the primitives must now be positioned across three dimensions, the work required to create desired scenes significantly increases.

There also exists a middle ground between the two representations – each point on a flat plane could hold a series of layers that are present above it together with their heights, representing a narrow column of the resulting terrain. This approach could either be used in conjunction with the standard two-dimensional methods or be extended with the ability to create a layer containing only air, allowing us to model things like caves and overhangs [12].

What most of the described methods have in common is that they treat the placement of water bodies in the result more like an afterthought rather than a significant step of the algorithm. Rivers and lakes are usually generated using heuristic methods to fit the generated environment, or their placement is dictated purely by the input data, which allows for setting their positions by the user almost entirely.

To design our algorithm, we needed to choose a terrain generation algorithm that would serve as a base for our water flow simulation. We have decided on using the method by Cordonnier at al. [9], which – apart from using the previously mentioned uplift maps – also considers the flow of water to be a significant factor in the process of erosion that is being simulated. This allowed us to use some of the water flow data structures already present in the solution as a jumping-off point for our method.

III. Method

A. Starting off point

The base method [9] operates over domain $\Omega \subset \mathbb{R}^2$ and aims to compute the function $h\colon \Omega \to \mathbb{R}$ representing the height of the terrain at each domain point. During the initialization step the method generates a terrain graph G defined over a discretization of the domain achieved by computing the Voronoi cells of points distributed over the domain using the Poisson distribution. In order to perform the fluvial erosion step the algorithm also computes a directed stream graph G_S, based on the same nodes as graph G, representing the water flow in the system, which serves as the foundation of our method.

Each node v in the graph G_S is represented by the cell's surface area a_v and a generating point p_v. The edges in the graph G_S represent the flow of water. Each node is connected to its lowest (in terms of height during this iteration) neighboring region, apart from points on the very edge of the domain where we assume the water flows out of the system. This creates disjointed components in the graph G_S (each in a form of a directed tree with the lowest - in terms of height - node forming the root), which in the original method are called lakes. Connection between these trees are stored in a separate lake graph G_L with edges defining the lowest connection between them - the minimum height at which one lake would start overflowing to the other.

The base method performs an additional step to merge the trees in G_S (representing lakes) into a complete graph. This is done by connecting them according to the connections in G_L. The algorithm allows itself to do that based on the assumption that all lakes are fully filled at all times – an assumption that we reject in the design of our method. That is why our solution is based on a disconnected stream graph; any mentions of it from now on will refer to this version.

At the base level of the solution, we have also incorporated different types of rock layers into the terrain representation [13]. Each point in the domain bears the characteristics of all layers below it with appropriate weights. This modification improves the overall realism of the result but does not influence the structure of the proposed water accumulation algorithm.

B. Algorithm modification

Algorithm 1 presents a general description of our modified fluvial erosion algorithm. A water accumulation step, performed during each iteration, and a lakes computation step have been added. Since a stream graph built for the current elevation of nodes is required for the correct computation of lakes, it is necessary to recreate this graph after the last change in elevation caused by erosion.

Algorithm 1 Base fluvial erosion algorithm [9] extended by adding water accumulation and lakes computation (underlined sections)

```
Require: $u$ {uplift map}, $i_{max}$ {maximum number of iterations} and $P$ {terrain sample points}
compute terrain graph $G$
for $i = 1$ to $i_{max}$ do
    compute stream graph $G_S$
    update accumulated water level in $G$
    update $G_S$ with edges resulting from lake overflow
    compute the new elevation values $h(v)$ after uplift and erosion for $v \in V(G)$
end for
compute stream graph $G_S$
update accumulated water level in $G$
compute the lakes list $L$
determine the elevation values $h(p)$ for $p \in P$ by interpolating values $h(v)$ for $v \in V(G)$
return $h$ {elevation map} and $L$ {lakes list}
```

1) Water accumulation: Each vertex v of the terrain graph G stores information about the volume V_v of water collected on the part of the terrain represented by this vertex. The initial value of V_v is 0.

In this model, it was assumed that two factors had the most significant impact on the change in the accumulated water level: the increase in water resulting from precipitation and the evaporation phenomenon. Let us consider the first one. Due to the immense time scale of the simulation (millions of years), we assume that the amount of water that will accumulate at the bottom of the lake represented by its root v during a given iteration will be proportional to the time δt of the iteration and the basin area A_v, i.e.,

$$V_v^{rain}(t + \delta t) = \delta t \cdot rA_v(t + \delta t),$$

where r is a constant describing the amount of precipitation per unit area.

The evaporation phenomenon leads to a partial loss of water in each iteration. We assume that the amount of water that has evaporated will be proportional to the iteration time and the area of the considered lake. Studies on the characteristics of natural lakes [14] have shown that the relationship between the volume V of a lake and its water surface S is approximately $V \sim S^{6/5}$. Thus, the volume of water lost can be written as:

$$V_v^{evap}(t + \delta t) = \delta t \cdot eS_v = \delta t \cdot eV_v^{5/6}(t),$$

where e is a constant controlling the evaporation rate. Using this approximation allows us to estimate the water loss due to evaporation without repeatedly determining the actual areas of the lakes.

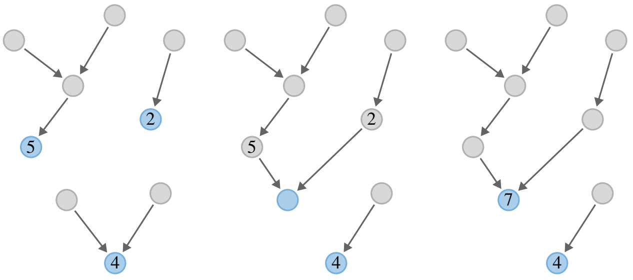
The formula for changing the volume of water during an iteration of δt is then:

$$\Delta V_v(t + \delta t) = \delta t \left(rA_v(t + \delta t) - eV_v^{5/6}(t) \right).$$

It is also worth noting that because each iteration of the stream graph G_S contains different connections, it is necessary to traverse all the vertices belonging to a given subtree and move the water accumulated in them so far to the current root. This step is performed before calculating the change in water volume so that water accumulated at other vertices can also participate in evaporation. The water-shifting process is illustrated in Figure 1.

In the stream graph G_S created during each iteration of the algorithm, roots of trees contained in G_S do not have receivers – water does not flow from them to another node. If the tree's root is at the edge of the terrain, we can assume that rainfall water flows out of the domain. Otherwise, the water should stagnate in the area of the tree (starting from the root), creating a lake.

2) Computing and merging lakes: The key element of the described modification is the algorithm for determining the water level in each lake. The non-trivial nature of this task results from the fact that more water can be accumulated in the lake than the surrounding area can hold. Therefore, a solution is needed that will allow for modeling the flow of water between lakes, as well as combining several lakes into



(a) i-th iteration (b) $(i + 1)$-th iteration (c) result

Fig. 1. The process of water shifting to its current roots in a stream graph. The numbers in the vertices indicate the volume of water accumulated in them. Different edge structure between iterations is caused by the erosion process which changes the height of the nodes resulting in the need to update the stream graph.

Algorithm 2 Determining the water level in lakes

```
Require: $G_S$ {stream graph}
$G_L \leftarrow$ lakes graph determined by $G_S$
$L \leftarrow V(G_L)$ {merged lakes set}
for all lakes $k$ in $V(G_L)$ do
    $v \leftarrow$ root of $k$
    $V_k^{rem} \leftarrow V_v$ {the water remaining to fill up is all the water collected in the root}
    $V_k^{acc} \leftarrow 0$ {water accumulated in a given lake}
    $l_k \leftarrow$ neighbor of $k$, to which the lowest passage from $k$ leads
    $V_k^{miss} \leftarrow$ the amount of water needed to reach the crossing height $(k, l_k)$
end for
while the flow is non-zero or no lakes have merged do
    $s \leftarrow$ network source, $t \leftarrow$ network sink
    $N \leftarrow$ empty lake network; $V(N) = \{L, s, t\}$
    for all lakes $k$ in $L$ do
        if $V_k^{rem} > 0$ then
            append to $N$ arc $(s, k)$ with capacity $V_k^{rem}$
        end if
        if $V_k^{miss} > 0$ then
            append to $N$ arc $(k, t)$ with capacity $V_k^{miss}$
        else
            append to $N$ arc $(k, l_k)$ with capacity $\infty$
        end if
    end for
    $SCC \leftarrow$ set of strongly connected components in $N$
    for all strongly connected components $H$ in $SCC$ such that $|H| > 1$ do
        collapse the vertices belonging to $H$ into one pool $m$
        determine $l_m$ and $V_m^{miss}$
        remove the component lakes and add a new merged lake $m$ to $L$
    end for
    $F \leftarrow$ maximal flow in $N$
    for all arcs coming from the source $(s, k)$ in $F$ do
        subtract the flow value at $(s, k)$ from $V_k^{rem}$
    end for
    for all arcs entering the sink $(k, t)$ in $F$ do
        add the flow value at $(k, t)$ to $V_k^{acc}$
        subtract the flow value at $(k, t)$ from $V_k^{miss}$
    end for
end while
for all lakes $k$ in $V(G_L)$ do
    $h^{acc} \leftarrow$ water level based on $V_k^{acc}$
end for
return $G_L$ {graph of lakes supplemented with information about the water level $h^{acc}$}
```

one. The pseudocode of the developed algorithm is described in Algorithm 2.

a) Computation of water volume and surface height: During the algorithm, it is necessary to determine the volume of water required to reach a certain surface height. We can approximate the volume of the lake by the sum of prisms whose bases are the Voronoi cells of individual vertices of the terrain graph located underwater, and the height is the difference between the height of the surface h^{acc} and the height of the vertex h_v. The volume approximation method is illustrated in Figure 2 and described in Algorithm 3.

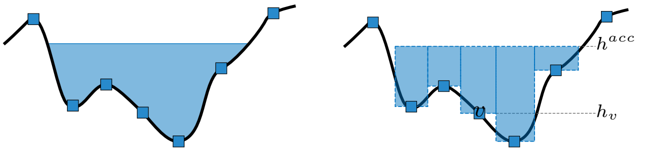
Algorithm 3 Determining the volume of water needed to reach the given height of the surface

```
Require: $G_S$ {stream graph}, $k$ {lake}, $V_k^{acc}$ {volume of water accumulated in a given lake} and $h^{target}$ {target surface height}
$V^{total} \leftarrow 0$
$Q \leftarrow$ empty queue
for all lakes $l$ forming part of the combined lake $k$ do
    if root of $l$ is below $h^{target}$ then
        append root of $l$ to $Q$
    end if
end for
while $Q$ is not empty do
    remove vertex $v$ from $Q$
    $V^{total} = V^{total} + a_v \cdot (h^{target} - h_v)$
    for all nodes $w$ in the set of children of $v$ do
        if $h_w < h^{target}$ then
            append $w$ to $Q$
        end if
    end for
end while
$V_k^{miss} \leftarrow V^{total} - V_k^{acc}$
return $V_k^{miss}$ {the amount of water needed to reach a given height}
```

It is worth noting that the lake for which calculations are made can be either a single tree in the stream graph or a combination of several trees into one merged lake.

At the end of the lake computation algorithm, a reverse operation is required, i.e., calculating the height of the water surface based on the volume of collected water. The method chosen for this purpose consists of gradually flooding the tops of the lake to increase the height until the lowest unflooded top is above the existing water surface. A detailed description of this step is provided in Algorithm 4.

b) Finding strongly connected components: Strongly connected components are essential in constructing a network describing the water flow between lakes. They represent



(a) The actual volume of water in the lake

(b) Approximating the volume of water by the sum of the prisms

Fig. 2. Cross-section of an example lake filled with water to the level of h^{acc}

Algorithm 4 Determining the height of the lake surface for a given volume of water

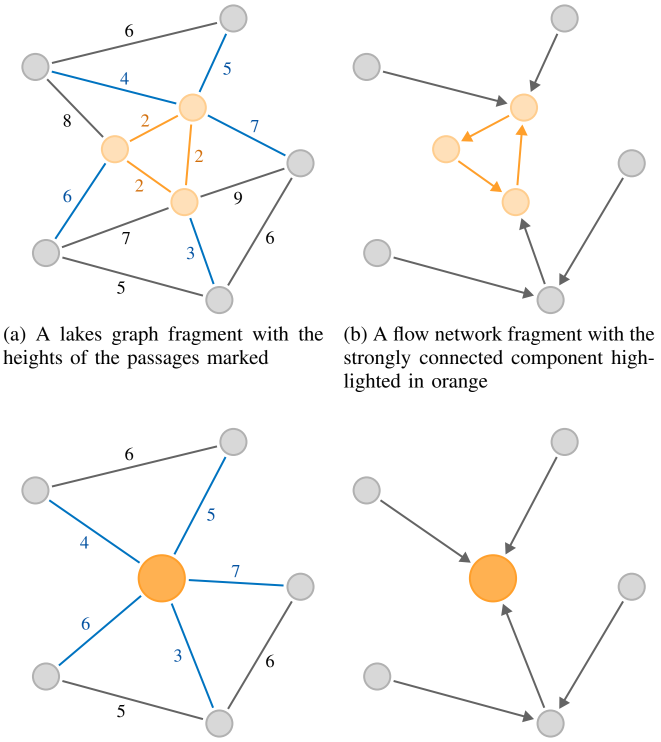
```
Ensure: $G_S$ {stream graph}, $k$ {lake} and $V_k^{acc}$ {volume of water accumulated in a given lake}
  $Q \leftarrow$ empty priority queue {priority – lowest height}
  $V_k^{below} \leftarrow 0$ {volume of land under the lake}
  $A_k^{acc} \leftarrow 0$ {surface area}
  $h_k^{acc} \leftarrow \infty$
  for all lakes $l$ forming part of the combined lake $k$ do
    append root of $l$ to $Q$
  end for
  while $Q$ is not empty and the lowest vertex of $Q$ is below $h_k^{acc}$ do
    remove the lowest vertex $v$ from $Q$
    $A_k^{acc} \leftarrow A_k^{acc} + a_v$
    $V_k^{below} \leftarrow V_k^{below} + a_v \cdot h_v$
    $h_k^{acc} \leftarrow \frac{V_k^{acc} + V_k^{below}}{A_k^{acc}}$
    for all nodes $w$ in the set of children of $v$ do
      if $h_w < h_k^{acc}$ then
        append $w$ to $Q$
      end if
    end for
  end while
  return $h_k^{acc}$ {surface height}
```

groups of two or more lakes connected by passages at the same height and filled with water at least up to that height. Adding water to the system of lakes thus connected would result in a combined rise in the level of all the constituent lakes. To model this behavior, lakes belonging to the strongly connected component are represented by one vertex when determining the flow. This means that the water flow network and the lake graph G_L must be modified at each iteration, as shown in Figure 3.

Tarjan's algorithm [15] was used to find strongly connected components in the directed lake graph G_L. It allows us to determine all such components in time $O(V+E)$. Since each vertex except sources and sinks has precisely one outgoing arc in the considered network, the complexity, in this case, will be $O(M)$, where M is the number of lakes.

c) Computing the maximum flow: Flow in a network of lakes allows the determination of the amount of water flowing out of each lake and into each lake. The arcs from the source to the lakes with the capacity of the collected water model the water that directly flows into the basin of a given lake due to precipitation. The arcs from the lakes to the estuary with a capacity representing the volume to the lowest passage represent the water accumulating in the basin. The arches between the lakes allow water to flow through passages.

Push-relabel is the chosen algorithm for determining the maximum flow in the network. An appropriate implementation of this algorithm [16] allows obtaining a complexity of $O(V^2\sqrt{E})$, which in the case of the considered network is $O(M^2\sqrt{M})$.



(a) A lakes graph fragment with the heights of the passages marked

(b) A flow network fragment with the strongly connected component highlighted in orange

(c) A lakes graph fragment after the collapse of the strongly connected component. Only those edges incident with the component, which have the smallest height, are left

(d) A flow network fragment after the collapse of the strongly connected component

Fig. 3. Modification of the lakes graph and the water flow network as a result of replacing the strongly connected component with one merged lake

d) Optimizations: The first optional optimization of the lake computation algorithm is the step of joining pairs of lakes for which the lowest passage is at the same level as both of their roots. This situation occurs primarily in the first iterations of the erosion algorithm when many vertices are at the same level. This optimization will connect many lakes that consist of only one vertex, significantly reducing the lake network's size.

The second optimization used consists of the lakes' initial filling with the water collected in them. This step is done after executing the 8 line in Algorithm 2. For lake k, V_k^{acc} becomes $\min(V_k^{rem}, V_k^{miss})$, and V_k^{rem} and V_k^{miss} will be reduced by same amount. This means that if more water has accumulated in the lake than is necessary to fill it, then already in the first iteration of the lake computation algorithm, water from it can flow to the remaining reservoirs.

3) Computational complexity: The water accumulation step involves traversing the stream graph to move water from the non-root vertices to the roots and determining a new value for each lake's water volume. Hence the complexity of this step is $O(N+M) = O(N)$, where N is the node count in stream graph G_S. Since each iteration of the basic fluvial erosion algorithm requires $O(L \cdot N + M \log M)$ operations, where L

is the terrain layer count, adding a water accumulation step does not affect the complexity of the iterations.

In order to determine the complexity of the lake determination algorithm, all steps described in the Algorithm 2 should be considered. Determining the lakes graph in line 1 has a complexity of $O(N)$ due to the need to traverse all edges of the terrain graph and its planarity. The loop in line 3 executes M times, calling Algorithm 3 for each lake. However, since within Algorithm 3, each vertex of the terrain graph will be queued at most once, the total execution time for all iterations of this loop will be $O(N)$. The situation for the loop in line 38 is analogous, so this fragment also has a complexity of $O(N)$.

Let us now consider the execution time of one iteration of the lake determination algorithm. The complexity of the loops in 13, 30 and 33 is $O(M)$. Finding strongly connected components using Tarjan's algorithm also requires $O(M)$ operations. Collapsing them into one vertex is also $O(M)$ complex. This step involves traversing all the vertices of the found components and updating the arcs in the network incident with those vertices. We assume that we can find a specific arc and then modify or delete it in constant time, which is possible, for example, by using hash tables as collections of arcs. The most demanding operation, with a complexity of $O(M^2\sqrt{M})$, is finding the maximum flow in the network.

The last key issue is the number of iterations performed by the lake determination algorithm. The chosen stopping condition means that in each iteration except the last one, there must be either a combination of at least one strongly connected component with a cardinality greater than 1 into one lake or a non-zero flow.

Each iteration in which a newly merged lake is created reduces the number of network vertices representing the lakes by a minimum of one. This means that at most $M-1$ of such iterations is possible.

The lake merging operation is the algorithm's only element that modifies the network structure between successive iterations. This means that if the lakes do not connect, the network in the next iteration will be the same network with the volumes of the arcs minus the value of the last maximum flow, resulting in zero flow. There can therefore be at most M iterations in which the lakes do not merge. Hence the total number of iterations is limited by $O(M)$.

The total computational complexity of the lake determination step is $O(N + M^3\sqrt{M})$. It is worth noting, however, that the number of vertices in the network decreases with each strongly connected component collapse, so in practice, later iterations of the water level determination algorithm run faster.

4) Computation of lakes and the basic erosion algorithm: In our modified fluvial erosion algorithm, water overflow between lakes is a separate step independent of lake overflow during the iteration. This approach proved necessary to preserve the realistic and varied resulting terrain.

The first attempts to extend the erosion algorithm consisted of replacing the original overflow of lakes with a model that would consider water stagnating in depressions. However, this approach did not produce the expected results. Since the terrain was almost flat in the first iterations, the water stagnated at numerous points and led to the formation of craters. It was, therefore, necessary to separate the overflowing and the computation of lakes. The first of these stages allows us to shape the relief, while the second only determines which places in the existing area should be flooded with water.

5) Elevation computation for vertices on the boundary of the domain: The original version of the fluvial erosion algorithm described in [9] assumes that vertices of the terrain graph located on the edge of the domain never change the height. As a result, the edge of the resulting terrain is always at 0. As the rest of the land is uplifted, this leads to the formation of a steep slope along the entire length of the border.

This problem can be eliminated by determining the vertices' elevations on the domain's border in the same way as for the other vertices. However, since some of these vertices do not have receivers, it is necessary to modify the original formula [9, eq. (2)].

The proposed modification consists in replacing the value $h_w(t + \delta t)$ with 0 and replacing the distance $\|\mathbf{p}_v - \mathbf{p}_w\|$ with a large constant, for example, the domain size. This is equivalent to introducing artificial vertices of constant elevation 0 into the terrain graph, which are ignored when determining rivers and lakes. So the formula for vertices without a receiver is:

$$h_v(t + \delta t) = \frac{h_v(t) + \delta t u_v}{1 + \dfrac{k\sqrt{A_v}}{D}\delta t},$$

where D is the domain size.

Thanks to this modification, the resulting terrain has a natural-looking edge. It gives the impression of a fragment cut out from a larger area, allowing it to be used in various applications without additional processing. In addition, the terrain more accurately reflects the features marked on the uplift speed map, giving the user more control over the outcome.

IV. Results analysis

The new proposed version of the whole terrain generation algorithm has been implemented in C++ as an *Unreal Engine 5* plugin. The plugin allows the user to generate terrain based on required input parameters and save it in one of the available formats. All reported results were obtained on a PC with an AMD Ryzen 7 1700 3.0 GHz and 16 GB RAM, supplied with an NVIDIA GeForce GTX 1080.

The algorithm's parameters allow the user to control the amount of water accumulating in lakes as well as decide the threshold for considering a stream of water to be a river. Figure 4 shows an example of the result of our method with generated lakes and rivers on display.

Figure 5 demonstrates the influence of the water accumulation parameters on resulting lakes. An increase in precipitation creates an increase in water flow in the system, which raises the water levels of the lakes, possibly even causing overflow

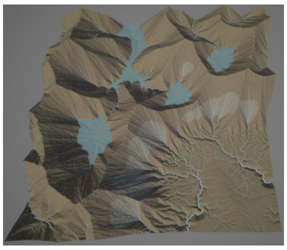
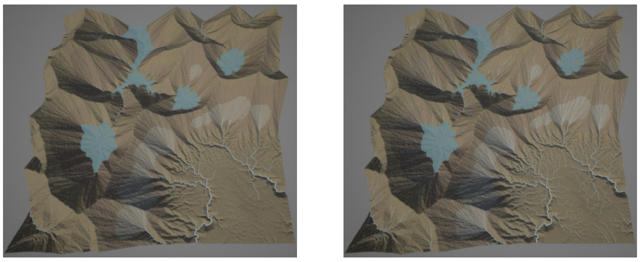
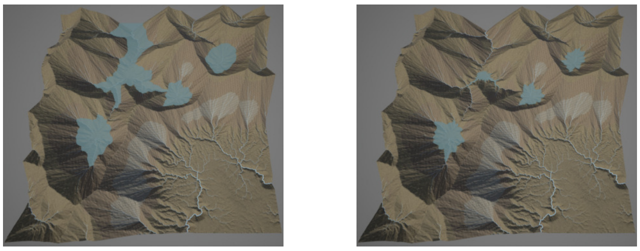


Fig. 4. Terrain with generated lakes and rivers



(a) Terrain with a high river threshold (b) Terrain with a low river threshold

Fig. 6. The impact of the river threshold parameter on the resulting terrain



(a) Terrain with increased precipitation (b) Terrain with increased evaporation rate

Fig. 5. The impact of water accumulation parameters on the resulting terrain

and joining some of them together. On the other hand, increasing the evaporation rate causes the water levels to fall, resulting in some of the lakes completely drying out.

Figure 6 shows the river network for different inclusion threshold levels. Setting a high value for the threshold results in highlighting only the most pronounced rivers in the basin, while keeping the value low allows us to keep even the small streams in the scene.

The computation time of the algorithm has been tested for multiple input terrain parameter presets:

- *Basic* – a constant terrain uplift on the whole domain
- *Perlin* – uplift of the domain defined by 2D Perlin noise
- *MountainSimple*, *MountainSteep* – uplift map typical for mountain regions; the second preset allows for steeper slopes
- *MountainLayers* – uplift map typical for mountain regions but with each layer subdivided into 5 additional sublayers

The key aspect of the algorithm is the time required for generating the terrain (Figure 7). The performed tests show that generating the terrain in low resolution (a 5 km × 5 km region with mesh vertices about 2.5 m from each other) takes from a dozen to several dozen seconds (Figure 7a). Such a resolution is enough to give the user a general idea of the topographical aspects of the resulting terrain. It allows for a quick iterative process of refining the result. The highest resolution (a 5 km × 5 km region with mesh vertices about 0.5 m from each other) in most cases computes in less than 30 minutes (Figure 7c) and allows the user to glimpse the precise shape of the result.

In one specific case, the time is substantially lengthened. When the uplift map is constant, the terrain remains mostly flat, resulting in a significant amount of puddles instead of a handful of big lakes resulting in the apparent rise of processing time (Fig. 7c) caused by the parts of the method that are dependant on the number of lakes. The other parameter set that was an outlier in terms of processing time, but this time by not that big of a margin, was the Mountain Layers preset. This is caused by the extended calculation time of the erosion step since multiple layers need to be considered.

It is worth noting that since the node and edge counts in the terrain graph are not dependent on the water accumulation input data but only on the size of the terrain and the sampling rate, the execution time of the lake generation process is invariant to different input parameter sets.

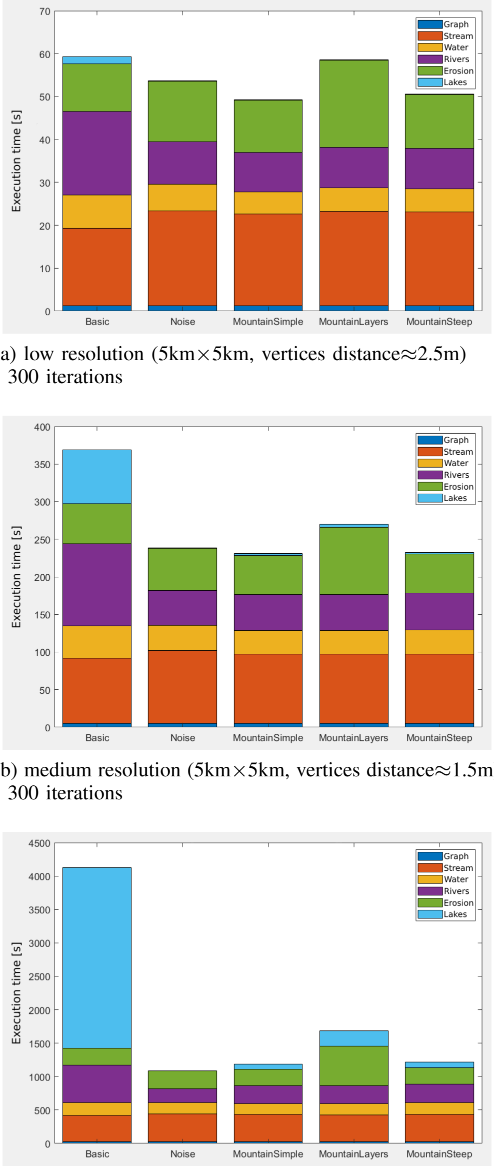
The extension of the terrain generation algorithm by adding the water accumulation and lake generation step impacts the total processing time marginally. The only exception is the *Basic* preset. However, since it is an artificial example created only for the purposes of testing the processing time and the method aimed at generating realistic environments, this result can be disregarded. For the highest resolution examples presented in the paper, the additional lake calculations only take up from 15% to 23% of the total computation time.

V. Conclusion

The modification presented in this paper to the base fluvial erosion algorithm [9] can allow the user to obtain diversified terrains in a few seconds to several dozen minutes. While the base method was only capable of generating terrains consisting of topographical features such as mountains, hills, plains the extension developed by us makes it possible to also add bodies of water to that list – rivers and lakes. This creates a much more natural looking environment since not only do we end up with more diverse features in the result but all of these features are also simulated simultaneously, interacting with each other during the generation process and in turn leading to a more lifelike result.

Rivers and lakes are created fully automatically. Rivers flow from higher to lower terrain points and naturally join together. Lakes form in natural depressions where water accumulates. The user can suggest the desired locations of water reservoirs by using the uplift velocity map.

During further research, the presented new version of the algorithm, could also allow us to diversify the terrain's shape by changing the erosion result for submerged vertices. Such



(a) low resolution (5km×5km, vertices distance≈2.5m) - 300 iterations

(b) medium resolution (5km×5km, vertices distance≈1.5m) - 300 iterations

(c) high resolution (5km×5km, vertices distance≈0.5m) - 300 iterations

Fig. 7. Execution time results for different resolutions

a change could depend on a total change in height for all nodes belonging to a given lake or the phenomenon of sedimentation. It is worth noting that in the described version of the modified algorithm, it is not necessary to compute lakes after each iteration. However, any method improvement that would affect the erosion result for underwater vertices will also require information about the water level during the algorithm's evaluation.

Further improvements of the method may also relate to the user's level of control over the final terrain – for instance the ability to determine the location of water reservoirs. Currently, the user can suggest where rivers and lakes should appear using the uplift map, but their occurrence is not guaranteed. An alternative is to add the option of forcing some connections in the flow graph, freezing the height of selected vertices, or imposing additional conditions on them, which would facilitate the insertion of selected topographic features. This however, could lead to some loss of realism over more control given to the user.

Acknowledgment

The research was supported by Warsaw University of Technology faculty grant.

References

[1] J.-D. Champagnac, P. Molnar, C. Sue, and F. Herman, "Tectonics, climate, and mountain topography," *Journal of Geophysical Research: Solid Earth*, vol. 117, no. B2, 2012. doi: 10.1029/2011JB008348

[2] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, "A survey on procedural modelling for virtual worlds," *Computer Graphics Forum*, vol. 33, no. 6, pp. 31–50, 2014. doi: 10.1111/cgf.12276

[3] J.-D. Génevaux, E. Galin, A. Peytavie, E. Guérin, C. Briquet, F. Grosbellet, and B. Benes, "Terrain modelling from feature primitives," *Computer Graphics Forum*, vol. 34, no. 6, pp. 198–210, 2015. doi: 10.1111/cgf.12530

[4] M. Luckner and K. Rzążewska, "3D model reconstruction and evaluation using a collection of points extracted from the series of photographs," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, 2014. doi: 10.15439/2014F304 pp. 669–677.

[5] E. Michel, A. Emilien, and M.-P. Cani, "Generation of folded terrains from simple vector maps," in *Eurographics 2015 short paper proceedings*. The Eurographics Association, 2015. doi: 10.2312/egsh.20151019

[6] O. Argudo, E. Galin, A. Peytavie, A. Paris, J. Gain, and E. Guérin, "Orometry-based terrain analysis and synthesis," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–12, 2019. doi: 10.1145/3355089.3356535

[7] D. B. Adams, "Feature-based interactive terrain sketching," Master's thesis, Brigham Young University, 2009. [Online]. Available: hdl.lib.byu.edu/1877/etd3221

[8] S. T. Teoh, "River and coastal action in automatic terrain generation," in *Proceedings of the 2008 International Conference on Computer Graphics and Virtual Reality*, 2008, pp. 3–9. [Online]. Available: citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=316be57e56662a0113a5678eb29dd5b3b951694a

[9] G. Cordonnier, J. Braun, M.-P. Cani, B. Benes, E. Galin, A. Peytavie, and E. Guérin, "Large scale terrain generation from tectonic uplift and fluvial erosion," *Computer Graphics Forum*, vol. 35, no. 2, pp. 165–175, 2016. doi: 10.1111/cgf.12820

[10] M. Becher, M. Krone, G. Reina, and T. Ertl, "Feature-based volumetric terrain generation and decoration," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 2, pp. 1283–1296, 2019. doi: 10.1109/TVCG.2017.2762304

[11] A. Paris, E. Galin, A. Peytavie, E. Guérin, and J. Gain, "Terrain amplification with implicit 3d features," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–15, 2019. doi: 10.1145/3342765

[12] A. Peytavie, E. Galin, J. Grosjean, and S. Mérillou, "Arches: a framework for modeling complex terrains," *Computer Graphics Forum*, vol. 28, no. 2, pp. 457–467, 2009. doi: 10.1111/j.1467-8659.2009.01385.x

[13] G. Cordonnier, M.-P. Cani, B. Benes, J. Braun, and E. Galin, "Sculpting mountains: Interactive terrain modeling based on subsurface geology," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 5, pp. 1756–1769, 2017. doi: 10.1109/TVCG.2017.2689022

[14] B. B. Cael, A. J. Heathcote, and D. A. Seekell, "The volume and mean depth of Earth's lakes," *Geophysical Research Letters*, vol. 44, no. 1, pp. 209–218, 2017. doi: 10.1002/2016GL071378

[15] R. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972. doi: 10.1109/SWAT.1971.10

[16] J. Cheriyan and S. N. Maheshwari, "Analysis of preflow push algorithms for maximum network flow," *SIAM Journal on Computing*, vol. 18, no. 6, pp. 1057–1086, 1989. doi: 10.1137/0218072

DOI: 10.15439/2023F960
ISSN 2300-5963 ACSIS, Vol. 35

Applying Knowledge Distillation to Improve Weed Mapping With Drones

Giovanna Castellano, Pasquale De Marinis, Gennaro Vessio
0000-0002-6489-8628
0000-0001-8935-9156
0000-0002-0883-2691
Department of Computer Science
University of Bari Aldo Moro
Bari, Italy
Email: {giovanna.castellano, pasquale.demarinis, gennaro.vessio}@uniba.it

***Abstract*—In precision agriculture, non-invasive remote sensing using UAVs can be employed to observe crops in visible and non-visible spectra. This paper investigates the effectiveness of state-of-the-art knowledge distillation techniques for mapping weeds with drones, an essential component of precision agriculture that employs remote sensing to monitor crops and weeds. The study introduces a lightweight Vision Transformer-based model that achieves optimal weed mapping capabilities while maintaining minimal computation time. The research shows that the student model effectively learns from the teacher model using the WeedMap dataset, achieving accurate results suitable for mobile platforms such as drones, with only 0.5 GMacs compared to 42.5 GMacs of the teacher model. The trained models obtained an F1 score of 0.863 and 0.631 on two data subsets, with a performance improvement of 2 and 7 points, respectively, over the undistilled model. The study results suggest that developing efficient computer vision algorithms on drones can significantly improve agricultural management practices, leading to greater profitability and environmental sustainability.**

I. Introduction

PRECISION agriculture has become increasingly important due to the growth of the world's population—which is expected to reach nine billion people by 2050—and the resulting need to increase food production [1]. However, the resources that sustain agriculture are becoming increasingly scarce, degraded, and vulnerable to climate change. This has led to the need for more sustainable and efficient agricultural practices that make optimal use of available resources.

Unmanned aerial vehicles (UAVs), also known as drones, have emerged as valuable tools for precision agriculture due to their versatility and affordability [2]. They can capture high-resolution images and data from agricultural fields, which can be used to monitor crop growth, identify diseases and pests, and optimize irrigation. By providing farmers with accurate and timely information, drones can help reduce costs, increase yields, and minimize the use of inputs such as water, fertilizer, and pesticides. Traditional ground-based methods, such as manual exploration or satellite remote sensing, cannot match the level of detail that drones can provide. Drones can fly over crops and capture images and data in real-time, enabling farmers to make informed decisions quickly. Another benefit of drones is their ability to quickly and efficiently cover large areas. Drones can fly over fields and acquire data in hours, which would otherwise take days or weeks with traditional methods. This can help farmers save time and resources and make more timely decisions. Newer techniques now allow their application in swarm configurations, significantly improving their efficiency and the range of tasks they can perform [3].

One area of particular interest in precision agriculture that could benefit significantly from using drones is weed mapping. Weed mapping is critical in precision agriculture because it allows farmers to apply herbicides accurately and reduce overuse, which can cause environmental and health problems. Convolutional Neural Network (CNN)-based models have been recently proposed to perform semantic segmentation and identify weeds in images captured by drones or other aerial vehicles [4], [5], [6]. However, these models are typically computationally expensive, making them difficult to implement on drones with limited processing power and limited battery. These limitations and the need for real-time responses call for lightweight solutions.

In cases like this, knowledge distillation (KD) can help. KD is a technique that allows a smaller model to learn from a larger, more complex model, often referred to as a *teacher* [7]. The goal is to transfer the knowledge learned from the teacher to a smaller, lighter *student*, which can be used on resource-limited devices such as drones.

In this study, we explore the application of different knowledge distillation techniques to evaluate their effectiveness in the context of drone-based weed mapping. The focus is on using the WeedMap dataset [8] and proposing specific, extra-lightweight architectural designs that aim to achieve superior weed mapping capabilities while maintaining minimal computational time. This technology can improve weed management practices, leading to more sustainable and efficient agriculture.

The rest of this paper is structured as follows. Section II reviews related work. Sections III and IV describe materials and methods. Section V reports and discusses the experimental results. Section VI concludes the paper and outlines future developments in our research.

II. Related Work

Many precision agriculture tasks have been addressed thanks to the recent development of computer vision techniques and remote sensing data collection methodologies. Recent tasks include disease and pest identification, abiotic stress assessment, growth monitoring, crop yield prediction, and weed mapping.

A. Weed Mapping

Weed mapping is a semantic segmentation task in which each pixel of an image is assigned a class. Deep learning algorithms have significantly outperformed more traditional techniques in this task. Dos Santos et al. [4] were among the first to demonstrate the superiority of CNNs, particularly AlexNet, over traditional machine learning approaches such as SVM and Random Forest. Lottes et al. [9] used a CNN with two decoders, one for detecting stem position and the other for plant segmentation. They used the BoniRob dataset and one collected with a UAV for evaluation. They obtained an mAP of 79.2% for stem detection and 75.3% for segmentation. SegNet with ResNet50 as an encoder was used in [6], achieving an F1 score of 64.6%.

Depending on the bands acquired, multispectral images can contain information on the growth and health status of a plant and its species. Consequently, they can improve the accuracy of deep learning models compared to models trained only with RGB. Furthermore, multispectral image sensors can be easily integrated into UAVs. The popular U-Net was used on a dataset available on the Internet to separate weeds from crops and soil, achieving an F1 score of 89% and a mIoU of 98% [10]. WeedNet, a semantic segmentation network based on SegNet, was developed and trained on the WeedMap dataset and achieved an F1 score of 80% [5]. WeedMap contains two sets of images of sugar beet fields collected in Germany (Rheinbach) and Switzerland (Eschikon). Both were collected using UAVs equipped with multispectral cameras. The former used a 5-channel RedEdge-M camera, while the latter used a 4-channel Sequoia camera. The authors also trained SegNet using various combinations of the acquired channels, resulting in an AUC of 84.3% [8]. On the WeedMap dataset, the DeepLabV3 architecture for semantic segmentation was compared with SegNet and U-Net, achieving an F1 score of 81% on the Rheinbach subset [11]. Mozzam et al. [12] used patch-based training with a modified VGG model on the same dataset, also using the Eschikon subset. The patches were chosen by hand, and those that contained both classes were removed. On the Rheinbach subset, the accuracy reached 92%, and on Eschikon 90%. WeedMap has been used here for benchmarking purposes, as it has become the preferred dataset in several works due to its volume and quality.

B. Knowledge Distillation

Knowledge distillation is a popular method for "compressing" neural models [7]. However, previous studies have shown a significant size gap between student and teacher networks, which limits the effectiveness of KD. Mirzadeh et al. [13] showed that the gap between student and teacher could not be arbitrary and proposed a solution for this problem, called *teacher assistant knowledge distillation*. This method requires training one or more teaching assistant networks and is computationally expensive. In addition, errors of the teaching assistant can accumulate and transfer to the student. To mitigate these problems, Jafari et al. [14] introduced *annealing KD*, which achieves state-of-the-art performance in natural language understanding and computer vision tasks. In annealing KD, the teacher's goals are annealed to convey the information provided by the teacher to the student gradually. The predictions are annealed using a temperature parameter that gradually decreases during training. After this first phase, the student is trained with the ground truth. Although it can handle the capability gap problem, annealing KD is still vulnerable to noisy data and teachers' results. In addition, the training requires deciding when to switch from the first to the second phase, which can be challenging. Inspired by continuation optimization, Jafari et al. [15] tried to solve the above problems by introducing *continuation KD*. This method starts with an easy-to-train objective function that becomes increasingly complex as the training progresses, allowing the student model to learn and gradually improve its performance.

Several works have already used KD to obtain lightweight models suitable for UAVs. For example, Li et al. [16] have applied this technique for video saliency estimation, while Liu et al. [17], Yu [18], Ding et al. [19], and Luo et al. [20] used it for object detection, object recognition, action recognition, and UAV delivery, respectively. However, to our knowledge, no work has investigated knowledge distillation to produce efficient and accurate models tailored for UAVs in the context of weed mapping.

III. Materials

This section will discuss the dataset and the preprocessing and augmentation techniques implemented.

A. Dataset

Discrimination between weeds and crop plants is a significant challenge in agricultural imaging. To address this problem, the present study relied on the WeedMap dataset proposed by Sa et al. [8], which consists of orthomosaic maps of sugar beet fields (variety Beta vulgaris "Samuela") with three classes: background, crop, and weeds. Despite the limited number of classes, the dataset demonstrates a level of complexity comparable to larger semantic segmentation datasets, such as Cityscapes [21]. This complexity stems from the subtle differences between crop and weed classes and the limited number of examples. In particular, cultivated plants occupy 15-20 pixels, while individual weeds occupy only 5-10 pixels. Therefore, using pre-trained models or additional techniques, such as data augmentation, is necessary to improve the performance of the segmentation model.

More specifically, the dataset used in this study includes eight orthomosaic maps divided into two subsets based on the location of the fields: Rheinbach in Germany and Eschikon in Switzerland. The orthomosaic maps were further divided into tiles, resulting in 971 tiles for the Rheinbach subset and 700 tiles for the Eschikon subset. Data were acquired using two unmanned aerial vehicles: a DJI Inspire2 equipped with a RedEdge-M camera for the Rheinbach subset and a DJI Mavic Pro with a Sequoia camera for the Eschikon subset. The RedEdge-M camera acquired five channels of raw image data, including red, green, blue, near-infrared (NIR), and red edge (RE). On the other hand, the Sequoia camera acquired the same channels except for the blue channel.

B. Data Preprocessing

Although the dataset has already been thoroughly processed by the authors [8], further preprocessing is necessary. First, since the orthomosaic maps are not rectangles, they have some black areas at the edges, which generate many completely black tiles. As a first preprocessing step, these tiles were removed, reducing the dataset to 557 tiles for the Rheinbach subset and 561 for the Sequoia subset. In addition, the height of each tile of 360 is quite problematic because it must be divisible by $2^i, i > 3$ as some convolutional filters would require. For this reason, four crops of size 256×256 were extracted from each image. This also reduces the computational load.

C. Data Augmentation

The authors of the dataset implemented a *random horizontal flip* during their experiments, a commonly used augmentation technique because it does not distort the image. However, a *vertical flip* can also be used without problems for images acquired from a nadir direction. Similarly, random rotations can be applied to images with a degree range of 0 to 360. To resolve the class imbalance, *selective random rotation* was used, applying the increment only to examples containing at least one pixel of the minority class (i.e., weeds). This approach helped to increase the number of images containing weeds, improving the model's ability to learn the minority class. This technique was applied only to the Eschikon subset, which had a weed representation of only 0.166% and later increased to 0.499%. The Rheinbach subset, on the other hand, already had a weed representation of 0.706%.

IV. Methods

Two different architectures, both Transformers, were used. The first, used as a teacher, is HRNet+OCR+PSA [22], [23]. The second is a modified version of Lawin [24], which is a Vision Transformer (ViT) [25] suitable for semantic segmentation. In particular, we lightened the architecture by obtaining an extra-light model, which we named "Lawin-L0". Both architectures have achieved state-of-the-art results on the Cityscapes, ADE20K, and COCO-Stuff reference datasets. However, they cannot be directly applied as-is. Weed mapping, like other precision agriculture tasks, benefits from some bands of the non-visible spectrum, particularly the NIR and RE bands. This hinders the application of deep learning models, which are typically suited to be fed RGB images. A concatenation layer with a modified first convolution layer is needed to handle other channels besides RGB.

A. Teacher

The network used as a teacher in this paper is a modified version of the HRNet+OCR+PSA architecture [23]. It comprises the HRNet+PSA backbone, a version of HRNetV2 [26] with the Polarized Self Attention (PSA) block as the attention block. High Resolution Net V2 (HRNetV2) is an ad-hoc architecture for semantic segmentation, derived from HRNet. It consists of four stages, each of which produces high-resolution features. The stages consist of repeated multi-resolution blocks. Each block consists of a multi-resolution group convolution and a multi-resolution convolution. The multi-resolution group convolution is an extension of the group convolution. It separates the input channels into multiple subsets of channels and applies a standard convolution to each subset at different spatial resolutions. In a multi-resolution convolution, on the other hand, the input and output subsets are fully connected, and each connection is a standard convolution. The output channels for each subset are the sum of the results of the convolutions on each subset of input channels.

The HRNet features are then transmitted to the decoder, which serves as the OCR (Object Contextual Representation) module [27]. The central concept of the OCR module is that the label assigned to each pixel must match the label of the object containing it. To achieve this goal, the OCR module first extracts soft object regions from feature maps and then, using an attention mechanism, computes representations of the object regions together with the pixel representations. These representations are used to improve the final representations employed to predict the segmentation map.

Starting from the basic architecture, we modified it specifically to solve the weed mapping problem (see Fig. 1). In particular, to handle additional input channels, we modified the first input layer so that it can accept not only visible channels but also non-visible channels.

B. Student

Like the teacher and other semantic segmentation models, Lawin includes an encoder and a decoder. The encoder is a type of architecture called Mix Transformer (MiT) [28], explicitly designed for semantic segmentation as an alternative to the original ViT. MiT can produce multilevel features with different resolutions, similar to CNNs, and outputs a feature map for each Transformer block. This hierarchical representation provides high-level coarse-grained and low-level fine-grained features that generally improve performance in semantic segmentation. For example, starting from an RGB image of size $3 \times H \times W$, the first Transformer block generates a feature map of size $C_1 \times \frac{H}{4} \times \frac{W}{4}$, where C_1 is the chosen embedding dimension. Then, each subsequent transformer block takes as input the feature maps of the previous block and produces a feature map $F_i \in \mathbb{R}^{C_i \times \frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}}}$, where i is the index of the block.

The decoder uses a technique called Large Window Attention Spatial Pyramid Pooling (LawinASPP), which consists of five different branches, including a pooling layer, a shortcut connection, and three large window attentions with different

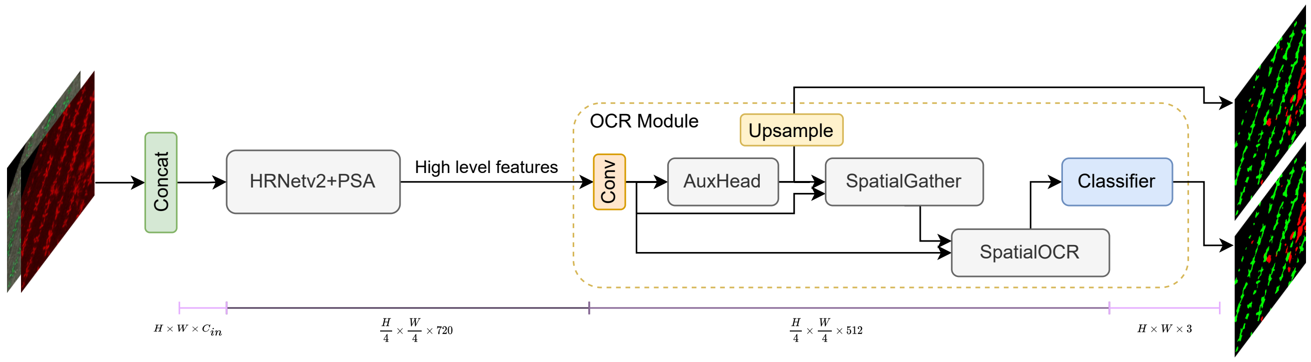


Fig. 1. HRNet+OCR+PSA modified for weed mapping ("Conv" is a convolutional layer with a 3×3 kernel, while H, W, and C_{in} are the height, width, and channels of the input images, respectively). Unlike the original version, the model accepts both visible and non-visible channels as input.

context sizes. The pooling branch handles the global context, while the three window attentions serve as local context extractors. The last two outputs of the encoder are processed with a standard multilayer perceptron and an upsampling operation. However, the first output is not processed by LawinASPP but concatenated with its output. A final linear transformation is applied to create the final segmentation map, followed by an upsampling operation. The resulting map is a probability distribution that assigns each pixel to a specific class.

As for HRNet+OCR+PSA, starting from the basic architecture of Lawin, we modified it to accept visible and non-visible channels. Moreover, to further improve performance, we propose a lighter variant of Lawin, Lawin-L0, which uses SegFormer as the encoder [28]. SegFormer, in turn, has five variants based on embedding size and model depth (B0, B1, B2, B3, B4). Lawin-L0 has a halved embedding size and a halved number of blocks in each phase compared with Lawin-B0. In addition, Lawin-B0 repeats each of the four stages twice, while Lawin-L0 repeats them only once. The embedding size in Lawin-B0 is (32, 64, 160, 256), while in Lawin-L0 it is (16, 32, 80, 128). The decoder also reflects these sizes, further reducing the computational cost. Lawin-L0 is shown in Fig. 2.

C. Vanilla Knowledge Distillation

In precision agriculture using drones, obtaining lightweight models is critical. Knowledge distillation can help achieve higher accuracy on lighter models. In particular, for weed mapping, we want to show that lightweight models can achieve comparable performance to large models when adequately trained.

In *vanilla KD*, the loss is a weighted sum of a task loss and a distillation loss:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{task} + \alpha\mathcal{L}_{KD}$$

where $\mathcal{L}_{task}$ is the task-specific loss function for the student model, $\mathcal{L}_{KD}$ is the distillation loss, and α is a hyperparameter controlling the relative weighting between the two losses.

D. Teacher Assistant Knowledge Distillation (TAKD)

This variant of KD consists of two distillation stages [13], where the first stage involves the teacher model distilling its knowledge to an intermediate assistant model, and the second stage involves the assistant model further distilling knowledge to the final student model. This approach is designed to leverage the expertise of the teacher model while mitigating the impact of the capability gap between the teacher and student models. Introducing the assistant model as an intermediary aims to minimize the loss of information and enable a more effective transfer of knowledge to the student model. As an assistant, we used Lawin, specifically the B0 variant, which falls between the teacher and student models in terms of complexity.

E. Annealing Knowledge Distillation

This technique tries to solve the capacity gap problem by modifying the KD loss and introducing a dynamic temperature function to make the student's training gradual and smooth [14]. The process is divided into two phases: Stage I, gradual training of the student to imitate the teacher using the annealing KD loss; Stage II, fine-tuning the student with hard labels using the task loss. The resulting loss can be defined as:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{KD}^{annealing}(i), & \text{Stage I: } 1 \leq \mathcal{T}_i \leq \tau_{max} \\ \mathcal{L}_{task}, & \text{Stage II: } \mathcal{T}_n = 1 \end{cases}$$

where i denotes the epoch index in the training process with n maximum epochs for Stage I and $\mathcal{T}_i$ the corresponding temperature value. At epoch (i), $\mathcal{L}_{KD}^{annealing}(i)$ is defined as:

$$\mathcal{L}_{KD}^{annealing}(i) = ||z_s(x) - \Phi(\mathcal{T}_i)z_t(x)||_2^2$$

$$\Phi(\mathcal{T}) = 1 - \frac{\mathcal{T}-1}{\tau_{max}}, 1 \leq \mathcal{T} \leq \tau_{max}, \mathcal{T} \in \mathbb{N}$$

In this case, the distillation loss is a mean squared error (MSE) between the student's logits ($z_s(x)$) and an annealed version of the teacher's logits ($z_t(x)$), obtained by multiplying them by the annealing function Φ. The annealing function is a monotonically decreasing function $\Phi : [1, \tau_{max}] \in \mathbb{N} \to [0,1] \in \mathbb{R}$. τ_{max} represents the hyperparameter for the maximum temperature.

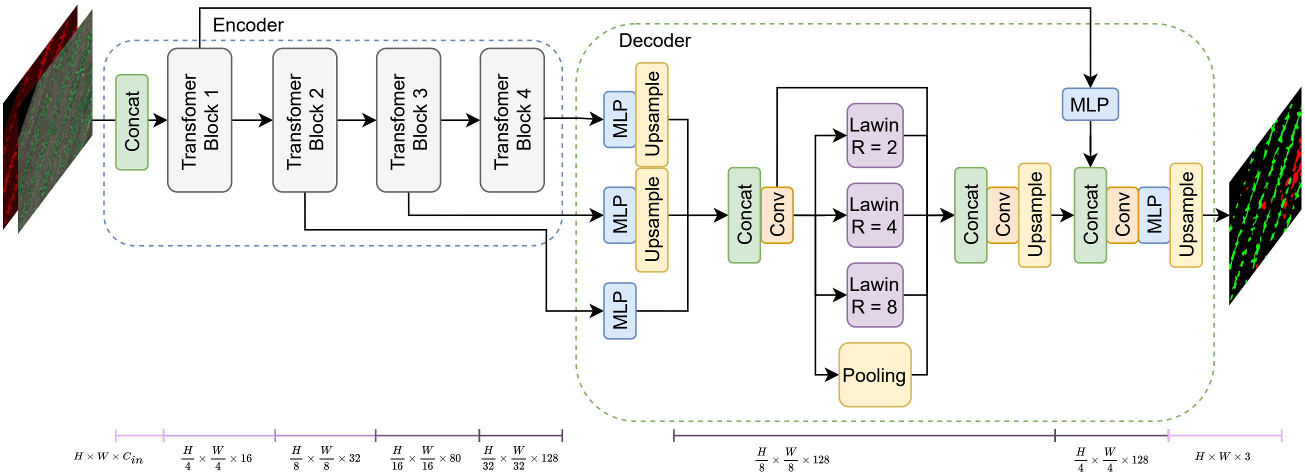

Encoder
Concat
Transfomer Block 1
Transfomer Block 2
Transfomer Block 3
Transfomer Block 4
Decoder
MLP
Upsample
MLP
Upsample
MLP
Concat
Conv
Lawin R = 2
Lawin R = 4
Lawin R = 8
Pooling
Concat
Conv
Upsample
MLP
Concat
Conv
MLP
Upsample
$H \times W \times C_{in}$
$\frac{H}{4} \times \frac{W}{4} \times 16$
$\frac{H}{8} \times \frac{W}{8} \times 32$
$\frac{H}{16} \times \frac{W}{16} \times 80$
$\frac{H}{32} \times \frac{W}{32} \times 128$
$\frac{H}{8} \times \frac{W}{8} \times 128$
$\frac{H}{4} \times \frac{W}{4} \times 128$
$H \times W \times 3$

Fig. 2. Lawin-L0 (“Conv” is a convolutional layer with a 3 × 3 kernel, “MLP” stands for fully-connected layer, and H, W, and C_{in} are the height, width, and channels of the input images, respectively). As before, unlike the original version, the model accepts both visible and non-visible channels as input and has a reduced number of hidden channels.

F. Continuation Knowledge Distillation

This technique is based on continuation optimization, a method for solving optimization problems by gradually increasing the complexity of the objective function [15]. The idea is to start with an easy-to-train objective function that becomes increasingly complex as the training progresses, allowing the student model to learn and gradually improve its performance. The loss function is defined as:

$$\mathcal{L} = \psi(i)\mathcal{L}_{task} + (1 - \psi(i))\mathcal{L}_{KD}$$

where $\psi(i)$ is a monotonically increasing linear function $\psi : \mathbb{N} \to [0, 1] \in \mathbb{R}$. The ψ function is defined as:

$$\psi(i) = \begin{cases} \frac{i}{N_{epochs}} & \text{if } i \leq N_{epochs} \\ 1 & \text{if } i > N_{epochs} \end{cases}$$

where i is the epoch index and N_{epochs} is the number of epochs the student model will learn from the teacher.

$\mathcal{L}_{KD}$ is the distillation loss, defined as the MSE between the student's logits $(z_s(x))$ and the annealed teacher's logits $(z_t(x))$ similarly to annealing KD, but with a defined margin m:

$$\mathcal{L}_{KD} = \max\{0, ||z_s(x) - \Phi(\mathcal{T}_i)z_t(x)||_2^2 - \Phi(\mathcal{T}_i)m\}$$

where $\Phi(\mathcal{T}_i)$ is the annealing function.

V. Experiments

This section presents our experimental setup, followed by the quantitative and qualitative results of crop and weed segmentation.

A. Experimental Setup

For the Rheinbach subset, we used the same train-test subdivision applied in [8] and [29], namely [000, 001, 002, 004]–[003]. Due to the limited number of images containing weeds in the Eschikon subset test set, we opted for a different split to ensure more reliable results, i.e., [005, 007]–[006]. All channels provided in the two subsets are fed to the models. We used Adam as an optimizer for model training, with batch size 6, a maximum number of epochs of 500, and an early stop with patience 25. Specifically, the validation sets were randomly extracted from the training sets for early stopping. We used the regional mutual information [30] as the task loss, weighted by the frequency of pixel classes, as done in [8]:

$$\mathcal{L}_{RMI} = \lambda w_c \mathcal{L}_{ce}(y, p) + (1 - \lambda)\frac{1}{B}\sum_{b=1}^{B}\sum_{c=1}^{C}(-I_l^{b,c}(\mathbf{Y}; \mathbf{P}))$$

where $\lambda \in [0, 1]$ is a weight factor, $\mathcal{L}_{ce}$ is the cross-entropy, B denotes the batch size, C the number of classes, $I_l^{b,c}(\mathbf{Y}; \mathbf{P})$ is the mutual information between the ground truth and the prediction, and $\mathbf{Y}$ and $\mathbf{P}$ are the ground truth and the prediction, respectively. w_c are the class weights, calculated as:

$$w_c = \frac{FoA(c)}{\widetilde{FoA(c)}}$$

$$FoA(c) = \frac{I_c}{I}$$

where $f(x)_c$ is the probability of the true class c predicted by the model, $\widetilde{FoA(c)}$ is the median of $FoA(c)$ by varying c, I_c is the number of pixels in c, and I is the total number of pixels. The eventual application of these weights is a hyperparameter in the experiments.

In addition, we used Kullback-Leibler divergence and MSE for vanilla KD as the distillation loss, with $\alpha = 0.8$. The same hyperparameters were used for TAKD. As for annealing KD, we used an initial temperature of 0.9. In addition, it is not possible to use early stopping because the temperature is a function of epochs, so we set 50, 100, and 150 as the maximum epochs. For continuation KD, we used the same hyperparameters as for annealing KD, but given the way ψ

TABLE I
COMPARISON OF DIFFERENT KD TECHNIQUES WITH THE TEACHER AND THE UNDISTILLED MODEL FOR BOTH SUBSETS OF DATA

	Rheinbach				**Eschikon**			
	F1	**F1 Background**	**F1 Crop**	**F1 Weed**	**F1**	**F1 Background**	**F1 Crop**	**F1 Weed**
Teacher	0.868	0.990	0.877	0.737	0.656	0.995	0.816	0.155
No distillation	0.843	0.989	0.847	0.696	0.565	0.995	0.653	0.046
Vanilla KD	0.855	0.989	0.855	0.719	0.624	0.990	0.668	0.215
TAKD	0.850	0.988	0.842	0.720	**0.631**	0.992	0.763	0.138
Annealing KD	0.853	0.989	0.849	0.722	0.581	0.994	0.691	0.059
Continuation KD	**0.863**	0.990	0.862	0.736	0.553	0.987	0.666	0.005

is defined, the early stopping technique can be used. The experiments were performed on an RTX 3080 Ti with 12 GB of VRAM.

We used the F1 score for each class and macro-averaged to assess the models quantitatively. It was calculated as follows:

$$F1 = \frac{2TP}{2TP + FP + FN}$$

where TP stands for true positives, FP for false positives, and FN for false negatives.

B. Quantitative Results

The study's results are presented in Table I, which compares the performance of the teacher model used alone, the student model without teacher knowledge distillation, and the student model with knowledge distillation. We do not show the results obtained with all possible hyperparameter configurations for better readability, but only the best ones obtained. Furthermore, it is worth noting that the different KD methods did not modify the number of parameters of the resulting models but only the training procedure.

The study found that in both subsets, the use of knowledge distillation improved performance. Interestingly, the vanilla KD approach was sufficient to improve the score. Continuation KD outperformed the other models for the Rheinbach subset, with an average F1 score of only 0.005 lower than the teacher model and 0.001 lower for the weed class. On the other hand, the Eschikon subset presented difficulties due to the differences between the crops of the three collected fields, highlighting the difficulty of generalization in weed mapping. However, the teacher assistant technique showed promise, improving performance by up to 7%. The F1 score for weeds also showed a substantial increase, from 0.046 without KD to 0.138 with TAKD. In addition, the F1 score for crop class showed a significant increase from 0.653 to 0.753. Although there was a reduction in the background class score, the score was still high enough to make the reduction insignificant. State-of-the-art models for the WeedMap dataset include DeepLabV3 [11], which obtained an F1 score of 0.81 on the Rheinbach subset. Instead, while not using F1 in their experiments [8], replicating SegNet scores 0.445 on Eschikon and 0.836 on Rheinbach. Therefore, our lightweight model outperforms even the dataset's state-of-the-art.

Regarding computational time and complexity, Lawin-L0 has a relatively low amount of computational operations, measured in GMacs (giga multiply-accumulated operations), equal to 0.5 GMacs, and a relatively low number of parameters, measured in millions, equal to 0.98 million parameters. On the other hand, HRNet+OCR+PSA has significantly higher computational requirements, with 42.04 GMacs and 75.74 million parameters. Despite the substantial reduction in parameters, a satisfactory level of accuracy can be achieved.

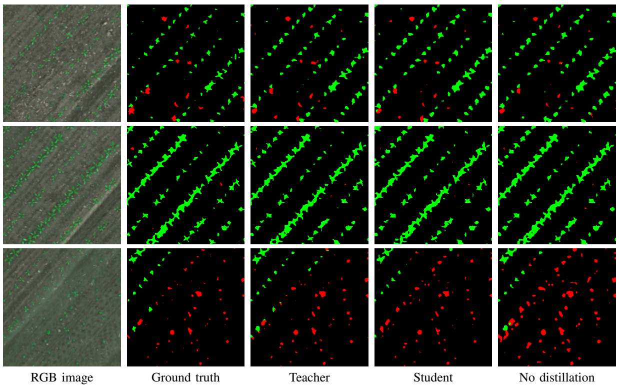


Fig. 3. Segmentation examples performed on test field [003] (black = background, green = crop, red = weed).

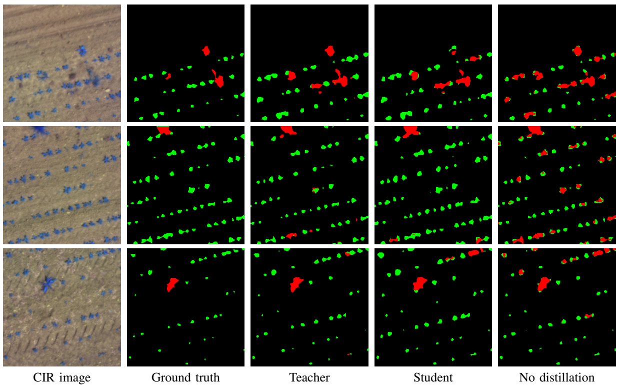


Fig. 4. Segmentation examples performed on test field [006] (black = background, green = crop, red = weed). CIR stands for color infrared.

C. Qualitative Evaluation

A qualitative assessment of weed mapping found that the segmentation maps obtained from the students' model were the same quality as those obtained from the teacher model. This is reflected in the F1 score obtained from both models. This indicates that the students' model can learn effectively from the teacher model and produce accurate weed mapping results. Examples of segmentation maps obtained as output from the best execution of student Lawin-L0 and teacher HRNet+OCR+PSA on Rheinbach are shown in Fig. 3. In the Rheinbach subset, the segmentation maps reveal no apparent visual difference in the F1 score, despite all models performing well. However, distinct differences are observed for the weed class in the Eschikon subset, shown in Fig. 4. The models show an imbalance toward the weed class, with high recall, low precision, and many false positives. In particular, this phenomenon is more pronounced in the undistilled model than in the distilled model. The segmentation maps produced by the distilled model resemble those of the teacher model, indicating the significant influence of the teacher model on students' predictions.

VI. Conclusion

Our study demonstrated that knowledge distillation in the context of drone-based weed mapping could be effectively used to train an extremely lightweight model with only 0.5 GMacs. Our results indicate that this model can provide high-level performance while maintaining a short inference time. This makes them ideal for mobile platforms such as drones or ground control stations, which can also be smartphone devices. In particular, we have shown that the student model can learn from the teacher model and produce accurate results. Applying knowledge distillation to the Rheinbach subset resulted in a relatively modest 2% increase in the F1 score. However, the technique proved more effective for the more challenging Eschikon subset, where a significant 7% improvement was achieved. This highlights the practical value of knowledge distillation in this particular context. A potential future direction of this research could be to apply knowledge distillation to other tasks similar to weed mapping. This would allow us to evaluate the effectiveness of our approach further and explore its potential applications in a broader range of contexts where lightweight models are critical (for example, in crowd flow detection [31]).

In conclusion, developing effective and efficient computer vision algorithms on drones can significantly improve weed management practices, leading to more sustainable and efficient farming practices. By enabling farmers to quickly and easily identify infested areas and prioritize control efforts, this technology has significant implications for precision agriculture, ultimately increasing profitability and environmental sustainability.

Acknowledgment

The research of Pasquale De Marinis is funded by a Ph.D. fellowship within the framework of the Italian "D.M. n. 352, April 9, 2022" - under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 - Ph.D. Project "Computer Vision techniques for sustainable AI applications using drones", co-supported by "Exprivia S.p.A." (CUP H91I22000410007).

References

[1] FAO, "How to Feed the World in 2050. Insights from an Expert Meet," *FAO*, 2009.

[2] S. G. Vougioukas, "Agricultural Robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 365–392, 2019.

[3] K. Danilchenko and M. Segal, "An efficient connected swarm deployment via deep learning," in *Proceedings of the 16th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 25. IEEE, 2021, p. 1–7. [Online]. Available: http://dx.doi.org/10.15439/2021F001

[4] A. dos Santos Ferreira, D. M. Freitas, G. G. da Silva, H. Pistori, and M. T. Folhes, "Weed Detection in Soybean Crops Using ConvNets," *Computers and Electronics in Agriculture*, vol. 143, pp. 314–324, 2017.

[5] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. Nieto, and R. Siegwart, "Weednet: Dense Semantic Weed Classification Using Multispectral Images and Mav for Smart Farming," *IEEE robotics and automation letters*, vol. 3, no. 1, pp. 588–595, 2017.

[6] B. Hobba, S. Akıncı, and A. H. Göktogan, "Efficient Herbicide Spray Pattern Generation for Site-Specific Weed Management Practices Using Semantic Segmentation on UAV Imagery," in *Australasian Conference on Robotics and Automation (ACRA-2021)*, 2021, pp. 1–10.

[7] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021.

[8] I. Sa, M. Popović, R. Khanna, Z. Chen, P. Lottes, F. Liebisch, J. Nieto, C. Stachniss, A. Walter, and R. Siegwart, "WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming," *Remote Sensing*, vol. 10, no. 9, p. 1423, 2018.

[9] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss, "Joint Stem Detection and Crop-Weed Classification for Plant-Specific Treatment in Precision Farming," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 8233–8238.

[10] M. Á. Chicchón Apaza, H. M. B. Monzón, and R. Alcarria, "Semantic Segmentation of Weeds and Crops in Multispectral Images by Using a Convolutional Neural Networks Based on U-Net," in *International Conference on Applied Technologies*. Springer, 2019, pp. 473–485.

[11] W. Ramirez, P. Achanccaray, LF. Mendoza, and MAC. Pacheco, "Deep Convolutional Neural Networks for Weed Detection in Agricultural Crops Using Optical Aerial Images," in *2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS)*. IEEE, 2020, pp. 133–137.

[12] S. I. Moazzam, U. S. Khan, W. S. Qureshi, M. I. Tiwana, N. Rashid, W. S. Alasmary, J. Iqbal, and A. Hamza, "A Patch-Image Based Classification Approach for Detection of Weeds in Sugar Beet Crop," *IEEE access : practical innovations, open solutions*, vol. 9, pp. 121 698–121 715, 2021.

[13] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved Knowledge Distillation via Teacher Assistant," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5191–5198, Apr. 2020.

[14] A. Jafari, M. Rezagholizadeh, P. Sharma, and A. Ghodsi, "Annealing Knowledge Distillation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 2493–2504.

[15] A. Jafari, I. Kobyzev, M. Rezagholizadeh, P. Poupart, and A. Ghodsi, "Continuation KD: Improved Knowledge Distillation through the Lens of Continuation Optimization," Dec. 2022.

[16] J. Li, K. Fu, S. Zhao, and S. Ge, "Spatiotemporal Knowledge Distillation for Efficient Estimation of Aerial Video Saliency," *IEEE Transactions on Image Processing*, vol. 29, pp. 1902–1914, 2020.

[17] B.-Y. Liu, H.-X. Chen, Z. Huang, X. Liu, and Y.-Z. Yang, "ZoomInNet: A Novel Small Object Detector in Drone Images with Cross-Scale Knowledge Distillation," *Remote Sensing*, vol. 13, no. 6, p. 1198, Jan. 2021.

[18] G. Yu, "Data-Free Knowledge Distillation for Privacy-Preserving Efficient UAV Networks," in *2022 6th International Conference on Robotics and Automation Sciences (ICRAS)*, Jun. 2022, pp. 52–56.

[19] M. Ding, N. Li, Z. Song, R. Zhang, X. Zhang, and H. Zhou, "A Lightweight Action Recognition Method for Unmanned-Aerial-Vehicle Video," in *2020 IEEE 3rd International Conference on Electronics and Communication Engineering (ICECE)*, Dec. 2020, pp. 181–185.

[20] "KeepEdge: A Knowledge Distillation Empowered Edge Intelligence Framework for Visual Assisted Positioning in UAV Delivery," https://ieeexplore.ieee.org/abstract/document/9732222/.

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[22] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical Multi-Scale Attention for Semantic Segmentation," May 2020.

[23] H. Liu, F. Liu, X. Fan, and D. Huang, "Polarized Self-Attention: Towards High-quality Pixel-wise Regression," Jul. 2021.

[24] H. Yan, C. Zhang, and M. Wu, "Lawin Transformer: Improving Semantic Segmentation Transformer with Multi-Scale Representations via Large Window Attention," *arXiv preprint arXiv:2201.01615*, 2022.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929*, 2020.

[26] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-Resolution Representations for Labeling Pixels and Regions," Apr. 2019.

[27] Y. Yuan, X. Chen, and J. Wang, "Object-Contextual Representations for Semantic Segmentation," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, vol. 12351, pp. 173–190.

[28] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[29] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[30] S. Zhao, Y. Wang, Z. Yang, and D. Cai, "Region Mutual Information Loss for Semantic Segmentation," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[31] G. Castellano, E. Cotardo, C. Mencar, and G. Vessio, "Density-Based Clustering with Fully-Convolutional Networks for Crowd Flow Detection from Drones," *Neurocomputing*, 2023.

DOI: 10.15439/2023F378
ISSN 2300-5963 ACSIS, Vol. 35

Mutual Learning Algorithm for Kidney Cyst, Kidney Tumor, and Kidney Stone Diagnosis

Sabrina Tarin Chowdhury
ORCID ID: 0009-0005-5854-8161
Indiana University-Purdue University
Computer and Information Science
IN 46202, Indianapolis, USA
Email : sabchow@iupui.edu

Snehasis Mukhopadhyay
ORCID ID : 0009-0000-0836-2901
Indiana University-Purdue University
Computer and Information Science
IN 46202, Indianapolis, USA
Email : smukhopa@iupui.edu

Kumpati S. Narendra
Yale University
Center for Systems Science
CT 06520, New Haven, USA
Email : kumpati.narendra@yale.edu

Abstract— **Mutual learning is a machine learning algorithm where multiple machine learning algorithms share knowledge among themselves to improve themselves. The utilization of mutual learning algorithms can effectively enhance the efficiency of machine learning and neural networks within a multi-agent system. This approach is particularly useful in scenarios where the system cannot be adequately trained with a large dataset. By exchanging data in a dynamic teacher-student network system, mutual learning can result in efficient learning outcomes. Typically, a large network serves as a static teacher and transfers data to smaller networks, referred to as student networks, to improve their efficiency. In this study, we aim to demonstrate that two small networks can dynamically alternate between the roles of teacher and student to share knowledge, resulting in improved efficiency for both networks. To exemplify this concept, we apply a mutual learning algorithm using convolutional neural networks (CNNs) and Support Vector Machine (SVM) to accurately identify the kidney diseases – cyst, tumor and stone using image classification algorithm.**

Index Terms—**Mutual learning, teacher-student network, CNN, model distillation, Kidney Disease, Cyst, Tumor, Stone**

I. Introduction

MACHINE learning has a great potential to revolutionize the medical science. It can be a big aid to the current medical system specially in disease diagnosis in early stage. Moreover, in many third world countries, there is extreme shortage of doctors and hence, the doctors do not have the ample time and energy to invest behind a patient. In those cases, machine learning algorithm can provide work as a 'second brain' for the doctor to aid him in disease diagnosis. Even in the first world countries the machine learning algorithms can provide a third eye to the doctors. Couple of recent studies [1][2] shows that an estimated 5% of the outpatients get wrong diagnosis in US every year. Particularly when a patient is in serious medical condition, the misdiagnosis is common. A study shows that almost 20% of the serious patients are misdiagnosed at the level of primary care [3]. Misdiagnosis can result in serious harm of the patient and almost one-third of the misdiagnosed patient face harmful consequences [4].

Nevertheless, use of machine learning in medical diagnosis is still limited due to several facts. Experiments proved that for attaining considerable accuracy level, machine learning training dataset requires abundant amount of patient data [5][6][7][8][9][10]. However, the machine learning diagnostic algorithms could not reach the accuracy of the doctors in differential diagnosis [11][12][13] yet specially where there can be multiple possible causes of a patient disease symptoms. Lots of research has been done in disease diagnosis but few has shown considerable accuracies (accuracy>90%) because as stated earlier, wrong diagnosis can be potentially dangerous for the patient. For example, machine learning algorithms for heart disease detections show accuracies in the range between 80% to 90% [14][15][16][17] while only one result shows accuracy of 94% [16] using SVM algorithm. Machine learning algorithm for diabetes detection shows accuracies between the 70% to 80% [18][19][20] [21][22] while only one result using Naïve Bayes algorithm shows accuracy of 95%. Lever disease detection algorithms [23][24] [25] shows even poorer accuracies (around 70%) while only couple of results shows accuracies over 96% [25] using Naïve Bayes and functional tree algorithm. Research [26] [27][28] shows poor accuracies for Hepatitis detection also (ranges between 70% to 90%) while only one result [26] shows accuracy of 96% using Naïve Bayes algorithm.

Medical diagnosis AI must have very high accuracy (>97%) on unknown dataset [29][30]. For that, medical diagnosis AI must have training dataset greater than 10000 to build reliable system [29][30]. But building a big and comprehensive dataset in medical sector is not easy because patient data sharing has lots of confidentiality and legal bindings. Moreover, a key step in machine learning is to train the algorithm/network properly to achieve good accuracy. Most often, in order to achieve good enough accuracy, machine learning algorithms have to be trained using fairly large number of training datapoints. It may require large memory to execute and get power and computation resource hungry training algorithms which can be a tremendous problem in many systems. Hence, there is a big demand to find small and fast training mechanisms. Mutual learning [31][32][33] is one of the interesting concepts explored to execute faster and efficient training of machine learning algorithm and share knowledge among the algorithms.

Mutual learning algorithm is a machine learning algorithm where multiple machine learning algorithms learns from different sources and then share their knowledge among themselves (fig 1) so that all the agents can improve their classification and prediction accuracies simultaneously. Mutual learning algorithm can be an efficient mechanism for improving the machine learning and neural network efficiency in a multi-agent system. Most of the model distillation systems use a big network, known as teacher network, to pass its learning to a smaller network to train the later [34][35][36]. Static teacher-student network data passing is one-way which incurs several issues like mimicry loss [37]. Furthermore, the teacher network does not see any improvement in efficiency. On the other hand, in mutual learning, a variation of model distillation, there is no static teacher-student network. Rather, role of teacher and student network can change dynamically based on the training sample, and both networks can train each other (fig 1). Thus, efficiencies and accuracies of both

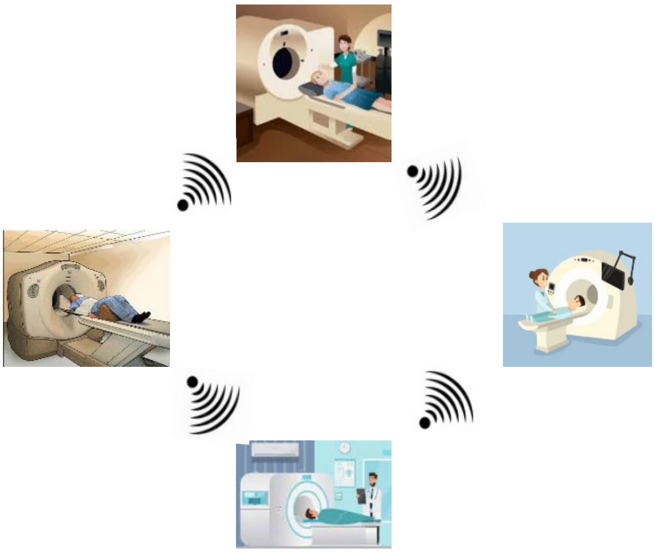


Figure 1: Medical diagnosis machines sharing knowledge.

networks improve. The concept can be particularly useful to increase the efficiency of multiple small networks simultaneously which can be used to do parallel processing. Furthermore, mutual learning can be particularly useful when a big single training dataset is not available. Rather, small, distributed sets of training data are available and some of the training data may need to be relabeled. In this way mutual learning can be very helpful in the field of machine learning for medical diagnosis. Since the machines are sharing data among themselves in non-human readable format, the issue of privacy breaching can also be avoided and gradually over the time the machine will improve their accuracy levels to an acceptable threshold for medical diagnosis.

In this paper, we demonstrate the concept of such mutual learning via the different kidney disease (cyst, tumor and stone in kidneys) detection using scanned kidney images in ref[38] dataset. We used CNN and SVM algorithm to implement the kidney image pattern recognition. We show how the mutual learning can improve the efficiency of both the networks simultaneously and how it can reduce the overall training time significantly. The accuracy keeps improving over the time as more and more training data is shared between machine learning algorithms. We also show that the increasing the number of networks in mutual learning can significantly improve the efficiency of all the networks involved without significantly increasing the training time. The mutual learning is implemented between both homogeneous and heterogeneous agents and comparison between relative accuracy improvements are discussed and analyzed in detail.

II. Background on Mutual Learning

For centuries, philosophers have delved into the study of learning theory, while psychologists, engineers, and computer scientists have joined this exploration over the past seventy years. As a vast, multidisciplinary field, learning theory has been the subject of investigation using a multitude of methods. Historically, the majority of research has focused on a single agent, typically a learner or student, operating in a deterministic or stochastic environment. However, this report marks a significant departure from traditional learning approaches as it investigates the dynamics of multiple agents learning from each other.

The fundamental inquiry, posed in various iterations, revolves around how two or more agents or entities, operating within the same or similar environment and attempting to solve the same or similar problem, can share information to increase operational efficiency.

Mutual learning problems are prevalent and encompass a wide spectrum, ranging from straightforward deterministic optimization to exceedingly complex ones that are challenging to articulate accurately. The problems addressed in this report span multiple areas, such as deterministic optimization in high-dimensional spaces, stochastic reinforcement learning in static/stationary environments (learning automata), employing both deterministic and stochastic schemes, learning in dynamic environments, such as those defined by Markov Decision Processes, and learning/adaptation by multiple agents in dynamic environments described by deterministic or stochastic difference and differential equations.

Mutual learning can occur between two humans, a human and a machine, or between two machines. Researchers in fields such as social psychology are particularly interested in the former. However, the importance of human-machine interactions has become increasingly evident, especially in the context of interactions between human-driven and fully autonomous vehicles. We anticipate that machine-machine learning will lead to complex yet intriguing problems that will keep investigators occupied for many years. The quantitative approach used in this and future reports will not only facilitate efficient collaboration between machines, but also shed light on the limitations of such collaboration. Specifically, the study aims to address the question of whether two agents, each utilizing schemes that result in optimal behavior in stationary environments, may arrive at an incorrect conclusion when learning from one another.

A. *Related Research*

In the study conducted by Ikemoto *et al* [39], human-robot mutual learning and co-adaptation were explored, inspired by human parenting behavior. In the context of artificial neural networks, Zhang *et al* [37] examined the problem of a group of deep neural networks learning from each other for a classification task. The researchers concluded that small neural networks with mutual learning could outperform a single powerful teacher network. Nie *et al* [40] investigated mutual learning to achieve superior performance in two related yet distinct computer vision tasks, namely human parsing and pose estimation. Another relevant research theme is multi-agent learning systems, where agents focus on different subtasks of a complex problem and work together to solve it, similar to mathematical game theory. Panait and Luke [41] provide an overview of this well-established field, emphasizing inter-agent communication, task decomposition, and scalability in multi-agent systems. In contrast to multi-agent systems, mutual learning involves agents that collaborate to solve the same or similar tasks and act as (partial) teachers to each other to enhance their learning.

III. Kidney Disease Diagnosis with Machine Learning : Literature Review

In literature, many machine learning studies has been done on pattern recognition-based kidney diseases detection.

However, most of the work focused on chronic kidney disease detection [42][43][44] since it can be fatal for the patient. Few works [45][46][47] has been done on machine learning based kidney cyst, tumor and stone detection. Ref [45] used CNN to show an accuracy of 99.52% while ref [46] showed 99.30% accuracy using VGG16. Ref [47] showd impressive 99.98% accuracy using DenseNet201. All these works are conducted on a certain dataset and parameters are optimized for maximum accuracy. The principle concern is, how these trained networks would behave for a new unknown set of data. It is unlikely that they will show similar accuracy for unknown dataset. For achieving good universal accuracy, the algorithms are needed to be trained over times by datasets from various sources and types.

IV. THE PATTERN RECOGNITION PROBLEM

For every pattern recognition problem, there is a sample space S consisting of elements. These elements, also known as pattern samples or samples, are the focus of a specific problem. For example, in character recognition, a sample would refer to a specific character, while in medical diagnosis, it would be a set of symptoms. The goal of a pattern recognizer is to develop a rule that divides the sample space into partitions where all elements belonging to the same partition are equivalent. Essentially, the sample space S is divided into equivalence classes.

A. Design of a Pattern Recognizer

The basic structure of the pattern recognizer consists of the following three stages:

1) Physical Measurement: In the first stage, each sample (converted from physical measurements) corresponds to a set of ordered numbers.

2) Feature Extraction: In the second stage those features which are judged to be important for the recognition problem are derived from the elements in stage 1 (this is more of an art than a science).

3) Classification : This is the crucial part of the procedure in which the elements are classified on the basis of their features.

The above separation of the problem into the three stages of physical measurement, feature extraction and classification is mainly for convenience. The choice of the features is critical to the success of the classification process, but the former depends on the physical measurements made on the samples. If the original set S can be expressed as $S = \cup_i \ C(i)$, where C(i) is an equivalence class, the objective of pattern recognition is to find a mapping such that all elements of C(i) are mapped to the same class.

B. Methods for Pattern Recognition

Historically, the methods proposed for pattern recognition , belong to two distinct periods. During the 1960s,70s, and part of 80s most of the methods assumed that the two sets could be separated by a hyperplane in the feature space. Hence the problem was to determine the orientation of the hyperplane based on the test samples. A very large number of outstanding text books exist in which the convergence of the hyperplane to the desired orientation, based on the information contained in the training samples, is rigorously proved.

The rise of methods based on artificial neural networks followed the period referred to earlier. Significantly more complex decision surfaces than hyper-planes (manifolds in the feature space) could be used to perform pattern classification. The methods were significantly less analytic in nature, but the success of the methods in real problems eventually made them the preferred methods in practical applications. In much of the literature in the 1960s, 70s, and 80s, the discriminant surfaces were linear hyperplanes and the classification rule was based on whether a sample lies above or below the hyperplane (i.e., whether the projection of the sample on the normal to the hyperplane is positive or negative). In such situations, classification is the process by which the hyperplane is determined by the training samples, and, if a solution exists, using the hyperplane to classify test samples whose classifications are unknown.

Pattern recognition based on the above methods are well known. When 'Mutual Learning' is used for such problems, it is assumed that agents (or machines) with different training sets (datasets) are attempting to solve the same problem. Our interest lies in the questions that can arise when they communicate with each other and whether they can improve their performance in some sense by such communication.

V. THE MUTUAL LEARNING ALGORITHM

In this section, we describe our main contribution of the paper, i.e., the mutual learning algorithm. As stated earlier, we propose two different algorithms for mutual learning.

A. Algorithm I - Similarity Matching Based Mutual Learning

In this algorithm, the two agents take turns in serving the other agent's teacher, i.e., there are no predefined assigned roles as teacher and student between the two agents. When an agent encounters a novel data-point that is not present in either agent's dataset, the two agents engage in mutual communication where each one looks for points that are 'similar' in its training set, and the corresponding data labels. The agent that wins this competition, i.e., has (labeled) examples that are more 'similar' to the novel data point than the other agent, serves as the teacher and the other agent takes the role of the student for the novel data point. The two agents augment their training with their respective (labeled) training datasets intermittently with such mutual learning with exploratory 'novel' datapoints not included in the training set.

The expectation in such mutual learning is that, by leveraging the 'expertise' of the other agent on specific 'unseen' examples, an agent may be able to overcome the inadequacy of its training data, and will be able to learn faster. That is, each agent will be able to achieve superior classification performance than what is possible without such mutual learning. The following describes the proposed mutual learning algorithm in a more precise manner.

Let D be a (training) dataset of ground truth with N examples. At each iteration k, each agent picks an example from D with probability p(k). With probability (1-p(k)) they choose a random input with unknown class label. If they choose an example from D, learning proceeds as in the isolated learning case. If they choose a random input X, each agent A_1 and A_2 determines their output classification for M <= N examples in D that are closest to X. Whichever agent A_i has higher number of correct labels for these M ground truth examples, is considered the teacher, with the other agent A_j

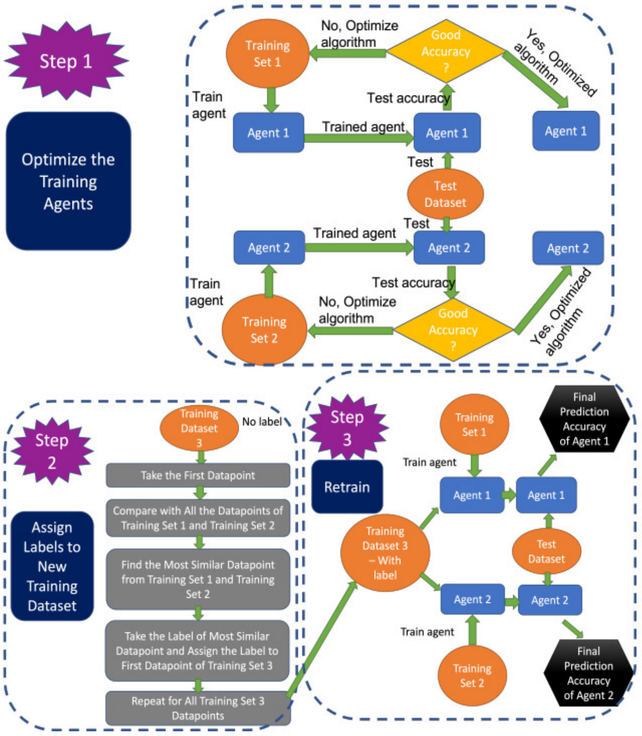


Figure 2: Pictorial representation of Algorithm 1

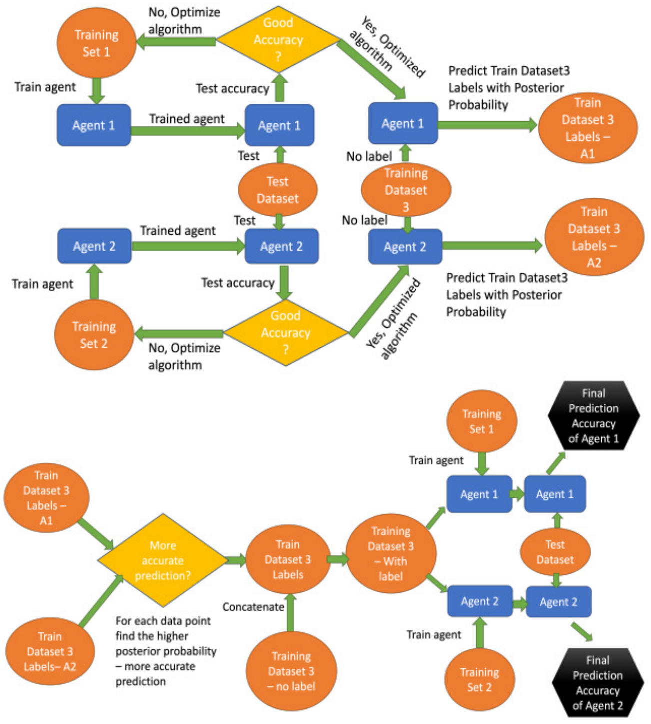

Training Set 1
No, Optimize algorithm
Good Accuracy ?
Yes, Optimized algorithm
Train agent
Test accuracy
Agent 1
Trained agent
Agent 1
Agent 1
Predict Train Dataset3 Labels with Posterior Probability
Train Dataset 3 Labels – A1
Test
Test Dataset
No label
Training Dataset 3
No label
Test
Agent 2
Trained agent
Agent 2
Agent 2
Train Dataset 3 Labels – A2
Train agent
Test accuracy
Predict Train Dataset3 Labels with Posterior Probability
Training Set 2
No, Optimize algorithm
Good Accuracy ?
Yes, Optimized algorithm
Training Set 1
Final Prediction Accuracy of Agent 1
Train agent
Train Dataset 3 Labels – A1
More accurate prediction?
Train 3 Dataset Labels
Training Dataset 3 – With label
Agent 1
Agent 1
Test Dataset
Concatenate
Agent 2
Agent 2
Train Dataset 3 Labels– A2
For each data point find the higher posterior probability – more accurate prediction
Training Dataset 3 – no label
Train agent
Training Set 2
Final Prediction Accuracy of Agent 2

Figure 3: Pictorial representation of Algorithm II

being the student. The student A_j then updates itself for X treating the output label generated by the teacher A_i for X as the target. X can be viewed as an off-line experiment, while choosing any example from D is an on-line experiment. p(k) starts at k=1 with a value of 0.5 (say) but is increased towards 1 with increasing k as, say,

$$p(k) = 0.5*(2- 1/\sqrt{k})$$

Therefore, eventually both agents only use ground truths. This provides the guarantee that eventually both agents will only use on-line experiments, i.e., isolated learning with ground truths, and therefore is guaranteed to perform no worse than isolated learning. What we hope to demonstrate that with an appropriate scheduling of p(k), the mutually learning team can achieve a given high level of accuracy with fewer total online experiments than that required by an isolated learning agent, by making used of the off-line experiments.

Fig.2 shows a pictorial representation of algorithm 1. Fist, two agents are trained with different training datasets first (fig 2 top). When an agent encounters a novel datapoint, it matches the novel data point with all the datapoints in both the training datasets of agent1 and agent2. In this way it tries to find the ’most similar’ and previously seen datapoint. The new novel datapoint is labeled the same as the ’most similar’ datapoint (fig 1 bottom). All the novel datapoints are labelled in the same way and both the agents are retrained with new novel dataset and their respective old training dataset (fig 2 bottom).

B. Algorithm II - Previous Knowledge Based Mutual Learning

In this algorithm the two agents are trained with different training datasets first (fig 3 top). When an agent encounters a novel data point, both the agents predict the label of the data point based on their own knowledge and previous training. The confidence level or the prediction accuracy probability for each agent are also calculated at the same time (fig 3 top). The agent with higher confidence wins the competition and the novel data point is labeled according to the winner agent’s prediction (fig 3 bottom). The winner agent then works as the teacher and the other agent becomes the student for that particular new data point. In this way, for a particular unseen data point, we are trying to see which agent has the most capability to predict based on previous knowledge and training. In this way the ‘less capable’ agent for that particular agent learns something new and as a result, it will be able to predict for similar datapoint faster and with better accuracy in future. The two agents augment their training with their respective (labeled) training datasets intermittently with such mutual learning with exploratory ‘novel’ datapoints not included in the training set.

Assuming there are total ‘N’ number of classes in a multiclass classification problem. For a particular unseen new data point the probability of predictions from output neurons can be written as follows according to Gibb’s measure

$$\sum_{k=1}^{N} Pr(Y_i = k) = \sum_{k=1}^{N} \frac{1}{Z} e^{\beta_k \cdot X_i} = \frac{1}{Z} \sum_{k=1}^{N} e^{\beta_k \cdot X_i} = 1$$

Here we assumed softmax activation function. Solving for Z (normalization constant) gives

$$\sum_{k=1}^{N} e^{\beta_k \cdot X_i} = Z$$

Therefore, the prediction confidence for different classes are given as

$$Pr(Y_i = 1) = \frac{e^{\beta_1 \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}, \quad Pr(Y_i = 2) = \frac{e^{\beta_2 \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}$$

$$\dots\dots\dots\dots\dots\dots\dots\dots$$

$$Pr(Y_i = N) = \frac{e^{\beta_N \cdot X_i}}{\sum_{k=1}^{N} e^{\beta_k \cdot X_i}}$$

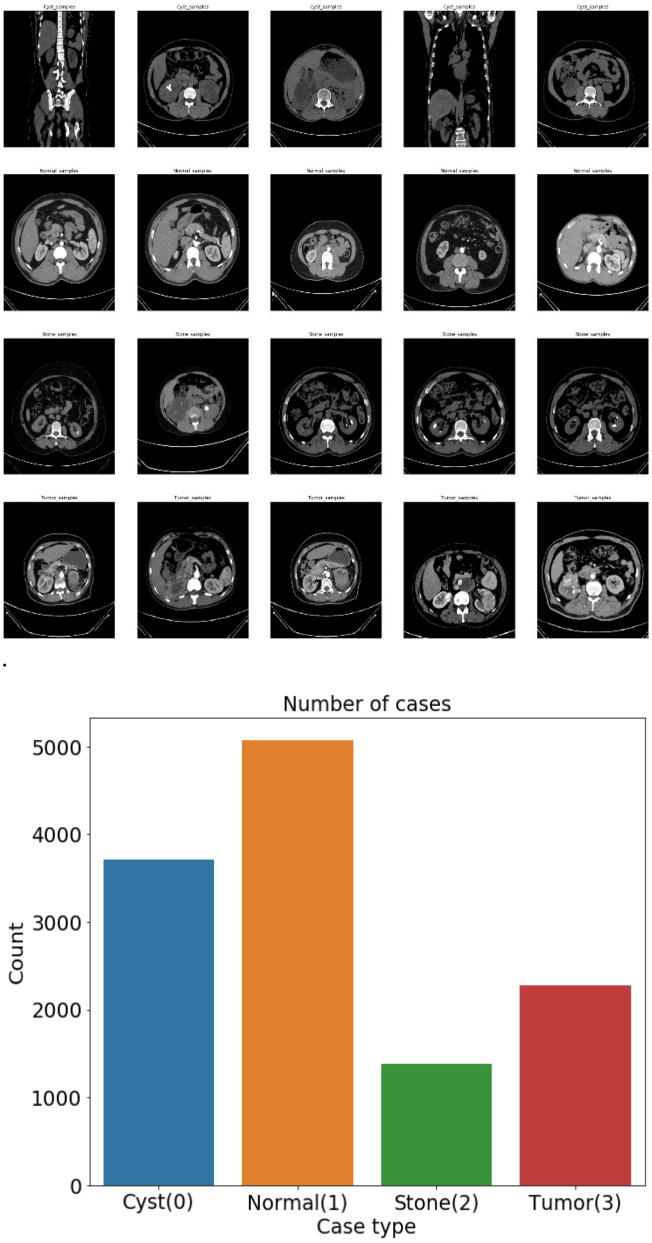


Figure 4 : (top) Samples of Kidney images. The first row shwos kidney images with Cyst. Second row are the scans of nomal kidneys. The third row contains the scans of kidneys with stone. The last row shows the scans of kidneys with tumors. (bottom) Histogram plot showing the number of samples in different data types in full training dataset.

The maximum probability is the predicted class for the unseen new data point. In this algorithm we determine the probabilities of predicted class from different agents and consider the prediction of the most ‘confident’ agent.

VI. Simulation Framework for Mutual Learning Demonstration in CNN

A. *Training Data*

The dataset is a collection of 12,446 unique data within it in which the cyst contains 3,709, normal 5,077, stone 1,377, and tumor 2,283 [38]. The size of each image is 865x700 pixels (figure 4-top). The training dataset contains 10,000 images and the testing dataset contains 2,446 images. The images roes through one hot encoding and the pixel values are normalized. The dataset is not well distributed especially if you consider the individual data points for different kidney diseases. Nevertheless, in our real world, we rarely have well distributed dataset rather it is expected that the we will see skewed data distribution. We intentionally kept the raw dataset in the same way so that we can demonstrate that our proposed mutual learning algorithm is not strongly affected by the skewness of the data.

The training dataset is equally divided into 4 smaller datasets – each containing 2,500 training points. The CNN is trained separately with the full training dataset and small training datasets and accuracy is tested against the testing dataset.

B. *CNN Model*

The CNN model consists of two parts - the data preprocessing part and the artificial neural network. The CNN preprocessing steps contains several layers. First, there is a 2D convolution layers both with 28 output filter with 3x3 kernel and ReLU activation function. The output of the convolution layer goes through a 2x2 maxpool layer. The input images again goes though a convolution layers both with 64 output filter with 3x3 kernel and ReLU activation function. The output from the convolution layer goes through a 2x2 maxpool layer, another convolution layers both with 64 output filter with 3x3 kernel and ReLU activation function. Then finally there is a flatten layer to convert the 2D data to 1D array.

The second layer in CNN is a fully connected artificial neural network (ANN). The ANN basically consist of four layers -three hidden dense layers and the output layer. The first hidden dense layer contains 640 neurons. Output from the first hidden layer goes through a dropout layer to avoid overfitting. Second hidden dense layer consists of 264 neurons, third hidden dense layer consists of 64 neurons and the output layer consists of 4 neurons. Each output neuron indicates a probability of kidney with cyst, normal kidney, kidney with stone and kidney with tumor respectively. The output neuron indicating highest probability is the final result.

C. *SVM Model*

In this work we used two types of SVM network – SVM with linear kernel and non-linear (sigmoid) kernel. The hyperparameters are optimized for both linear and non-linear SVM to achieve good accuracy with full training dataset.

VII. Results and Discussion

First, the CNN network is trained with the complete training dataset and tested for efficiency using the testing dataset. Next, the CNN network is trained with each of the smaller training datasets, and efficiency is tested against the testing dataset each time. Two CNN networks are then trained simultaneously using different small and randomly chosen training datasets, followed by interaction between the networks for data exchange and mutual training. The efficiency of both CNN networks is then tested using the testing dataset. The experiment is further repeated with four teacher-student networks.

A. *CNN Network Testing Efficiency with Full Training Dataset and One-fourth Training Dataset*

The CNN is first trained with the full training dataset first and the accuracy is tested against the testing dataset. Training with the full dataset is a time and memory hungry procedure. The whole process took almost 35 minutes in a machine with 1.6 GHz dual-core processor and 8GB 2,133 MHz RAM. The

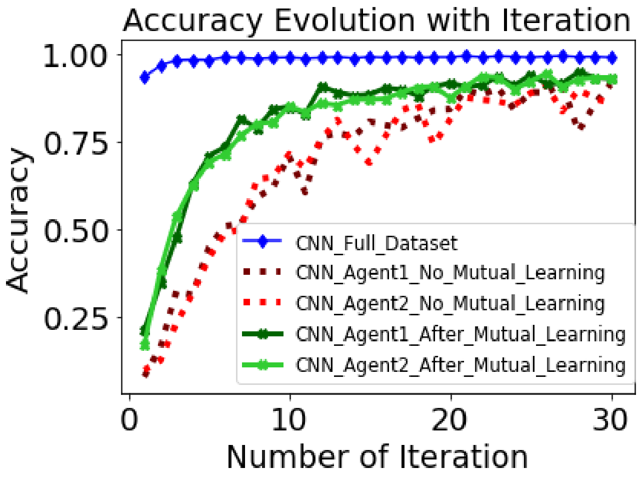


Fig. 5. CNN network accuracy plot with iteration for mutual learning. The top blue plot indicates the accuracy when the CNN is trained with the full training dataset. The light green and dark green plots indicates the CNN network accuracy with iteration after mututal learning. The light red and dark red lines are the accuracy plots before mutual learning.

maximum accuracy is 99.54%. The accuracy plot with iteration for full training dataset is shown in fig. 5 (top blue curve). The CNN is then trained with the each of the small datasets and accuracy is tested every time. In fig. 5 accuracy plots are shown for two of the small training sets. The maximum accuracy for all the training sets is found to be 84.1%.

B. Mutual Learning with Two Teacher-Student Networks Using Algorithm I

Two CNN networks are trained with two randomly selected small training sets (dataset 1 and dataset 2) first. The two networks then share their knowledge with each other and get trained further. For that, the labels are removed from another small training dataset (dataset 3).

Each datapoint (kidney scnas) are compared with the data points of dataset 1 and dataset 2 that are used to train the two CNNs. The comparison is done by comparing each corresponding pixels of the image and calculating the root mean square value of the difference. The closest data point from the two training datasets is assumed to contain the right label, The rest of the data points in dataset 3 are relabeled in this way and the two CNNs are trained accordingly.

Fig. 5 shows the accuracy plots before and after the mutual learning. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy before mutual learning was 84.1% while the maximum accuracy increased to 90.45% after mutual learning. In this way, mutual learning can help when system cannot be trained with big single dataset or there is scarcity of single big training dataset. Here both the network accuracy increases simultaneously which adds to the benefit of dynamic teacher-student network instead of static teacher-student network.

C. Mutual Learning with Two Teacher-Student Networks Using Algorithm II

Two CNN networks are trained with two different, randomly selected small one-fourth training sets again (dataset 1 and dataset 2). The two networks then share their knowledge with each other using the second mutual learning algorithm and get trained further. For that, the labels are removed from the another small training dataset (dataset 3).

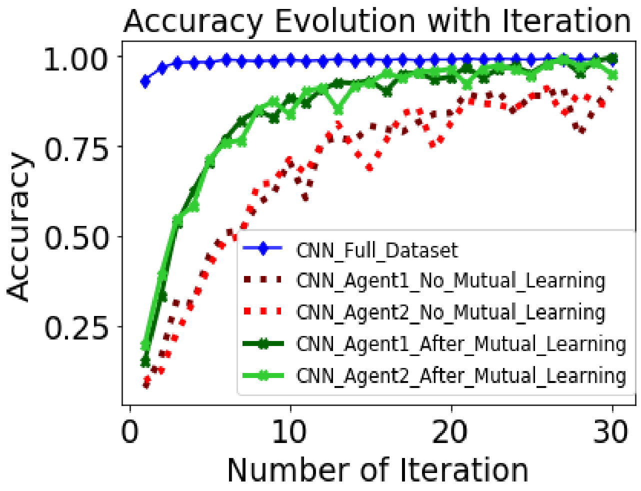


Fig. 6. CNN network accuracy plot with iteration for mutual learning. The top blue plot indicates the accuracy when the CNN is trained with the full training dataset. The light green and dark green plots indicates the CNN network accuracy with iteration after mututal learning. The light red and dark red lines are the accuracy plots before mutual learning.

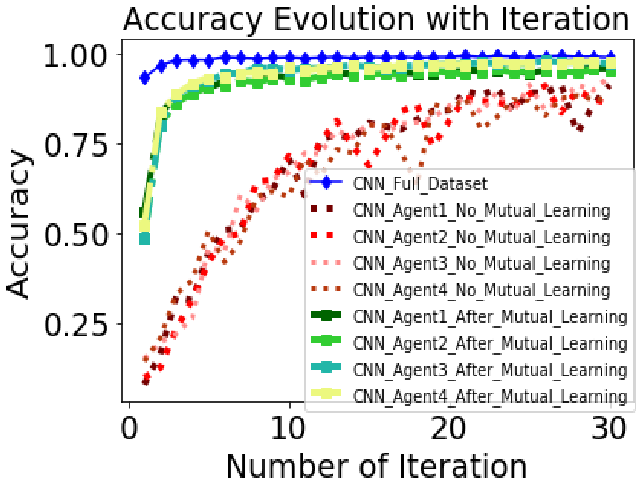


Fig 7. Accuracy plot with iteration for four CNN agents with mutual learning and without mutual learning

The new training data point labels are predicted using both the CNN agents along with their prediction confidence. Each data point in the dataset 3 set is relabeled according to more confident agent and both the agents are retrained after the labeling is finished. Fig. 6 shows the accuracy plots before and after the mutual learning. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy before mutual learning was 84.1% while the maximum accuracy increased to 93.6% after mutual learning. The accuracy is improvement is better than first algorithm. This is because the proximity calculation between figures can be more prone to error. Since, the accuracy of agents are already quite high, the confidence based relabeling is more accurate. But if the pre-mutual learning accuracy is low for the agents then algorithm I should work better than algorithm II. The most appropriate algorithm therefore depends on the pre-mutual learning accuracies of the agents and input types.

D. Mutual Learning with Four Teacher-Student Networks Using Algorithm I

Model distillation can be more efficient and the accuracy can be further improved if more agents share information

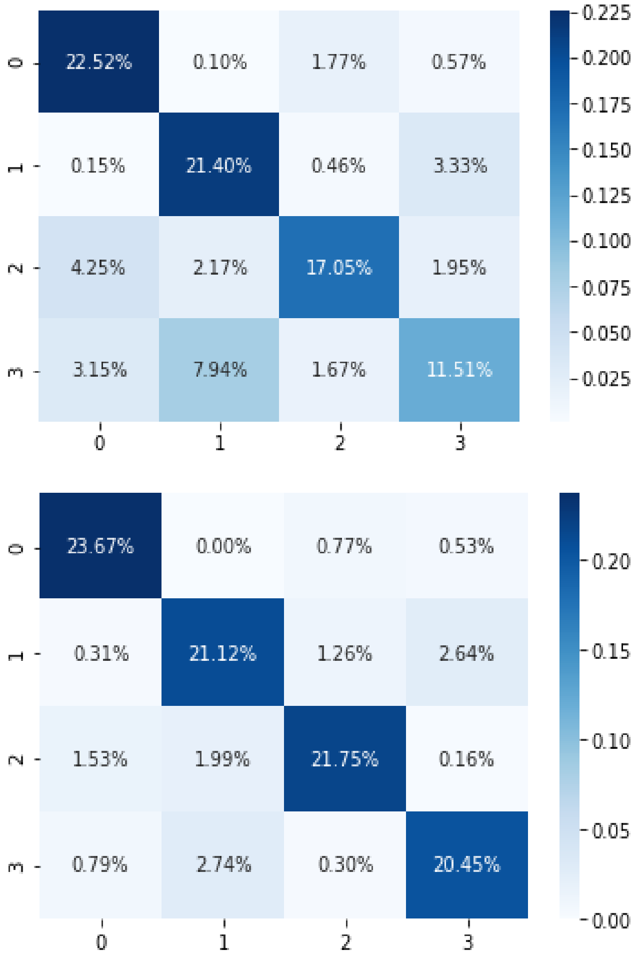


Fig. 8. Heatmap of confusion matrix for linear and non-linear SVN trained with full training dataset.

among themselves. To demonstrate the fact, we repeated the mutual learning algorithm with four networks. All the networks dynamically play the role of teacher and student. When one network plays the role of teacher, the three other networks play the role of student. Since four networks can share a lot more information with each other compared with two networks, all the networks become more well-trained and hence, the efficiency of all the networks increases simultaneously.

Fig. 7 shows the accuracy plots before and after machine learning for ten agents. The maximum accuracy achieved in this case is 94.78% compared to 90.45% for two network mutual learning. The green curve set represents the mutual learning with ten teacher-student networks in fig. 7.

E. SVM Network Testing Efficiency with Full Training Dataset and One-fourth Training Dataset

The SVM with linear and non-linear kernel is first trained with the full training dataset first and the accuracy is tested against the testing dataset. Training with the full dataset is a time and memory hungry procedure. The whole process took couple of hours in a machine with 1.6 GHz dual-core processor and 8GB 2,133 MHz RAM. The maximum accuracy for SVM with linear kernel is 76.59% while maximum accuracy for SVM with non-linear kernel is 85.9% after 30 iteration. The heatmap of the confusion matrix for linear and non-linear SVM is shown in fig 8.

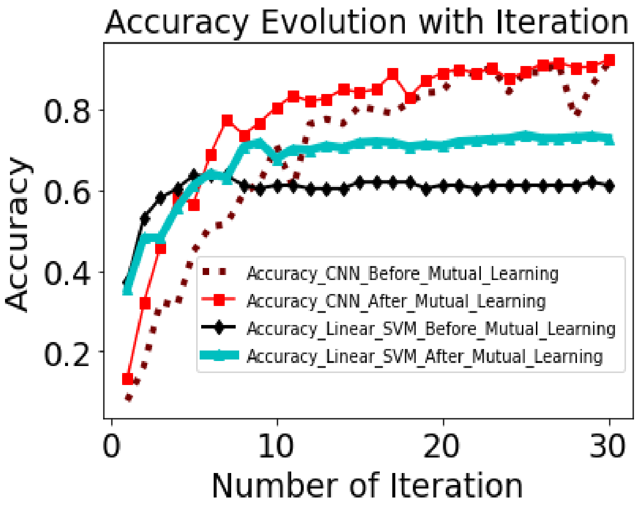


Fig. 9. Linear SVM anc CNN network accuracy plot with iteration for mutual learning. The brown and black plots indicates the CNN and linear SVM network accuracy cosecutively with iteration before mututal learning. The red and pastle lines are the accuracy plots after mutual learning for CNN and linear SVM network cosecutively.

Both SVM agents are then trained with small dataset and accuracy is tested every time. The maximum accuracy for linear SVM with small training set is 59.63%. while the accuracy is 44.1% for non-linear SVM.

F. Mutual Learning with Two SVM Networks Using Algorithm I

Homogeneous mutual learning is applied between two linear SVMs in the same ways as it was applied between two CNN agents. The linear SVM accuracy went up to 61.97% after mutual learning. Similarly, the accuracy went up to 55.74% after mutual learning between two non-linear SVMs.

G. Heterogeneous Mutual Learning with CNN and SVM Networks Using Algoritm II

Model distillation between homogeneous agents has been shown in literature. We have shown here that mutual learning is possible between heterogenous agents and the result is exciting. In figure 9, we have shown the accuracy plots before and after the mutual learning for linear SVM and CNN. The accuracy clearly got better after applying the mutual learning algorithm II. The maximum accuracy for linear SVM before mutual learning was 59.63% while the maximum accuracy increased to 74.04% after mutual learning. The maximum accuracy for CNN before mutual learning was 84.1% while the maximum accuracy increased to 85.76% after mutual learning. In this way, mutual learning can help both the agents get better accuracy.

One worth mentioning point is that we chose simple CNN and SVM algorithm because the basic purpose of the paper is not to show impressive accuracy of kidney disease diagnostic with machine learning. Rather our intention is to show that our proposed mutual learning algorithm works with different machine learning and neural network algorithm. Also most often we see that a particular algorithm is showing excellent accuracy for a particular dataset but might show poor accuracy for other dataset. A comprehensive way to avoid this fundamental issue is to keep training the algorithm with new datasets over the time. The mutual learning enables the machine learning algorithm to keep learning over time.

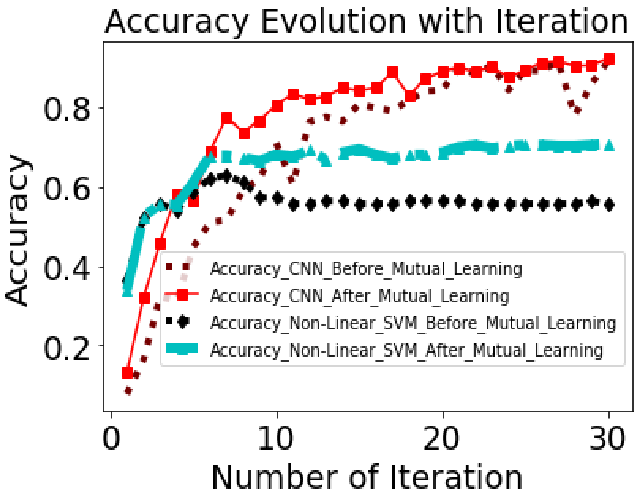


Fig. 10. Non-linear SVM anc CNN network accuracy plot with iteration for mutual learning. The brown and black plots indicates the CNN and non-linear SVM network accuracy cosecutively with iteration before mututal learning. The red and pastle lines are the accuracy plots after mutual learning for CNN and non-linear SVM network cosecutively.

In figure 10, we have shown the accuracy plots before and after the mutual learning for non-linear SVM and CNN. The accuracy clearly got better after applying the mutual learning algorithm. The maximum accuracy for non-linear SVM before mutual learning was 44.1% while the maximum accuracy increased to 72.21% after mutual learning. The maximum accuracy for CNN before mutual learning was 84.1% while the maximum accuracy increased to 84.79% after mutual learning.

An important observation is that the accuracy of SVM linear agent increased by 14.41% while accuracy of CNN only increased by 1.66%. Similarly, the accuracy of SVM linear agent increased by 28.11% while accuracy of CNN only increased by 0.69%. This is because CNN has already a much higher accuracy that SVM. Therefore, when they are engaged in mutual learning and trying to teach each other, the SVM learns a lot from CNN. But since the SVM accuracy was not high before mutual learning, it is not able to teach the CNN much and hence the CNN is less benefited from the mutual learning. Furthermore, the accuracy of CNN increases less in figure 9 vs in figure 10 because the non-linear SVM agent used in figure 10 has lower accuracy than the linear SVM agent used in figure 9. The machines here replicate our real life experience quite nicely.

H. Accuracy and Timing Comparison

The timing and accuracy comparison is shown below in table I. Clearly mutual learning gives a great advantage rather than training with big dataset because it significantly reduces the time and computational resource. It can give the flexibility to train multiple networks in parallel.

TABLE I. ACCURACY AND TIMING COMPARISON FOR CNN

	Single big training dataset	*Two small one-fourth Dataset*	*Mutual learning two agents Algorithm I*	*Mutual learning two agents Algorithm II*	*Mutual learning with four agents*
Maximum Accuracy	99.54%	84.1%	90.45%	93.6%	94.78%
Execution Time	~35 minute s	~6 minutes	~ 17 minutes	~ 15 minutes	~ 30 minutes

TABLE II. ACCURACY AND TIMING COMPARISON FOR SVM

	Single big training dataset	*Two randomly selected small one-fourth Dataset*	*Mutual learning with two SVM agents*	*Mutual learning with CNN and SVM agents*
Linear SVM Maximum Accuracy	76.59%	59.63%	61.97%	74.04%
Non-linear SVM Maximum Accuracy	85.9%	44.1%	55.74%	72.21%
Execution Time	~8 minutes	~2.5 minutes	~ 6 minutes	~ 6.5 minutes

VIII. CONCLUSION

This paper explores mutual learning in pattern classification, where two agents, P and Q, have separate sets of learning samples (patterns A and B) that require classification. The primary objective is to ensure that both agents classify all learning samples correctly, and the exchange of all samples is one possible solution. However, the paper seeks more efficient ways to determine misclassified samples between the two agents.

The paper concludes that detailed discussion between the two agents is necessary for successful classification, particularly regarding samples near the discriminant surfaces of the classifiers. When one agent misclassifies a learning sample of the other, the latter must continue its learning process until it correctly classifies the sample. Both agents then store different learning samples to accelerate the mutual learning process.

Furthermore, the paper presents a detailed description of a classification problem with simulation results that demonstrate the proposed mutual learning algorithm significantly enhances the participating agents' performance compared to isolated learning without mutual learning.

To summarize, the mutual learning algorithm has several benefits for machine learning systems. It not only improves the accuracy of all the networks involved, but it also enhances the speed of learning. This feature makes it a suitable option for practical systems with memory and computation resource constraints. Moreover, it enables many small networks to operate simultaneously, making it compatible with GPU-based systems. By increasing the number of networks, the accuracy of the system can be enhanced, providing the flexibility to adjust the system size as per the need. In short, the mutual learning algorithm can make machine learning faster, more flexible, and require fewer memory and computing resources.

ACKNOWLEDGMENT

The research reported here was supported by the National Science Foundation under grant numbers 1930601 (to Yale) and 1930606 (to IUPUI).

REFERENCES

[1] Singh, H., Meyer, A. N. & Thomas, E. J. "The frequency of diagnostic errors in outpatient care: estimations from three large observational studies involving us adult populations". *BMJ Qual. Saf.* **23**, 727–731 (2014).

[2] Singh, H., Schiff, G. D., Graber, M. L., Onakpoya, I. & Thompson, M. J. "The global burden of diagnostic errors in primary care", *BMJ Qual. Saf.* **26**, 484–494 (2017).

[3] Graber, M. L. "The incidence of diagnostic error in medicine", *BMJ Qual. Saf.* **22**, ii21–ii27 (2013).

[4] Singh, H., Giardina TD, Meyer AN, Forjuoh SN, Reis MD, Thomas EJ., "Types and origins of diagnostic errors in primary care settings". *JAMA Intern. Med.* **173**, 418–425 (2013).

[5] Liang, H., Tsui BY, Ni H, Valentim CCS, Baxter SL, Liu G, Cai W, Kermany DS, et al. "Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence". *Nat. Med.* **1**, 433–438 (2019).

[6] Topol, E. J. "High-performance medicine: the convergence of human and artificial intelligence". *Nat. Med.* **25**, 44 (2019).

[7] De Fauw J., Ledsam J.R., Romera-Paredes B., Nikolov S., Tomasev N., Blackwell S., Askham H., Glorot X., O'Donoghue B., Visentin D., van den Driessche G., Lakshminarayanan B., Meyer C., Mackinder F., Bouton S., Ayoub K., Chopra R., King D., Karthikesalingam A., Hughes C.O., Raine R., Hughes J., Sim D.A., Egan C., Tufail A., Montgomery H., Hassabis D., Rees G., Back T., Khaw P.T., Suleyman M., Cornebise J., Keane P.A., Ronneberger O.. "Clinically applicable deep learning for diagnosis and referral in retinal disease". Nat Med. 2018 Sep;24(9):1342-1350. *doi: 10.1038/s41591-018-0107-6.* Epub 2018 Aug 13. PMID: 30104768.

[8] Yu, K.-H., Beam, A. L. & Kohane, I. S. "Artificial intelligence in healthcare". *Nat. Biomed. Eng.* **2**, 719 (2018).

[9] Jiang F, Jiang Y, Zhi H, Dong Y, Li H, Ma S, Wang Y, Dong Q, Shen H, Wang Y. "Artificial intelligence in healthcare: past, present and future." Stroke Vasc Neurol. 2017 Jun 21;2(4):230-243. *doi: 10.1136/svn-2017-000101.* PMID: 29507784; PMCID: PMC5829945.

[10] Esteva A, Robicquet A, Ramsundar B, Kuleshov V, DePristo M, Chou K, Cui C, Corrado G, Thrun S, Dean J. "A guide to deep learning in healthcare". Nat Med. 2019 Jan;25(1):24-29. *doi: 10.1038/s41591-018-0316-z.* Epub 2019 Jan 7. PMID: 30617335.

[11] Semigran, H. L., Levine, D. M., Nundy, S. & Mehrotra, A. "Comparison of physician and computer diagnostic accuracy". JAMA Intern. Med. 176, 1860–1861 (2016).

[12] Miller, R. "A history of the internist-1 and quick medical reference (qmr) computer-assisted diagnosis projects, with lessons learned". Yearb. Med. Inform. 19, 121–136 (2010).

[13] Razzaki, S., Baker, A., Perov, Y., Middleton, K., Baxter, J., Mullarkey, D., Sangar, D., Taliercio, M., Butt, M., Azeem Majeed, DoRosario, A., Mahoney, M., Saurabh, J., "A comparative study of artificial intelligence and human doctors for the purpose of triage and diagnosis". Preprint at *https://arxiv.org/abs/1806.10698* (2018).

[14] Vembandasamy, K., Sasipriya, R. and Deepa, E. (2015), "Heart Diseases Detection Using Naive Bayes Algorithm", IJISET-International Journal of Innovative Science, Engineering & Technology, 2, 441-444.

[15] Chaurasia, V. and Pal, S. (2013) "Data Mining Approach to Detect Heart Disease", International Journal of Advanced Computer Science and Information Technology (IJACSIT), 2, 56-66.

[16] Parthiban, G. and Srivatsa, S.K. (2012) "Applying Machine Learning Methods in Diagnosing Heart Disease for Diabetic Patients". International Journal of Applied Information Systems (IJAIS), 3, 25-30.

[17] Tan, K.C., Teoh, E.J., Yu, Q. and Goh, K.C. (2009) "A Hybrid Evolutionary Algorithm for Attribute Selection in Data Mining. Journal of Expert System with Applications", 36, 8616-8630. *https://doi.org/10.1016/j.eswa.2008.10.013*

[18] Iyer, A., Jeyalatha, S. and Sumbaly, R. (2015) "Diagnosis of Diabetes Using Classification Mining Techniques". International Journal of Data Mining & Knowledge Management Process (IJDKP), 5, 1-14. *https://doi.org/10.5121/ijdkp.2015.5101*

[19] Sen, S.K. and Dash, S. (2014) "Application of Meta Learning Algorithms for the Prediction of Diabetes Disease". International Journal of Advance Research in Computer Science and Management Studies, 2, 396-401.

[20] Kumari, V.A. and Chitra, R. (2013) "Classification of Diabetes Disease Using Support Vector Machine". International Journal of Engineering Research and Applications (IJERA), 3, 1797-1801.

[21] Sarwar, A. and Sharma, V. (2012) "Intelligent Naïve Bayes Approach to Diagnose Diabetes Type-2". Special Issue of International Journal of Computer Applications (0975-8887) on Issues and Challenges in Networking, Intelligence and Computing Technologies-ICNICT 2012, 3, 14-16.

[22] Ephzibah, E.P. (2011) "Cost Effective Approach on Feature Selection using Genetic Algorithms and Fuzzy Logic for Diabetes Diagnosis". International Journal on Soft Computing (IJSC), 2, 1-10. *https://doi.org/10.5121/ijsc.2011.2101*

[23] Vijayarani, S. and Dhayanand, S. (2015) "Liver Disease Prediction using SVM and Naïve Bayes Algorithms". International Journal of Science, Engineering and Technology Research (IJSETR), 4, 816-820.

[24] Gulia, A., Vohra, R. and Rani, P. (2014) "Liver Patient Classification Using Intelligent Techniques". (IJCSIT) International Journal of Computer Science and Information Technologies, 5, 5110-5115.

[25] Rajeswari, P. and Reena,G.S. (2010) "Analysis of Liver Disorder Using Data Mining Algorithm". Global Journal of Computer Science and Technology, 10, 48-52

[26] Ba-Alwi, F.M. and Hintaya, H.M. (2013) "Comparative Study for Analysis the Prognostic in Hepatitis Data: Data Mining Approach". International Journal of Scientific & Engineering Research, 4, 680-685.

[27] Karlik, B. (2011) "Hepatitis Disease Diagnosis Using Back Propagation and the Naive Bayes Classifiers". Journal of Science and Technology, 1, 49-62.

[28] Sathyadevi, G. (2011) "Application of CART Algorithm in Hepatitis Disease Diagnosis". IEEE International Conference on Recent Trends in Information Technology (ICRTIT), MIT, Anna University, Chennai, 3-5 June 2011, 1283-1287.

[29] Park C, Awadalla A, Kohno T, Patel S. "Reliable and trustworthy machine learning for health using dataset shift detection". In: Proceedings of the conference on NeurIPS, 2021, pp.1

[30] Hasani N, Morris MA, Rhamim A, Summers RM, Jones E, Siegel E, Saboury B. Trustworthy "Artificial Intelligence in Medical Imaging". PET Clin. 2022 Jan;17(1):1-12. *doi: 10.1016/j.cpet.2021.09.007.* PMID: 34809860; PMCID: PMC8785402.

[31] Hinton, Geoffrey E., Vinyals, O., Dean, J.,. "Distilling the Knowledge in a Neural Network." *ArXiv abs/1503.02531* (2015): n. pag.

[32] K. S. Narendra and S. Mukhopadhyay, "Mutual Learning: Part I - Learning Automata," 2019 American Control Conference (ACC), 2019, pp. 916-921, *doi: 10.23919/ACC.2019.8814751.*

[33] K. S. Narendra and S. Mukhopadhyay, "Mutual Learning: Part II -- Reinforcement Learning," 2020 American Control Conference (ACC), 2020, pp. 1105-1110*, doi: 10.23919/ACC45564.2020.9147838*

[34] Jimmy Ba and Rich Caruana. "Do deep nets really need to be deep?," In Advances in Neural Information Processing Systems. 2014.

[35] Adriana, R., Nicolas, B., Ebrahimi, K. S., Antoine, C., Carlo, G., & Yoshua, B. (2015). "Fitnets: Hints for thin deep nets". *Proc. ICLR*, *2*, 3.

[36] David Lopez-Paz, Ankit Singh Rawat Sashank J. Reddi Seungyeon Kim Sanjiv Kumar, "Unifying distillation and privileged information," International Conference on Learning Representations, 2016.

[37] Ying Zhang, Xiatian Zhu, Mao Ye,. 2018. "Deep mutual learning". In Conference on Computer Vision and Pattern Recognition, (CVPR), pages 4320–4328.

[38] Islam MN, Hasan M, Hossain M, Alam M, Rabiul G, Uddin MZ, Soylu A. "Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography". Scientific Reports. 2022 Jul 6;12(1):1-4.

[39] S. Ikemoto, H. B. Amor, T. Minato, B. Jung and H. Ishiguro, 2012. "Physical human-robot interaction: Mutual learning and adaptation". IEEE robotics & automation magazine, 19(4), pp.24-35.

[40] Nie, X., Feng, J. and Yan, S., 2018. "Mutual Learning to Adapt for Joint Human Parsing and Pose Estimation". In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 502-517).

[41] Panait, L. and Luke, S., 2005. "Cooperative multi-agent learning: The state of the art". *Autonomous agents and multi-agent systems*, *11*(3), pp.387-434.

[42] Bai, Q., Su, C., Tang, W, Li Y.. "Machine learning to predict end stage kidney disease in chronic kidney disease". *Sci Rep* **12**, 8377 (2022). *https://doi.org/10.1038/s41598-022-12316-z*

[43] Dashtban A, Mizani MA, Pasea L, Denaxas S, Corbett R, Mamza JB, Gao H, Morris T, Hemingway H, Banerjee A. “Identifying subtypes of chronic kidney disease with machine learning: development, internal validation and prognostic validation using linked electronic health records in 350,067 individuals”. EBioMedicine. 2023 Mar;89:104489. *doi: 10.1016/j.ebiom.2023.104489*. Epub 2023 Feb 27. PMID: 36857859; PMCID: PMC9989643..

[44] U. Ekanayake and D. Herath, "Chronic Kidney Disease Prediction Using Machine Learning Methods," *2020 Moratuwa Engineering Research Conference (MERCon)*, Moratuwa, Sri Lanka, 2020, pp. 260-265, *doi: 10.1109/MERCon50084.2020.9185249*.

[45] Bhandari M, Yogarajah P, Kavitha MS, Condell J. “Exploring the Capabilities of a Lightweight CNN Model in Accurately Identifying Renal Abnormalities: Cysts, Stones, and Tumors, Using LIME and SHAP”. *Applied Sciences*. 2023; 13(5):3125. *https://doi.org/10.3390/app13053125*

[46] Islam, M.N., Hasan, M., Hossain, M.K. *et al.* “Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from CT-radiography”. *Sci Rep* **12**, 11440 (2022). *https://doi.org/10.1038/s41598-022-15634-4*

[47] Badawy M, Abdulqader M. Almars, Hossam Magdy Balaha, Mohamed Shehata, Mohammed Qaraad, Mostafa Elhosseini, “A two-stage renal disease classification based on transfer learning with hyperparameters optimization”. Front Med (Lausanne). 2023 Apr 5;10:1106717. *doi: 10.3389/fmed.2023.1106717*. PMID: 37089598; PMCID: PMC10113505

DOI: 10.15439/2023F4672
ISSN 2300-5963 ACSIS, Vol. 35

A Graph Matching Algorithm to extend Wise Systems with Semantic

Abdelhafid Dahhani, Ilham Alloui
0000-0001-6314-662X
0000-0002-3713-0592
Université de Savoir Mont Blanc,
LISTIC laboratoty,
Polytech Annecy-Chambery,
5 Chem. de Bellevue, 74940 Annecy, France
Email: {abdelhafid.dahhani, ilham.alloui}@univ-smb.fr

Sébastien Monnet, Flavien Vernier
0000-0002-6036-3060
0000-0001-7684-6502
Université de Savoir Mont Blanc,
LISTIC laboratoty,
Polytech Annecy-Chambery,
5 Chem. de Bellevue, 74940 Annecy, France
Email: {sebastien.monnet, flavien.vernier}@univ-smb.fr

Abstract—**Software technology has exponentially evolved leading to the development of intelligent applications using artificial intelligence models and techniques. Such development impacts all scientific and social fields: home automation, medicine, communication, etc. To make those new applications useful to a larger number of people, researchers are working on how to integrate artificial intelligence into real world while respecting the notion of calm technology. This paper fits in the context of the development of intelligent systems termed "wise systems" that aim at satisfying the calm technology requirement. Those systems are based on the concept of "Wise Object": a software entity – object, service, component, application, etc. – able to learn by itself how it is expected to behave and how it is used by a human or another software entity. During its learning process, a Wise Object constructs a graph that represents its behavior and the way it is used. A major weakness of Wise Objects is that the numerical information that they generate is mostly meaningless to humans. Therefore the objective of the work presented in this paper is to extend Wise Objects with semantic that enable them communicate with humans whose attention will consequently be less involved. In this paper, we address the issue of how to relate two different views using two state-based formalisms: State Transition Graph for views generated by the Wise Objects and Input Output Symbolic Transition System for conceptual views. Our proposal extends previous work done to extend the generated information with the conceptual knowledge using a matching algorithm founded on graph morphism. The first version of the algorithm has several limitations and constraints on the graphs that make it difficult to use in realistic cases. In this paper, we propose to generalize the algorithm and raise those restrictions. To illustrate the complete process, the construction of a sample graph matching on a home-automation system is considered.**

I. Introduction

To ENABLE usability and accessibility of intelligent systems to a large number of people, researchers are working on how to integrate artificial intelligence (AI) into real world while respecting the notion of calm technology. Calm technology represented by Mark Weiser and John Seely Brown [1] in 1995, intends to lightly involve humans within the work process by requiring the smallest possible amount of their attention [2], minimizing therefore system intrusion into their life. Furthermore as software systems usage varies depending on users and time, they should be able to autonomously adapt to evolving such usages. To meet those requirements, we realised a software framework to develop intelligent systems termed "Wise Systems" (WS), based on the concept of "Wise Object" (WO) [3].

A WO is a software entity – object, service, component, application – able to learn by itself how it is expected to behave and how it is used by a human or another software entity. This is enabled by introspection and reflection mechanisms (more details in [4]). WOs compose a WS that may be considered as a multi-agent system [5] [6] where a WO is a self-learning agent that autonomously monitors its internal changes and that does not know the other WOs in the system. Thus, a WO informs the WS about its state changes, so that other WOs react accordingly.

Based on the behaviour of the WS and the internal monitoring of each WO, the collected monitoring data can feed a learning process able to determine usual and unusual behaviors (for instance). As the development of WSs is non-trivial, we developed an object based framework, known as the Wise Object Framework (WOF [4]) to help developers design, deploy and evolve WSs. Generally, knowledge in AI-enabled systems can be provided according to two approaches: (*i*) describing a priori the arrangement of activities to be performed by the system, or, (*ii*) letting the system acquire by its own the required knowledge using different learning mechanisms:

- *In the former approach*, ontologies or scenarios are usually used to describe the arrangement of activities to achieve a goal as in [7] and [8]. In [7], functional behavior as well as inter-operation of system entities are described a priori using state-diagrams. In [8], the authors go a step forward combing ontologies to design ambiant assisted living systems with an unsupervised learning before system deployment to create relevant scenarios. In those approaches, the end user is at the heart of the scenario creation process as described in [9] and [10].
- *In the second approach*, knowledge is provided by the AI-enabled system in representations and views not necessarily understandable by humansand to the distance

between the business domain and technological domain views [11].

According to the calm technology [12], the WO embedded AI fits in the second approach. By self-monitoring, a WO acquires raw knowledge (logs) about its behavior. Based on IBM's 4 state loop (MAPE-K) [13] [14] architecture, WOF provides a plugin system to add analyzers that produce new knowledge from existing knowledge (i.e., logs or knowledge produced by the analyzers). Different analyzers already exist like statistical, Markovian, etc. In this paper, we focus on the State Transition Graph (STG) analyzer, it analyzes logs and produces an STG. This graph is useful for a WO to know in which state it stands and to determine the consequences on its state when it performs an action (a method invocation in the object oriented paradigm of WOF). This graph can also be used by the WO to determine the sequence of methods to call to reach a specific state. As previously stated, and because it is built in a completely unsupervised way, this graph does not carry useful semantic, the states only carry numbers, that is a problem to communicate with end-users.

Although initiated in the 18th century with Euler's work on the famous problem of Königsberg bridges [15], graph theory remains a powerful tool for software-intensive system development. Graphs are present at the software design stage, such as Input/Output Labelled Transitions Systems (IOLTS) or Input/Output Symbolic Transitions Systems (IOSTS) which are often used to model the system behavior, especially to build systems based on oracle, controller synthesis [16] [17] or to test a system by executing the various possible behaviors of this system [18], as well as during the data collection phase, to connect data from different sources [19]. As these graphs are conceptual, dedicated to human to human communication, they bring semantic. Since the STG built at runtime by a WO is close to IOLTS, a WO can use knowledge from its design stage, the IOSTS for example, to attach semantic to its STG. These graphs are obtained by combining two research methods (quantitative and qualitative). On the machine side, behavioral data (i.e., logs) is autonomously collected by WOs which dynamically analyse it and build an STG. On the human side, a developer/expert expresses the expected behavior of a system in a conceptual model using a human vocabulary. A qualitative behavioral view is provided using an IOSTS.

Our proposal in this paper is to present a new matching algorithm between both types of graphs. As the problem was simplified in our previous work [20], it has many limitations, the strongest being the number of equivalent attributes/variables in STG/IOSTS, another limitation is the constraint on the existence of only one matching between an STG and an IOSTS. This new algorithm will extend more further the generated knowledge with conceptual knowledge, based on the graph morphism [21] [22]. Thus, many variables from both knowledge will be taken into account this time, resulting in many different matching. This provides the ability to make WSs' generated knowledge understandable by human and to enable human evaluation of WSs' outputs. Explicitly, the contribution presented in this paper attempts to relate both views: (a) a conceptual view relying on knowledge given by developers to either describe or control the system behavior, and, (b) behavior-related knowledge acquired during WS's learning process, this process is illustrated through a concrete example. Consequently, we are in the process of establishing a machine-human communication (MHC). In this way, we use two state-based formalisms:

- *STG* for representing behavior-related knowledge generated by the WSs,
- *IOSTS* for modelling conceptual views of developers/experts.

This paper is organised as follows: Section II presents the basic idea, describes the architectural overview and gives the definition of important terms. Section III presents STG and IOSTS formalisms, and illustrates them through examples. Finally, Section IV presents our graph matching algorithm, before Section VI concludes the paper.

II. Basic idea & architectural view

The basic idea underlying the WO concept is to give a software entity the core mechanisms for learning behaviors through introspection and analysis. Our aim is to go further by enabling software to execute MAPE-K loops [4]. On the top of this concept, we built the WOF [3] with design decisions mainly guided by reusability and genericity requirements: the framework should be maintainable and used in different application domains with different strategies (e.g., analysis approaches).

Seeking clarity, we borrowed some terms used for humans to refer to abilities a WO possesses. Awareness and wisdom both rely on knowledge. Inspired by [23], we give some definitions of those terms commonly used for humans and present those we chose for WOs.

Knowledge: refers to information, inference rules and information deduced from them, for instance: "Turning on a heater will cause temperature change".

Awareness: represents the ability to collect - ability to provide internal data - on itself by itself. For instance, it is when an entity/object/device collects information and data about capabilities *(what is intended to do)* and its use *(what it is asked to do)*. Capabilities are the services and functionalities the WO may render.

Wisdom: is the ability to analyse collected information and stored knowledge related to their capabilities and usage to output useful information for end users. It is worth noticing that a WO is highly aware, while the converse is false. Wisdom implies awareness, but awareness does not imply wisdom.

Semantic: is the meaning given to something so that it can be understood by humans as mentioned in the Cambridge dictionary. This definition also applies to objects/devices, as semantic is used to communicate with humans.

From the conceptual view, according to the target application, a WO may be considered as:

- a stand-alone software entity (object, component, etc.),
- a software avatar designed to be a proxy for a physical device (e.g., a heater, vacuum cleaner, light bulb),

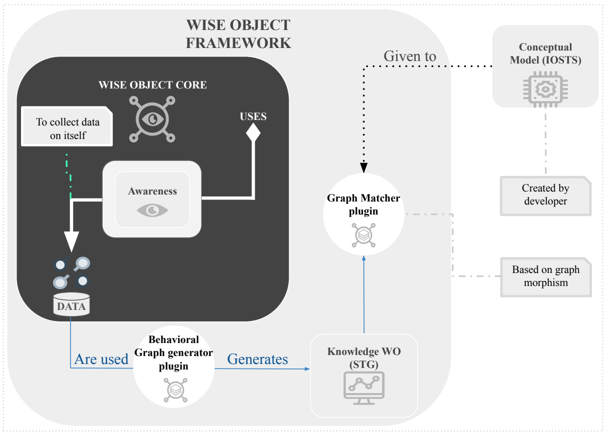


Fig. 1: Generic functional architecture of a WO in the WOF [20].

- a software avatar designed to be a proxy for an existing software entity (object, component, etc.).

A WO is characterised by its:

- autonomy: it is able to operate with no human intervention,
- adaptability: it changes its behavior when its environment evolves,
- ability to communicate (send its state changes and receive requests): with its environment according to a publish-subscribe paradigm.

Fig. 1 illustrates a partial view of a WO's functional architecture defined in the WOF. As depicted, the WO uses awareness to collect data on itself. It analyses those data thanks to the behavioral graph generator plugin (Also termed analyzer) and generates a behavioral graph represented by an STG (Section III-A). On the other hand, when designing an application, developers can provide a conceptual model describing/specifying the way they view the behavior of the system's entities associated to WOs. Such models are represented using IOSTS and contain the semantic given by developers to WOs (Section III-B). The IOSTS formalism is mostly known in simplifying system modelling by allowing symbolic representation of parameters and variable values instead of concrete data values enumeration [24].

The semantic carried by the IOSTS will be used by the graph matcher plugin to extend the STG using the algorithm proposed in this paper.

III. Complex Behavioral Models, Definitions and Illustrations

Modeling the behavior of a system is enabled by tools and languages that result in informal, semi-formal or formal representations: semi-formal notations like UML or more abstract behavior representations based on proven theories [25] like graph theory. In our case, STGs and IOSTSs respectively generated by WOs and provided by developers are used.

A. Definition of an STG

An STG is a directed graph where vertices represent the different states of an object and transitions represent the execution of its methods. Let us consider an object defined by its set of attributes A and its set of methods M. According to this information (A and M) on the object, the STG definition is given in Definition 1.

***Definition** 1:*

An STG is defined by the triplet $G(V, E, L)$ where:

- V is the set of vertices, with $|V| = n$ where each vertex represents a unique state of the object, and conversely, each state of the object is represented by a unique vertex. Therefore $v_i = v_j \Leftrightarrow i = j$ with $v_i, v_j \in V$ and $i, j \in [0, n[$.
- E is the set of directed edges where $\forall e \in E$, e is defined by the triplet $e = (v_i, v_j, m_k)$, such that $v_i, v_j \in V$ and $m_k \in M$. This triplet is called a transition labeled by m_k. The invocation of method m_k from state v_i switches the object to state v_j.
- L is a set of vertex labels where any label $l_i \in L$ is associated to v_i. A label l_i is the set of pairs $(att_j, value_{i,j})$ $\forall att_j \in A$, with $value_{i,j}$ the value of att_j in the state v_i and $Dom(att_j)$ the value domain of att_j, i.e., the set of $value_{i,j}$ for all i. By definition, 2 states v_i and v_j are different $v_i \neq v_j$, $iff \quad \exists att_k \in A$, such that $value_{i,k} \neq value_{j,k}$. Conversely, if $\forall k \in [0, |A|[\quad value_{i,k} = value_{j,k}$, the states v_i and v_j are considered the same, i.e., $v_i = v_j$, thus $i = j$.

The STG must comply with certain constraints:

- The number of attributes is finite:

$$|A| \in \mathbb{N}^*.$$

- The domain $Dom(att_i)$ of each attribute $att_i \in A$ is bounded and discrete:

$$\begin{gathered} \forall att \in A, \min(Dom(att)) \neq -\infty, \\ \max(Dom(att)) \neq \infty, \\ |Dom(att)| \in \mathbb{N}^*. \end{gathered}$$

The matching algorithm we propose in Section IV takes as input an STG with a specific property we name exhaustiveness. Definition of "exhaustive STG" is given in Definition 2

***Definition** 2:* An exhaustive STG is an STG such that from each vertex v_i there exist $|M|$ transitions, each labeled by a method m_k in M:

$$\begin{gathered} \forall v_i \in V, \forall m_k \in M, \\ \exists v_j \in V | (v_i, v_j, m_k) \in E. \end{gathered} \tag{1}$$

It is worth noting that v_i and v_j may be different or same states ($v_i \neq v_j$ or $v_i = v_j$).

According to Definition 2, an exhaustive STG is deterministic, i.e., from any state, on any method invocation, the destination state is known. Moreover, the number of transitions $|E|$ in an exhaustive STG depends on the numbers of vertices $|V|$ and methods $|M|$ such that:

$$|V| \times |M| = |E|. \tag{2}$$

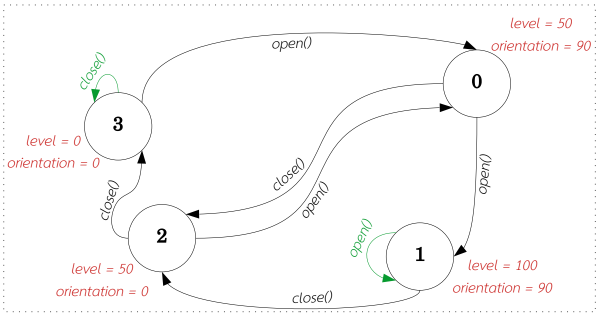


Fig. 2: An exhaustive STG presentation of a roller shutter.

As each vertex represents a state and a unique set of attribute values, an attribute is defined according a discrete and bounded domain and the set of attributes is naturally finite, the number of vertices $|V|$ is bounded:

$$\prod_{i=1}^{|A|} |Dom(att_i)| \geq |V|. \tag{3}$$

The inequality is due to the fact that some attribute value combinations may not be compatible. In other words the number of possible states is $\prod_{i=1}^{|A|} |Dom(att_i)|$, but all are not necessary reachable.

Fig. 2 illustrates an exhaustive STG of the behavior of a simplified roller shutter with adjustable slats. It is defined by the attributes "level" and "orientation" ($A = \{level, orientation\}$) and 2 methods "open" and "close" ($M = \{open(), close()\}$). The methods "open" and "close" respectively increase and decrease the level by 50, the slats orientation is adjusted automatically to have values 0 or 90. This STG was automatically generated and State 0 corresponds to the first discovered state, State 1 the second, etc., consequently, except the methods that give semantic to the transitions, the states have no semantic. According to the domains of the level and orientation, and Property 3, the STG has 4 reachable states, and according to Property 2 the number of edges is 8. Point out that State 0 has nothing to do with the initial values of the roller shutter, they correspond to the first state found during the automatic generation of the STG.

Let us note that a WO, never stores the whole STG due to an evident combinatorial explosion. Only a useful sub-graph is stored and mechanisms for forgetting sub-parts that are no longer useful are implemented. These memory problems are out of the scope of this paper, and we consider here only exhaustive STG to highlight the algorithm.

B. Definition of an IOSTS

An IOSTS is a directed graph whose vertices, called localities represent different states of the system (in our case, the system is a software object) and whose edges are transitions. The localities are connected by transitions triggered by actions. An IOSTS allows the definition of an infinite state transition system in a finite way, unlike an STG. In the literature, IOSTS are used to verify, test and control systems. Verification and testing are formal techniques for validating and comparing two views of a system while control is used to constrain the system behavior [16] [26]. The definition of IOSTS given in Definition 3, is taken from [27] [16] and especially from the use case given in [24].

Definition 3: An IOSTS is a sixfold $\langle D, \Theta, Q, q_0, \Sigma, T\rangle$ such as:

- D is a finite set of typed data consisting of two disjoint sets of: variables X and action parameters P. Let an element $d \in D$, $Dom(d)$ determines the value domain of d.
- Θ an initial condition expressed as a predicate on variables X.
- Q is a non-empty finite set of localities with $q_0 \in Q$ the initial locality. A locality q is a set of states such that statesOf(q) $\subseteq Dom(X)$, with $Dom(X)$ the cartesian product of the domains $Dom(x)$ of each $x \in X$:
$$Dom(X) = \prod_{x \in X} Dom(x). \tag{4}$$
Let us note that a state is defined by a unique tuple of values for the whole variables.
- Σ is the alphabet, a finite, non-empty set of actions. It consists of the disjoint union of the set $\Sigma^?$ of input actions, the set $\Sigma^!$ of output actions, and the set $\Sigma^\mathcal{T}$ of internal actions. For each action a in Σ, its signature $sig(a) = \langle p_1, \dots, p_k\rangle | p_i \in P$ is a tuple of parameters. The signature of internal actions is always an empty tuple.
- T is a finite set of transitions, such that each transition is a tuple $t = \langle q_o, a, G, A, q_d\rangle$ defined by:
  - a locality $q_o \in Q$, called the origin of the transition,
  - an action $a \in \Sigma$, called the action of the transition,
  - a boolean expression G on $X \cup sig(a)$ related to the variables and the parameters of the action, called the transition guard, transition guards allows us to distinguish transitions that have same origin and action but disjoint conditions to their triggering,
  - an assignment of the set of variables, of the form $(x := A^x)_{x \in X}$ such that for each $x \in X$, A^x is an expression on $X \cup sig(a)$, it defines the change of variable values during the transition,
  - a locality q_d, called the transition destination.

According to this definition, each variable has a subdomain in each locality. Thus, let us define the function $dom(q, x)$ that returns the definition domain of the variable $x \in X$ in the locality $q \in Q$, consequently $dom(q, x) \subseteq Dom(x)$. By extension, $dom(q, X)$ is the cartesian product of domains of all x in q:

$$\begin{gathered} dom(q, X) = \textstyle\prod_{x \in X} dom(q, x), \\ dom(q, X) \subseteq Dom(X). \end{gathered} \tag{5}$$

Fig. 3 illustrates the IOSTS given by a developer to control a roller shutter with adjustable slats. This IOSTS expresses that the roller shutter expects an input $up?/down? \in \Sigma^?$ carrying the parameter $step \in]0, 100]$, the relative elevation to increase or decrease the shutter level. In addition, the roller shutter slats can be rotated between 0 and 90 degrees during the elevation.

Let us mention that the roller shutter used in this example adapts automatically its $angle$, while the elevation is between 0 and 100 steps. Thus, end-users only control the elevation (height). The IOSTS contains two localities:

- The locality where the system is closed (i.e., both variables are equal to 0, see Equation 6). In this locality, if the system receives the $up?(step)$ command, the transition will be made from the $Closed$ to $Open$ locality by increasing the value of the $height$ variable by $step$. If the system receives the $down?(step)$ action, it will perform no operation (NOP).
- The locality where the system is open (i.e., one of the variables is different from 0, see Equation 7). In this locality, if the system receives the action $up?(step)$, the transition will be reflexive from $Open$ to itself and will compute the value of the variable $height$ by executing this assignment $height := min(height + step, 100)$, the shutter elevation cannot be increased more than the maximum of elevation. If it receives the $down?(step)$ action and the action closes the shutter less than it is open ($step < height$), $height$ is decreased by $step$, otherwise the transition will be from the locality $Open$ to the locality $Closed$ by assigning 0 to the variable $height$.

As illustrated in Fig. 3, and according to the Definition 3, the IOSTS is composed of:

- $Q = \{Closed, Open\}$, the set of localities.
- $X = \{height, angle\}$, the set of variables.
- $P = \{step\}$, the set of parameters.
- $\Sigma = \{up?, down?\}$, the set of actions where the signatures of the actions are $sig(up?) = sig(down?) = \langle step\rangle$.
- $Dom(height) = [0, 100]$, $Dom(angle) = [0, 90]$ and $Dom(step) =]0, 100]$ are respectively the domains of $height$, $angle$ and $step$.
- According to Equation 4, $Dom(X)$ is:

$$Dom(X) = [0, 100] \times [0, 90].$$

- The states of closed locality, defined by Equation 5, are:

$$\begin{gathered} statesOf(Closed) = \{(0, 0)\}, \\ \{(0, 0)\} \subset Dom(X). \end{gathered} \tag{6}$$

- The states of open locality are defined as follows:

$$\begin{gathered} statesOf(Open) = Dom(X) \backslash statesOf(Closed), \\ Dom(X) \backslash statesOf(Closed) \subset Dom(X). \end{gathered} \tag{7}$$

Thus:

$$statesOf(Closed) \cap statesOf(Open) = \emptyset. \tag{8}$$

As IOSTS is classical reference modeling formalism for model-based testing of reactive systems [28], it provide a convenient abstraction of the behaviors of such systems, which are beneficial and playing an important role in the matching algorithm.

IV. Graph matching algorithm

This section introduces our algorithm that relates the generated STG to developers' semantic expressed in an IOSTS formalism. The generated STG in Fig. 2 is composed of states automatically labelled by the WO: 0, 1, 2 and 3 and the localities of the IOSTS are labelled "Open" and "Closed". As both represent the same roller shutter, the main challenge is how to match states 0, 1, 2 and 3 to the localities of the IOSTS, in other words, which attribute of the STG corresponds to which variable of the IOSTS.

A. Matching constraints

The STG and IOSTS must meet certain criteria to correctly apply the matching algorithm.

1) Considering that we have in the set of attributes A a non empty subset called $A_e \neq \emptyset$ and in the set of variables X a non empty subset called $X_e \neq \emptyset$. Furthermore, let us consider $\mathcal{R}$ the binary relation of A_e in X_e, which is a bijection ($\rightarrowtail\!\!\!\!\twoheadrightarrow$) [29]:
   - each member of A_e must be linked exactly to one element of X_e,
   - each element of X_e must be linked exactly to one member of A_e.

   $$\begin{gathered} \exists! A_e \subseteq A, \exists! X_e \subseteq X | A_e \rightarrowtail\!\!\!\!\twoheadrightarrow X_e \\ \Leftrightarrow \\ (A_e, X_e), \end{gathered} \tag{9}$$

   where (A_e, X_e) means that both A_e and X_e represent the same information. The matching solution is given by $(A_e, X_e, \mathcal{R})$. Therefore, any couple att_e, x_e such that $att_e \mathcal{R} x_e$ will be featured by (att_e, x_e) as they represent the same information, thus:

   $$Dom(att_e) \subseteq Dom(x_e). \tag{10}$$

   Let us note that $Dom(att_e)$ is a subset of $Dom(x_e)$ due to the fact that $Dom(x_e)$ can be defined as a continuous domain and $Dom(att_e)$ is defined as a discrete and bounded set of values. Furthermore, from Equation 10 we deduce the following:
   - In case of $Dom(att_e) \subset \mathbb{R}$:

   $$\begin{gathered} (att_e, x_e) \\ \Leftrightarrow \\ min(Dom(att_e)) \geq min(Dom(x_e)) \\ \wedge \\ max(Dom(att_e)) \leq max(Dom(x_e)). \end{gathered}$$

   - In case of $Dom(att_e) \subset \mathbb{S} \mid \mathbb{S}$ is the set of strings:

   $$\begin{gathered} (att_e, x_e) \\ \Leftrightarrow \\ \forall value_i \in Dom(att_e), value_i \in Dom(x_e). \end{gathered}$$

2) Every locality in the IOSTS must be unique taking into account just the set variables X_e. Thus, the domains of X_e in the different localities in the IOSTS are disjoint:

   $$\begin{gathered} \forall q, q' \in Q \mid q \neq q' \\ \Leftrightarrow \\ dom(q, X_e) \cap dom(q', X_e) = \emptyset. \end{gathered}$$

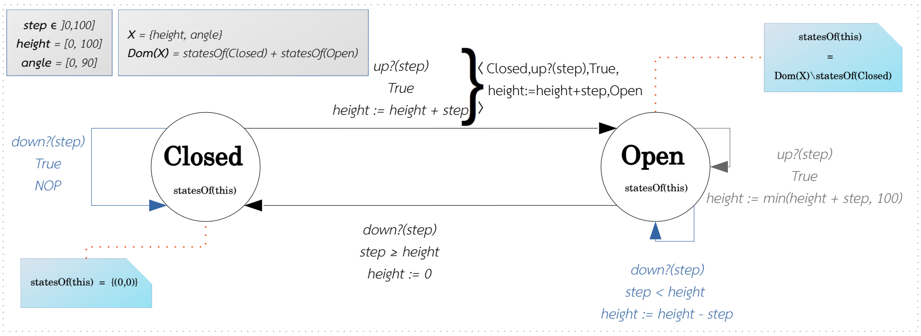


Fig. 3: An IOSTS representation of a roller shutter.

B. Algorithm

The matching algorithm will automatically run through several steps summarized in Fig. 4. The algorithm will receive two types of knowledge representation (STG and IOSTS). It is divided into three parts. The former produces all possible attribute-variable matching pairs P. The result will be used in the second part to build all longest combinations $\mathcal{P}_m$, which will be used to construct the matching between states and localities in the third part.

As validation of the algorithm is NP-complete, reducing the matching cost is planned as future work through the use of ontology and the matrix structure of graphs.

Let us detail the algorithm:

- As a starting point, P is the set that contains all potential equivalent attribute-variable pairs according to their domains:
$$P = \{(att, x) \mid att \in A, x \in X, Dom(att) \subseteq Dom(x)\}.$$
- $\mathcal{P}(P)$ is the power set of P [30], it contains all subsets c of P:
$$\forall c \subseteq P, c \in \mathcal{P}(P),$$
thus, $|\mathcal{P}(P)| = 2^{|P|}$. As A_e and X_e are not empty, the empty set can be removed from $\mathcal{P}(P)$:
$$\mathcal{P}\not\emptyset = \mathcal{P}(P) \backslash \{\emptyset\}.$$
- $\mathcal{P}_v$ is all valid combinations c of pairs attribute-variable in $\mathcal{P}\not\emptyset$. A combination c_i is valid if and only if it represents a bijection between its attributes and variables.
$$\mathcal{P}_v = \{c \mid c \in \mathcal{P}\not\emptyset, \forall (att_i, x_j) \in c,$$
$$(att_i, x_k) \notin c \wedge (att_l, x_j) \notin c\},$$
with $i \neq l$ and $j \neq k$.
- $\mathcal{P}_m$ is all maximized combinations[1] c_i of pairs attribute-variable in $\mathcal{P}_v$:
$$\mathcal{P}_m = \{c \mid c \in \mathcal{P}_v, \forall c_j \in \mathcal{P}_v, c \not\subset c_j\}. \tag{11}$$
$\mathcal{P}_m$ stores the maximized combinations according to the definition domain of attributes and variables.

As for each vertex it exists a unique locality such that for any couple attribute-variable of a combination, the values of the attribute are included in the domain of the locality, we keep from $\mathcal{P}_m$ only the combinations that satisfy such property. Therefore, $\mathcal{P}_{VQ}$ stores the possibles combinations c_i that correspond to a valid matching between vertices V and localities Q.

$$\begin{aligned}\mathcal{P}_{VQ} = \{c \mid c \in \mathcal{P}_m, \exists! v \in V, \exists! q \in Q, \forall (att, x) \in c,\\ dom(v, att) \subseteq dom(q, x)\}.\end{aligned} \tag{12}$$

From $\mathcal{P}_{VQ}$, all the possible matching $\mathcal{M}_{VQ}$ that stores the sets of vertex-locality couples can be deduced:

$$\begin{aligned}\mathcal{M}_{VQ} = \{m_i \mid \forall c_i \in \mathcal{P}_{VQ}, \exists! v \in V, \exists! q \in Q,\\ \forall (att, x) \in c_i, dom(v, att) \subseteq dom(q, x),\\ (v, q) \in m_i\}.\end{aligned} \tag{13}$$

These matching $\mathcal{M}_{VQ}$ are valid regarding the couples attribute att variable x and the couples vertex v locality q. Since the matching algorithm is based on the graph morphism, it needs to respect the structure of the matched graphs [31]. In our context, the images of the vertices of the STG in the IOSTS – the localities – must respect the *adjacency relations (neighboring)* present in the STG (i.e., the transitions). In other words, two adjacent vertices must match the same or two adjacent localities. Consider $\mathcal{S}$ is the surjective application of the STG in the IOSTS respectively between the vertices V and localities Q (see Equation 14), i.e., each vertex matches one locality and a locality is matched by at least one vertex. For any transition $(u, v) \in E$ of STG, then $(\mathcal{S}(u), \mathcal{S}(v)) \in T$ is a transition of the IOSTS. The STG → IOSTS matching is a surjective homomorphism, i.e., an epimorphism [31].

$$\begin{aligned}\mathcal{S} \colon STG &\to IOSTS,\\ V &\twoheadrightarrow Q = \mathcal{S}(V),\end{aligned} \tag{14}$$

implies:

$$\begin{gathered}\mathcal{S}_E \colon E \to T,\\ \mathcal{S}_E((u, v)) = (\mathcal{S}(u), \mathcal{S}(v)),\\ (\mathcal{S}(u), \mathcal{S}(v)) \subset T.\end{gathered} \tag{15}$$

[1] The longest combinations such that any subset of a combination does not exist in the set of combinations.

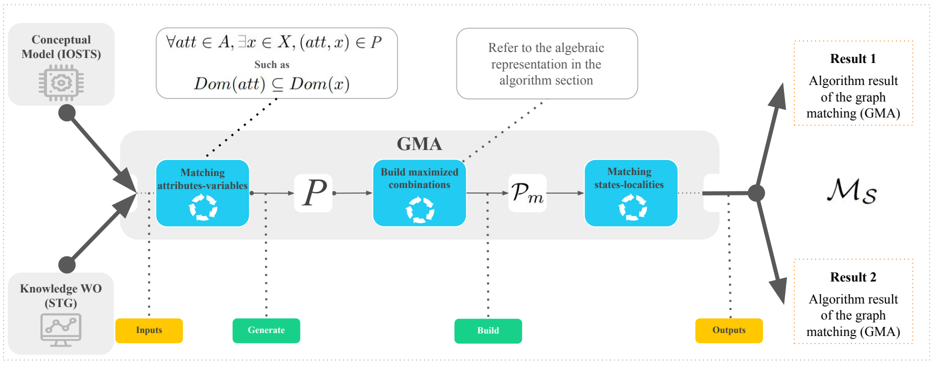

Conceptual Model (IOSTS)
Knowledge WO (STG)
$\forall att \in A, \exists x \in X, (att, x) \in P$
Such as
$Dom(att) \subseteq Dom(x)$
Refer to the algebraic representation in the algorithm section
GMA
Matching attributes-variables
P
Build maximized combinations
$\mathcal{P}_m$
Matching states-localities
Inputs
Generate
Build
Outputs
Result 1
Algorithm result of the graph matching (GMA)
$\mathcal{M}_{\mathcal{S}}$
Result 2
Algorithm result of the graph matching (GMA)

Fig. 4: Illustration of stepwise matching algorithm (outputs in Fig. 5 and Fig. 6)
.

According to Equation 15, if state v' is the neighbor of v and the locality q matches with v, v' must match with q or a neighbor of q:

$$\mathcal{M}_{\mathcal{S}} = \{m \mid m \in \mathcal{M}_{VQ}, \forall (v, q) \in m, \forall v' \in neighbors(v), \\ \exists q' \in neighbors(q) \cup \{q\}, (v', q') \in m\}. \tag{16}$$

C. Matching illustration

In the previous example, the STG in Fig. 2 is automatically generated by a WO and the IOSTS in Fig. 3 is provided by a developer. Both represent the same simplified roller shutter behavior. The behavior is simplified to highlight the algorithm, the implementation can deal with complex behaviors. The STG uses discrete values with a level of opening of 50% and a slats orientation of 90%, while the IOSTS use continuous intervals, without any constraint on the step that is a real value.

Fig. 5 and Fig. 6 illustrate all possible matching results of the API developed in the LISTIC laboratory of both (STG and IOSTS) graphs. Localities in the IOSTS are *Closed* and *Open*, each contains variables with disjoint domains (see Equation 8), in our example, both variables *height* and *angle*.

According to the constraints of the matching algorithm given in Section IV-A:

1) there is potential equivalent attributes/variables between the STG and the IOSTS, more precisely between the attributes "level, orientation" and the variables "height, angle". According to Equation 10, the following pairs represent potential attributes/variables equivalences:

$$(level, height), \\ (orientation, angle), \\ (orientation, height),$$

2) the domains of both localities *Closed* and *Open* respect Equation 8. Thus, *Closed* and *Open* are disjoint.

On the STG side, there are four vertices, each one labeled with a set of attribute-value pairs $(att, value)$. In our case, the pair $(level, orientation)$ takes the values $[(50, 90), (100, 90), (50, 0), (0, 0)]$ respectively for

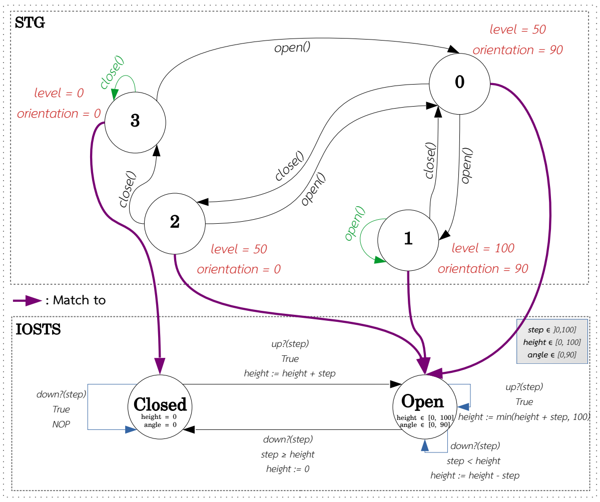


Fig. 5: Algorithm result of the graph matching $\mathcal{P}_m^1 = \{(level, height), (orientation, angle)\} \in \mathcal{P}_{VQ}$.

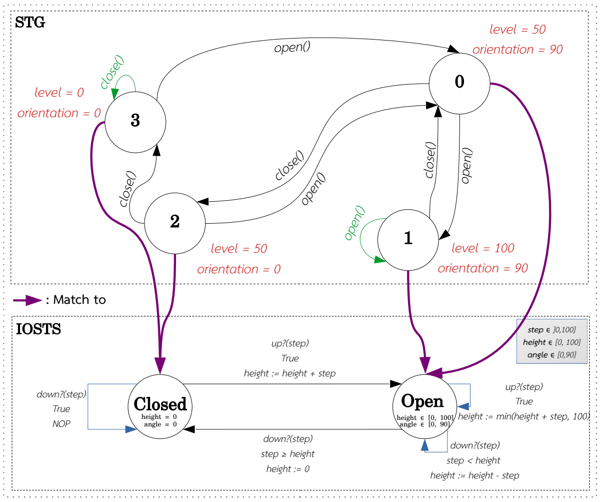


Fig. 6: Algorithm result of the graph matching $\mathcal{P}_m^2 = \{(orientation, height)\} \in \mathcal{P}_{VQ}$.

(v_0, v_1, v_2, v_3). Therefore, to establish a correspondence between the two graphs, for all combinations $\mathcal{P}_m(P)$, the definition domain of the pair $(level, orientation)$ in each vertex of the STG must be compared to the definition domain of the variables pair $(height, angle)$ in each locality of the IOSTS. This comparison gives $\mathcal{P}_{VQ}$, which consequently implies $\mathcal{M}_{\mathcal{S}}$. In detail, the following sets are obtained:

$$P = \{(level, height), (orientation, angle),\\ (orientation, height)\}.$$

The powerset of P:

$$\mathcal{P}\emptyset(P) = \{\{(level, height)\}, \{(orientation, angle)\},\\ \{(orientation, height)\},\\ \{(level, height), (orientation, angle)\},\\ \{(level, height), (orientation, height)\},\\ \{(orientation, angle), (orientation, height)\},\\ \{(level, height), (orientation, angle),\\ (orientation, height)\}\}.$$

Therefore:

$$\mathcal{P}_v = \{\{(level, height)\}, \{(orientation, angle)\},\\ \{(orientation, height)\},\\ \{(level, height), (orientation, angle)\}\}.$$

Thus:

$$\mathcal{P}_m = \{\{(orientation, height)\},\\ \{(level, height), (orientation, angle)\}\}.$$

As $\mathcal{P}_{VQ} \equiv \mathcal{P}_m$ in this illustration, it contains two combinations, which gives two matches

$$\mathcal{M}_{VQ} = \{\\ \{(v_0, Open), (v_1, Open), (v_2, Open), (v_3, Closed)\},\\ \{(v_0, Open), (v_1, Open), (v_2, Closed), (v_3, Closed)\}\\ \}.$$

Since both matching are surjective in this illustration, $\mathcal{M}_{VQ} \equiv \mathcal{M}_{\mathcal{S}}$.

This example provides two possible matches and the algorithm cannot determine, which one corresponds to (A_e, X_e). More information is required to determine the good matching. This information can be provided by the end user or, as we intend to do in future work, determined from the meaning of attributes, variables, methods and actions using an ontology.

V. Related work

For many years, graphs have been used in several fields to represent complex problems in a descriptive way (e.g., maps, relationships between people profiles, public transportation, scene analysis, chemistry, molecular biology, the quest for evolutionary conserved pathways thought protein-protein alignment, etc.). This has been done for various purposes: analysis, operation, knowledge modeling, pattern detection, etc. Although initiated in the 18th century with Euler's work on the problem of Königsberg bridges [15], graph theory remains a powerful tool for software-intensive system development and an effective way to represent objects as in [32]. Since then, several approaches of graph matching have been developed and the first formulation of the graph matching problem was proposed by [33] and dates back to 1979. Afterwards, several formulations appeared like convex-concave programming formulation, maximum common subgraph (MCS), the use of the Frobenius norm that uses the adjacency matrices of corresponding graphs to express the maximization or the minimization of the non-overlapping edges between two graphs, graph matching using dummy vertices that consist of finding a matching with the exception of some vertices in the data graph, which have no correspondence at all. In general, there exist two major formulations of the graph matching problem [34] [35]:

- *Exact Matching* is divided into two categories, (a) graph isomorphism, checks whether two graphs are the same. (b) subgraph isomorphism, checks whether the smallest graph is a subgraph of the biggest one. Both techniques are overly complex and rely on graph/subgraph isomorphism, whether or not they check the one-to-one or many-to-one matching.
- *Inexact Matching* is a term used where it is impossible to find an isomorphism between two graphs, and it comprises many approaches:
  - the maximum common subgraph, used in searching the similarity between the graphs to know how different they are instead of a binary answer [36]
  - least square formulation, used in the case of weighted graphs to search for a match that minimizes the total difference between all aligned edges through the use of the Frobenius norm for instance [36].
  - graph edit distance, used to find in a low cost the sequence of operations (i.e., deletion, insertion and substitution of vertices and edges) that transform one graph into another [37]. As this procedure is a hard combinatorial problem, another alternative called "beam search" is explained in details in [38]

In real applications, we often want to match graphs of different sizes, which results in new techniques and norms as depicted in [34]. Moreover, the problem is more extensive than one might imagine, as graphs are used to represent objects, images, regions in maps, also, many formalisms have emerged so far, such as the correspondence between different representations of knowledge as an STG and an IOSTS, which, to our knowledge, no paper has addressed. Until now, the most well-known operations on graphs is the comparison of two or more graph representations that requires many theoretical and complex concepts [21], like graph matching, which is a more constrained version of the graph isomorphism problem that is at the basis of our proposal. Finally, we mention that graph/sub-graph isomorphism and homomorphism are considered to be the most complex problems in graph matching, they are NP-complete. These problems have been studied in [39] [21] [40]. It is worth to mention that for certain types of graphs under certain constraints, the complexity of the isomorphism has been proven to be of polynomial type with a huge cost [41].

The matching of two knowledge representations (STG - IOSTS) led us to two problems: exact matching and inexact matching. To understand the problems, we need to see the matching from both perspectives: machine (i.e., numerical and structural) and human (i.e., semantic). According to the machine, and since the matching preserves the structure and the transitions between both formalisms, the matching is always exact between "states" and "localities", which gives an epimorphism (Equations 14, 15). However, from the human perspective, in most real-life cases, there will be at most one exact matching according the semantic. As illustrated in the illustrations (Fig. 5 and Fig. 6), only one matching is exact. The exact matching problem is a great challenge, our work focuses on this problem by taking into account, in addition to the numerical perspective, the semantic perspective.

VI. Conclusion and future work

Our research work on software intelligent systems, namely WSs, tackles the issue of bringing closer knowledge generated by AI and human semantic. Indeed a major weakness of WOs composing a WS is that they generate numerical information mostly meaningless to humans. Our proposal is to extend knowledge issued by WOs (expressed in STGs) at runtime with knowledge (expressed in IOSTs) provided by developers at design time. Such extension is based on graph matching.

We have proposed in this paper an algorithm to face the problem of matching an STG and an IOSTS. The goal of this work is to extend WOs with the behavior semantic defined at software design time. From the end user's perspective, the algorithm provides the system with the ability to communicate with him using human semantic. From the developer's perspective, the resulted matching may help him discover errors or inconsistencies between the conceptual view and the system implementation. The results of the algorithm provide a set of valid matching according to the numerical and structural information stored in both behavioral graphs.

As illustrated in the paper, the algorithm provides a set of valid matching, however, it cannot determine the valid one from the end user perspective. Besides, the algorithm has neither an idea of the meaning of variables that represent the same information at different scales, nor of variable names. Thus, one of our future work will address this problem: How to take semantic into account in the matching problem? From the end user perspective, this problem is a semantic one, the natural solution is therefore to rely on ontologies. Approaches we are currently considering consist in semantic graph matching based on an ontology merged with the results presented in this paper.

In this respect, we have initiated a France-Canada innovation project with the University of Sherbrooke to investigate matching algorithms based on other semantic formalisms than IOSTSs, such as ontologies and scenario-based [8] [7]. The main idea is to bind the algorithm matching results with an ontology to provide a human level communication with users according to the context of the application. Moreover, this collaboration brings us the medical context as a new application domain, namely ambient assistance systems for elderly people.

Acknowledgment

This research was supported by French National Research Agency (ANR), AI Ph.D funding project.

References

[1] M. Weiser and J. S. Brown, "Designing calm technology," *POWERGRID JOURNAL*, vol. 1, 1996.

[2] A. Tugui, "Calm technologies in a multimedia world," *Ubiquity*, vol. 2004, 03 2004. doi: 10.1145/985616.985617

[3] I. Alloui, D. Esale, and F. Vernier, "Wise objects for calm technology," in *Proceedings of the 10th International Conference on Software Engineering and Applications - ICSOFT-EA, (ICSOFT 2015)*, INSTICC. SciTePress, 2015. doi: 10.5220/0005560104680471 pp. 468–471 – Section 2.

[4] S. Lejamble, I. Alloui, S. Monnet, and F. Vernier, "A new software architecture for the wise object framework: Multidimensional separation of concerns," in *Proceedings of the 17th International Conference on Software Technologies - ICSOFT,*, INSTICC. SciTePress, 2022. doi: 10.5220/0011355000003266 pp. 567–574.

[5] R. A. Flores-Mendez, "Towards a standardization of multi-agent system framework," *XRDS*, vol. 5, no. 4, p. 18–24, jun 1999. doi: 10.1145/331648.331659

[6] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 1–1, 2018. doi: 10.1109/ACCESS.2018.2831228

[7] D. Bonino and F. Corno, "Dogont - ontology modeling for intelligent domotic environments," in *The Semantic Web - ISWC 2008*, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-88564-1_51 pp. 790–803.

[8] H. K. Ngankam, H. Pigot, M. Frappier, O. Souza, H. Camila, and S. Giroux, "Formal specification for ambient assisted living scenarios," *UCAmI*, pp. 508–519, 11 2017. doi: 10.1007/978-3-319-67585-5_51

[9] J.-B. Woo and Y.-K. Lim, "User experience in do-it-yourself-style smart homes," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '15. New York, NY, USA: Association for Computing Machinery, 2015. doi: 10.1145/2750858.2806063 p. 779–790.

[10] R. Radziszewski, H. Ngankam, H. Pigot, V. Grégoire, D. Lorrain, and S. Giroux, "An ambient assisted living nighttime wandering system for elderly," in *Proceedings of the 18th International Conference on Information Integration and Web-Based Applications and Services*, ser. iiWAS '16. New York, NY, USA: Association for Computing Machinery, 2016. doi: 10.1145/3011141.3011171 p. 368–374.

[11] R. S. Michalski, "A theory and methodology of inductive learning," *Artificial Intelligence*, vol. 20, no. 2, pp. 111–161, 1983. doi: 10.1016/0004-3702(83)90016-4

[12] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 66–75, January 1991. doi: 10.1145/329124.329126

[13] Y. Brun, G. D. M. Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Muller, M. Pezzè, and M. Shaw, *Engineering Self-Adaptive Systems through Feedback Loops*. Springer Berlin Heidelberg, 2009, pp. 48–70.

[14] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, pp. 41 – 50, 02 2003. doi: 10.1109/MC.2003.1160055

[15] H. Sachs, M. Stiebitz, and R. Wilson, "An historical note: Euler's königsberg letters," *Journal of Graph Theory*, vol. 12, pp. 133–139, 10 2006. doi: 10.1002/jgt.3190120114

[16] C. Constant, T. Jéron, H. Marchand, and V. Rusu, "Integrating Formal Verification and Conformance Testing for Reactive Systems," *IEEE Transactions on Software Engineering*, vol. 33, no. 8, pp. 558–574, Aug. 2007. doi: 10.1109/TSE.2007.70707

[17] C. Camille, J. Thierry, M. Hervé, and R. Vlad, "Validation of Reactive Systems," in *Modeling and Verification of Real-TIME Systems - Formalisms and software Tools*, S, Merz, N, and Navet, Eds. Hermès Science, Jan. 2008, pp. 51–76. ISBN 978-1848210134

[18] I. Boudhiba, C. Gaston, L. G. Pascale, and V. Prevosto, “Input Output Symbolic Transition Systems Enriched by Program Calls and Contracts: a detailed example of vending machine,” Laboratoire MAS - Centrale-Supelec, Research Report, 2015.

[19] J. Dörpinghaus, T. Hübenthal, and J. Faber, “A novel link prediction approach on clinical knowledge graphs utilizing graph structures,” in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, vol. 30. IEEE, 2022. doi: 10.15439/2022F36 p. 43–52.

[20] A. Dahhani, I. Alloui, S. Monnet, and F. Vernier, “Towards a semantic model for wise systems - a graph matching algorithm,” in *ADVCOMP 2022, The Sixteenth International Conference on Advanced Engineering Computing and Applications in Sciences*, S. Laura Garcia, Universitat Politecnica de Valencia, Ed., vol. 34, Nov. 2022. ISBN 2308-4499 pp. 27–34.

[21] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*. W.H. Freeman and Co., 1979. ISBN 0716710455

[22] V. A. Cicirello, “Survey of graph matching algorithms,” Geometric and Intelligent Computing Laboratory, Drexel University, Technical Report, March 1999.

[23] T. Davenport and L. Prusak, *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, 01 1998, vol. 1. ISBN 1578513014

[24] P. Moreaux, F. Sartor, and F. Vernier, “An effective approach for home services management,” in *2012 20th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, 2012. doi: 10.1109/PDP.2012.45 pp. 47–51.

[25] M. N. Nicolescu and M. J. Mataric, “Extending behavior-based systems capabilities using an abstract behavior representation,” in *AAAI 2000*, North Falmouth, MA, 2000, pp. 27–34.

[26] V. Rusu, H. Marchand, V. Tschaen, T. Jéron, and B. Jeannet, “From safety verification to safety testing,” in *Testing of Communicating Systems*, 03 2004. doi: 10.1007/978-3-540-24704-3_11. ISBN 978-3-540-21219-5 pp. 160–176.

[27] V. Rusu, H. Marchand, and T. Jéron, “Automatic verification and conformance testing for validating safety properties of reactive systems,” in *Formal Methods 2005 (FM05)*, ser. Lecture Notes in Computer Science, vol. 3582. Newcastle, United Kingdom: Springer-Verlag, Jul. 2005. doi: 10.1007/11526841_14 pp. 189–204.

[28] C. Gaston, P. Le Gall, N. Rapin, and A. Touil, “Symbolic execution techniques for test purpose definition,” in *Testing of Communicating Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. doi: 10.1007/11754008_1 pp. 1–18.

[29] M. Mashaal, *Bourbaki*, ser. Bourbaki : A secret society of mathematicians. American Mathematical Society, Jun. 2006. ISBN 9780821839676

[30] A. Salomaa, I. N. Sneddon, H. M. Stark, and J.-P. Kahane, *Theory of Automata : International Series of Monographs in Pure and Applied Mathematics*, ser. International series in pure and applied mathematics. - page.1-2. London : Elsevier Science, 2015., 1969-2015. ISBN 978-1483121970

[31] G. Hahn and C. Tardif, *Graph homomorphisms: structure and symmetry*. Dordrecht: Springer Netherlands, 1997, pp. 107–166. ISBN 978-94-015-8937-6

[32] M. Eshera and K.-S. Fu, “An image understanding system using attributed symbolic representation and inexact graph-matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 604–618, 1986. doi: 10.1109/TPAMI.1986.4767835

[33] W.-H. Tsai and K.-S. Fu, “Error-correcting isomorphisms of attributed relational graphs for pattern analysis,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 12, pp. 757–768, 1979. doi: 10.1109/TSMC.1979.4310127

[34] M. Zaslavskiy, “Graph matching and its application in computer vision and bioinformatics,” Theses, École Nationale Supérieure des Mines de Paris, Jan. 2010.

[35] E. Bengoetxea, “Inexact graph matching using estimation of distribution algorithms,” Ph.D. dissertation, Ecole Nationale Supérieure des Télécommunications, Paris, France, Dec 2002.

[36] J. R. Ullmann, “An algorithm for subgraph isomorphism,” *J. ACM*, vol. 23, no. 1, p. 31–42, jan 1976. doi: 10.1145/321921.321925

[37] H. Bunke and G. Allermann, “Inexact graph matching for structural pattern recognition,” *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245–253, 1983. doi: 10.1016/0167-8655(83)90033-8

[38] M. Neuhaus, K. Riesen, and H. Bunke, “Fast suboptimal algorithms for the computation of graph edit distance,” in *Structural, Syntactic, and Statistical Pattern Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. ISBN 978-3-540-37241-7 pp. 163–172.

[39] D. A. Basin, “A term equality problem equivalent to graph isomorphism,” *Information Processing Letters*, vol. 51, no. 2, pp. 61–66, 1994. doi: 10.1016/0020-0190(94)00084-0

[40] M. A. Abdulrahim, *Parallel algorithms for labeled graph matching*. Colorado School of Mines1500 Illinois St. Golden, CO, 1998.

[41] J. E. Hopcroft and J. K. Wong, “Linear time algorithm for isomorphism of planar graphs (preliminary report),” in *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, ser. STOC ’74. Association for Computing Machinery, 1974. doi: https://dx.doi.org/10.1145/800119.803896 p. 172–184.

DOI: 10.15439/2023F6191
ISSN 2300-5963 ACSIS, Vol. 35

Modelling and Solving the Precedence-Constrained Minimum-Cost Arborescence Problem with Waiting-Times

Mauro Dell'Amico, Jafar Jamal, Roberto Montemanni
0000-0002-3283-6131
0009-0003-9368-1094
0000-0002-0229-0465
Department of Sciences and Methods for Engineering
University of Modena and Reggio Emilia
Via Amendola 2, 42122 Reggio Emilia, Italy
Email: mauro.dellamico@unimore.it;
268108@studenti.unimore.it;
roberto.montemanni@unimore.it

***Abstract*—A polynomial-size mixed integer linear programming model for the Precedence-Constrained Minimum-Cost Arborescence Problem with Waiting-Times was recently proposed in the literature, that uses a smaller number of variables and constraints compared to previously proposed polynomial-size models. In this work, we extend this model with constraint programming constructs to further enhance its performance. An extensive computational study support that modern constraint programming solvers are the best tool available at solving the models proposed. Several improvements to state-of-the-art results are finally reported.**

***Index Terms*—Combinatorial Optimization; Arborescences; Precedence-Constraints.**

I. Introduction

THE *Minimum-Cost Arborescence* (MCA) problem involves finding a directed minimum-cost spanning tree, rooted at vertex r, in a given input directed graph. Jack Edmonds [1], and Yoeng-Jin Chu and Tseng-Hong Liu [2] independently introduced the first polynomial time algorithm for solving the problem. Gabow and Tarjan [3] improved the running time of the algorithm by using *disjoint-sets* and a special implementation of *Fibonacci heaps*.

Several variations of the MCA problem with different objective function and/or constraints were introduced in the literature since its introduction. Given a finite resource associated with each vertex in the input graph, the *Resource-Constrained Minimum-Weight Arborescence problem* [4] is an $\mathcal{NP}$-hard problem which asks to find an arborescence with minimum total cost where the sum of the costs of outgoing arcs from each vertex is at most equal to the resource of that vertex. Given an integer Q and non-negative integer vertex demand q_j associated with each vertex, the *Capacitated Minimum Spanning Tree problem* [5] is an $\mathcal{NP}$-hard problem which asks to find a directed minimum spanning tree rooted at r, such that the sum of the weights of the vertices in any subtree off the root is at most Q. Given a weighted directed acyclic graph with each vertex having a specified color from a set of colors, the *Maximum Colorful Arborescence problem* [6] is an $\mathcal{NP}$-hard problem which asks to find an arborescence of maximum weight, in which no color appears more than once. Given an integer rank associated with each vertex, the *Restricted Fathers Tree problem* [7] asks to find a minimum-cost arborescence rooted at r, such that the path between each vertex and the root contains only vertices with same rank or higher.

Constraint programming (CP) is paradigm for solving combinatorial problems by representing them as *constraint satisfaction problems* (CSP) [8]. A CSP is represented as a set of variables each with a defined domain of values, and a set of relations/constraints on the subsets of these variables. A CP solver takes a CSP and finds an assignment to all the variables that satisfies the constraints, and can also extend the problem to finding optimal solutions according to an optimization criteria. A CP solver searches the solution space systematically using a branch-and-bound algorithm with inference techniques which consists of propagating the information contained in one constraint to the neighboring constraints. Such techniques reduce the size of the solution space that needs to be explored [9]. CP has been used to solve a wide range of problems in the literature. Hande [10] proposed a CP model for the *Open Vehicle Routing problem with Heterogeneous Vehicle Fleet* (HFOVRP). In [10] the CP model is compared with a mixed-integer linear programming (MILP) model of the HFOVRP, and they showed that the CP model is effective for providing good-quality solutions for small-sized instances of the HFOVRP in short computational times compared to the MILP model. Kasapidis et al. [11] presented a MILP model and a CP model for the *Multi-Resource Flexible Job-Shop Scheduling problem with Arbitrary Precedence Graphs*. The computational experiments conducted in [11] has shown that the CP model is more effective and achieves the best

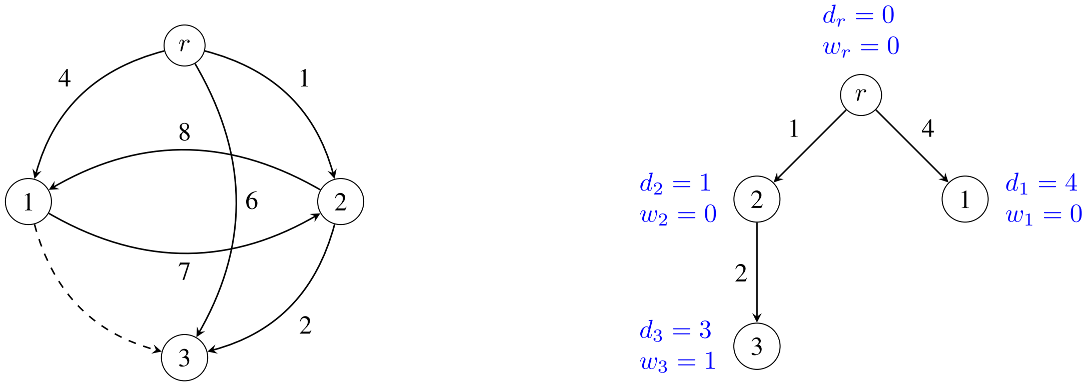


Fig. 1: Example of an instance solved as a PCMCA-WT. The graph on left shows the instance with its respective arc costs, and the precedence relationship $(1, 3) \in R$ marked as a dashed arrow. The graph on the right shows an optimal PCMCA-WT solution of cost 8.

results compared to the MILP model, although more time-consuming on some instances. Kirac et al. [12] proposed a CP approach for solving the *Team Orienteering problem with Time Windows and Mandatory Visits*, and they showed that the CP-based approach finds 99 of the best-known solutions and explores 64 new best-known solutions for the benchmark instances. Kizilay et al. [13] proposed a novel CP model for the *Mixed-Blocking Permutation Flow Shop Scheduling problem with Batch Delivery* that minimizes the total tardiness and batch cost. The results of their study has shown that due to the complexity of the problem, the developed CP model can solve only small-sized instances in reasonable computational time. Montemanni and Dell'Amico [14] proposed a CP model for the *Parallel Drone Scheduling Traveling Salesman problem*, and showed that by exploiting multi-threading computation, the method was able to optimally solve all the instances considered in the literature.

The *Precedence-Constrained Minimum-Cost Arborescence* (PCMCA) problem is an $\mathcal{NP}$-hard problem [15] that was first introduced by Dell'Amico et al. [16]. The PCMCA problem is an extension to the MCA problem, in which precedence constraints must be satisfied as follows. Given a set R of ordered pairs of vertices, then for each precedence relationship $(s, t) \in R$, a path in the solution which covers both s and t, must visit vertex s before visiting vertex t. The objective is to find an arborescence of minimum total cost satisfying the precedence constraints. The PCMCA problem has applications in the design of commodity distribution networks where certain paths are not allowed in the network due to logistical constraints [16]. Several MILP models of the problem were proposed in [15], [16], [17].

The *Precedence-Constrained Minimum-Cost Arborescence Problem with Waiting-Times* (PCMCA-WT) is an $\mathcal{NP}$-hard problem that was recently introduced by Chou et al. [15]. The PCMCA-WT is a variation on the PCMCA problem characterized by the following differences. Given arc costs indicating the time required to traverse an arc, suppose there is a flow which starts at the root vertex r, that must reach every vertex in an arborescence T. For each precedence relationship $(s, t) \in R$, the flow must enter vertex s at the same time step, or before entering vertex t, which means that the flow can stop at any vertex and wait. The waiting time at vertex t is defined as the difference between the time at which the flow enters s and the time at which the flow reaches t. The objective of the problem is to find an arborescence T of minimum total cost, plus total waiting times, where the flow never enters s after entering t for all $(s, t) \in R$. Several MILP models for solving the problem were proposed in [15].

The PCMCA-WT problem can be formally defined as follows. Given a directed graph $G = (V, A, R, r)$, where $V = \{1, \dots, n\}$ is the set of vertices, $A \subseteq V \times V$ is the set of arcs, $R \subset V \times V$ is the set of precedence relationships, and $r \in V$ is the root of the arborescence. Let c_{ij} be a cost associated with each arc $(i, j) \in A$ which represents the time required for the flow to travel from vertex i to vertex j. Let d_j be the time step at which the flow enters vertex $j \in V$, and let w_j be the waiting time at vertex $j \in V$. The objective of the problem is to find an arborescence T rooted at vertex r, that has a minimum total cost plus total waiting time, where the flow never enters t before entering s for all $(s, t) \in R$ (i.e. $d_t \geq d_s$ for all $(s, t) \in R$).

Figure 1 presents an example of an instance solved as a PCMCA-WT. The instance graph (left graph) shows the precedence relationship $(1, 3) \in R$ marked as a dashed arrow, while the solution graph (right graph) shows an optimal solution of that instance, with the corresponding d_i and w_i value written next to each vertex. The graph on the right shows an optimal PCMCA-WT solution of cost 8 (sum of all the arcs cost plus total waiting time at each vertex), with a resulting waiting time of value 1 at vertex 3, since $d_1 = 4$, while $d_3 = 3$ and $(1, 3) \in R$.

The rest of this paper is organized as follows. Section II introduces the MILP model used in this study. Section III introduces a CP model that extends the MILP model introduced in Section II by introducing redundant constraints for a subset of the original inequalities and describing them in terms of

CP constructs, in order to further exploit the capabilities of the CP solver. Section IV summarizes computational results, while some conclusions are outlined in Section V.

II. A Mixed Integer Linear Programming Model

A polynomial-size MILP model for the PCMCA-WT was recently proposed by Dell'Amico et al. [18]. The model extends a classical formulation for the MCA problem [19], through the addition of precedence-enforcing constraints. The precedence-enforcing constraints detect a precedence violating path by propagating a value along all the paths of the solution starting from t for all $(s,t) \in R$ [16], [17].

A different version of the model that contains a smaller number of variables and constraints was also proposed in [18]. The reduction is achieved by exploiting the special property of the PCMCA-WT, that is for any precedence relationship $(s,t) \in R$, the flow must enter vertex t at the same time step or after entering vertex s. This implies that it is possible to remove a precedence relationship $(s,t) \in R$ when the input graph does not contain a zero-cost path that starts from t and ends in s. The reduced model for the PCMCA-WT proposed in [18] is summarized as follows. For further details the reader can refer to [18].

Let x_{ij} be a variable associated with every arc $(i,j) \in A$ such that $x_{ij} = 1$ if $(i,j) \in T$, and 0 otherwise. Let y_i be a variable associated with every vertex $i \in V$ that indicates the order in which vertex i is visited on the path connecting vertex i to the root r. Let u_j^t be a variable associated with every vertex $j \in V$, and vertex $t \in V$ where t is part of a precedence relationship (i.e. $\exists (s,t) \in R$). Let d_j be the time at which the flow enters vertex $j \in V$, and let w_j be the waiting time before the flow enters vertex j. Let $P_{ij} \subset A$ be a simple directed path that starts from i and ends at j, and let $c(P_{ij}) = \sum_{(i,j)\in P} c_{ij}$ be the cost of that path. For each $s \in V$, let $V_s = \{t \in V \backslash \{r\} \,|\, \exists\, (s,t) \in R, c(P_{ts}) = 0\}$. The PCMCA-WT can be formulated as the following MILP model.

$$\text{minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{i\in V} w_i \qquad (1)$$

$$\text{s.t.:} \quad \sum_{(i,j)\in A} x_{ij} = 1 \qquad \forall j \in V\backslash\{r\} \qquad (2)$$

$$y_i - y_j + 1 \leq n(1 - x_{ij}) \qquad \forall (i,j) \in A : j \neq r \qquad (3)$$

$$u_s^t = 0 \qquad \forall (s,t) \in R : t \in V_s \qquad (4)$$

$$u_t^t = 1 \qquad \forall t \in V_s \qquad (5)$$

$$u_j^t - u_i^t - x_{ij} \geq -1 \quad \forall (s,t) \in R : t \in V_s, (i,j) \in A \qquad (6)$$

$$d_r = 0 \qquad (7)$$

$$w_r = 0 \qquad (8)$$

$$d_j \geq d_i - M + (M + c_{ij})x_{ij} \qquad \forall (i,j) \in A \qquad (9)$$

$$w_j \geq d_j - d_i - M + (M - c_{ij})x_{ij} \qquad \forall (i,j) \in A \qquad (10)$$

$$d_t \geq d_s \qquad \forall (s,t) \in R \qquad (11)$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (12)$$

$$y_i \geq 0 \qquad \forall i \in V \qquad (13)$$

$$u_j^t \geq 0 \qquad \forall t \in V_s, j \in V \qquad (14)$$

$$d_i, w_i \in \mathbb{R}^+_{\leq M} \qquad \forall i \in V \qquad (15)$$

The set of constraints (2) impose that every vertex excluding the root must have exactly one parent. Constraints (3) are the subtour elimination constraints, which enforce that any feasible solution is acyclic. The set of constraints (2) and (3) guarantee that any feasible solution is an arborescence rooted at vertex $r \in V$. Constraints (4) and (5) fix the values of u_s^t and u_t^t to 0 and 1 respectively, for all $(s,t) \in R$, where $t \in V_s$. Constraints (6) impose that if $x_{ij} = 1$ then $u_j^t \geq u_i^t$ (see Figure 2 for further explanation). Constraint (7) sets the time step at which the flow enters the root to 0. Constraint (8) sets the waiting time at the root r to be equal to 0. Constraints (9) impose that when arc $(i,j) \in A$ is selected to be part of the solution, then the flow enters vertex j at a time step that is greater than or equal to the time step at which the flow enters vertex i plus c_{ij}. Constraints (10) enforce that the waiting time at vertex j is greater than or equal to the difference between the time at which the flow enters vertex j and the time at which the flow enters vertex i plus c_{ij}. Constraints (11) enforce that the time at which the flow enters vertex t is greater than or equal to the time at which the flow enters vertex s for all $(s,t) \in R$. Finally, constraints (12)-(15) define the domain of the variables, and M is an upper bound on the value of an optimal solution.

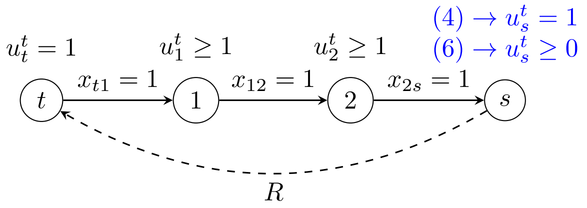


Fig. 2: An example on how a precedence relationship $(s,t) \in R$ can be enforced by propagating the value of u_t^t along every path starting from t, and if the solution contains a path from t to s, then we are propagating a value of one to vertex s and imposing that $u_s^t \geq 1$. However, we enforce $u_s^t = 0$, and therefore the solution violates the precedence relationship $(s,t) \in R$.

III. A New Constraint Programming Model

The CP solver used in this study, CP-SAT [20] is a solver that utilizes integer programming techniques (linear relaxation, presolve, cuts, and branching heuristics) to enhance its performance [21] and has recently been shown to successfully deal with different combinatorial optimization problems [22], [23]. Furthermore, the computational results in Section IV show that for the model considered in this work, the CP solver outperforms the MILP solver on a subset of the instances considered in terms of achieved average optimality gap, solution time, and the quality of the solutions obtained. Therefore, we introduce a CP model in this section that extend the MILP model introduced in Section II by adding the set of constraints

(3) and (6) formulated as logical constraints, and merging the two sets of constraints (9) and (10) into one set of logical constraints. By doing so, we further exploit the capabilities of the CP solver. Since the CP solver used utilizes integer programming techniques, it is beneficial to include both the logical and linear form of the constraints in the model, so that when the logical constraint is not enforced by the SAT solver (i.e. the logical constraint is not included in the model by the solver), their equivalent linear constraint is included in the program when computing its linear relaxation.

Using a set of implication constraints which enforce the implied constraint when the value of the variable is true, the MILP model introduced in Section II can be extended using the following set of constraints.

$$x_{ij} \implies y_j = y_i + 1 \qquad \forall (i,j) \in A : j \neq r \qquad (16)$$

$$x_{ij} \implies u_j^t \geq u_i^t \qquad \forall t \in V_t, j \in V \backslash \{r\} \qquad (17)$$

$$x_{ij} \implies d_j = d_i + w_j + c_{ij} \qquad \forall (i,j) \in A \qquad (18)$$

Constraints (16) are the subtour elimination constraints modelling the nonlinear relationship $y_j = (y_i + 1)x_{ij}$. Constraints (17) are the precedence-enforcing constraints modeling the nonlinear relationship $u_j^t \geq u_i^t x_{ij}$. Constraints (18) combine the two constraints (9) and (10) into a single equality constraint that model the nonlinear relationship $(d_j - d_i - w_j - c_{ij})x_{ij} = 0$. Note that variables d_i and w_i are defined as integers (compared to the MILP model), since a CP solver only accepts integer variables and coefficients. This means that c_{ij} for all $(i,j) \in A$ should be integer or to be discretized before solving the model. The value of c_{ij} can be discretized by multiplying every c_{ij} by a constant k, and then considering only the integer part of the result. In order to compute the correct solution cost, the objective function value should be divided by k. A higher k value leads to higher numerical precision, whereas a low k value leads to a lower numerical precision and thus faster execution. Therefore, a k value which balances the two factors should be considered. In this study we only consider instances with integer coefficients. However, the interested reader can refer to [14] where the authors show how changing the k value can affect the computation time.

IV. Experimental Results

The computational experiments are based on the benchmark instances of TSPLIB [24], SOPLIB [25], [26], and COMPILERS [27], originally proposed for the *Sequential Ordering Problem* (SOP) [28], [29], [30]. The benchmark instances are the same instances previously adopted in [15], [18] for the PCMCA-WT with the following characteristics. The benchmark sets contain a total of 116 instances (81 open instances) ranging in size between 9 and 700 vertices, with an average of 248 vertices. Finally, all instances have integer coefficients (i.e. the weight of the arcs of the cost graph is integer). All the experiments are performed on an Intel Xeon Platinum 8375C processor with 8 cores running at 2.9 GHz with 16 GB of RAM. For all instances an upper bound on the value of the optimal solution (i.e. M), is set to the value of the solution cost of solving the instance as a SOP, using a nearest neighbor algorithm [27]. This is a valid upper bound for the cost of the optimal solution of the PCMCA-WT, being a feasible solution for the SOP a simple directed path that includes all the vertices of the graph, such that t never precede s for all $(s,t) \in R$. This implies that $d_t \geq d_s$ for all $(s,t) \in R$, with a waiting time equal to zero at each vertex by definition.

The computational results are generated using two solvers: a *MILP Solver* and a *CP Solver*. The MILP Solver is CPLEX v12.8 [31], and is run with 8 thread standard B&C algorithm, with the two parameters *NodeSelect* and *MIP emphasis* are set to *BestBound* and *MIPEmphasisOptimality* respectively. The CP Solver is Google OR-Tools [20] v9.5 CP-SAT solver, and is run with its default parameters with all 8 threads available are allocated for the solver. A time limit of 1 hour is set on the computation time of both solvers. For the rest of this section we will be referring to the MILP model introduced in Section II as *BM* (Basic Model), while the CP model introduced in Section III will be referred to as *RM* (Reinforced Model).

Tables I, II and III show the complete results of each model and solving method, where we report the following. For each instance, columns *Name* and *Size* report the name and size of the instance. Column $\rho(R)$ reports the density of arcs in the set of precedence relationships computed as $\frac{2 \cdot |R|}{|V|(|V|-1)}$. Column *Best-Known* reports the best-known bounds on the optimal solution for each instance as $[LB, UB]$, where *LB* is the lower bound on the optimal solution, and *UB* is the best-known solution. The best-known solutions are obtained from the results appeared in [18], generated using the same computational setup and configuration used in this study. For each model solved with the corresponding solver, we report the following columns. Columns *LB* and *UB* report the lower/upper bound on the optimal solution achieved by the corresponding solving method of that model. Column *Gap* reports the optimality gap computed as $\frac{UB-LB}{UB}$. Column *Branches* reports the number of branches created in the search-decision tree, and is only reported when the models are solved with the CP Solver. Finally, column *Time [s]* reports the solution time in seconds and is only reported for the instances that are solved optimally within the time limit. In the tables, bold numbers indicate that a new best-known lower/upper bound is found.

A. Multi-threading Computation

The performance of CP solvers can often be greatly improved by the use of multi-threading computation, usually more than MILP solvers due to the different approaches used to solve the mathematical model. In this section, we assess the effect of multi-threading on the performance of the CP Solver and MILP Solver at solving the model introduced in Section II. The four instances *ft53.1, prob.42, ESC78*, and *jpeg.4753.54* were selected as both solvers are able to optimally solve those instances within the time limit using 8 threads.

Figure 3 reports the time required to optimally solve the different instances considered using a number of threads

TABLE I: Computational results for TSPLIB instances.

				MILP Solver				CP Solver									
Instance				BM				BM					RM				
Name	Size	$\rho(R)$	Best-Known	LB	UB	Gap	Time [s]	LB	UB	Gap	Branches	Time [s]	LB	UB	Gap	Branches	Time [s]
br17.10	18	0.314	44	40	44	0.091	-	44	44	0.000	56849	43.544	44	44	0.000	40829	34.568
br17.12	18	0.359	44	41	44	0.068	-	44	44	0.000	1225655	40.911	44	44	0.000	86038	36.960
ESC07	9	0.611	1906	1906	1906	0.000	0.070	1906	1906	0.000	153	0.050	1906	1906	0.000	102	0.020
ESC11	13	0.359	2174	2174	2174	0.000	0.114	2174	2174	0.000	339	0.059	2174	2174	0.000	339	0.076
ESC12	14	0.396	1138	1138	1138	0.000	0.030	1138	1138	0.000	326	0.044	1138	1138	0.000	302	0.046
ESC25	27	0.177	1158	1158	1158	0.000	1.945	1158	1158	0.000	16928	0.634	1158	1158	0.000	1451	0.578
ESC47	49	0.108	747	747	747	0.000	22.153	747	747	0.000	90693	3.008	747	747	0.000	5473	2.856
ESC63	65	0.173	56	56	56	0.000	57.347	56	56	0.000	7842	1.089	56	56	0.000	7708	1.289
ESC78	80	0.139	1196	1196	1196	0.000	257.609	1196	1196	0.000	297687	22.228	1196	1196	0.000	16383	7.453
ft53.1	54	0.082	4089	4089	4089	0.000	2023.553	4089	4089	0.000	47285	109.466	4089	4089	0.000	835469	110.951
ft53.2	54	0.094	[4161, 4284]	4161	4334	0.040	-	**4284**	**4284**	0.040	2513278	1284.742	**4284**	**4284**	0.000	237937	1522.620
ft53.3	54	0.225	[4799, 5279]	4799	5279	0.091	-	4474	5322	0.159	2596272	-	4428	5481	0.192	2275894	-
ft53.4	54	0.604	[5923, 6420]	5923	6420	0.077	-	5490	6489	0.154	2229608	-	5397	6420	0.159	2112150	-
ft70.1	71	0.036	[33101, 33298]	32827	33308	0.014	-	33105	33298	0.006	1130106	-	**33111**	33298	0.006	6020308	-
ft70.2	71	0.075	[33089, 33670]	33089	33916	0.024	-	**33261**	33824	0.017	5506591	-	33259	33670	0.012	4140282	-
ft70.3	71	0.142	[34423, 36932]	34423	38351	0.102	-	33813	**36152**	0.065	2653144	-	33773	36372	0.071	3190167	-
ft70.4	71	0.589	[36850, 36939]	36850	38771	0.050	-	35838	39706	0.097	4827165	-	35813	39897	0.102	3417417	-
rbg048a	50	0.444	263	259	264	0.019	-	263	263	0.000	14834	17.308	263	263	0.000	14896	17.679
rbg050c	52	0.459	225	225	225	0.000	36.673	225	225	0.000	8549	1.363	225	225	0.000	5436	2.196
rbg109	111	0.909	[366, 401]	366	407	0.101	-	358	402	0.109	1795755	-	357	**400**	0.108	172527	-
rbg150a	152	0.927	[463, 509]	461	509	0.094	-	454	556	0.183	153864	-	420	556	0.245	124524	-
rbg174a	176	0.929	[463, 553]	463	553	0.163	-	454	**537**	0.155	1356878	-	455	551	0.174	1226907	-
rbg253a	255	0.948	[532, 718]	532	718	0.259	-	515	672	0.234	888937	-	511	**665**	0.232	110453	-
rbg323a	325	0.928	[1009, 1891]	974	2466	0.605	-	996	1636	0.391	152276	-	889	**1544**	0.424	104927	-
rbg341a	343	0.937	[780, 1457]	761	2907	0.738	-	643	1283	0.499	640718	-	668	**1213**	0.449	1928843	-
rbg358a	360	0.886	[788, 1150]	755	2453	0.692	-	758	**1052**	0.279	219152	-	701	1130	0.380	130991	-
rbg378a	380	0.894	[678, 1126]	648	2191	0.704	-	582	**1070**	0.456	543767	-	639	1087	0.412	352856	-
kro124p.1	101	0.046	[32630, 33962]	32630	36099	0.096	-	32576	34235	0.048	422907	-	32544	34433	0.055	643136	-
kro124p.2	101	0.053	[33006, 35860]	33006	39931	0.173	-	32781	36284	0.097	10775155	-	32800	36687	0.106	12286675	-
kro124p.3	101	0.092	[34005, 42416]	34005	46764	0.273	-	33716	40958	0.177	7997226	-	33621	**40814**	0.176	527262	-
kro124p.4	101	0.496	[39333, 49590]	39333	53456	0.264	-	38268	48940	0.218	7075173	-	38333	**48035**	0.202	1783421	-
p43.1	44	0.101	[2860, 3955]	2656	3955	0.328	-	2860	3980	0.281	5139272	-	2852	3955	0.279	2323809	-
p43.2	44	0.126	[2870, 4160]	2705	4210	0.357	-	2837	4105	0.309	7941383	-	**2877**	**4020**	0.284	3390171	-
p43.3	44	0.191	[2966, 4255]	1383	4440	0.689	-	2880	4350	0.338	3165556	-	2929	4425	0.338	1828845	-
p43.4	44	0.164	[3125, 4495]	3125	4605	0.321	-	3048	4495	0.322	1603271	-	3047	4540	0.329	1316102	-
prob.100	100	0.048	[677, 738]	677	741	0.086	-	674	738	0.087	221458	-	674	**734**	0.082	178127	-
prob.42	42	0.116	171	171	171	0.000	230.506	171	171	0.000	230646	37.298	171	171	0.000	28930	78.549
ry48p.1	49	0.091	[13200, 13670]	13200	13670	0.034	-	13197	**13665**	0.034	692909	-	13157	13670	0.038	754011	-
ry48p.2	49	0.103	[13336, 14305]	13336	14305	0.068	-	**13370**	**14224**	0.060	23506295	-	13360	**14224**	0.061	16644814	-
ry48p.3	49	0.193	[13994, 15477]	13994	15840	0.117	-	13757	15477	0.111	17599209	-	13949	**15439**	0.097	554056	-
ry48p.4	49	0.588	[17180, 19495]	17180	19583	0.123	-	15867	19544	0.188	393528	-	15848	19656	0.194	380840	-
Average						0.167	263.000			0.125	2822894	111.553			0.127	1687825	129.703

TABLE II: Computational results for SOPLIB instances.

Instance				MILP Solver				CP Solver									
				BM				BM					RM				
Name	Size	$\rho(R)$	Best-Known	LB	UB	Gap	Time [s]	LB	UB	Gap	Branches	Time [s]	LB	UB	Gap	Branches	Time [s]
R.200.100.1	200	0.020	29	29	29	0.000	6.017	29	29	0.000	202206	29.711	29	29	0.000	83586	26.963
R.200.100.15	200	0.847	[589, 979]	525	1033	0.492	-	563	**972**	0.421	2282206	-	550	1035	0.469	2155929	-
R.200.100.30	200	0.957	[838, 1761]	774	1761	0.560	-	843	1714	0.508	140133	-	809	**1708**	0.526	1378303	-
R.200.100.60	200	0.991	[8861, 16197]	8861	16930	0.477	-	8057	15595	0.483	577845	-	7628	**15176**	0.497	108865	-
R.200.1000.1	200	0.020	887	887	887	0.000	15.635	887	887	0.000	83329	29.587	887	887	0.000	79196	33.442
R.200.1000.15	200	0.876	[7231, 12601]	6895	12601	0.453	-	6205	12788	0.515	5742971	-	7033	14254	0.507	161527	-
R.200.1000.30	200	0.958	[10512, 22781]	10512	22781	0.539	-	9959	24172	0.588	3764158	-	9802	**22693**	0.568	123982	-
R.200.1000.60	200	0.989	[12042, 19934]	12042	21993	0.452	-	10122	21435	0.528	114847	-	9910	21377	0.536	8963959	-
R.300.100.1	300	0.013	13	13	13	0.000	35.012	13	13	0.000	68802	45.037	13	13	0.000	161824	54.610
R.300.100.15	300	0.905	[811, 2056]	669	2259	0.704	-	749	**1865**	0.598	113431	-	**813**	2080	0.609	50986	-
R.300.100.30	300	0.970	[1157, 2590]	1102	3163	0.652	-	1132	2785	0.594	81595	-	1130	2931	0.614	74613	-
R.300.100.60	300	0.994	[991, 1865]	949	1954	0.514	-	932	2293	0.594	199836	-	953	2344	0.593	342126	-
R.300.1000.1	300	0.013	715	715	715	0.000	64.683	715	715	0.000	212106	130.421	715	715	0.000	758190	87.442
R.300.1000.15	300	0.905	[8768, 24047]	7832	24047	0.674	-	7544	**20895**	0.639	149209	-	7524	22818	0.670	3176756	-
R.300.1000.30	300	0.965	[12269, 31618]	12071	40863	0.705	-	10706	43534	0.754	5185819	-	10415	41310	0.748	428501	-
R.300.1000.60	300	0.994	[10408, 21623]	10275	25323	0.594	-	8561	25887	0.669	106918	-	8655	**21294**	0.594	684540	-
R.400.100.1	400	0.010	6	6	6	0.000	995.137	6	6	0.000	67605	129.029	6	6	0.000	4181266	76.868
R.400.100.15	400	0.927	[963, 3591]	856	22767	0.962	-	955	2488	0.616	2480098	-	849	**2830**	0.700	125348	-
R.400.100.30	400	0.978	[1084, 3061]	1010	26438	0.962	-	997	**2678**	0.628	89327	-	1001	2839	0.647	1890510	-
R.400.100.60	400	0.996	[966, 2069]	861	2652	0.675	-	666	2135	0.688	139991	-	736	2084	0.647	123574	-
R.400.1000.1	400	0.010	780	780	780	0.000	124.990	780	780	0.000	209665	76.386	780	780	0.000	56921	105.197
R.400.1000.15	400	0.930	[9976, 35160]	9083	85878	0.894	-	7804	**29188**	0.733	2821927	-	7642	34339	0.777	897078	-
R.400.1000.30	400	0.977	[12337, 57272]	11783	127290	0.907	-	9727	**34781**	0.720	120018	-	9927	48659	0.796	5057396	-
R.400.1000.60	400	0.995	[9954, 22376]	9877	36662	0.731	-	7237	24990	0.710	182264	-	7620	22521	0.662	128489	-
R.500.100.1	500	0.008	3	3	3	0.000	1881.297	3	3	0.000	1665985	1184.061	3	3	0.000	2352	66.186
R.500.100.15	500	0.945	[1250, 5508]	1018	11452	0.911	-	1044	**4700**	0.778	1977102	-	907	5370	0.831	137132	-
R.500.100.30	500	0.980	[1099, 4841]	976	14273	0.932	-	808	3687	0.781	120015	-	717	**3326**	0.784	130094	-
R.500.100.60	500	0.996	[931, 2723]	840	6357	0.868	-	560	5309	0.895	2325233		560	**2544**	0.780	4962333	-
R.500.1000.1	500	0.008	297	297	297	0.000	85.459	297	297	0.000	2551213	209.776	297	297	0.000	6327	63.832
R.500.1000.15	500	0.940	[10628, 45356]	9461	107776	0.912	-	8240	**35647**	0.769	149478	-	8445	41479	0.796	1363078	-
R.500.1000.30	500	0.981	[12694, 57330]	12694	156359	0.919	-	9458	**47859**	0.802	110401	-	9995	53765	0.814	1836393	-
R.500.1000.60	500	0.996	[8192, 20465]	8192	45696	0.821	-	6159	21832	0.718	559087	-	6159	22735	0.729	262935	-
R.600.100.1	600	0.007	1	1	55	0.982	-	1	1	0.000	95274	253.537	1	1	0.000	63402	118.1845
R.600.100.15	600	0.950	[938, 2443]	845	4044	0.791	-	549	3942	0.861	1233613		662	**2399**	0.724	2763030	-
R.600.100.30	600	0.985	[1099, 6467]	1099	18932	0.942	-	740	6868	0.892	1649053		789	**6346**	0.876	2955997	-
R.600.100.60	600	0.997	[778, 2494]	778	25214	0.969	-	538	3395	0.842	766227	-	538	**2833**	0.810	7723228	-
R.600.1000.1	600	0.007	322	322	322	0.000	140.645	322	322	0.000	5796029	373.208	322	322	0.000	634	81.753
R.600.1000.15	600	0.945	[10915, 65039]	10915	121877	0.910	-	9401	62114	0.849	1003612		9875	**53937**	0.817	832443	-
R.600.1000.30	600	0.984	[12431, 48775]	12431	190145	0.935	-	9356	73581	0.873	1242031	-	10008	**43929**	0.772	699944	-
R.600.1000.60	600	0.997	[8162, 42652]	8162	75269	0.892	-	6908	44310	0.844	1368338		6908	**40077**	0.828	1640120	-
R.700.100.1	700	0.006	2	-	-	-	-	2	2	0.000	4624	105.444	2	2	0.000	1345	75.207
R.700.100.15	700	0.957	[972, 2759]	972	5718	0.830	-	655	5914	0.889	909974	-	753	2995	0.749	631765	-
R.700.100.30	700	0.987	[983, 2531]	983	4218	0.767	-	588	2531	0.768	335440	-	756	2531	0.701	143114	-
R.700.100.60	700	0.997	[555, 1598]	555	1854	0.701	-	383	1598	0.760	316660	-	383	1598	0.760	236953	-
R.700.1000.1	700	0.006	611	611	616	0.008	-	611	611	0.000	31507	150.460	611	611	0.000	30830	113.285
R.700.1000.15	700	0.956	[5136, 6315]	5136	7145	0.281	-	2787	61078	0.954	365432		4316	6315	0.317	128446	-
R.700.1000.30	700	0.986	[4827, 6115]	4827	6981	0.309	-	2658	6200	0.571	294568	-	3906	6115	0.361	223725	-
R.700.1000.60	700	0.997	[2997, 5357]	2997	5842	0.487	-	1913	**5331**	0.641	284690	-	2022	5379	0.624	176457	-
Average						0.577	372.097			0.531	1047748	226.388			0.505	1211365	75.247

TABLE III: Computational results for COMPILERS instances.

				MILP Solver				CP Solver									
Instance				BM				BM					RM				
Name	Size	$\rho(R)$	Best-Known	LB	UB	Gap	Time [s]	LB	UB	Gap	Branches	Time [s]	LB	UB	Gap	Branches	Time [s]
gsm.153.124	126	0.970	[280, 311]	269	311	0.135	-	276	311	0.113	3919288	-	**283**	311	0.090	345832	-
gsm.444.350	353	0.990	[2456, 4310]	2405	4856	0.505	-	**2482**	4625	0.463	127304	-	2453	**4214**	0.418	207341	-
gsm.462.77	79	0.840	[419, 465]	402	477	0.157	-	419	471	0.110	3145801	-	419	466	0.101	4649930	-
jpeg.1483.25	27	0.484	87	87	87	0.000	18.556	87	87	0.000	1278	0.150	87	87	0.000	966	0.173
jpeg.3184.107	109	0.887	[518, 656]	510	715	0.287	-	**520**	686	0.242	2786312	-	517	**623**	0.170	2220962	-
jpeg.3195.85	87	0.740	[23, 25]	17	25	0.320	-	23	25	0.080	488728	-	22	25	0.120	506891	-
jpeg.3198.93	95	0.752	[181, 188]	180	188	0.043	-	181	188	0.037	2762954	-	181	188	0.037	1116829	-
jpeg.3203.135	137	0.897	[629, 750]	618	751	0.177	-	622	809	0.231	273519	-	629	**709**	0.113	258220	-
jpeg.3740.15	17	0.257	33	33	33	0.000	0.839	33	33	0.000	532	0.058	33	33	0.000	490	0.069
jpeg.4154.36	38	0.633	90	90	90	0.000	60.924	90	90	0.000	5347	1.010	90	90	0.000	7836	1.855
jpeg.4753.54	56	0.769	164	164	164	0.000	1790.269	164	164	0.000	234891	33.849	164	164	0.000	55605	214.247
susan.248.197	199	0.939	[805, 1320]	802	1370	0.415	-	**810**	**1200**	0.325	1205085	-	807	1557	0.482	1065206	-
susan.260.158	160	0.916	[598, 897]	573	938	0.389	-	586	969	0.395	1146515	-	588	934	0.370	1170422	-
susan.343.182	184	0.936	[636, 776]	622	776	0.198	-	641	817	0.215	1478092	-	639	796	0.197	1208246	-
typeset.10192.123	125	0.744	[293, 379]	282	379	0.256	-	**300**	379	0.208	2212422	-	292	393	0.257	425307	-
typeset.10835.26	28	0.349	[110, 111]	100	112	0.107	-	109	111	0.018	925171	-	108	111	0.027	841850	-
typeset.12395.43	45	0.518	146	141	146	0.034	-	146	146	0.000	369091	475.072	146	146	0.000	82439	250.703
typeset.15087.23	25	0.557	97	97	97	0.000	29.118	97	97	0.000	1469	0.228	97	97	0.000	814	0.336
typeset.15577.36	38	0.555	125	125	125	0.000	43.164	125	125	0.000	55895	2.485	125	125	0.000	18202	11.596
typeset.16000.68	70	0.658	[79, 80]	66	80	0.175	-	74	80	0.075	2083457	-	74	80	0.075	1029470	-
typeset.1723.25	27	0.245	60	60	60	0.000	86.068	60	60	0.000	1550	0.260	60	60	0.000	2537	0.362
typeset.19972.246	248	0.993	[1525, 2509]	1452	2509	0.421	-	1514	3001	0.496	591545	-	1515	2692	0.437	1047705	-
typeset.4391.240	242	0.981	[1154, 1905]	1137	2476	0.541	-	**1172**	1379	0.150	1261056	-	1165	**1324**	0.120	147004	-
typeset.4597.45	47	0.493	154	151	154	0.019	-	154	154	0.000	465417	497.584	154	154	0.000	127268	380.478
typeset.4724.433	435	0.995	[2679, 6131]	2673	6131	0.564	-	**2701**	**5738**	0.529	98263	-	2676	6275	0.574	521523	-
typeset.5797.33	35	0.748	113	113	113	0.000	28.504	113	113	0.000	1902	0.600	113	113	0.000	1286	0.752
typeset.5881.246	248	0.986	[1406, 2084]	1396	2426	0.425	-	1426	**2067**	0.310	207231	-	1417	2082	0.319	164597	-
Average						0.191	257.180			0.148	957412	101.129			0.145	637955	86.057

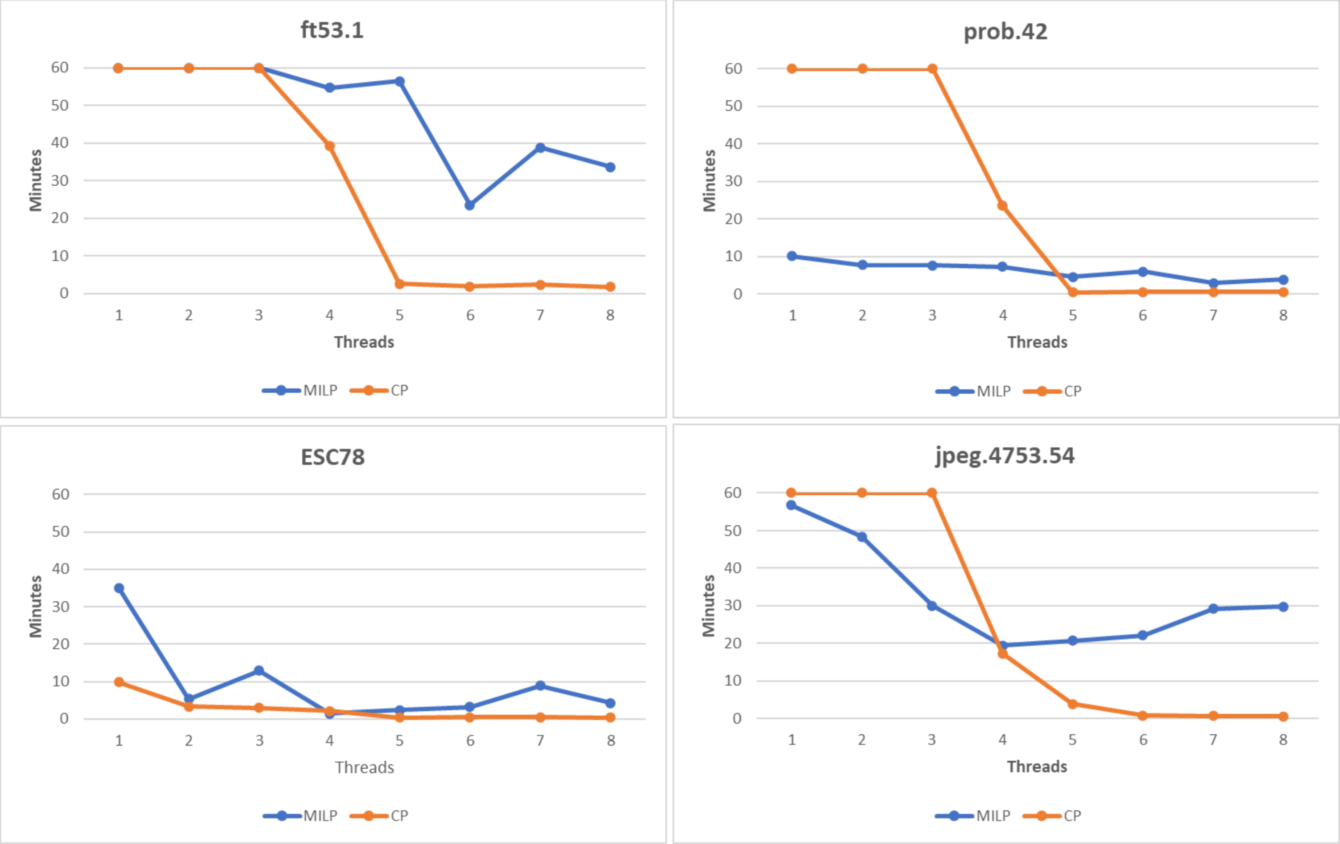


Fig. 3: Time required by the MILP Solver and CP Solver to optimally solve different instances with different number of threads. A time of 60 minutes reported means that the respective solver was not able to optimally solve the instance within the time limit.

between 1 and 8. In the figure, a time of 60 minuets reported means that the respective solver was not able to optimally solve the instance within the time limit of one hour.

The results reported in Figure 3 show that the CP Solver substantially benefits from the use of multi-threading computation. Furthermore, the results show that the CP Solver is not able to optimally solve three out of four instances within the time limit when less than four threads are allocated for the solver. However, when allocating four or more threads, the CP Solver is able to optimally solve those instances. Furthermore, a drastic change in performance can be observed between four and five threads, reaching a speedup up to 93.5%. On the other hand, the MILP Solver does not seem to benefit as much from multi-threading for the instances considered, possibly due to the overhead of task distribution, and the waiting time incurred by the variety of methods run in parallel. Furthermore, we can notice less consistent gain when increasing the number of threads used by the MILP Solver compared to the CP Solver. It should be noted that the differences between the two solvers might be less extreme when more challenging instances are considered, but this is difficult to investigate as most instances are hard to solve optimally, even with longer computational time limit (hours) is allowed, and with eight threads allocated for the solvers.

In conclusion, the CP Solver appears to greatly benefit from multi-threading computation; therefore, all the experiments reported in this section were run on eight cores.

B. *Analysis of the Results*

In this section, we first compare and discuss the results achieved by the MILP and CP Solvers by solving the model *BM*. We then compare and discuss the results achieved by the CP Solver by solving the models *BM* and *RM*.

TABLE IV: Summary of the results achieved by solving the model *BM* with the MILP Solver and CP Solver.

	MILP Solver	CP Solver
Average optimality gap	0.340	0.301
Average solution time	297.6	89.7
New best-known lower bounds	0	9
New best-known upper bounds	0	19
New optimal solution	0	1

Table IV summarizes the results of solving the model *BM* by the MILP Solver and CP Solver, where we report the following. The *Average optimality gap* is computed with respect to all the instances where both solvers find a feasible/optimal solution before reaching the time limit when solving the model *BM*. The *Average solution time* is computed on all the instances that are solved optimally by the both solvers. The *New best-known lower bounds* and *New best-known upper bounds* rows report the number of instances where solving the model by each solver resulted in an improved lower or upper bound. Finally, *New optimal solution* row reports the number of instances where an optimal solution is found for an instance that was previously open, by each solver.

Considering the model *BM*, the MILP Solver achieves an average optimality gap of 0.340 across all the instances, but fails to solve a single instance (marked bold in the table) as it runs out of memory while solving the linear relaxation of the model. On the other hand, the CP Solver achieves an average optimality gap of 0.301 (a 11.5% improvement) when excluding the instance that is not solved by the MILP Solver, and an average optimality gap of 0.298 (a 12.4% improvement) across all the instances. By further inspecting the results, we notice that the CP Solver achieves a smaller average optimality gap within the time limit for instances with density less than 0.85 and size smaller than 400.

For a total of 27 instances that are optimally solved by both solvers, the MILP Solver has an average solution time of 297.6 seconds, while the CP Solver has an average solution time of 89.7 seconds (a 69.9% improvement). We should note that the CP Solver generally finds the optimal solution in less time compared to the MILP Solver on small to medium sized instances.

Finally, out of a total of 81 open instances, the CP Solver is able to find an improved lower bound for 9 instances (11.1%), an improved upper bound for 19 instances (23.5%), and finds the optimal solution of one instance that was previously open. On the other hand, the MILP Solver is not able to improve the best-known solution of any instance. Based on the experiments performed on the model *BM* presented in Section II, we can conclude that the CP Solver has an overall better performance at solving the given MILP model.

TABLE V: Summary of the results achieved by solving each model with the CP Solver.

	BM	RM
Average optimality gap	0.298	0.287
Average solution time	146.9	99.4
New best-known lower bounds	9	4
New best-known upper bounds	19	27
New optimal solution	1	1

The rest of this section discusses the results achieved by the CP Solver by solving the two models *BM* and *RM*. The results are summarized in Table V where we report the following. The *average optimality gap* reports the Average optimality gap of all the instances where the solver finds a feasible/optimal solution before reaching the time limit by solving both models. The *Average solution time* reports the average solution time in seconds of all the instances that are solved optimally by the solver when solving both models. The *New best-known lower bounds* and *New best-known upper bounds* rows report the number of instances where solving each model resulted in an improved lower/upper bound. Finally, *New optimal solution* report the number of instances where an optimal solution is found for an instance that was previously open.

In terms of achieved average optimality gap, the CP Solver achieves an average optimality gap of 0.287 (a 3.7% improvement) when solving the model *RM*, compared to solving the model *BM*. Furthermore, for a total of 36 instances that are solved optimally when solving both models, the CP Solver generates 57.9% less branches in the search-decision tree when solving the model *RM* compared to solving the model *BM*. By further inspecting the results, we notice that the CP Solver achieves a smaller average optimality gap within the time limit when solving the model *RM* for instances with density less than 0.89 and size less than 500, which means that the CP Solver performs better on a larger subset of the instances compared to solving the model *BM*.

For a total of 36 instances that are solved optimaly by the CP Solver when solving both models, the CP Solver has an average solution time of 146.9 seconds when solving the model *BM*, and an average solution time of 99.4 seconds (a 32.4% improvement) when solving the model *RM*.

Finally, out of a total of 81 open instances, when solving the model *BM* the CP Solver finds an improved lower bound for 9 instances (11.1%), an improved upper bound for 19 instances (23.5%), and finds the optimal solution for one instances that was previously open. On the other hand, when solving the model *RM* the CP Solver finds an improved lower bound for 4 instances (4.9%), an improved upper bound for 27 instances (33.3%), and finds the optimal solution for the same instance that was previously open. Based on the computational experiments and the improvements in the results achieved by the CP Solver when solving the model *RM*, we can conclude that duplicating the constraints can indeed improve the performance of the solver for the given model.

V. Conclusions

The computational experiments has shown that the CP Solver outperforms the MILP Solver at solving instances with sizes up to 500 with precedence relationships density that is less than 0.89. Furthermore, the CP Solver achieves a smaller average optimality gap and solution time compared to the MILP Solver. By adding constraint programming constructs to the MILP model, we were able to further exploit the capabilities of the CP Solver, and improve its performance at solving the instances. In terms of solution quality, and out of a total of 81 open instances, the CP Solver was able to find the optimal solution to an instance that was previously open, provide new best-known lower bounds for 13 instances, and establish new best-known solution for 46 instances. Based on the computational experiments performed, we have shown that the CP Solver performs better on average for the given models. Furthermore, duplicating constraints by defining them in their linear form and logical form further pushes the performance of the CP Solver. Future work will consider investigating new valid constraints/inequalities for the PCMCA-WT that can be used within a constraint programming paradigm to further utilize its potential.

REFERENCES

[1] J. Edmonds, "Optimum branchings," *Journal of Research of the National Bureau of Standards*, vol. B 71, no. 4, pp. 233–240, 1967.

[2] Y. J. Chu and T. Liu, "On the shortest arborescence of a directed graph," *Scientia Sinica*, vol. 14, pp. 1396–1400, 1965.

[3] H. N. Gabow, Z. Galil, T. Spencer, and R. E. Tarjan, "Efficient algorithms for finding minimum spanning trees in undirected and directed graphs," *Combinatorica*, vol. 6, no. 2, pp. 109–122, 1986.

[4] M. Fischetti and D. Vigo, "A branch-and-cut algorithm for the resource-constrained minimum-weight arborescence problem," *Networks: An International Journal*, vol. 29, no. 1, pp. 55–67, 1997.

[5] L. Gouveia and M. J. Lopes, "The capacitated minimum spanning tree problem: On improved multistar constraints," *European Journal of Operational Research*, vol. 160, no. 1, pp. 47–62, 2005.

[6] G. Fertin, J. Fradin, and G. Jean, "Algorithmic aspects of the maximum colorful arborescence problem," in *Theory and Applications of Models of Computation. TAMC 2017*. Springer, 2017, pp. 216–230.

[7] N. Guttmann-Beck and R. Hassin, "On two restricted ancestors tree problems," *Information processing letters*, vol. 110, no. 14-15, pp. 570–575, 2010.

[8] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems: Algorithms and applications," *European journal of operational research*, vol. 119, no. 3, pp. 557–581, 1999.

[9] F. Rossi, P. Van Beek, and T. Walsh, "Constraint programming," *Foundations of Artificial Intelligence*, vol. 3, pp. 181–211, 2008.

[10] H. Öztop, "A constraint programming model for the open vehicle routing problem with heterogeneous vehicle fleet," in *Towards Industry 5.0: Selected Papers from ISPR2022, October 6–8, 2022, Antalya*. Springer, 2023, pp. 345–356.

[11] G. Kasapidis, S. Dauzère-Pérèz, D. Paraskevopoulos, P. Repoussis, and C. Tarantilis, "On the multi-resource flexible job-shop scheduling problem with arbitrary precedence graphs," *Production and Operations Management*, 2023.

[12] E. Kirac, R. Gedik, and F. Oztanriseven, "Solving the team orienteering problem with time windows and mandatory visits using a constraint programming approach," *International Journal of Operational Research*, vol. 46, no. 1, pp. 20–42, 2023.

[13] D. Kizilay, Z. A. Çil, H. Öztop, and İ. Bağcı, "A novel mathematical model for mixed-blocking permutation flow shop scheduling problem with batch delivery," in *Towards Industry 5.0: Selected Papers from ISPR2022, October 6–8, 2022, Antalya*. Springer, 2023, pp. 453–461.

[14] R. Montemanni and M. Dell'Amico, "Solving the parallel drone scheduling traveling salesman problem via constraint programming," *Algorithms*, vol. 16, no. 1, p. 40, 2023.

[15] X. Chou, M. Dell'Amico, J. Jamal, and R. Montemanni, "Precedence-constrained arborescences," *European Journal of Operational Research*, vol. 307, no. 2, pp. 575–589, 2022.

[16] M. Dell'Amico, J. Jamal, and R. Montemanni, "A mixed integer linear program for a precedence-constrained minimum-cost arborescence problem," *In Proc. The 8^{th} International Conference on Industrial Engineering and Applications (Europe)*, pp. 216–221, 2021.

[17] M. Dell'Amico, J. Jamal, and R. Montemanni, "Compact models for the precedence-constrained minimum-cost arborescence problem," in *Advances in Intelligent Traffic and Transportation Systems*. IOS Press, 2023, pp. 112–126.

[18] M. Dell'Amico, J. Jamal, and R. Montemanni, "Compact models for the precedence-constrained minimum-cost arborescence problem with waiting-times," *Annals of Operations Research. Submitted*, 2023.

[19] N. Kamiyama, "Arborescence problems in directed graphs: Theorems and algorithms," *Interdisciplinary information sciences*, vol. 20, no. 1, pp. 51–70, 2014.

[20] Google, "Google OR-Tools," 2015, [last accessed 7-March-2023]. [Online]. Available: https://developers.google.com/optimization

[21] L. Perron, "CP-SAT over CBC for MIP, is it worthwhile?" 2020, [last accessed 7-March-2023]. [Online]. Available: https://or.stackexchange.com/questions/4119/cp-sat-over-cbc-for-mip-is-it-worthwhile

[22] R. Montemanni, G. H. Carraretto, U. J. Mele, and L. M. Gambardella, "A constraint programming model for the b-coloring problem," *International Conference on Industrial Engineering and Applications, to appear*, 2023.

[23] R. Montemanni and M. Dell'Amico, "Solving the parallel drone scheduling traveling salesman problem via constraint programming," *Algorithms*, vol. 16, no. 1, p. 40, 2023.

[24] G. Reinelt, "TSPLIB–A travelling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991.

[25] R. Montemanni, D. H. Smith, and L. M. Gambardella, "A heuristic manipulation technique for the sequential ordering problem," *Computers & Operations Research*, vol. 35, no. 12, pp. 3931–3944, 2008.

[26] R. Montemanni, D. H. Smith, A. E. Rizzoli, and L. M. Gambardella, "Sequential ordering problems for crane scheduling in port terminals," *International Journal of Simulation and Process Modelling*, vol. 5, no. 4, pp. 348–361, 2009.

[27] G. Shobaki and J. Jamal, "An exact algorithm for the sequential ordeing problem and its application to switching energy minimization in compilers," *Computational Optimizations and Applications*, vol. 61, no. 2, pp. 343–372, 2015.

[28] N. Ascheuer, N. Jünger, and G. Reinelt, "A branch & cut algorithm for the asymmetric traveling salesman problem with precedence constraints," *Computational Optimization and Applications*, vol. 17, no. 1, pp. 61–84, 2000.

[29] V. Papapanagiotou, J. Jamal, R. Montemanni, G. Shobaki, and L. M. Gambardella, "A comparison of two exact algorithms for the sequential ordering problem," in *IEEE Conference on Systems Process and Control (ICSPC)*, 2015.

[30] J. Jamal, G. Shobaki, V. Papapanagiotou, L. M. Gambardella, and R. Montemanni, "Solving the sequential ordering problem using branch and bound," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.

[31] IBM, "IBM CPLEX Optimizer," 1988, [last accessed 7-March-2023]. [Online]. Available: https://www.ibm.com/de-de/analytics/cplex-optimizer

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 431–440

DOI: 10.15439/2023F5436
ISSN 2300-5963 ACSIS, Vol. 35

Optimum Large Sensor Data Filtering, Networking and Computing

Joanna Berlińska
0000-0003-2120-2595
Adam Mickiewicz University,
Poznan, Poland
Joanna.Berlinska@amu.edu.pl

Maciej Drozdowski
0000-0001-9314-529X
Poznan University of Technology
Poznan, Poland
Maciej.Drozdowski@put.poznan.pl

Thomas Robertazzi
0000-0002-2382-1843
Stony Brook University
Stony Brook NY. USA
Thomas.Robertazzi@stonybrook.edu

Abstract—**In this paper we consider filtering and processing large data streams in intelligent data acquisition systems. It is assumed that raw data arrives in discrete events from a single expensive sensor. Not all raw data, however, comprises records of interesting events and hence some part of the input must be filtered out. The intensity of filtering is an important design choice because it determines the complexity of filtering hardware and software and the amount of data that must be transferred to the following processing stages for further analysis. This, in turn, dictates needs for the following stages communication and computational capacity. In this paper we analyze the optimum intensity of filtering and its relationship with the capacity of the following processing stages. A set of generic filtering intensity, data transfer, and processing archetypes are modeled and evaluated.**

I. Introduction

SENSOR technology has developed rapidly over the past decades. Numerous surveys can be found, each limited to a specific sensor technology type. An important research topic is the computational and communication integration of multiple sensors in a network [1], [2], [3], [4], [5]. There is an implicit assumption here that the economics of constructing, deploying and operating such a system are such that the use of multiple sensors is plausible within project budget constraints.

In this paper we envision a problem with a different emphasis where there is a single, very expensive “sensor”. Examples of such a device include detectors for particle accelerator experiments such as the CERN Large Hadron Collider [6], or the recently announced Electron Ion Collider [7] to be built at Brookhaven National Laboratory. Another example is the use of a large radar equipped drone for monitoring ocean traffic. A final example is the use of imaging satellites for ocean, weather, environmental monitoring and earth resources sensing [8]. Note that what we generically consider a “sensor", may consist of a large number of individual sensing elements (as in a particle detector) but we refer to the ensemble collection as a sensor.

The commonality in all these examples is that the sensor increasingly can generate raw data at rates that are much faster than the data can be downloaded over a communication channel(s) onto servers for processing. Moreover, not all data collected by the sensor is valuable enough to be retained for further processing. The generally proposed solution for this problem is to have onboard pre-processing of the data at the detector, the drone or the satellite. This processing is not classical data compression but more so the use of data analysis, machine learning (ML) or other type of broadly understood artificial intelligence techniques to pre-process the raw data for a much smaller processed summary that is more amenable to transmit. Thus, only interesting particle tracks, ship tracks and weather patterns may be transmitted, but the majority of the raw data isn’t. For simplicity of exposition we will say that the ML algorithm serves as a filter on the raw data, independently of the field of origin of the actually applied filtering technique. Clearly this situation has similarities with edge computing where sensors/actuators at the network edge do local processing in order to reduce the amount of network traffic to distant cloud facilities. Again, the specification of a single sensor here makes our problem have a unique character.

In this paper we analyze the intensity of data filtering in the first stages of data processing necessary for the shortest time to obtain results. The intensity of filtering is important for the following reasons: (i) filtering algorithms are often hardware-implemented, and consequently, have costly realization, their changes are less flexible than in software (especially in remote posts like satellites); (ii) the higher intensity of filtering, the more complex algorithm is applied which results in longer filtering time and/or more extensive hardware system; (iii) the size of data emerging from the filtering stage determines needs for capacity in the further stages of data processing pipeline where more sophisticated algorithms are executed; (iv) and vice versa the speed of communication between the stages of data pipeline and processing in the stages following the initial filtering have impact on the required intensity of raw data filtering. For the purpose of analyzing systems of the above nature, an extensive set of the single expensive sensor (SES) models for filtering and processing problem will be examined. These cases are meant to illustrate the modeling and solution possibilities and be representative in a generic way. A common sense expectation is that high intensity of initial data filtering reduces amounts of the data analyzed in the pipeline and thus reduces the time to obtaining the results. However, more intensive filtering is also more time-consuming. Thus, due to interaction between nonlinear speed of filtering and complexity of processing the data in

TABLE I
SUMMARY OF NOTATIONS

α_i	size of load part assigned to stage 2 processor i [byte]
A_0	reciprocal of the first stage processing speed [e.g. s/byte]
A_i	reciprocal of the heterogeneous second stage processing speed on processor i [e.g. s/byte]
A	reciprocal of the second stage processing speed for identical processors [e.g. s/byte]
C_i	reciprocal of the communication speed between stage 1 and heterogeneous stage 2 processor i
C	reciprocal of the communication speed between stage 1 and stage 2 for identical processors [e.g. s/byte]
F	fraction of retained data
m	number of machines (a.k.a. servers, processors) in the second stage
T	execution time of the whole filtering and processing workflow
V	size of input data (at the front-end of the system)

the later stages, the processing time may have a minimum at a certain filtering intensity. We investigate such minima in this paper. Furthermore, options for combining advanced processing algorithms of various complexity classes in the data processing workflow are analyzed.

Our models are largely tractable. Parts of the evaluation of the models use concepts from the theory of divisible (i.e. partitionable) loads, a well established concept, that provides elegant solutions particularly for linear models [9], [10], [11], [12], [13]. The divisibility of the loads means that big volumes of data are processed and the discrete units of data are small in relation to the whole data size. It is also assumed that the loads can be divided into parts processed independently in parallel.

Further organization of this paper is the following. In the next section related literature is outlined. The filtering and parallel processing problem is formally defined in Section III. Section IV is dedicated to analytical derivation of the formulas guiding selection of the optimum filtering intensity. Results of numerical modeling of filtering and processing systems are provided in Section V. The last section is dedicated to conclusions. The notations are summarized in Table I.

II. Related Work

In the literature on sensor networks, particularly wireless sensor networks, there are general surveys [14] and surveys on specific technological aspects of sensor networks such as transport and routing [15], fault detection [16], security solutions [17], optimizing sensor-source geometries and minimizing the number of sensors [18], the use of swarm intelligence for performance optimization [19] and numerous applications.

There has been some work on analytical models of sensor data generation, communication and computation. For instance, an early work is [1] which examined scheduling for measurement and data reporting in wireless sensor networks. Data gathering networks have been the subject of research by Berlińska and recently by Luo et. al. Data gathering networks have been studied in connection with background communication [3], limited base station memory [4], data compression [5], [20], [21], energy minimization [22] and in the case of tree data gathering networks [2]. The case when the load is processed in a pipeline fashion has been studied in [23], [24].

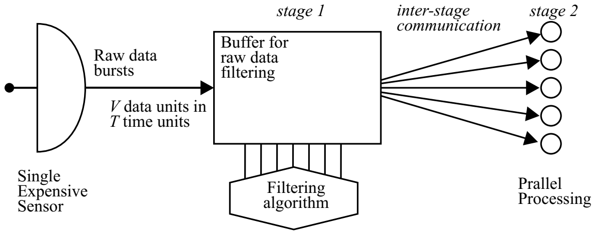


Fig. 1. Data filtering and processing system architecture.

Most work to date has involved multiple sensors, unlike the single expensive sensor paradigm of this paper. An LHC data acquisition system [6] is a good example of a single expensive sensor with data filtering and parallel processing. In LHC protons circulate in bunches and opposing beams that cross each other resulting in collisions with 40MHz frequency. Only data from particle collisions with sufficient energy and momentum are allowed to proceed from the so-called level-1 trigger to the second stage of processing (so called high-level trigger) for further reconstruction of particle trajectories and analysis.

III. Problem formulation

It is assumed that there is a two-stage workflow: (1) the first stage is related to sensor data filtering, (2) the second stage conducts further data processing. An overview of the system architecture is shown in Fig.1. The data from the sensor arrive in discrete events each delivering V units of data. The arriving chunk of data is intercepted in the input buffer, and all filtering algorithms use this buffer. The same buffer is used to send the filtered data to the second stage of processing. The use of a single buffer in this model is an aggregate representation of specialized buffer architectures that may be needed in practice to handle the influx, staging and filtering of massive amounts of data. The data chunks may arrive repetitively, then it is assumed that at most V units of data arrive in a chunk once in T time units. The first stage filtering is done in linear time, but the intensity of filtering, i.e. fraction F of the initial data transferred to the second stage, is related to the speed of filtering.

In the second stage more intricate, than filtering, data processing is conducted requiring machines with substantial computing power, memory and storage. Hence, these can be dedicated data centers or cloud systems. The machines running in the second stage will be referred to as servers or processors. Many different algorithms may be executed in parallel in the second stage. For example, different algorithms discovering unrelated artifacts may be run in parallel. In such a case it is assumed that each specific data-processing algorithm receives the same data set, has its own set of processors, and all the many algorithms are executed independently of each other. The longest path in the data-processing would always go to the processor(s) running the most time-consuming algorithm. Consequently, in the following we analyze only one of the possibly many parallel paths in data-processing. Namely, the longest one.

A. First Stage Filtering Intensity and Complexity

The intensity of filtering is expressed by fraction $F \in (0, 1]$ controlling the amount of produced results. The amount of results delivered from stage 1 to stage 2 is FV, where V is the size of data injected into the first stage from the sensor. This volume of data is filtered in time A_0V. It is assumed that the intensity of filtering F and speed of filtering are interdependent. Precisely, the inverse of filtering speed is some function $A_0(F)$. Depending on the application, $A_0(F)$ may assume various forms. Here we list a few possibilities:

Case 1: $\mathbf{A_0(F) = 1/F^c}$, where $c > 0$ is some constant. This kind of relationship may emerge as a consequence of iterative filtering. Let i be the number of iterations that are executed on each data unit, then A_0 and F can be expressed as

$$F = f^i \tag{1}$$

$$A_0 = a^i, \tag{2}$$

where $f \in (0, 1)$ is the fraction of data remaining after each iteration of filtering, and f and $a > 1$ are some given constants determined by the filtering algorithm. This means that an iterative filtering algorithm is executed on *each* data unit, and extending the algorithm by each new iteration takes exponentially longer to process a data unit. This can be the case when each data unit (e.g. a picture) is rectified with increasing resolution. From equation (1), we get $i = \frac{\ln F}{\ln f}$. From (2), $\ln A_0 = i \ln a$, and hence, $\ln A_0 \ln f = \ln F \ln a$. Equivalently, we have $\ln A_0 = \ln F \ln a/\ln f$, and hence, $A_0 = F^{\frac{\ln a}{\ln f}}$. Since $f \in (0, 1), a > 1$, we have $\frac{\ln a}{\ln f} < 0$ and $A_0(F) = \frac{1}{F^c}$, where $c = -\frac{\ln a}{\ln f} > 0$.

Case 2: $\mathbf{A_0(F) = c\ln(1/F)}$. Again all filtering iterations are executed on each data unit, each iteration takes the same time and reduces output data size $f \in (0, 1)$ times. Then as in the previous case $F = f^i$, $i = \frac{\ln F}{\ln f}$, but since each iteration takes the same time, we have $A_0 = ai$, and hence we obtain $A_0(F) = c\ln(1/F)$, where $c = -\frac{a}{\ln f} > 0$.

Case 3: $\mathbf{A_0(F) = c(1 - \ln F)}$. Suppose the filtering is a *sieve*, i.e., the filtering algorithm sifts data in some buffer and with each iteration part of the data is dropped. Suppose $\frac{1}{j}$-th of the initial data is removed in iteration j. Thus, after i iterations $V \times \frac{1}{2} \times \frac{2}{3} \times \cdots \times \frac{i-1}{i} = V/i = FV$ data units remain. Hence, $F = 1/i$. The filtering time T_1 is proportional to the diminishing data sizes: $T_1 = aV(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{i}) = aH_iV = A_0V$, where H_i is the ith harmonic number. Since $H_i \le 1 + \ln i$, $A_0(F) = aH_i \le a(1 + \ln 1/F) = c(1 - \ln F)$, where $a = c$ is some constant.

Case 4: $\mathbf{A_0(F) = c(1 - F)}$. The data filtering is a sieve again, and after i iterations the remaining data size is $FV = f^iV$ where $f \in (0, 1)$. Filtering time is $T_1 = A_0V = aV\sum_{j=1}^{i} f^i = aV\frac{1-f^i}{1-f}$. Hence, $F = f^i$, $A_0 = a\frac{1-f^i}{1-f}$. Consequently, $A_0(F) = \frac{a(1-F)}{1-f} = c(1 - F)$ where $\frac{a}{1-f} = c$ is constant.

Let us observe that in all the above cases $A_0(F)$ decreases with F which means that more intensive filtering (F decreases) requires longer computation ($A_0(F)$ increases).

B. Inter-Stage Communication

The flow of results from stage 1 to stage 2 can be organized in a number of ways. Here we assume two alternatives:

A. Sequential communication – time to transfer x_i bytes to server i and x_j bytes to server j is $C_ix_i + C_jx_j$.

B. Parallel communication – stage 1 to server connections are mutually independent. Time to transfer x_i bytes from stage 1 to server i and x_j bytes to server j is equal to $\max\{C_ix_i, C_jx_j\}$.

C. Computational Complexity of the Second Stage

Computational complexity of the second stage depends on the executed algorithms, necessary result integration and storage. We adopt divisible load theory assumption [10], [12], [13] that the data processed in the second stage is arbitrarily divisible and can be processed in parallel. Potential ways of parallelization are determined by a particular algorithm. Since the variety of possible second stage algorithms and ways of parallelizing them seems unlimited, we will consider a limited set of archetype algorithms as examples of typical computational complexity functions:

Linear – The time to process x units of data is A_ix on processor i. Typical examples are searching for patterns, compression, message digest (e.g. MD5) calculation or scoring data units (like in the LHC example). Here we assume that result collection time is negligible because the size of output data is small or the results are left on the servers and storing is included in the algorithm run time. Since result return is neglected, it can be shown that in the optimum schedule all servers must finish computations at the same moment [10], [12], [13].

Loglinear – Sorting is a typical example of an algorithm with loglinear complexity. Classic sequential sorting algorithms like heapsort, quicksort have complexity $O(n \log n)$, where n is the number of sorted items. We will assume that a parallel version of these algorithms consists in splitting the volume of data into parts, sorting the parts in parallel, and then sequentially merging the results. Thus, on m identical processors the complexity of this method would be $O((n/m)\log(n/m) + n \log m)$.

Quadratic - Computing a similarity matrix can serve as an example of an algorithm with quadratic computational complexity. Its sequential complexity is $O(n^2)$. In the case of parallel processing, the square area of work, e.g. a similarity matrix, may be partitioned into $\ell \times \ell$ squares distributed to processors, where integer ℓ is a tunable parameter of the algorithm. There are ℓ^2 tiles each of of size n^2/ℓ^2. Sending and processing one square takes $O(n^2/\ell^2)$ time. If equal numbers of ℓ^2/m tiles are assigned to each of m processors, then receiving and processing them can be executed in time $O(n^2/\ell^2 \times \ell^2/m)$ which is $O(n^2/m)$. Note that tile distribution may be different, that is, it may depend on the communication and computing speeds of the processors.

IV. Optimum Filtering Intensity

Our goal in this section is to derive close-form solutions (i.e. formulas) linking optimum filtering intensity F with other system parameters to minimize the time required to transmit and process all data. In many cases the obtained formulations are not amenable to analytic solutions. In such a situation further study is delegated to numerical modeling described in the next section.

A. Parallel Communication, Linear Second Stage

In the optimum schedule, all servers communicate and process in parallel, finishing at the same time T. Hence, we have

$$\alpha_1(A_1+C_1) = \alpha_i(A_i+C_i) \qquad i=2,\dots,m \quad (3)$$

$$FV = \sum_{i=1}^{m}\alpha_i \quad (4)$$

In the above equation system all processors communicate and compute in the same interval by (3), and all work is done by (4). From (3) we get $\alpha_i = (A_1+C_1)/(A_i+C_i)\alpha_1$ and

$$FV = (A_1+C_1)\sum_{i=1}^{m}\frac{\alpha_1}{A_i+C_i} = K(A_1+C_1)\alpha_1, \quad (5)$$

where

$$K = \sum_{i=1}^{m}\frac{1}{A_i+C_i} \quad (6)$$

is a constant. The schedule length is a sum of filtering, communication and processing times:

$$T = A_0(F)V + (A_1+C_1)\alpha_1 = A_0(F)V + FV/K. \quad (7)$$

Thus, in order to minimize T, we have to minimize the function

$$t(F) = A_0(F) + F/K. \quad (8)$$

We will now compute the optimum value of F for the considered functions $A_0(F)$. Note that practical values of F belong to some interval $[F_{min}, F_{max}] \subset (0,1]$. Thus, if the computed optimum value F^* is larger than F_{max}, we should set $F = F_{max}$. Similarly, if $F^* < F_{min}$, then the smallest possible value of F should be chosen.

1) $\mathbf{A_0(F) = 1/F^c}$, where $c > 0$ is a constant. Then,

$$t(F) = F^{-c} + F/K, \quad (9)$$

$$t'(F) = -cF^{-(c+1)} + 1/K \quad (10)$$

and

$$t''(F) = c(c+1)F^{-(c+2)} > 0. \quad (11)$$

Thus, $t(F)$ is minimized when $t'(F) = 0$, i.e., for

$$cF^{-(c+1)} = 1/K, \quad (12)$$

$$F = (Kc)^{1/(c+1)}. \quad (13)$$

2) $\mathbf{A_0(F) = c\ln(1/F)}$, where $c > 0$ is a constant. We have

$$t(F) = -c\ln F + F/K, \quad (14)$$

$$t'(F) = -c/F + 1/K \quad (15)$$

and

$$t''(F) = c/F^2 > 0. \quad (16)$$

Hence, $t(F)$ is minimized when

$$F = cK. \quad (17)$$

3) $\mathbf{A_0(F) = c(1 - \ln F)}$, where $c > 0$ is a constant. Then,

$$t(F) = c(1 - \ln F) + F/K, \quad (18)$$

$$t'(F) = -c/F + 1/K \quad (19)$$

and

$$t''(F) = c/F^2 > 0. \quad (20)$$

The minimum value of $t(F)$ is obtained for

$$F = cK. \quad (21)$$

4) $\mathbf{A_0(F) = c(1 - F)}$, where $c > 0$ is a constant. Now we have

$$t(F) = c(1-F) + F/K, \quad (22)$$

$$t'(F) = -c + 1/K. \quad (23)$$

Thus, $t'(F)$ does not depend on F. If $c > 1/K$, then $t(F)$ is decreasing, and the maximum possible value of F should be chosen. If $c < 1/K$, then $t(F)$ is increasing and the smallest possible F should be selected.

B. Sequential Communication, Linear Second Stage

We assume that the sensor communicates with the processors in the order of their identifiers. Hence, we have

$$\alpha_i(A_i+C_i) = \alpha_{i-1}A_{i-1} \qquad i=2,\dots,m \quad (24)$$

$$FV = \sum_{i=1}^{m}\alpha_i \quad (25)$$

Equations (24) mean that communication to and computation on processor i is preformed in parallel with computation on processor $i-i$. It follows implicitly that processor i is started after activating processor $i-1$. Hence, we obtain

$$\alpha_i = \frac{\alpha_1\prod_{j=1}^{i-1}A_j}{\prod_{j=2}^{i}(A_j+C_j)} \qquad i=2,\dots,m, \quad (26)$$

$$FV = \alpha_1\sum_{i=1}^{m}\frac{\prod_{j=1}^{i-1}A_j}{\prod_{j=2}^{i}(A_j+C_j)} \quad (27)$$

and

$$T = A_0(F)V + (A_1+C_1)\alpha_1 = A_0(F)V + FV/L, \quad (28)$$

where

$$L=\sum_{i=1}^{m}\frac{\prod_{j=1}^{i-1}A_j}{\prod_{j=1}^{i}(A_j+C_j)}. \tag{29}$$

Equation (28) has the same form as (7), and hence, the considerations from Section IV-A can be also applied in the case of sequential communication.

C. Parallel Communication, Loglinear Second Stage

In the case of loglinear complexity of the second stage, it is assumed that processing consists of three steps: parallel communication, parallel processing chunks of data, and sequential merging of the results. The latter can be executed in time $FVC_M \log m$, where FV is the amount of data that has to be collected, C_M is reciprocal of merging speed (e.g. in sec/byte) which is taking into account the speed of communication between the servers providing data to merge and the merging server, $\log m$ is a factor representing time to elect the smallest value among m servers in the merging step. The former two steps (parallel communication and processing) take the same time T_1 on all servers. Hence we have:

$$T_1 = C_i\alpha_i + A_i\alpha_i \ln\alpha_i \qquad i=1,\dots,m. \tag{30}$$

Let us define

$$y_i = \frac{C_i}{A_i} + \ln\alpha_i. \tag{31}$$

We have

$$\begin{aligned} y_i e^{y_i} &= \left(\frac{C_i}{A_i} + \ln\alpha_i\right) e^{\frac{C_i}{A_i}}\alpha_i = \\ &= \frac{1}{A_i}\left(C_i\alpha_i + A_i\alpha_i \ln\alpha_i\right) e^{\frac{C_i}{A_i}} = \frac{T_1}{A_i} e^{\frac{C_i}{A_i}}. \end{aligned} \tag{32}$$

Recall that if $ye^y = x$ then $y = W(x)$, where W is the Lambert function [25]. Thus, we have

$$y_i = W\left(\frac{T_1}{A_i} e^{\frac{C_i}{A_i}}\right), \tag{33}$$

and (30) can be written as

$$T_1 = A_i\alpha_i W\left(\frac{T_1}{A_i} e^{\frac{C_i}{A_i}}\right). \tag{34}$$

Hence, the load chunk sizes are:

$$\alpha_i = \frac{T_1}{A_i W\left(\frac{T_1}{A_i} e^{\frac{C_i}{A_i}}\right)}. \tag{35}$$

The Lambert function cannot be expressed in terms of elementary functions, but T_1 can be found numerically by solving

$$FV = \sum_{i=1}^{m} \frac{T_1}{A_i W\left(\frac{T_1}{A_i} e^{\frac{C_i}{A_i}}\right)}. \tag{36}$$

The above equation is easier to solve in homogeneous systems because all processors have the same parameters and load to process is split equally. Then, each processor receives load of size $\alpha_i = \frac{FV}{m}$ and equation (36) becomes:

$$FV = \frac{mT_1}{AW\left(\frac{T_1}{A} e^{\frac{C}{A}}\right)}. \tag{37}$$

Moreover, we get from (33)

$$\begin{aligned} W\left(\frac{T_1}{A} e^{\frac{C}{A}}\right) &= \frac{C}{A} + \ln\alpha_i = \frac{C}{A} + \ln\left(\frac{FV}{m}\right) = \\ &= \ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right) \end{aligned} \tag{38}$$

and hence,

$$T_1 = \frac{AFV}{m} \ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right). \tag{39}$$

The schedule length including filtering, communication and processing is

$$T(F) = A_0(F)V + \frac{AFV}{m} \ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right) + FVC_M \ln m \tag{40}$$

We will now compute the optimum value of F for which T is minimum.

1) $\mathbf{A_0(F) = 1/F^c}$, where $c > 0$ is a constant. Then,

$$\begin{aligned} T'(F) = -cVF^{-(c+1)} + \frac{AV}{m}\left[\ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right) + 1\right] + \\ + VC_M \ln m \end{aligned} \tag{41}$$

$$T''(F) = c(c+1)VF^{-(c+2)} + \frac{AV}{mF} > 0 \tag{42}$$

Hence, $T(F)$ is a minimum when $T'(F) = 0$.

2) $\mathbf{A_0(F) = c\ln(1/F)}$, where $c > 0$ is a constant. We have

$$\begin{aligned} T'(F) = -cV/F + + \frac{AV}{m}\left[\ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right) + 1\right] + \\ + VC_M \ln m \end{aligned} \tag{43}$$

and

$$T''(F) = cV/F^2 + \frac{AV}{mF} > 0 \tag{44}$$

Again, for F satisfying $T'(F) = 0$, $T(F)$ is minimum.

3) $\mathbf{A_0(F) = c(1 - \ln F)}$, where $c > 0$ is a constant, is dealt in the same way as the previous case because $[c\ln(1/F)]' = c(1 - \ln F)' = -c/F$.

4) $\mathbf{A_0(F) = c(1 - F)}$, where $c > 0$ is a constant.

$$T'(F) = -cV + \frac{AV}{m}\left[\ln\left(\frac{FVe^{\frac{C}{A}}}{m}\right) + 1\right] + VC_M \ln m \tag{45}$$

and

$$T''(F) = \frac{AV}{mF} > 0 \tag{46}$$

and $T(F)$ is a minimum when $T'(F) = 0$.

D. Sequential Communication, Loglinear Second Stage

In this case communications and parts of processed load are linked by the system of equations:

$$C_{i+1}\alpha_{i+1} + A_{i+1}\alpha_{i+1}\ln\alpha_{i+1} = A_i\alpha_i\ln\alpha_i \qquad i = 1,\dots,m-1 \tag{47}$$

$$FV = \sum_{i=1}^{m}\alpha_i \tag{48}$$

Equations (47) ensure that work on processor $i-1$ is processed in parallel with communication to and computation on processor i. This set of nonlinear equations does not seem to have an easy analytical solution. Therefore, we will recourse to numerical methods to solve (47)-(48) and find F for which the processing time is minimum.

E. Parallel Communication, Quadratic Second Stage

As mentioned in Section III we assume that the quadratic amount of work is shared between the m processors. This amount of work can be split into ℓ^2 work units, each of size $(FV/\ell)^2$. We will assume that ℓ is large and hence work can be sufficiently flexibly divided as in the linear case. Yet, mind that the amount of work, i.e. data to be processed, grows proportionately to $(FV)^2$. Furthermore, a homogeneous system is considered. Similarly to the linear case (Section IV-A), results are not explicitly merged. We have

$$T_1 = \alpha_i\left(A\left(\frac{FV}{\ell}\right)^2 + 2C\left(\frac{FV}{\ell}\right)\right) \qquad i = 2,\dots,m \tag{49}$$

$$\ell^2 = \sum_{i=1}^{m}\alpha_i \tag{50}$$

Equations (49) mean that communication and processing is performed in the same interval on all processors. Since the system is homogeneous, $\alpha_i = \ell^2/m$, for $i = 1,\dots,m$, the whole schedule length is

$$T(F) = A_0(F)V + \frac{\ell^2}{m}\left(A\left(\frac{FV}{\ell}\right)^2 + 2C\left(\frac{FV}{\ell}\right)\right) \tag{51}$$

We will now compute the optimum value of F for which T is minimum.

1) $\mathbf{A_0(F) = 1/F^c}$, where $c > 0$ is a constant. Then,

$$T'(F) = -cF^{-(c+1)}V + \frac{2FAV^2}{m} + \frac{2CV\ell}{m} \tag{52}$$

$$T''(F) = c(c+1)VF^{-(c+2)} + \frac{2AV^2}{m} > 0 \tag{53}$$

Hence, $T(F)$ is minimum when $T'(F) = 0$. Unfortunately, equation (52) does not seem to have an easy analytical solution for $T'(F) = 0$ and has to be solved numerically.

2) $\mathbf{A_0(F) = c\ln(1/F)}$, where $c > 0$ is a constant. We have

$$T'(F) = -cV/F + \frac{2FAV^2}{m} + \frac{2CV\ell}{m} \tag{54}$$

$$T''(F) = cV/F^2 + \frac{2AV^2}{m} > 0 \tag{55}$$

Again, $T(F)$ is minimum when $T'(F) = 0$ and

$$F = \frac{\sqrt{4C^2V^2\ell^2/m^2 + 8cAV^3/m} - 2CV\ell/m}{4AV^2/m} \tag{56}$$

3) $\mathbf{A_0(F) = c(1 - \ln F)}$, where $c > 0$ is a constant, is dealt in the same way as the previous case because $[c\ln(1/F)]' = c(1-\ln F)' = -c/F$.

4) $\mathbf{A_0(F) = c(1 - F)}$, where $c > 0$ is a constant.

$$T'(F) = -cV + \frac{2FAV^2}{m} + \frac{2CV\ell}{m} \tag{57}$$

and

$$T''(F) = \frac{2AV^2}{m} > 0. \tag{58}$$

Hence, $T(F)$ is minimum when $F = \frac{mc-2C\ell}{2AV}$.

F. Sequential Communication, Quadratic Second Stage

We have a set of equations determining work distribution:

$$2C\alpha_{i+1}\left(\frac{FV}{\ell}\right) + A\alpha_{i+1}\left(\frac{FV}{\ell}\right)^2 = A\alpha_i\left(\frac{FV}{\ell}\right)^2 \qquad i = 1,\dots,m-1 \tag{59}$$

$$\ell^2 = \sum_{i=1}^{m}\alpha_i \tag{60}$$

Unfortunately, this set of nonlinear equations does not seem to have an easy analytical solution for F minimizing the schedule length. Therefore, we will recourse to numerical methods to solve (59)-(60) and find F for which the processing time is minimum.

V. Numerical Modeling

This section is dedicated to showing tendencies in the system parameters when the filtering intensity is optimum with respect to the minimum total processing time. In cases not amenable to representation with a closed-form formula, the optimum value of F was found by use of Python method `scipy.optimize.fsolve`. We assume that the amount of input data is $V = 1\mathrm{E}6$. We will analyze recurring patterns in performance rather than particular numbers. Therefore, only representative examples of the cases introduced in Section III-A are extensively discussed. For simplicity, only homogeneous systems are analyzed.

A. Parallel Communication, Linear Second Stage

Fig. 2 presents the relationship between the retained data fraction F and the schedule length T in case 2, i.e. $A_0(F) = c\ln(1/F)$, for $m = 100$ and several combinations of A, C and c values. The smallest value of F for which T is shown in Fig. 2 is 0.01, because F must be greater than 0. When $A = C = 5$ and $c = 0.001$, filtering is fast, while data transfer and processing in the second stage are rather slow. Hence, the smaller amount of data is retained, the shorter schedule is obtained. Contrarily, when $A = C = 1$ and $c = 0.3$, filtering

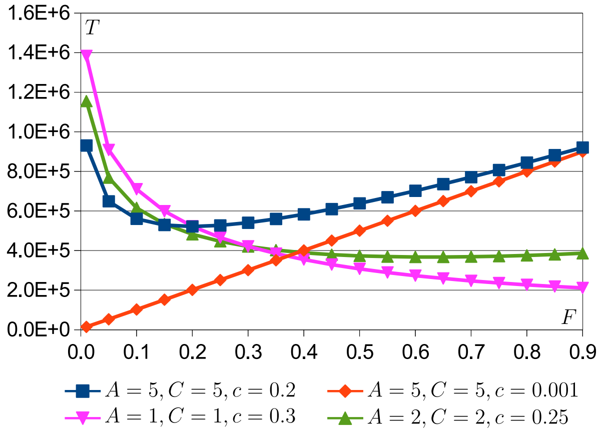


Fig. 2. T vs. F for parallel communication, linear second stage, case 2, $m = 100$.

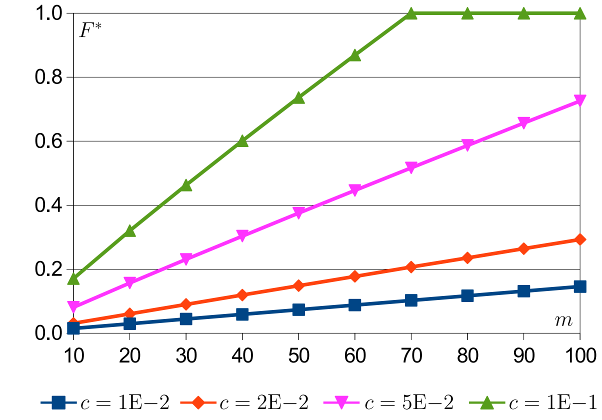


Fig. 3. F^* vs. m for parallel communication, linear second stage, case 1, $A = 5$, $C = 2$.

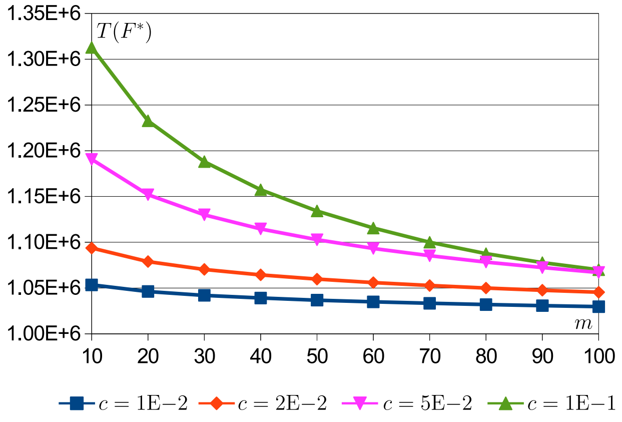


Fig. 4. $T(F^*)$ vs. m for parallel communication, linear second stage, case 1, $A = 5$, $C = 2$.

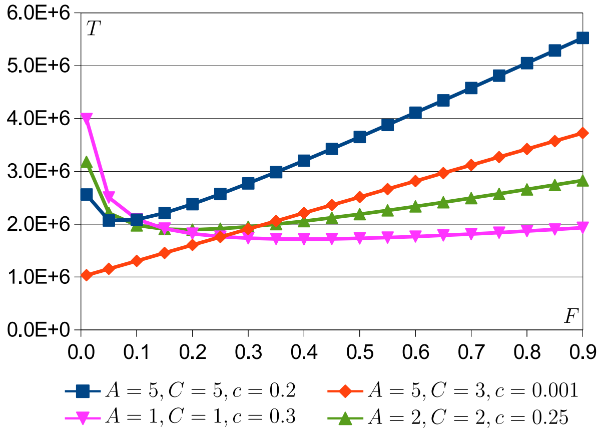


Fig. 5. T vs. F for sequential communication, linear second stage, case 1, $m = 10$.

is slow, while data transfer and processing are rather fast. In consequence, larger F (i.e. lower filtering intensity) results in a smaller schedule length T. In the remaining two presented cases, the optimum value of F is neither the minimum possible (close to 0) nor the maximum possible (1). For $A = C = 5$ and $c = 0.2$, the best value of F is 0.2, and for $A = C = 2$, $c = 0.25$ it is 0.65.

Fig. 3 shows how the optimum value of retained data fraction F^* depends on the number m of second stage processors, for case 1 ($A_0(F) = 1/F^c$) with $A = 5$ and $C = 2$. When m grows, parallel data transfer and processing take less time in comparison to the filtering stage. Therefore, a larger fraction of data should be retained, in order to decrease the filtering time. Naturally, the optimal filtering intensity decreases when filtering is slow, i.e., for large c. In particular, when $c = 1\text{E}{-1}$ and $m \geq 70$, no filtering should take place.

The total processing time resulting from filtering the optimum size of data for different values of c and m is depicted in Fig. 4. Naturally, the schedule length decreases when more processors are used. This effect is stronger when c is large. Indeed, in this case, decreasing filtering intensity (F increases, $A_0(F)$ decreases), which is possible because of using a larger number of processors, has a large impact on the filtering time.

B. Sequential Communication, Linear Second Stage

When communication is sequential, a smaller number of second stage processors can be effectively used than in the case of parallel communication. Therefore, in Fig. 5, we present the schedule lengths obtained for different values of F and network parameters, $m = 10$, and for filtering case 1. In general, the visible tendencies are similar to the ones in Fig. 2. However, even when $A = C = 1$ and $c = 0.3$, the optimum value of F is much smaller than 1, i.e. filtering must be more intensive than for parallel communication. The best among values analyzed here is 0.4. Indeed, sequential communication is a bottleneck, and even very costly filtering can be beneficial, because it decreases the communication time.

The optimum data fractions F^* for $A = 5$ and $C = 2$ are presented in Fig. 6. The values are much smaller than in the case of parallel communication (see Fig. 3). As we already explained, intensive filtering decreases the communication time, which dominates in the schedule length. The fraction

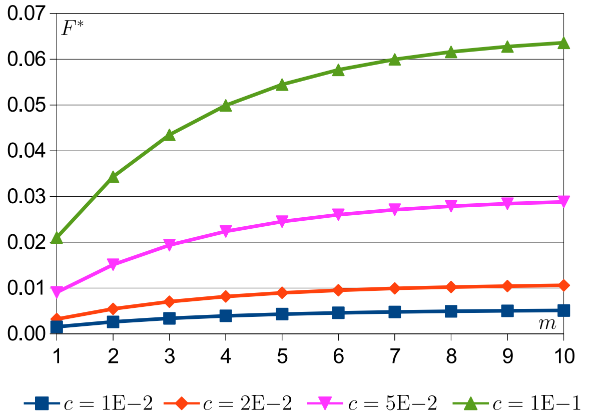


Fig. 6. F^* vs. m for sequential communication, linear second stage, case 1, $A = 5$, $C = 2$.

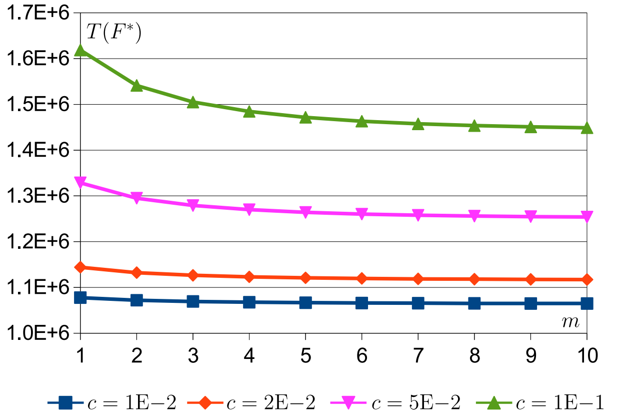


Fig. 7. $T(F^*)$ vs. m for sequential communication, linear second stage, case 1, $A = 5$, $C = 2$.

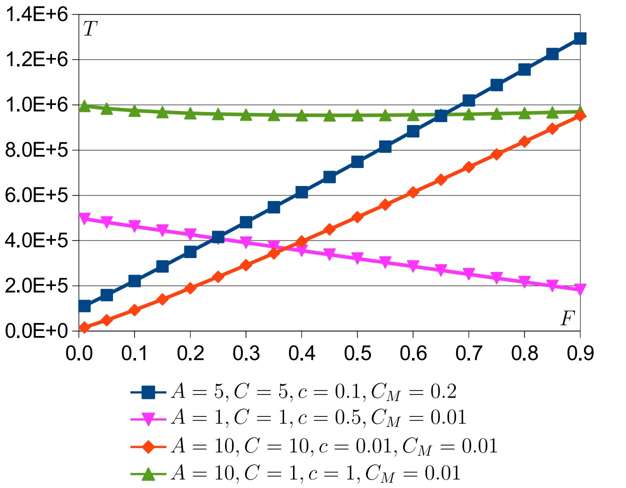


Fig. 8. T vs. F for parallel communication, loglinear second stage, case 4, $m = 100$.

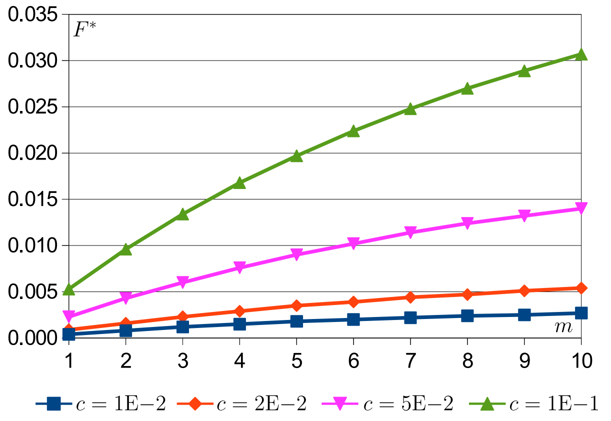


Fig. 9. F^* vs. m for sequential communication, loglinear second stage, case 1, $A = C = 3$, $C_M = 1\text{E}{-}2$.

of retained data grows with increasing m at a slower pace than in the case of parallel communication.

The optimum schedule lengths are shown in Fig. 7. Using a larger number of servers results in a shorter processing time, but the impact of changing m is smaller than in Fig. 4, because using more processors m does not decrease the communication time.

C. Parallel Communication, Loglinear Second Stage

Fig. 8 presents the schedule lengths T obtained for different values of F and network parameters, for parallel communication, loglinear second stage case 4. i.e. $A_0(F) = c(1 - F)$. Recall that in the case of linear second stage and case 4, the function $T(F)$ was always monotonous (Section IV-A, equations (22, 23)). It can be seen in Fig. 8 that when the second stage complexity is loglinear, $T(F)$ is also monotonous for many choices of network parameters, but not for all of them. In particular, when $A = 10$, $C = 1$, $c = 1$ and $C_M = 0.01$, the best among analyzed values of F is 0.45.

In cases 1 and 2 of data filtering complexity, the impact of increasing the number of processors m on the optimum value of F and schedule length is similar as for linear processing complexity. The main difference is that the time of merging the results also influences the results. When C_M is big, the merging stage becomes a bottleneck. Hence, more intensive filtering is required to reduce its duration and thus to obtain an optimum schedule.

D. Sequential Communication, Loglinear Second Stage

The differences between systems with sequential and parallel communication in the case of loglinear second stage are similar to those present when the second stage is linear. Since sequential communication is a bottleneck, optimum schedules are obtained by more intensive data filtering. Fig. 9 shows that even if m is large and filtering is slow, the fraction of retained load should be at most several percent in case 1 of filtering complexity. The optimum fractions obtained for cases 2, 3 and 4 are even smaller.

E. Parallel Communication, Quadratic Second Stage

When the second stage complexity is quadratic, intensive data filtering is required to obtain a short schedule by de-

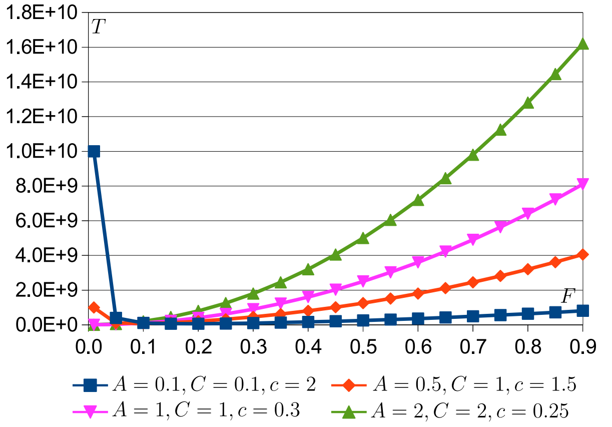


Fig. 10. T vs. F for parallel communication, quadratic second stage, case 1, $m = 100$.

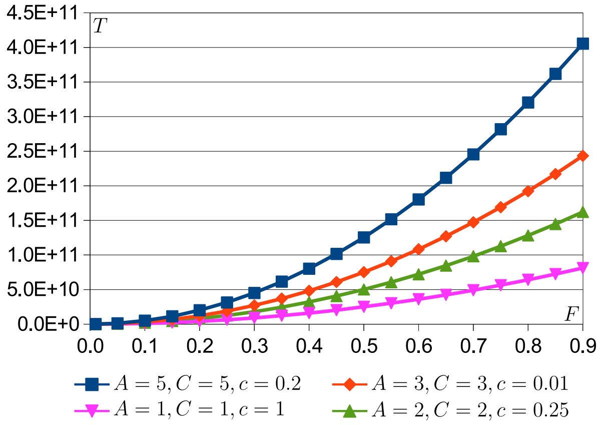


Fig. 11. T vs. F for sequential communication, quadratic second stage, case 1, $m = 10$.

creasing the duration of processing. It can be seen in Fig. 10, representing case 1 of the filtering complexity, that only when c is really large (i.e. $c > 1$), it may not be beneficial to decrease the data size as much as possible. In cases 2, 3 and 4 the fraction of retained data should be practically always as small as possible.

F. Sequential Communication, Quadratic Second Stage

When communication is sequential and the second stage complexity is quadratic, very intensive filtering should be used, even if it is costly. For all combinations of parameter values we studied, $T(F)$ is an increasing function (see Fig. 11). Although the optimum fraction F^* increases slightly with growing m, it stays below 0.01 for all settings we tested. Taking into account the practical limitations on F, this means that the smallest possible amount of data should be retained.

VI. Conclusions

In this paper we analyzed impact of data filtering intensity on the performance of systems with single expensive sensors. The analysis covered two-stage systems with generic representations of filtering algorithms, communication patterns and data processing methods. It appears that due to the interaction of nonlinear complexity of filtering, transmission time and further data processing stages, there exists filtering intensity which is optimum for the overall processing performance. These optima were investigated both analytically and computationally. It appeared that the communication subsystem and the second stage algorithm complexity have a large impact on the first stage filtering intensity. The ability of certain combinations of the system designs to scale is very limited. In systems with sequential communication gains from using parallel processors in the second stage are quickly diminishing because data transfer easily becomes a bottleneck. Processing with algorithms of high complexity (loglinear, quadratic) should be delegated to even further stages of data processing workflows because they incur needs for filtering intensities which may be hard to realize. Thus, by exposing scalability issues we demonstrated in this paper that designers of workflows with data filtering and distributed processing should strive for parallel data transfers and linear processing algorithms when handling large volumes of data from the sensors.

References

[1] M. Moges and T. Robertazzi, “Wireless sensor networks: Scheduling for measurement and data reporting,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 327–340, 2006. doi: 10.1109/TAES.2006.1603426

[2] J. Berlińska, “A comparison of priority rules for minimizing the maximum lateness in tree data gathering networks,” *Engineering Optimization*, vol. 54, pp. 218–231, 2022. doi: 10.1080/0305215X.2020.1861263

[3] ——, “Scheduling in data gathering networks with background communication,” *Journal of Scheduling*, vol. 23, pp. 681–691, 2020. doi: 10.1007/s10951-020-00648-5

[4] ——, “Heuristics for scheduling data gathering with limited base station memory,” *Annals of Operations Research*, vol. 285, pp. 149–159, 2020. doi: 10.1007/s10479-019-03185-3

[5] ——, “Scheduling for data gathering networks with data compression,” *European Journal of Operations Research*, vol. 246, pp. 744–749, 2015. doi: 10.1016/j.ejor.2015.05.026

[6] T. Colombo, “Trigger & DAQ at the LHC, filtering data from 50 TB/s to 1 GB/s,” https://indico.cern.ch/event/825688/attachments/1872900/3082664/trigger_daq_at_lhc.pdf, CERN EP/LBC, July 2019, accessed 31/3/2022.

[7] Committee on U.S.-Based Electron-Ion Collider Science Assessment, *An Assessment of U.S.-Based Electron-Ion Collider Science*. Washington D.C.: The National Academy of Science, Engineering and Medicine, The National Academy Press, 2018.

[8] C. Toth and C. Jóźków, “Remote sensing platforms and sensors: A survey,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 115, pp. 22–36, 2016. doi: 10.1016/j.isprsjprs.2015.10.004

[9] Y.-C. Cheng and T. Robertazzi, “Distributed computation with communication delay,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 6, pp. 700–712, 1988. doi: 10.1109/7.18637

[10] V. Bharadwaj, D. Ghose, V. Mani, and T. Robertazzi, *Scheduling Divisible Loads in Parallel and Distributed Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1996.

[11] T. Robertazzi, “Ten reasons to use divisible load theory,” *IEEE Computer*, vol. 36, no. 5, pp. 63–68, 2003. doi: 10.1109/MC.2003.1198238

[12] M. Drozdowski, *Scheduling for Parallel Processing*. London: Springer, 2009.

[13] H. Casanova, A. Legrand, and Y. Robert, *Parallel Algorithms*. London, UK: CRC Press, Taylor and Francis, 2009.

[14] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, pp. 102–114, 2002. doi: 10.1109/MCOM.2002.1024422

[15] P. Pereira, A. Grilo, and F. Rocha, *End-to-End Reliability in Sensor Networks: Survey and Research Challenges in P. Pereira (ed), EuroFGI Workshop in IP Qos and Traffic Control*. Academia, 2007.

[16] T. Muhammed and A. Shaikh, “An analysis of fault detection strategies in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 78, pp. 267–287, 2017. doi: 10.1016/j.jnca.2016.10.019

[17] M. Dener, “Security analysis in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 10, p. 303501, 2014. doi: 10.1155/2014/303501

[18] W. Meng, L. Xie, and W. Xiao, “Optimality analysis of sensor-source geometries in heterogeneous sensor networks,” *IEEE Transactions on Wireless Communication*, vol. 12, pp. 1958–1967, 2013. doi: 10.1109/twc.2013.021213.121269

[19] L. Cao, Y. Cai, and Y. Yue, “Swarm intelligence-based performance optimization for mobile wireless sensor networks: Survey, challenges, and future directions,” *IEEE Access*, vol. 7, pp. 161 524–161 553, 2019. doi: 10.1109/access.2019.2951370

[20] W. Luo, B. Gu, and G. Lin, “Communication scheduling in data gathering networks of heterogeneous sensors with data compression: Algorithms and empirical experiments,” *European Journal of Operational Research*, vol. 271, pp. 462–473, 2018. doi: 10.1016/j.ejor.2018.05.047

[21] W. Luo, Y. Xu, B. Gu, W. Tong, R. Goebel, and G. Lin, “Algorithms for communication scheduling in data gathering network with data compression,” *Algorithmica*, vol. 80, pp. 3158–3176, 2018. doi: 10.1007/s00453-017-0373-6

[22] C. Li and W. Luo, “Exact and approximation algorithms for minimizing energy in wireless sensor data gathering network with data compression,” *American Journal of Mathematical and Management Sciences*, vol. 41, no. 4, pp. 305–315, 2022. doi: 10.1080/01966324.2021.1960226

[23] J. Berlińska and M. Drozdowski, “Scheduling divisible mapreduce computations,” *Journal of Parallel and Distributed Computing*, vol. 71, no. 3, pp. 450–459, 2011. doi: 10.1016/j.jpdc.2010.12.004

[24] ——, “Comparing load-balancing algorithms for mapreduce under zipfian data skews,” *Parallel Computing*, vol. 72, pp. 14–28, 2018. doi: 10.1016/j.parco.2017.12.003

[25] Wikipedia contributors, “Lambert W function,” https://en.wikipedia.org/wiki/Lambert_W_function, [Online; accessed 5-August-2022].

DOI: 10.15439/2023F8932
ISSN 2300-5963 ACSIS, Vol. 35

Resilient s-ACD for Asynchronous Collaborative Solutions of Systems of Linear Equations

Lucas Erlandson*, Zachary Atkins†, Alyson Fox*, Christopher J. Vogl*, Agnieszka Międlar‡, and Colin Ponce*
*Center for Applied Scientific Computing
Lawrence Livermore National Lab
ORCIDS: 0000-0003-4544-6148, None, 0000-0002-3855-694X, 0000-0002-6720-8805
Emails: {erlandson3,fox33,vogl2,ponce11}@llnl.gov
†University of Colorado Boulder
ORCID: 0000-0002-2491-0725
Email: zach.atkins@colorado.edu
‡Department of Mathematics, Virginia Tech
Blacksburg, VA
ORCID: 0000-0002-2995-7426
Email: amiedlar@vt.edu

***Abstract*—Solving systems of linear equations is a critical component of nearly all scientific computing methods. Traditional algorithms that rely on synchronization become prohibitively expensive in computing paradigms where communication is costly, such as heterogeneous hardware, edge computing, and unreliable environments. In this paper, we introduce an s-step Approximate Conjugate Directions (s-ACD) method and develop resiliency measures that can address a variety of different data error scenarios. This method leverages a Conjugate Gradient (CG) approach locally while using Conjugate Directions (CD) globally to achieve asynchronicity. We demonstrate with numerical experiments that s-ACD admits scaling with respect to the condition number that is comparable with CG on the tested 2D Poisson problem. Furthermore, through the addition of resiliency measures, our method is able to cope with data errors, allowing it to be used effectively in unreliable environments.**

I. Introduction

Solving a system of linear equations $A\mathbf{x} = \mathbf{b}$ is a critical kernel in many applications, studied in great detail across applications, as well as for both iterative [1], [2] and direct solves [3], [4], [5], [6]. However, even iterative methods such as Krylov subspace methods, which have reduced serialization [7], require global synchronization. One of the most popular of such methods, the Conjugate Gradient (CG) method, computes global inner product at each iteration [8], [9]. The burden of this synchronization cost is increasing in modern computing environments due to two reasons: 1) as the number of parallel processes rapidly increases, the cost of global synchronization does too; 2) new environments are being considered for computationally expensive tasks, e.g. distributed (drones, power grid) and heterogeneous (accelerators) computing. Due to these factors, there is a critical need for asynchronous algorithms that can operate without the need for synchronization at every iteration and are able to handle the increase of data errors introduced by the increased number of failure points and unreliabilities.

The contributions of this paper are four-fold:

1) We introduce the s-step Approximate Conjugate Directions (s-ACD) method, which is a novel asynchronous Krylov-like linear system solver that achieves scaling with respect to the condition number that is comparable with CG on the tested 2D Poisson problem.
2) We introduce and investigate a demonstrative scenario that results in data errors by introducing unreliabilities into the calculations.
3) We develop resiliency measures that are used in s-ACD to detect corruption and could be used in other iterative methods.
4) We provide a comparative study using numerical experiments of the newly introduced developments.

In Section II, we begin by providing a background of iterative and asynchronous methods. Additionally, we briefly discuss *Skywing* [10], the collaborative autonomy framework that provides the agent-based approach that our implementation utilizes. Following this, Section III describes the s-ACD method for the solution of linear systems. Section IV discusses how the corruption of calculations in an unreliable environment are modeled and what situations the resulting corruption type and failure model might apply to. Then, Section V introduces methods that can be used for adding resiliency to iterative solvers, which results in the Resilient s-ACD method. Section VI provides a variety of numerical experiments developed to test the properties of the various methods to demonstrate their efficacy. Final concluding remarks are presented in Section VII.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 21-FS-007 and 22-ERD-045. The work of Agnieszka Międlar was supported by the National Science Foundation (NSF) under Grant DMS-2144181 and DMS-2324958. LLNL-CONF-849356.

II. Background

First, let us consider the Krylov subspace methods, a class of iterative methods where an initial guess of the solution to $A\mathbf{x} = \mathbf{b}$ is updated by iteratively building up a Krylov subspace span$\{\mathbf{b}, A\mathbf{b}, A^2\mathbf{b}, \ldots\}$. For symmetric positive definite matrices (SPD), the CG method, a particular Krylov solver, constructs a series of direction vectors $\mathbf{p}^\kappa$ that are A-conjugate to each other, as well as residual vectors $\mathbf{r}^\kappa$ that span the same Krylov subspace for iteration κ [8], [9]. Asymptotically, this method achieves convergence to a given tolerance in $\mathcal{O}(\sqrt{\operatorname{cond}(A)})$ iterations for a problem with condition number $\operatorname{cond}(A)$, making it the solver of choice for SPD matrices. However, global communication is needed to ensure the orthogonality necessary for the method to be robust.

To reduce the cost of global communication, methods such as the communication-avoiding s-step algorithms [11], [12] and communication hiding pipeline methods [13], [14] have been proposed in the synchronous case [15]. To address the issue of unreliable computing environments, some fault-tolerant or resilient CG methods are also available [16], [17], [18]. However, both of these classes of methods still require a high level of global synchronization for orthogonality to be preserved. In sufficiently distributed environments, these costs may become too restrictive, leading to the need of methods that do not require global synchronization and exact orthogonality.

For the solution of linear systems, chaotic or asynchronous methods [19], [20], [21] such as asynchronous Jacobi [22], [23], [24] have been developed to provide asynchronicity to already existing solvers. Although resiliency has been added to asynchronous Jacobi [25] to make it more fault-tolerant, the number of iterations scales proportional to the condition number, driving the need for more powerful asynchronous linear solvers.

A. Skywing

Edge computing, in which many small devices exist and work together in an unstructured setting, is a rapidly growing field in computing. Edge computing applications pose a unique set of challenges:

1) Both the physical and cyber environments are highly unreliable, as devices are placed in uncontrolled locations, e.g. homes or along power lines. As such, they can readily and unexpectedly break, get unplugged, or become compromised by cyberattacks.
2) The collection of participating devices is often quite heterogeneous, with a range of vendors and device capabilities.
3) The computational workflows are frequently streaming workflows that continually monitor and respond to some needs, rather than being a single computational task that terminates upon completion.

While traditional parallel computing paradigms, such as HPC or database computing, each have some of these challenges in common, the combination is unique to edge computing.

This paper details a new method in the *collaborative autonomy* paradigm, a class of methods in which multiple computational units work independently of each other but towards a common goal. Through adapting to unreliability present in the environment, these methods can provide reliable computing in unreliable environments.

Existing software platforms like Apache Hadoop [26] and Apache Spark [27] are designed for large-scale, “big data” computing work, but they largely implement leader-follower patterns and perform computing in batches. These approaches, while effective in controlled cluster environments, lack the resilience necessary to withstand common faults in edge computing applications such as hardware faults and, increasingly, cyber intrusions. Other parallel computing frameworks, such as OpenMP and MPI, do not necessarily rely on leader-follower paradigms, but are more naturally designed for well-controlled environments and terminating computational tasks.

Skywing is a software platform developed at Lawrence Livermore National Lab, which follows a publication/subscription paradigm. This allows any agent involved in the computation to subscribe or publish to a stream of data, and any data on a stream an agent is subscribed to is considered agnostic. Because of the unstructured nature, this enables increased flexibility, particularly for consensus based methods. Skywing aims to provide *method composition* to enable a modular approach, allowing users to utilize appropriate levels of resiliency for each module. The source code of Skywing is available on GitHub [10].

III. s-ACD

A. Problem Statement

Consider solving the linear system

$$A\mathbf{x} = \mathbf{b} \tag{1}$$

for $\mathbf{x} \in \mathbb{R}^m$, where $A \in \mathbb{R}^{m\times m}$ and $\mathbf{b} \in \mathbb{R}^m$. Assume the linear system is distributed across N agents according to a non-overlapping partition. Denote $A_i \in \mathbb{R}^{m_i\times m}$, $m_i < m$, as the block of rows of matrix A that are stored on agent i, $i = 1, \ldots, N$. Then

$$A = [A_1^T \cdots A_N^T]^T. \tag{2}$$

Given a SPD matrix A, two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$ are A-conjugate if and only if

$$\langle \mathbf{u},\ \mathbf{v}\rangle_A := \langle \mathbf{u},\ A\mathbf{v}\rangle = \mathbf{u}^T A\mathbf{v} = 0. \tag{3}$$

This paper establishes an asynchronous iterative method for solving $A\mathbf{x} = \mathbf{b}$, where agent i computes successive approximations to the solution vector $\mathbf{x}$, denoted $\mathbf{x}^0$, $\mathbf{x}^1$, etc…

B. Conjugate Directions Algorithm

Due to the high communication costs of edge computing environments, the classical Conjugate Gradient (CG) method cannot be directly used due to the need for significant synchronization when computing orthogonal direction vectors. However, we can utilize a variant called the Conjugate Directions

(CD) method by relaxing the global orthogonality constraints. The CD method, introduced by Hestenes and Stiefel in [9], is a generalization of the classical CG method. It solves the problem iteratively by computing a sequence of conjugate direction vectors. CG defines the new search direction based on a residual vector and the previously computed search directions, while the CD uses only the previous search directions. In this section, a short introduction to the CD method is given. For more details, see Hestenes and Stiefel [9].

Denote a vector $\mathbf{z} \in \mathbb{R}^m$ at iteration κ as $\mathbf{z}^\kappa$. Let $\mathbf{x}^0$ be an initial guess to the solution, then set the initial residual $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0 \in \mathbb{R}^m$ and select an arbitrary initial direction $\mathbf{p}^0 \in \mathbb{R}^m$. At each iteration $\kappa = 0, 1, \ldots$, the new solution approximation and the residual are computed as

$$\alpha^\kappa = \frac{\langle \mathbf{p}^\kappa,\ \mathbf{r}^\kappa \rangle}{\langle \mathbf{p}^\kappa,\ A\mathbf{p}^\kappa \rangle}, \tag{4}$$

$$\mathbf{x}^{\kappa+1} = \mathbf{x}^\kappa + \alpha^\kappa \mathbf{p}^\kappa, \tag{5}$$

$$\mathbf{r}^{\kappa+1} = \mathbf{r}^\kappa - \alpha^\kappa A\mathbf{p}^\kappa. \tag{6}$$

A new direction vector $\mathbf{p}^{\kappa+1}$ is chosen such that

$$\left\langle \mathbf{p}^{\kappa+1},\ \mathbf{p}^\iota \right\rangle_A = 0, \quad \iota = 0, \ldots, \kappa. \tag{7}$$

In the special case of the Conjugate Gradient (CG) method, we initialize the first direction vector as $\mathbf{p}^0 = \mathbf{r}^0$ and compute the subsequent direction vectors using a three-term recurrence relation, i.e.,

$$\beta^\kappa = -\frac{\|\mathbf{r}^{\kappa+1}\|_2}{\|\mathbf{r}^\kappa\|_2} = \frac{\langle \mathbf{r}^{\kappa+1},\ \mathbf{p}^\kappa \rangle_A}{\langle \mathbf{p}^\kappa,\ \mathbf{p}^\kappa \rangle_A}, \tag{8}$$

$$\mathbf{p}^{\kappa+1} = \mathbf{r}^{\kappa+1} - \beta^\kappa \mathbf{p}^\kappa. \tag{9}$$

The second formulation for β^κ in equation (8) represents the coefficient used to orthogonalize the new residual vector $\mathbf{r}^{\kappa+1}$ against the prior direction vector $\mathbf{p}^\kappa$ using Gram-Schmidt orthogonalization with the A-norm. In other words, $\mathbf{p}^{\kappa+1}$ is computed by A-orthogonalizing the new residual vector $\mathbf{r}^{\kappa+1}$ against the prior direction vector $\mathbf{p}^\kappa$. Our method combines the Conjugate Direction (CD) method globally while allowing each device to perform Conjugate Gradient (CG) steps locally. This approach achieves improved scaling compared to asynchronous Jacobi (for which some convergence is presented in [23]) without requiring the synchronization at each iteration as CG does.

C. Asynchronous s-Approximate Conjugate Directions (s-ACD)

Within the framework of the Conjugate Directions (CD) method, our objective is to design a fully asynchronous method. First, we introduce the following notation. Let $\mathbf{z}^{\psi(i,j,\kappa)} \in \mathbb{R}^m$ denote the local copy of the vector $\mathbf{z}$ from agent j received by node i at iteration κ. Let $\mathbf{z}_i \in \mathbb{R}^{m_i}$ denote the subvector of $\mathbf{z}$ corresponding to the block of elements that agent i is approximating. Each agent has access to its local portion A_i of the matrix A, the full right-hand side vector $\mathbf{b}$, and maintains a set of local variables: a local residual vector $\mathbf{r}^\kappa \in \mathbb{R}^m$, a local solution vector $\mathbf{x}^\kappa \in \mathbb{R}^m$, and a local direction vector $\mathbf{p}^\kappa \in \mathbb{R}^m$ for each iteration κ. In this context, $\kappa = 1, 2, \ldots$ represents the local iteration count of agent i. It is important to emphasize that the iteration count may vary between agents due to the asynchronous nature. Therefore, the agents may be at different stages of the iterative process at any given time.

Each agent will initialize its local vectors as follows for $\kappa = 0$:

$$\mathbf{x}^0 = \mathbf{0}, \tag{10}$$

$$\mathbf{r}^0 = \mathbf{b}, \tag{11}$$

$$\mathbf{p}^0 = \mathbf{b}. \tag{12}$$

Then, each agent will asynchronously advance from local iteration κ to $\kappa + 1$ using the following steps:

1) Compute the local matrix-vector product $\mathbf{w}^\kappa := A_i^T \mathbf{p}_i^\kappa$, where $\mathbf{w}^\kappa \in \mathbb{R}^m$ and $\mathbf{p}_i^\kappa \in \mathbb{R}^m$ is the subvector of $\mathbf{p}^\kappa$ corresponding to the block of elements that agent i is responsible for.
2) Asynchronously send the vector $\mathbf{w}^\kappa$ from step 1 and the vector $\mathbf{p}_i^\kappa$ to the other agents.
3) Receive available updates from other agents
$$\left\{ \mathbf{w}^{\psi(i,j,\kappa)},\ \mathbf{p}_j^{\psi(i,j,\kappa)} \right\}_{j \in \mathcal{U}_i^\kappa},$$
where $\mathcal{U}_i^\kappa$ is the set of updates, i.e. $j \in \mathcal{U}_i^\kappa$ if and only if agent i during its local iteration κ received an update from agent j.

Note that for the restart mechanism introduced later, $\mathbf{r}^\kappa$ and $\mathbf{x}^\kappa$ are also sent and received during these communications. Note that the non-blocking communication allows agents to send and receive information in an asynchronous fashion, which enables parallelism while avoiding the need for synchronization at every iteration.

Once the updates have been received, each agent i will assemble the *asynchronous direction vector* $\widetilde{\mathbf{p}}^\kappa \in \mathbb{R}^m$ blockwise according to the partition:

$$\widetilde{\mathbf{p}}^\kappa = \begin{cases} \mathbf{p}_i^\kappa, & j = i; \\ \mathbf{p}_j^{\psi(i,j,\kappa)}, & j \in \mathcal{U}_i^\kappa; \\ \mathbf{0}_j, & \text{otherwise}, \end{cases}$$

where $\mathbf{p}_j^{\psi(i,j,\kappa)}$ represents the partial local direction vector agent i received from agent j at iteration κ. Since the matrix A is SPD,

$$A\mathbf{z} = A^T\mathbf{z} = \sum_{j=1}^{N} A_j{}^T \mathbf{z}_j \text{ for } \mathbf{z} \in \mathbb{R}^m.$$

Thus, the exact A matrix-vector product of the asynchronous direction vector, denoted $\widetilde{\mathbf{w}}^\kappa := A\widetilde{\mathbf{p}}^\kappa$, can be computed using only the received and local partial matrix-vector products:

$$\begin{aligned} \widetilde{\mathbf{w}}^\kappa := A\widetilde{\mathbf{p}}^\kappa &= A_i^T \mathbf{p}_i^\kappa + \sum_{j \in \mathcal{U}_i^\kappa} A_j{}^T \mathbf{p}_j^{\psi(i,j,\kappa)}, \\ &= \mathbf{w}^\kappa + \sum_{j \in \mathcal{U}_i^\kappa} \mathbf{w}^{\psi(i,j,\kappa)}. \end{aligned}$$

We use the asynchronous direction vector $\widetilde{\mathbf{w}}$ to construct the *s-conjugate direction vector*, denoted $\mathbf{d}^\kappa$. It is essential that $\mathbf{d}^\kappa$ is A-conjugate to the s prior s-conjugate direction vectors $\mathbf{d}^{\kappa-s-1}, \dots, \mathbf{d}^{\kappa-1}$. To achieve this, we can employ a method such as Gram-Schmidt orthogonalization. In order to ensure conjugacy with prior direction vectors, additional storage of the prior s conjugate direction vectors, $\left\{\mathbf{d}^{\kappa-\ell}\right\}_{\ell=1}^{s}$, and their A-products, $\left\{\mathbf{v}^{\kappa-\ell}\right\}_{\ell=1}^{s}$, is necessary. These vectors are defined recursively, with $\mathbf{d}^0 = \widetilde{\mathbf{p}}^0$, $\mathbf{v}^0 = \widetilde{\mathbf{w}}^0$, and for $\kappa > 0$,

$$\begin{aligned} \mathbf{d}^\kappa &= \widetilde{\mathbf{p}}^\kappa - \sum_{\ell=1}^{\min(s,\kappa)} \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big)\mathbf{d}^{\kappa-\ell}, \qquad (13)\\ \mathbf{v}^\kappa &= A\mathbf{d}^\kappa = \widetilde{\mathbf{w}}^\kappa - \sum_{\ell=1}^{\min(s,\kappa)} \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big)\mathbf{v}^{\kappa-\ell}, \end{aligned}$$

where $\mathrm{GS}_A(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell})$ is the magnitude of the projection of the vector $\widetilde{\mathbf{p}}^\kappa$ onto the vector $\mathbf{d}^{\kappa-\ell}$ under the A-inner product, i.e.,

$$\begin{aligned} \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big) &:= \frac{\left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A}{\left\langle \mathbf{d}^{\kappa-\ell},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A},\\ &= \frac{\left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{v}^{\kappa-\ell}\right\rangle}{\left\langle \mathbf{d}^{\kappa-\ell},\ \mathbf{v}^{\kappa-\ell}\right\rangle}. \end{aligned}$$

Note that the exact matrix-vector product $\mathbf{v}^\kappa = A\mathbf{d}^\kappa$ is ensured due to the definition of $\widetilde{\mathbf{w}}^\kappa := A\widetilde{\mathbf{p}}^\kappa$.

In the following theorem, we prove that $\mathbf{d}^\kappa$ is A-conjugate to the prior s conjugate direction vectors $\left\{\mathbf{d}^{\kappa-\ell-1}\right\}_{\ell=1}^{s}$.

Theorem 1. *Let $\mathbf{d}^\kappa, \mathbf{v}^\kappa$ be defined as in* (13). *Then for $\ell = 1, \dots, \min(s,\kappa)$,*

$$\left\langle \mathbf{d}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A = 0.$$

Proof. We proceed by induction on the iteration number κ. For $\kappa = 0$, the statement holds trivially. Suppose that the statement holds true for $\kappa = 0, \dots, \iota - 1$. We now show the statement holds true for $\kappa = \iota$.

By the definition of $\mathbf{d}^\kappa$, for $1 \le \ell \le \min(s,\kappa)$

$$\begin{aligned} &\left\langle \mathbf{d}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A\\ &= \left\langle \widetilde{\mathbf{p}}^\kappa - \sum_{\nu=1}^{\min(s,\kappa)} \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\nu}\big)\mathbf{d}^{\kappa-\nu},\ \mathbf{d}^{\kappa-\ell} \right\rangle\\ &= \left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A -\\ &\sum_{\nu=1}^{\min(s,\kappa)} \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-j}\big)\left\langle \mathbf{d}^{\kappa-\nu},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A. \end{aligned}$$

By the induction hypothesis, $\left\langle \mathbf{d}^{\kappa-\nu},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A = 0$ for all $\nu = 1, \dots, \min(s,\kappa)$ such that $\nu \neq \ell$. Hence,

$$\begin{aligned} &\left\langle \mathbf{d}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle\\ &= \left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A - \mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big)\left\langle \mathbf{d}^{\kappa-\ell},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A,\\ &= \left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A - \frac{\left\langle \widetilde{\mathbf{p}}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A}{\left\langle \mathbf{d}^{\kappa-\ell},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A}\left\langle \mathbf{d}^{\kappa-\ell},\ \mathbf{d}^{\kappa-\ell}\right\rangle_A,\\ &= 0. \end{aligned}$$

Thus at iteration κ, we have that $\left\langle \mathbf{d}^\kappa,\ \mathbf{d}^{\kappa-\ell}\right\rangle_A = 0$ for $\ell = 1, \dots, \min(s,\kappa)$. □

Using this definition of $\mathbf{d}^\kappa$, we can proceed in a manner similar to the Conjugate Directions (CD) method. Define the step size

$$\alpha^\kappa := \frac{\langle \mathbf{r}^\kappa,\ \mathbf{d}^\kappa\rangle}{\langle \mathbf{d}^\kappa,\ \mathbf{v}^\kappa\rangle}. \qquad (14)$$

Using the step size α^κ, the approximate solution and residual vectors are updated

$$\mathbf{x}^{\kappa+1} := \mathbf{x}^\kappa + \alpha^\kappa \mathbf{d}^\kappa, \qquad (15)$$

$$\mathbf{r}^{\kappa+1} := \mathbf{r}^\kappa - \alpha^\kappa \mathbf{v}^\kappa. \qquad (16)$$

Finally, the next local direction vector is computed by enforcing the new residual $\mathbf{r}^{\kappa+1}$ to be s-conjugate with the s-conjugate direction vectors $\mathbf{d}^{\kappa-\ell}$, $0 \le \ell \le \min(s,\kappa) - 1$,

$$\mathbf{p}^{\kappa+1} := \mathbf{r}^{\kappa+1} - \sum_{\ell=0}^{\min(s,\kappa)-1} \mathrm{GS}_A\big(\mathbf{r}^{\kappa+1}, \mathbf{d}^{\kappa-\ell}\big)\mathbf{d}^{\kappa-\ell}. \qquad (17)$$

Restarting: Due to the asynchronous nature of the s-ACD algorithm, it is possible that the direction vectors, and consequently the approximate solution vectors, differ between agents at each local iteration. To address the potential stagnation that can result from such a scenario, we incorporate an asynchronous restarting procedure. By introducing these asynchronous restarts, we provide an opportunity for the agents to realign their progress, mitigate the effects of asynchronicity, and make collective advancements towards the true solution. The frequency of the restarts can be adjusted based on the specific requirements and characteristics of the problem being solved. As mentioned earlier, during the s-ACD communication stage, each agent will send its local solution vector $\mathbf{x}^\kappa$ and local residual vector $\mathbf{r}^\kappa$. This exchange allows each agent to have an updated understanding of the current state of the (global) computation, enabling more efficient choices of direction vectors for the subsequent iterations.

The algorithm is restarted periodically after detecting stagnation in the residual norm. The detection of stagnation and following restart is purely a local decision and calculation. The restarting will be performed if a specified number of iterations have passed since the last restart and the residual norm has decreased less than a prescribed tolerance. This involves resetting the necessary variables, such as solution vectors, residual vectors, and direction vectors, to a common starting point. By doing so, the agents can restart from a more unified state and resume the algorithm to overcome the convergence stagnation. When a restart is deemed necessary, the local approximation to the asynchronous residual vector $\widetilde{\mathbf{r}}^\kappa$ is constructed by averaging the most recent updates $\left\{\mathbf{r}^{\psi(i,j,\kappa)}\right\}$ received from each neighbor. If no updates have been received from agent j, then set $\mathbf{r}^{\psi(i,j,0)} := \mathbf{b}$, i.e.,

$$\widetilde{\mathbf{r}}^\kappa = \frac{1}{N}\sum_{j=1}^{N}\mathbf{r}^{\psi(i,j,\kappa)},$$

where $\mathbf{r}^{\psi(i,i,\kappa)} = \mathbf{r}^\kappa$. Then, local approximation to the asynchronous solution vector $\widetilde{\mathbf{x}}^\kappa$ is computed by averaging the most recently received solution vectors $\left\{\mathbf{x}^{\psi(i,j,\kappa)}\right\}$. If no updates have been received from an agent j, then set $\mathbf{x}^{\psi(i,j,0)} = \mathbf{0}$, i.e.,

$$\widetilde{\mathbf{x}}^\kappa = \frac{1}{N}\sum_{j=1}^{N}\mathbf{x}^{\psi(i,j,\kappa)}, \tag{18}$$

where $\mathbf{x}^{\psi(i,i,\kappa)} = \mathbf{x}^\kappa$. Note that

$$\begin{aligned}\mathbf{b} - A\widetilde{\mathbf{x}}^\kappa &= \mathbf{b} - \frac{1}{N}\sum_{j=1}^{N}A\mathbf{x}^{\psi(i,j,\kappa)},\\ &= \frac{1}{N}\sum_{j=1}^{N}\Big(\mathbf{b} - A\mathbf{x}^{\psi(i,j,\kappa)}\Big),\\ &= \frac{1}{N}\sum_{j=1}^{N}\mathbf{r}^{\psi(i,j,\kappa)},\\ &= \widetilde{\mathbf{r}}^\kappa.\end{aligned}$$

Thus, the restarting procedure maintains the accuracy of the residual. Additionally, we found empirically that explicitly recomputing the local partial residual as

$$\widetilde{\mathbf{r}}_i^\kappa = \mathbf{b}_i - A_i\widetilde{\mathbf{x}}^\kappa$$

generally improves convergence and enables convergence for some non-symmetric matrices A. Modifications to (13) and (17) are required to account for the possibility that the s prior s-conjugate direction vectors may no longer be consistent after a restart. To address this issue, we introduce the set $\mathcal{S}$, which represents the subset of "active" s-conjugate direction vectors. After a restart, this set is reset to $\mathcal{S} = \emptyset$. After each iteration, we update $\mathcal{S}$ by taking the union of the set with the newly computed direction vector $\mathbf{d}^\kappa$, i.e., $\mathcal{S} = \mathcal{S} \cup \{\mathbf{d}^\kappa\}$. Only the vectors within the set $\mathcal{S}$ can be used in the s-step orthogonalization process. Thus, (13) and (17) need to be modified accordingly, i.e.,

$$\mathbf{d}^\kappa = \widetilde{\mathbf{p}}^\kappa - \sum_{\ell=1}^{\min(s,|\mathcal{S}|)}\mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big)\mathbf{d}^{\kappa-\ell}, \tag{19}$$

$$\mathbf{v}^\kappa = A\mathbf{d}^\kappa = \widetilde{\mathbf{w}}^\kappa - \sum_{\ell=1}^{\min(s,|\mathcal{S}|)}\mathrm{GS}_A\big(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\ell}\big)\mathbf{v}^{\kappa-\ell},$$

and

$$\mathbf{p}^{\kappa+1} := \mathbf{r}^{\kappa+1} - \sum_{\ell=0}^{\min(s,|\mathcal{S}|-1)}\mathrm{GS}_A\big(\mathbf{r}^{\kappa+1}, \mathbf{d}^{\kappa-\ell}\big)\mathbf{d}^{\kappa-\ell}. \tag{20}$$

To complete the restarting procedure, all the agents set their local vectors accordingly: $\mathbf{x}^\kappa = \widetilde{\mathbf{x}}^\kappa$, $\mathbf{p}^\kappa = \widetilde{\mathbf{r}}^\kappa$, and $\mathbf{r}^\kappa = \widetilde{\mathbf{r}}^\kappa$.

IV. Data Errors and Corruption Modeling

As increased parallelism and new environments are considered, the likelihood of errors increases and so too does the costs associated with data errors. Practical Krylov methods introduce restarts to handle errors introduced through finite precision calculations. However, the restarts alone are not enough when larger or more discrete data errors happen, the methods can lose the underlying subspace and orthogonality properties that they rely on. Thus, additional resiliency measures must be considered in environments where such large disruptions are expected. Even the s-ACD method above, which has the self-correcting restart mechanism, is still susceptible to errors introduced through numerous pathways including malicious injection, disconnection of agents, corruption introduced into the signal, delays in communication, agents entering a failed state, bit-errors introduced locally, etc. We focus on data corruption where the original data being communicated at a single iteration is replaced by other values.

One important note is that we only corrupt one vector at a time. Because the $\mathbf{x}$ and $\mathbf{r}$ vectors are only used during the restart, to accurately model the errors, we must force a restart after a corruption occurs, otherwise corruptions in $\mathbf{x}$ and $\mathbf{r}$ may be masked. To understand this, imagine we corrupt a transmission of vector $\mathbf{x}$ at iteration ι, and we do not force a restart. The receiving agent receives this corrupted $\mathbf{x}$ at iteration κ, stores it, but it does not use it in the calculation because a restart does not happen. However, at iteration $\kappa' > \kappa$ when a restart occurs, the transmitted vector $\mathbf{x}$ from iteration κ' is stored (overwriting the previous corrupted vector with an uncorrupted vector) and is used during the restart. Thus, the corrupted $\mathbf{x}$ that was stored at iteration κ was overwritten and never used, leading to the corruption being masked.

The failure model we consider is that of *one-off* corruption, meaning that at some point in time, one or more agents all have corruption applied to one of the vectors transmitted during that iteration. This failure model has been chosen for multiple reasons:

- such corruption clearly partitions time into a "before" and "after" corruption portions, where the "before" portion should be identical to the uncorrupted case,
- one can easily visually identify where the corruption occurs,
- it is simple to implement,
- it forms the basis for other forms of corruption and can relatively easily be generalized to the other forms.

V. Resilient s-ACD

While the s-ACD method has the self-correcting restart mechanism that allows it to be resilient to the presence of non-orthogonal directions introduced by the asynchronous approach (and somewhat resilient to other data errors), additional resiliency measures would be able to decrease the impact of other data errors. In this section, we introduce resiliency in

Algorithm 1 s-Approximate Asynchronous Conjugate Directions.

```
for node $i \leftarrow 1$ to $N$ do
  INPUT: global vector $\mathbf{b}$, local portion $A_i$ of the $A$ matrix
  OUTPUT: Each node $i$ has a local approximation $\mathbf{x}^\kappa$ to the solution vector $\mathbf{x}$
  for $j \leftarrow 1$ to $N, j \neq i$ do
    initialize $\mathbf{x}_j = \mathbf{0}$, $\mathbf{r}_j = \mathbf{b}_j, \mathbf{p}_j = \mathbf{b}_j$
  end for
  for $\kappa \leftarrow 0$ to $t_{\mathtt{max}}$ do
    $\mathbf{w}^\kappa \leftarrow A_i^T \mathbf{p}^\kappa$
    Send $\mathbf{w}^\kappa$, $\mathbf{p}_i^\kappa$, $\mathbf{x}^\kappa$, $\mathbf{r}^\kappa$
    $\left\{\mathbf{w}^{\psi(i,j,\kappa)}, \mathbf{p}_j^{\psi(i,j,\kappa)}, \mathbf{x}^{\psi(i,j,\kappa)}, \mathbf{r}^{\psi(i,j,\kappa)}\right\}_{j \in \mathcal{U}_i^\kappa} \leftarrow$ ReceiveAsync
    Set $\mathcal{U}_i^\kappa :=$ set of node indices from which updates were received
    $\widetilde{\mathbf{p}}_i^\kappa = \mathbf{p}_i^\kappa$
    $\widetilde{\mathbf{w}}^\kappa = \mathbf{w}^\kappa$
    for $j \in \mathcal{U}_i^\kappa$ do
      $\widetilde{\mathbf{p}}_j^\kappa = \mathbf{p}_j^{\psi(i,j,\kappa)}$
      $\widetilde{\mathbf{w}}^\kappa = \widetilde{\mathbf{w}}^\kappa + \mathbf{w}^{\psi(i,j,\kappa)}$
      $\mathbf{x}^{\psi(i,j,\ell)} = \mathbf{x}^{\psi(i,j,\kappa)}$
      $\mathbf{x}^{\psi(i,j,\ell)} = \mathbf{r}^{\psi(i,j,\kappa)}$
    end for
    $\mathbf{d}^\kappa = \widetilde{\mathbf{p}}^\kappa - \sum_{\nu=1}^{\min(s,|\mathcal{S}|)} \mathrm{GS}_A(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\nu})\mathbf{d}^{\kappa-\nu}$
    $\mathbf{v}^\kappa = \widetilde{\mathbf{w}}^\kappa - \sum_{\nu=1}^{\min(s,|\mathcal{S}|)} \mathrm{GS}_A(\widetilde{\mathbf{p}}^\kappa, \mathbf{d}^{\kappa-\nu})\mathbf{v}^{\kappa-\nu}$
    $\alpha^\kappa = \langle \mathbf{r}^\kappa,\ \mathbf{d}^\kappa \rangle / \langle \mathbf{d}^\kappa,\ \mathbf{v}^\kappa \rangle$
    $\mathbf{x}^{\kappa+1} = \mathbf{x}^\kappa + \alpha^\kappa \mathbf{d}^\kappa$
    $\mathbf{r}^{\kappa+1} = \mathbf{r}^\kappa - \alpha^\kappa \mathbf{v}^\kappa$

    $$\mathbf{p}^{\kappa+1} = \mathbf{r}_i^{\kappa+1} - \sum_{\nu=0}^{\min(s,|\mathcal{S}|-1)} \mathrm{GS}_A(\mathbf{r}^{\kappa+1}, \mathbf{d}^{\kappa-\nu})\mathbf{d}^{\kappa-\nu} \tag{21}$$

    if $\|\mathbf{r}^{\kappa+1}\| / \|\mathbf{b}\| < \epsilon$ then

      return $\mathbf{x}^{\kappa+1}$
    end if
    if ShouldRestart($\mathbf{r}^{\kappa+1}$, $\kappa$) then
      $\mathbf{x}^\kappa = \frac{1}{N}\left(\mathbf{x}^\kappa + \sum_{j=0,j\neq i}^{N} \mathbf{x}^{\psi(i,j,\ell)}\right)$
      $\mathbf{r}^\kappa = \frac{1}{N}\left(\mathbf{r}^\kappa + \sum_{j=0,j\neq i}^{N} \mathbf{x}^{\psi(i,j,\ell)}\right)$
      $\mathbf{r}_i^\kappa = \mathbf{b}_i - A_i \mathbf{x}^\kappa$
      $\mathbf{p}^\kappa = \mathbf{r}^\kappa$
      $\mathcal{S} = \emptyset$
    end if
  end for
end for
```

two stages: the detection stage and the correction stage. The benefit of this approach is to separate the tasks of identifying deviations from "normal" behavior and the ability to correct said behavior.

A. Detection Stage

To detect data errors, we need to be able to identify when the method is in a state that is not "normal." In CG, this can be done through the orthogonality conditions or monotonically decreasing quantities — which if violated indicate that something is not as expected. However, in order to achieve asynchronicity, s-ACD loses the orthogonality conditions. Instead, we can develop different detection schemes leveraging knowledge from CG.

1) Checksum: The checksum method is based off on idea of checksums, i.e., calculating a quantity locally, transmitting it with the information, and checking that the received quantity, when recalculated locally, is consistent. One simple way of doing this is using the inner-product of two vectors. The downside of this method is that it relies on a trustworthy sender (the sender can adjust both the checksum and sent vectors accordingly) and requires additional communication. However, it comes with a number of pros, e.g., it requires only a small amount of local computation (perhaps some that is being performed anyway), provides per-agent detection, puts constraints on the possible malicious vectors that can be used, and is cheap and easy to implement. In particular, we use $\mathbf{p}^T\mathbf{w}$, where only the local portions of both vectors are used (as only the local portion of $\mathbf{p}$ is sent and available to check with). This is done by calculating the checksum before an agent sends its vectors (Alg. 2), and comparing the received value against the recalculated value at the arrival (Alg. 3).

Algorithm 2 Checksum calculation before sending.

```
$\gamma^\kappa \leftarrow \mathbf{w}_i^{\kappa T} \mathbf{p}_i^\kappa$
Send $\mathbf{w}^\kappa$, $\mathbf{p}_i^\kappa$, $\mathbf{x}^\kappa$, $\mathbf{r}^\kappa$, $\gamma^\kappa$
```

Algorithm 3 Checksum calculation after receiving.

```
for $j \in \mathcal{U}_k^\kappa$ do
  if $\gamma^\kappa == \mathbf{w}^{\psi(i,j,\kappa)^T} \mathbf{p}_j^{\psi(i,j,\kappa)}$ then
    $\widetilde{\mathbf{p}}_j^\kappa = \mathbf{p}_j^{\psi(i,j,\kappa)}$
    $\widetilde{\mathbf{w}}^\kappa = \widetilde{\mathbf{w}}^\kappa + \mathbf{w}^{\psi(i,j,\kappa)}$
    $\mathbf{x}^{\psi(i,j,\ell)} = \mathbf{x}^{\psi(i,j,\kappa)}$
    $\mathbf{x}^{\psi(i,j,\ell)} = \mathbf{r}^{\psi(i,j,\kappa)}$
  else
    Mark update $\kappa$ from agent $j$ as corrupted
  end if
end for
```

2) General: As mentioned previously, one way of detecting corruption is to detect when the result from a calculation is different from what it should be. One can use "metrics" (also called *indicator variables*), which are simple scalars that change over time, and see when they change in unexpected ways. This allows us to adapt to different convergence speeds and parts of the convergence without relying too heavily on tunable parameters. While we lose precise orthogonality conditions and monotonically decreasing quantities in s-ACD, we do still have some relations that are roughly predictable. For example, as iterations progress, $\langle \mathbf{d}^\kappa,\ \mathbf{v}^\kappa\rangle$, $\langle \mathbf{p}^\kappa,\ \mathbf{p}^\kappa\rangle_A$, and $\langle \mathbf{r}^\kappa,\ \mathbf{r}^\kappa\rangle$ all tend to decrease. Thus, we can monitor these values and determine when they increase between successive iterations more than expected.

Because there is significant variation of the metrics over the course of the solve, we should not look directly at the successive difference of a timeseries metric ξ between iterations. Instead, we apply a smoothing step, take the difference between the smoothed values, and compare that against a smoothed version of the difference of smoothed values. When the ratio of these quantities gets above a specified value (which tends to be quite robust), we mark this iteration as corrupted.

To perform the smoothing we use a running average with a window size σ of 15 iterations, which allows us to perform these calculations online. If we let ξ be a $\mathbb{R}^\kappa$ timeseries with ξ^i being the value at point i in time, then we define the smoothed timeseries as

$$\mathrm{smooth}(\xi, \sigma, i) = \frac{\sum\limits_{j=\max(0,i-\sigma)}^{i} \xi^j}{\min(i, \sigma)}.$$

We define the relative successive difference to be

$$\mathrm{diff}(\xi, i) = \frac{\xi^{i+1} - \xi^i}{|\xi^i|}.$$

We consider a timeseries ξ to be corrupted at time i if $\mathrm{smooth}(\mathrm{diff}(\mathrm{smooth}(\xi, size, i), i), size, i) > \epsilon$ for some tolerance $\epsilon > 0$.

In particular, we track the timeseries defined by $\langle \mathbf{v}^\kappa,\ \mathbf{d}^\kappa\rangle$ and $\langle \mathbf{r}^\kappa,\ \mathbf{r}^\kappa\rangle$. The biggest drawbacks of this method are that it does not provide per-agent detection and introduces additional computational steps. However, it is easily generalizable to other methods, requires only local computation, doesn't rely on a trustworthy sender, and the computational complexity can be mitigated by updating the differences and smoothed timeseries at each time step rather than recalculating the entire timeseries.

3) Algorithm-Based: The final class of detection methods that we will discuss are the algorithm-based metrics. These rely on knowing specific analytical relations within the calculations, such as the orthogonality conditions in CG. Although we do not have the orthogonality conditions, we do know that, $A\mathbf{x} = \mathbf{b}$ and $\mathbf{r} = \mathbf{b} - A\mathbf{x}$. There are many solution approximations and associated residuals that can be calculated, all of which should be similar to each other. Thus, if one of these vectors differs significantly from the others (or what is expected via explicit calculation of the residual), this indicates that a corruption is likely to have occurred. We compare the incoming solution and residual vectors from other agents against the updated value of the currently considered solution and residual, as well as the most recent solution and residual that have been checked by the detection mechanisms and identified as uncorrupted.

Due to the many comparisons and matrix-vector products involved, this is a computationally expensive check. However, it doesn't require any additional communication and provides per agent detection. Furthermore, the adaptive properties of the general metrics can be explored, although they are not currently used in our implementation (Alg 4).

Algorithm 4 Algorithm-based check for agent i, which compares the proposed update when applied to the incoming solution and residual vectors from agent j, with baseline updated solution and residual vectors.

```
for j=1, ..., N do
  $\mathcal{X}_j := \mathbf{x}^{\psi(i,j,\kappa)} + \alpha^\kappa \mathbf{d}^\kappa$
  $\mathcal{R}^{exp} := \mathbf{b}_i - A_i \mathcal{X}_j$
  $\mathcal{R}^{iter} := \mathbf{r}^{\psi(i,j,\kappa)} + \alpha^\kappa \mathbf{d}^\kappa$
  $\mathbf{x}^S$ := the last $\mathbf{x}^{\psi(i,j,\iota)}$ vector considered to be uncorrupted
  $\mathbf{r}^S$ := the last $\mathbf{x}^{\psi(i,j,\iota)}$ vector considered to be uncorrupted
  for $(x, r) \in ((\mathbf{x}^\kappa, \mathbf{r}^\kappa), (\mathbf{x}^S, \mathbf{r}^S))$ do
    $\mathfrak{X}_j := x + \alpha^\kappa \mathbf{d}^\kappa$
    $\mathfrak{R}^{exp} := \mathbf{b}_i - A_i \mathfrak{X}_j$
    $\mathfrak{R}^{iter} := r - \alpha^\kappa \mathbf{v}^\kappa$
    if $(\frac{||\mathfrak{X}_j||-||\mathcal{X}_j||}{min(||\mathfrak{X}_j||,||\mathcal{X}_j||)} > \epsilon_1)$ or $(\frac{||\mathfrak{R}^{exp}||-||\mathcal{R}^{exp}||}{min(||\mathfrak{R}^{exp}||,||\mathcal{R}^{exp}||)} > \epsilon_2)$ or $(\frac{||\mathfrak{R}^{iter}||-||\mathcal{R}^{iter}||}{min(||\mathfrak{R}^{iter}||,||\mathcal{R}^{iter}||)} > \epsilon_3)$ then
      Agent $i$ considers agent $j$'s communication at iteration $\iota$ as corrupted.
    end if
  end for
end for
```

B. Correction Phase

Once a transmission has been identified to contain a data error, a correction must be performed. Under traditional methods, this might require restarting the entire computation [28], however, we are able to utilize a more nuanced approach which reduces the amount of redundant calculations. We introduce a simple rejection approach, which performs well for the class of investigated data errors. The simplest form of correction is to ignore the updates associated with an iteration that has been flagged as corrupted. If a specific agent has been identified as the source of the corruption, it is possible to ignore the updates from that agent only.

VI. Numerical Experiments

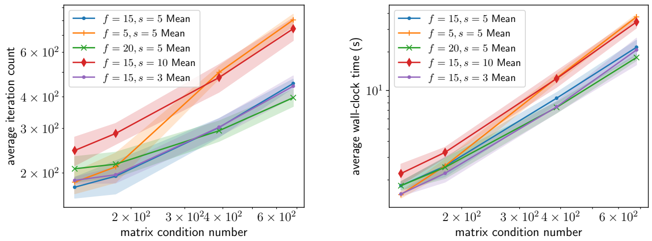
To demonstrate the performance of the newly proposed s-ACD and resiliency methods, a number of numerical experiments are conducted. These experiments are performed on a MacBook Pro (2019) with a 2.4GHz 8-Core Intel Core i9 CPU, and 64GB of 2667 MHz DDR4 Memory. Unless otherwise stated, experiments are run with four agents. When a 2D Poisson problem is used, it has homogeneous Dirichlet boundary conditions and is discretized with first order central finite differences with a right-hand side $g(x, y) = \sin(\pi x)\sin(\pi y)$ for a point at location (x, y). When a random SPD matrix is used, it has a condition number of 50, with a right-hand side defined by a vector of all ones.

Fig. 1 displays a variety of restarting frequencies f and s-step sizes s when using s-ACD over five runs, where the mean of the runs is plotted in a solid line and the 95% confidence interval displayed in the corresponding shaded region. We can see that having a large s-step size s relative to restart size f, such as the cases of $f = 15, s = 10$ and $f = 5, s = 5$ results in both a higher iteration count and a slower convergence. This is likely due to when the s-step size and restart sizes are similar, the number of full s sized s-step updates performed is small. There was not a significant difference in the other tested combinations. For the following tests, $f = 15$ and $s = 5$.

Fig. 2 demonstrates the scaling of the s-ACD method against the scaling of the asynchronous Jacobi and serial CG methods when changing the number of rows (and hence the condition number) of a 2D Poisson matrix with five runs, with the means plotted with solid lines and the 95% confidence interval plotted in the shaded region. We can see that our s-ACD method achieves better asymptotic scaling than the asynchronous Jacobi method, although not as good as serial CG, especially for larger condition numbers. The scaling achieved on systems with condition numbers in the range of 10^2 to 10^3 is comparable with CG, with a significant improvement in the absolute number of iterations compared to asynchronous Jacobi. The development of convergence theory could be used to better understand the results seen, and asynchronous preconditioning can improve the convergence further.

A 100×100 random SPD problem is used for Fig. 3, where different vectors are corrupted during communication, and the resulting l^2-norms are plotted for each agent, demonstrating the impact of these data errors on the metrics. The metrics are very noisy, due to the loss of orthogonality and independent nature of each agent. We see spikes around each restart, as the local residual and solution vectors are changed, potentially significantly, compared to the previous iteration. We observe that in all cases, the convergence slows down after the corruption at one second, while in the $\mathbf{x}, \mathbf{p}, \mathbf{r}$ cases, there is a significant spike in the observed metrics. This is because corrupting the $\mathbf{x}, \mathbf{r}, \mathbf{p}$ vectors (directly or indirectly) permanently destabilizes the subspace and has immediate consequences, while the $\mathbf{w}$ vector is recalculated at each iteration from the $\mathbf{p}$ vector.

The varying steps of the "generic" correction scheme are demonstrated in Fig. 4, which is a random SPD problem with a data error occurring after one second, separating out the global post-processed metrics from the metrics visible to each agent (Fig. 4a). We see that the global post-processed metric is very smooth, ignoring the spike at each restart, indicating that the overall behavior of the agents approximates is similar to the behavior of CG. We observe that the local metrics do correlate strongly with the trend of the global post-processed metrics, indicating that they can be used as a proxy for the overall convergence. While in synchronous CG, we would expect these metrics to be monotonically decreasing, the asynchronous algorithm removes these guarantees. Thus, the procedure of smoothing and successive differences must be used and through this adaptive method is able to detect when the corruption happens via local computations. It is clear that after these procedures (Fig. 4e) the aberration can be easily detected.



(a) Comparing by iteration count.

(b) Comparing by time.

Fig. 1: Comparing the scaling of different restarting frequencies f and s-step sizes s for s-ACD on a 2D Poisson problem discretized with finite differences with four agents and five runs proceeding until a tolerance of 1e-5 is reached. The shaded region represent a 95% confidence interval.

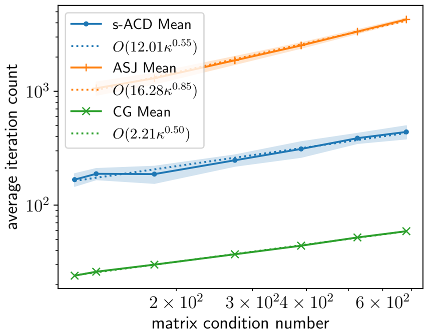
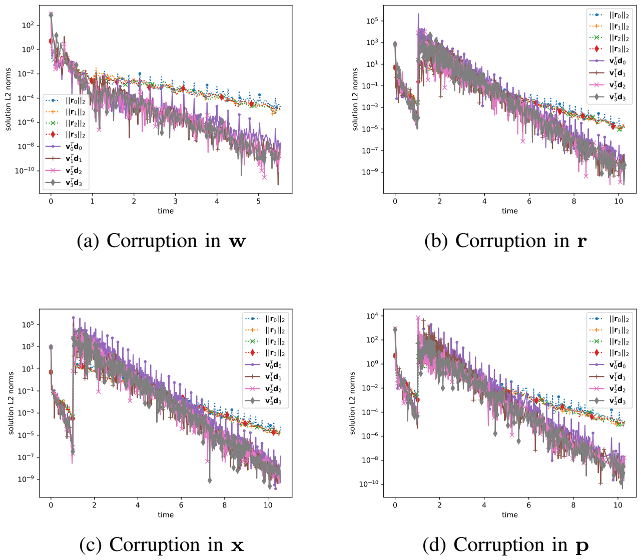


Fig. 2: Scaling of s-ACD vs asynchronous Jacobi (ASJ) and serial CG method on a 2D Poisson problem discretized with finite differences with four agents and five runs proceeding until a tolerance of 1e-5 is reached. The shaded region represents the 95% confidence interval.

Finally, Fig. 5 shows the residual of a 100×100 random SPD problem for four agents for the s-ACD method with and without the resiliency measures, for 30 runs with the mean of runs plotted as the solid line, while the dotted lines correspond to individual runs and the shaded region to the 95% confidence interval. By enabling all three sets of resiliency measure discussed above, the data errors are successfully detected and corrected. We can see that adding the resiliency measures annihilates the impact of the corruption, leading to four times faster convergence than without resiliency measures.



(a) Corruption in $\mathbf{w}$

(b) Corruption in $\mathbf{r}$

(c) Corruption in $\mathbf{x}$

(d) Corruption in $\mathbf{p}$

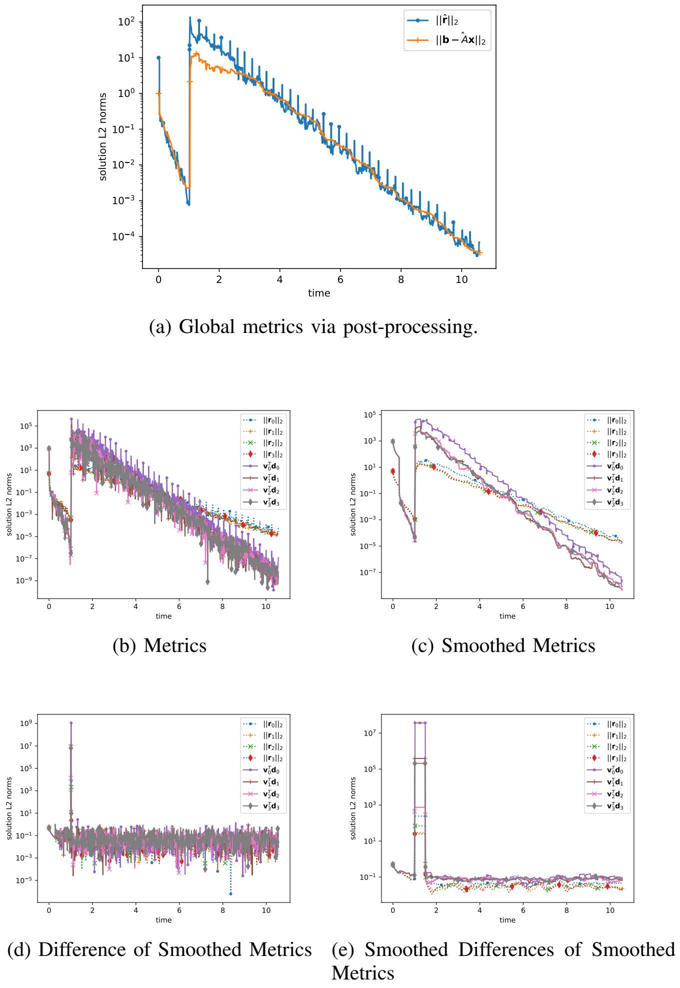
Fig. 3: Demonstrating the impact of errors introduced into different vectors of s-ACD after one second into all agents, where the metric of each agent is displayed. The problem considered is a 100×100 random SPD problem with condition number 50.

VII. Conclusion

We have seen that due to the increased parallelism and new computational paradigms, asynchronous and resilient methods should be developed. In this paper, we have developed the s-ACD method that combines the CD method globally with the CG method locally. This provides scaling with respect to the condition number comparable with CG on the tested 2D Poisson problem, while ensuring complete asynchronicity, as global orthogonalization is no longer required, as well as some resiliency. Furthermore, we developed three detection techniques: a "generic" detection scheme, a "checksum" detection scheme, and an algorithm-based detection scheme. These methods were applied to s-ACD, creating the resilient s-ACD method. Numerical experiments were performed to demonstrate that this resilient s-ACD method is able to handle the introduction of data errors into the communication pattern, resulting in a significant decrease of iterations compared to the uncorrupted case. Future improvements include developing theory for the s-ACD method and resilient variation, adding more elaborate correction methods such as rollback, as well as developing asynchronous preconditioners to allow the considered methods to scale to larger problems.

References

[1] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.
[2] C. T. Kelley, *Iterative methods for linear and nonlinear equations*. SIAM, 1995.
[3] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*. Oxford University Press, 2017.
[4] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006.



(a) Global metrics via post-processing.

(b) Metrics

(c) Smoothed Metrics

(d) Difference of Smoothed Metrics

(e) Smoothed Differences of Smoothed Metrics

Fig. 4: Different stages of the post-processing pipeline applied to the metrics when an error is introduced after one second into all agents for the s-ACD method (without resiliency measures). The problem considered is a 100×100 random SPD with condition number 50.

[5] J. J. Dongarra, I. S. Duff, D. C. Sorensen, H. A. Van der Vorst, and others, *Solving linear systems on vector and shared memory computers*. SIAM Philadelphia, 1991.
[6] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van der Vorst, *Numerical linear algebra for high-performance computers*. SIAM, 1998.
[7] Y. Saad, "Krylov subspace methods on supercomputers," *SIAM Journal on Scientific and Statistical Computing*, vol. 10, no. 6, pp. 1200–1232, 1989. doi: https://doi.org/10.1137/0910073
[8] C. Lanczos, "Solution of systems of linear equations by minimized iterations," *J. Res. Nat. Bur. Standards*, vol. 49, no. 1, pp. 33–53, 1952.
[9] M. R. Hestenes, E. Stiefel, and others, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.
[10] C. Ponce, K. Harter, A. Fox, and C. Vogl, "Skywing," [Computer Software] https://doi.org/10.11578/dc.20221110.2, nov 2022. [Online]. Available: https://doi.org/10.11578/dc.20221110.2
[11] E. C. Carson, "An adaptive s-step conjugate gradient algorithm with dynamic basis updating," *Applications of Mathematics*, vol. 65, no. 2, pp. 123–151, 2020. doi: https://doi.org/10.21136/AM.2020.0136-19

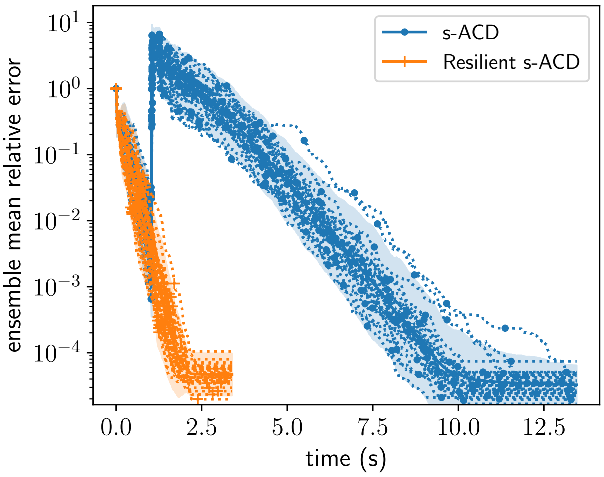


Fig. 5: Comparing the convergence of s-ACD ($f = 15, s = 5$) with and without correction where the $\mathbf{x}$ vector is corrupted on all four agents after one second for a random 100×100 SPD matrix with condition number 50 until a tolerance of 1e-5 is reached, averaged over 30 runs with the 95% confidence interval shown in the shaded region. The tail seen at the end is due to the agents ensuring that global convergence has been reached.

[12] A. Chronopoulos and C. Gear, "s-step iterative methods for symmetric linear systems," *Journal of Computational and Applied Mathematics*, vol. 25, no. 2, pp. 153–168, 1989. doi: https://doi.org/10.1016/0377-0427(89)90045-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042789900459

[13] S. Cools, J. Cornelis, P. Ghysels, and W. Vanroose, "Improving strong scaling of the conjugate gradient method for solving large linear systems using global reduction pipelining," *arXiv preprint arXiv:1905.06850*, 2019. doi: https://doi.org/10.48550/arXiv.1905.06850

[14] P. R. Eller and W. Gropp, "Scalable non-blocking preconditioned conjugate gradient methods," in *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2016. doi: https://doi.org/10.1109/SC.2016.17 pp. 204–215.

[15] M. Tiwari and S. Vadhiyar, "Pipelined Preconditioned s-step Conjugate Gradient Methods for Distributed Memory Systems," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021. doi: https://doi.org/10.1109/Cluster48925.2021.00061 pp. 215–225.

[16] M. Shantharam, S. Srinivasmurthy, and P. Raghavan, "Fault tolerant preconditioned conjugate gradient for sparse linear system solution," in *Proceedings of the 26th ACM international conference on Supercomputing*, 2012. doi: https://doi.org/10.1145/2304576.2304588 pp. 69–78.

[17] M. E. Ozturk, M. Renardy, Y. Li, G. Agrawal, and C.-S. Chou, "A Novel Approach for Handling Soft Error in Conjugate Gradients," in *2018 IEEE 25th International Conference on High Performance Computing (HiPC)*, 2018. doi: 10.1109/HiPC.2018.00030 pp. 193–202.

[18] A. Schöll, C. Braun, M. A. Kochte, and H.-J. Wunderlich, "Efficient online fault-tolerance for the preconditioned conjugate gradient method," in *2015 IEEE 21st International On-Line Testing Symposium (IOLTS)*, 2015. doi: 10.1109/IOLTS.2015.7229839 pp. 95–100.

[19] D. Chazan and W. Miranker, "Chaotic relaxation," *Linear algebra and its applications*, vol. 2, no. 2, pp. 199–222, 1969. doi: https://doi.org/10.1016/0024-3795(69)90028-7

[20] A. Frommer and D. B. Szyld, "On asynchronous iterations," *Journal of computational and applied mathematics*, vol. 123, no. 1-2, pp. 201–216, 2000. doi: https://doi.org/10.1016/S0377-0427(00)00409-X

[21] J. M. Bahi, S. Contassot-Vivier, and R. Couturier, *Parallel iterative algorithms: from sequential to grid computing*. CRC Press, 2007.

[22] J. Hook and N. Dingle, "Performance analysis of asynchronous parallel Jacobi," *Numerical Algorithms*, vol. 77, pp. 831–866, 2018. doi: https://doi.org/10.1007/s11075-017-0342-9

[23] J. Wolfson-Pou and E. Chow, "Convergence models and surprising results for the asynchronous Jacobi method," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2018. doi: 10.1109/IPDPS.2018.00103 pp. 940–949.

[24] ——, "Modeling the asynchronous Jacobi method without communication delays," *Journal of Parallel and Distributed Computing*, vol. 128, pp. 84–98, 2019. doi: https://doi.org/10.1016/j.jpdc.2019.02.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731518304751

[25] H. Anzt, J. Dongarra, and E. S. Quintana-Ortí, "Tuning stationary iterative solvers for fault resilience," in *Proceedings of the 6th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, 2015. doi: https://doi.org/10.1145/2832080.2832081 pp. 1–8.

[26] Apache Software Foundation, "Hadoop," 2021, version Number: 3.3.03. [Online]. Available: https://hadoop.apache.org

[27] ——, "Spark," 2021, version Number: 3.3.0. [Online]. Available: https://spark.apache.org

[28] A. Moody, G. Bronevetsky, K. Mohror, and B. R. De Supinski, "Design, modeling, and evaluation of a scalable multi-level checkpointing system," in *SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2010. doi: https://doi.org/10.1109/SC.2010.18 pp. 1–11.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 451–461
DOI: 10.15439/2023F6171
ISSN 2300-5963 ACSIS, Vol. 35

Risk-Based Continuous Quality Control for Software in Legal Metrology

Marko Esche, Levin Ho, Martin Nischwitz, Reinhard Meyer
Physikalisch-Technische Bundesanstalt,
Abbestraße 2-12, 10587 Berlin, Germany
Email: marko.esche@ptb.de, levin.ho@ptb.de, martin.nischwitz@ptb.de, reinhard.meyer@ptb.de

***Abstract*—Measuring instruments are increasingly defined by complex software while using simple hardware sensors. For such systems, software conformity between certified prototypes and devices in the field is usually demonstrated using version numbers and hashes over executable code. Legal requirements for regulated instruments could equally be satisfied if prototype and device in the field display identical functional behavior even if hashes differ. Such functional identification can give instrument manufacturers room for software patches and bugfixes without the need for recertification. Based on the L^* algorithm, which is used to learn the language which deterministic finite automata accept, a risk-based method is proposed that realizes automatic functional identification of software to a certain extent, thereby enabling quality control of regularly updated measuring instruments without the need for frequent manual inspections. Risk assessment may be used to identify critical state transitions in monitored devices, which can be used to trigger recertifications if needed.**

I. Introduction

MODERN communication infrastructure and the ubiquitous availability of significant computation power even in small devices like smart watches and smart sensors allow software developers to remotely and regularly fix bugs identified during use of an IT component in the field. The same mechanism can also be used to deliver upgraded software with new features to remote devices, enabling IT equipment manufacturers to sell devices based on long-living hardware which can be upgraded to customers' needs through software updates. However, this development also comes at a certain cost: Remote update capabilities have proven to introduce unexpected or unintended errors into otherwise stable systems [1]. Therefore, approaches to cover this gap (without forcing potential users of updated software to validate the complete source code of a device) have received significant attention in recent years [2], [3]. Such approaches enable users of IT equipment to monitor a device's behavior for potential anomalies without having to check each individual update, thus providing a high-level approach to identify software by means of its functionality rather than by means of its bit pattern.

Monitoring and identifying a device's functional behaviour becomes especially important if requirements on these devices are mandated by legal regulations, typically involving recertification of the entire system in case of modifications. One industry sector affected by such regulations is Legal Metrology covering all measurements conducted in the European Union for commercial or official use. These regulated instruments include, e.g., taximeters for taxi fare calculation, gas meters for measuring gas consumption and length measuring instruments to determine the dimensions of sold goods. Any such instrument put on the common EU market has to be subjected to a conformity assessment procedure according to Annex II of the Measuring Instruments Directive 2014/32/EU (MID) [4]. One conformity assessment body performing this task is Germany's national metrology institute Physikalisch-Technische Bundesanstalt (PTB).

During conformity assessment, manufacturers have to demonstrate that their instrument fulfills the essential requirements given in Annex I of the MID. During use, market surveillance authorities across the EU monitor devices and their usage to detect potential non-compliance. As an example, essential requirement 8.3 states that, ”Software that is critical for metrological characteristics shall be identified as such and shall be secured. Software identification shall be easily provided by the measuring instrument. Evidence of an intervention shall be available for a reasonable period of time.“ [4]. Not only does this entail the identifiability of software in general, but also the possibility to detect changes to said software and make them evident to all parties involved. Typically, both identifiability and detection of changes are achieved by using cryptographic hashes over the executable code to identify specific versions of the software and to detect unwanted modifications [5]. However, such an approach quickly may put serious strain on conformity assessment bodies and market surveillance authorities alike. For example, even a recompilation of otherwise unchanged source code my result in a different hash due to the inclusion of compile time stamps etc. Therefore, solutions that automatically evaluate software modifications and link them to a potential risk of non-compliance are needed.

To this end, a novel risk-focused method for remotely monitoring software in devices subject to legal control is proposed here. It is envisioned that the method will be used by market surveillance authorities and inspectors to automatically check certified devices in the field for potential non-compliant behavior. If a device is deemed to be in violation of legal requirements after a software modification, the manufacturer would then be requested to resubmit the modified software for a complete conformity assessment procedure. The main contributions of the paper are the following: The proposed method

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

constitutes a first step towards automatic remote quality control of devices subject to legal control. It enables automatic selection of risk scenarios based on remotely obtained behavioral data and thus also realizes functional identification of software to a certain extent.

The remainder of the paper is structured as follows: Section II provides some background on modelling and learning algorithms and presents the current state of the art in quality control for software as well as the risk assessment method currently used in Legal Metrology in the European Union. In Section III, the concept of modelling certain types of measuring instruments as deterministic finite automata (DFA) is investigated. The section also outlines which preconditions need to be fulfilled to justify such an approach. Afterwards, Section IV describes a novel risk-focused method of monitoring evolving software in measuring instruments based on the Active Continuous Quality Control (ACQC) approach from [2]. The method is then experimentally tested and evaluated in Section V. Finally, Section VI summarizes the paper and provides suggestions regarding further work.

II. Background and Related Work

Certain types of algorithms can be described as finite automata. The corresponding models and how to learn the behavior of such algorithms, which is of particular interest during monitoring of potentially modified software, are described in Section II-A. The methods proposed by Neubauer, Windmüller and Steffen together with Howar and Bauer [2], [3], which apply active automata learning to quality control for evolving systems, are briefly described in Sections II-B and II-C before recapitulating the previously published method of risk assessment for measuring instruments in Legal Metrology in Section II-D.

A. Active Automata learning

Simple state machines steered by input symbols from a finite alphabet that trigger internal state changes can be used to describe the behavior of certain types of algorithms such as used in controllers for elevators, household appliances, simple digital watches etc. [6]. From a mathematical point of view, these state machines, also referred to as DFAs, are defined as a 5-tuple $(Q, \Sigma, \delta, q_0, K)$ [6] where:

1) Q is a finite nonempty set of states.
2) Σ is a finite input alphabet.
3) $\delta : Q \times \Sigma \rightarrow Q$ is the transition function. (1)
4) $q_0 \in Q$ is the initial state.
5) $K \subset Q$ is the subset of accept states.

To indicate whether an arbitraty input sequence has successfully been processed, DFAs may contain accept states K which then trigger an *accept* message if such a state is reached. Otherwise, the output would be a *reject* message. It should be noted that the set K may also be empty, implying that the DFA does not contain any accept states triggering an *accept* message. In practice, the output of an algorithm is usually more complex than such binary feedback, requiring the existence of an output symbol from a finite output alphabet Γ. Such more general state machines are referred to as Mealy automata, which can be characterized as a 6-tuple $(Q, \Sigma, \Gamma, \delta, \gamma, q_0)$ [7] where:

1) Q is a finite nonempty set of states.
2) Σ is a finite input alphabet.
3) Γ is a finite output alphabet.
4) $\delta : Q \times \Sigma \rightarrow Q$ is the transition function. (2)
5) $\gamma : Q \times \Sigma \rightarrow \Gamma$ is the output function.
6) $q_0 \in Q$ is the initial state.

In addition to the transition function δ, describing state changes depending on the input symbol, these also possess an output function γ that associates an output symbol with each state change. Such Mealy automata were originally conceived to represent arbitrary logic circuits and can even mimic complex IT systems at a certain abstraction level [2]. For additional details, see Section II-B. To infer a DFA without having to know the exact implementation, the L^* algorithm was developed by Dana Angluin in 1987 [8]. The algorithm was later extended to the L^*_M algorithm to learn properties of the more general Mealy machines as well. Given that software changes in measuring instruments may have unknown effects and the instrument itself thus takes on the characteristics of a system with unknown behavior after an update, the basics of the L^* shall be briefly summarized here. See the original publication by Dana Angluin [8] for additional details of the L^* algorithm and the paper by Shahbaz and Groz [9] for an extended discussion including the L^*_M extension:

The aim of the L^* algorithm is to determine the properties of an unknown DFA by means of so-called membership and equivalence queries. To this end, the L^* learner communicates with a teacher T. The teacher abstracts the system under test (SUT), so that generic queries may be used by the learner to determine the SUT's internal DFA. If $L(A)$ is the set of strings a SUT A accepts, i.e., its language, and $Aut\,(A)$ is the set of all finite state machines with input alphabet Σ then the two types of generic queries used by the learner can be defined as follows:

- Membership queries $Q_M : \Sigma^* \rightarrow \{0,1\}$ where the learner asks the teacher to test the SUT with a given string x from the free monoid Σ^* that contains all words over Σ. If $x \in L(A)$ the response of the teacher is 1, otherwise 0.
- Equivalence queries $Q_E : Aut\,(\Sigma) \rightarrow \Sigma^* \cup \{\text{true}\}$ where the learner L^* asks the teacher T to perform an equivalence test between the current learned automaton representation $A' \in Aut(\Sigma)$ and the SUT A, resulting either in a counterexample $c \in \Sigma^*$ or confirmation of the equivalence.

Internally, the L^* algorithm operates on a so-called observation table that stores results of the queries in a systematic fashion. To this end, the learner continually performs membership queries until it has constructed an initial model A'. Subsequently, it issues an equivalence query to the teacher, which either confirms correspondence between A' and A or responds with a counterexample $c \in \Sigma^*$ that fulfills either $c \in L(A) \wedge c \notin L(A')$ or $c \notin L(A) \wedge c \in L(A')$. The algorithm finishes if the obtained data is sufficient to generate a system with the same algorithmic behavior as the SUT A. To illustrate the outcome of the L^* algorithm, an exemplary DFA is described in Section III together with a resulting transition function δ in tabular form in Table I.

B. Active Continuous Quality Control (ACQC)

In [2] Windmüller, Neubauer, Steffen, Howar and Bauer presented a novel approach for ensuring compliance of evolving complex applications through active automata learning technology. Their goal is to supervise and control modifications of applications during their entire life cycle. This is realized by establishing a consistent level for comparison via adaptive behavioral abstraction. Abstraction is achieved by means of a user-centric communication alphabet, where elements of the alphabet may correspond to entire (complex) use cases. One advantage of the method lies in its capability to identify bugs by simple examination of so-called ”difference views“ between consecutive models. The authors observe that software testing in general is not tailored to keep up with current, continuously evolving component-based software systems since repeatedly updating test suites for such systems is time-consuming and expensive. In [2] incremental active automata learning technology (also referred to as test-based modeling) is employed to address this issue.

To this end, daily system builds with an integrated fully automatic testing process are used, where the testing process is controlled by incremental active automata learning. The proposed approach aims to address the following main problems:

- ”Stable abstraction“: Downward compatibility is assumed, meaning users of the system should not change the way they interact with the system. Nevertheless, the source code etc. may change, but such modifications should not be apparent to the user. Therefore, the chosen abstraction mechanism is oriented on the level of use cases to facilitate comparisons between different software versions. Subsequently, the user-centric communication alphabet reflects distinct activities as part of the use cases.
- ”Bridging implementation“: A mechanism of the common abstraction level must ensure that any test is supported by a correct (version-dependent) implementation of an adapter for the symbols of the alphabet.
- ”Maximal reuse“: The central aspect of ACQC is based on the L^*_M learning algorithm for model inference. Based on selected counterexamples, the algorithm infers models from executed tests, see Section II-A. One drawback of the approach is the computationally expensive tests needed for the active learning process.

The authors observe that hypothesis models for new software releases are derived at the same level of detailedness as for the previous software release, which constitutes the main advantage of ACQC over similar approaches. Since identification of counterexamples is inherently ineffective, the derived system description will improve over time. Obviously, a precise initial model is needed to enable model-based testing. According to Windmüller, Neubauer, Steffen, Howar and Bauer, derivation of such models from source code is impractical for systems of a certain size. Indeed, any form of use-case-level modelling is difficult for such systems. Instead, active automata learning is used to extract models from live systems. The learned models then serve as the basis for regression tests. This approach will be reused in the method to be investigated here, see Section IV.

In [2] the proposed continuous quality control approach was validated by applying it to the Online Conference Service used for submitting and reviewing publications at Springer Verlag as an example with specific use cases as input symbols. Correspondingly, each input symbol represents processes like paper submission, reviewer selection or review submission. With such a high-level representation, a reasonably stable abstraction (as required above) was realized. The authors found that the chosen high-level modelling of input and output alphabets as abstraction of different use cases is well suited as a quality management facility for evolving IT systems. Not only is their method able to detect bugs, it also verifies if functional behavior of a system remains unchanged from one release to the next.

It should be noted, however, that the model learned by the L^*_M algorithm does not directly provide a link between the known set of states Q and the derived transition function δ. Instead, most L^* and L^*_M implementations assign input symbol sequences to the states they lead to. If the binary input 0 leads from a transition from the default empty state $\{\}$ to a state A, that state will be represented by the input sequence 0. If another input symbol 0 then leads to a transition from A to B, whereas the alternate input symbol 1 leads from A to C, B would be represented as 00 and C as 01. From a theoretical point of view, this corresponds to building the equivalence classes of the automata congruence relation for all states. It follows that an outside examiner can match the learned transition function δ against a known reference, but it is not guaranteed that the mapping between known states Q and learned states Q' is correct. This observation will be revisited again and illustrated by a more detailed example in Section V.

C. Risk-Based Testing via Active Continuous Quality Control

In [3] Neubauer, Windmüller and Steffen extended their approach to active automata learning and testing by adding a risk prioritization component. In this context, risk assessment is used to produce alphabet models which help to control the ACQC process to increase coverage of risk scenarios. The authors explain, that today’s complex IT systems usually consist of a combination of application servers with webinterfaces and third-party services. Due to the resulting heterogeneous structure, the subsequent system behavior becomes increasingly difficult to predict. During updates in particular, the mix of modified functionality and upgraded third-party components may have unintended effects. Their aim, therefore, was to continually perform automatic quality checks while using risk assessment to reduce the manual labor involved in regression testing.

In this regard, platform migrations are of particular interest since user experience may drastically change, even though the underlying functionality was not intended to be modified. Of course, potential risks resulting either from a platform change or from modified functionality cannot be automatically inferred. Therefore, the authors amended the original ACQC approach from [2] by enabling risk analysts to identify critical system aspects and prioritize them for error detection during the automatic model inference and checking steps. However, the paper [3] does not specify how risk levels are formally determined. Since risk analysts are typically not involved in software development itself, it becomes necessary to provide them with an abstraction layer that can be included in the original ACQC approach without performance loss. To this end, Neubauer, Windmüller and Steffen used the already abstract alphabet symbols from [2], which model different use cases of the SUT (see Section II-B).

The authors of [3] acknowledge that there are also model-driven approaches to risk-based testing such as the one described by Lund, Solhaug and Ketil Stølen in [10]. The so-called CORAS methodology offers the possibility to perform risk assessment using well-defined software models based on UML and the Unified Process (UP). However, CORAS and similar approaches only address the modelling aspect for risk assessment and are unable to monitor and perform comparisons between subsequent models of SUTs. Neubauer, Windmüller und Steffen also observe that it is unrealistic to assume that the internal number of states of a system will not change during its lifecycle. They therefore propose to continually repeat the learning process. This will be mirrored in the approach presented here, see Section IV.

D. Software Risk Assessment in Legal Metrology

One mandatory element of conformity assessment for measuring instruments consists of carrying out and evaluating a risk assessment for the instrument or type pattern to be assessed. In [11] Esche, Grasso Toro and Thiel described a method for software risk analysis particularly tailored for the software of such systems. The method is based on ISO 27005 [12] and ISO 18045 [13] and makes use of so-called assets, e.g., software, measurement data and parameters, and matching security properties, i.e., integrity, authenticity and availability, derived from the essential requirements from Annex I of the MID. These assets include the software, parameters and data of the instrument, but also the indication of the result, accompanying inscriptions and stored data. Within the frame of this paper, only the data during processing shall be considered. For such data, the MID requires integrity, authenticity and availability, i.e., it must be ensured that data cannot be modified or deleted without detection and that they are traceable to a known source. In short, any inadmissible influence on the processed data must be detectable.

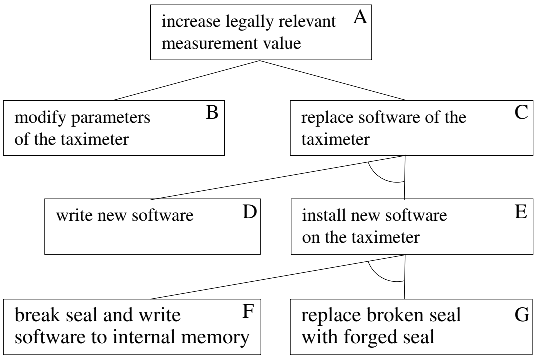

increase legally relevant measurement value A
modify parameters of the taximeter B
replace software of the taximeter C
write new software D
install new software on the taximeter E
break seal and write software to internal memory F
replace broken seal with forged seal G

Fig. 1. Graphical representation of an attack tree that illustrates necessary steps to manipulate the calculated fare of a taximeter during processing [14]. Child nodes must be read as OR-connected, unless they are connected by an arc, which represents an AND-connection [11].

The first step of a risk assessment then consists of formulating certain threats that constitute an invalidation of any security property for the assets. For example, such a formal threat might read, ”An attacker manages to invalidate integrity or authenticity of measurement data during processing.“ Given the known properties of the instrument, the assessor then identifies potential attack vectors which encompass practical technical steps to be implemented to realize the threat. Since such attacks tend to be made up of several steps which may even be shared between different threats, Esche, Grasso Toro and Thiel introduced the concept of Attack Probability Trees (AtPT) in [11]. An AtPT can be used to divide complex attacks into smaller subgoals by means of a tree representation, see Figure 1 for an example addressing attacks on the calculated fare of a taximeter. The AtPT method may be seen as an example of fault tree analysis (FTA) with an added layer that links the method to the vulnerability analysis from ISO 18045 enabling users of AtPTs to rank threats by means of a formalized and well-defined risk assessment. In the Figure, node A, which corrsponds to the threat of manipulating the measurement value, is divided into nodes B and C which represent the alternatives of either manipulating the measurement parameters or replacing the software of the instrument. These two child nodes may be split into further subtrees themselves. Once the tree has been established, all leaf nodes are assigned scores for required time, needed expertise, knowledge of the system, window of opportunity for an attacker and necessary equipment in accordance with the corresponding guidelines from ISO 18045 [13]. Finally these scores are propagated up the tree as prescribed by the rules from [11] to calculate probability of occurrence score and impact score of the original threat represented by the root node. The product of both, rounded to the next integer number, then becomes the (ideally) reproducible, numerical representation of the risk associated with the threat.

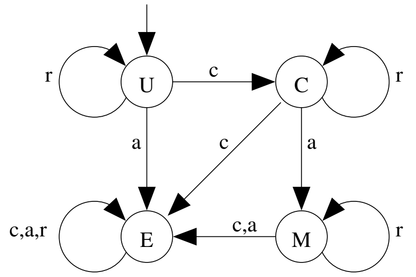


Fig. 2. DFA representing the different states of a heat meter and the state transitions. The heat meter states are: U for an unconfigured device, C for a configured yet inactive device, M for a measuring device and E for a an error state. The input alphabet consists of the symbol c for a configuration dataset, a for an activation signal and r for a request to retrieve measurement data from the device.

III. Modelling Measuring Systems as Deterministic Finite Automata

In principle, finding mathematical representations, i.e., functional identifications, even for simple measuring instruments is a complex task since various physical influences need to be taken into account and must be reflected in a corresponding uncertainty budget [15]. Automatic detection of unwanted behavior of complete instruments thus quickly becomes unfeasible. Nevertheless, many commonly used instruments, e.g., heat meters, typically contain internal state machines which ensure that the instrument behaves differently during installation/configuration than during permanent use. Among other qualities, heat meters have to guarantee that the installation point (either on supply side or return side of a heat generating device) can only be set during configuration and that the state cannot be reached again without physically tampering with the device. From this example, it should be clear that state machines within such instruments share many properties with DFAs and can thus be used to provide a simple form of high-level functional identification.

In the simple heat meter example, $Q = \{U, C, M, E\}$ would consist of the states U for an unconfigured device, C for a configured yet inactive device, M for a currently measuring device accumulating the consumed energy into a register and E for a device in an error state. A simple input alphabet would consist of three symbols $\Sigma = \{c, a, r\}$ where c represents a configuration datagram, a is the activation signal and r is a request to read measurement data from the device. For illustration purposes, a graphical representation of the complete DFA is given in Figure 2. It should be noted that the depicted DFA is only a simplistic exemplary representation of the possible software states of a heat meter. A real device will likely contain more states and more possible transitions. Also, the shown DFA only addresses the software aspects of the meter. For instance, if the permanent error state E is reached, recovery might still be possible via a hardware reset which is beyond the representation capabilities of the selected model. The corresponding transition function δ, which maps a current state to the next state given a specific input symbol, is shown in Table I. There exist some approaches to model both, the measuring function and the DFA of measuring instruments, resulting in a so-called digital twin describing an instruments behavior under arbitrary conditions. However, these are not suitable to monitor and evaluate frequent software changes. As stated in Section I, mechanisms are needed that can automatically identify and evaluate software modifications. It should be clear from the above-mentioned example for heat meters, that some measuring instruments contain state machines that control the interpretation of sensor data to produce the measured quantity value.

TABLE I

Transition function δ for the heat meter example in Fig 2. State U represents an unconfigured device, C a configured yet inactive device, M a measuring device and E a device in an error state. c is the input symbol for a configuration dataset, a for an activation signal and r for a request to retrieve measurement data from the device.

		symbol		
		c	**a**	**r**
state	**U**	C	E	U
	C	E	M	C
	M	E	E	M
	E	E	E	E

While certain measuring instruments include internal DFAs controlled by external input [5], such instruments usually also produce variable output data - namely the measurement result - either in a digital or visual representation. Therefore, such systems fulfill the criteria of the more general Mealy automata. Nevertheless, as illustrated by the heat meter example above, many measuring instruments already contain simple DFAs enabling the use of the original L^* algorithm without having to define additional output alphabets and resorting to the correspondingly more complex L^*_M algorithm for Mealy automata. This approach also mirrors the fact that evaluation of software security aspects in measuring instruments and evaluation of the measurement functionality are usually two separate tasks during conformity assessment of such devices. Section IV will revisit this aspect when elaborating on a possible quality control strategy for measuring instruments in the field.

IV. Risk-Based Continuous Quality Control for Measuring Systems

In [3], the authors used risk assessment to prioritize the input alphabet for the L^*_M algorithm applied to a Mealy machine to ensure quick detection of potential implementation or migration errors in evolving IT systems. In the scenario where software is updated in measuring instruments subject to legal control, a little more flexibility might be possible given that mere bugfixes, which do not affect the functionality of the instrument, should be covered by the original conformity assessment certificate without the need to revise the certificate. To achieve this, the focus shall not be put on the choice of the input alphabet but rather on the state transitions δ discovered by executing the L^* algorithm for a new or unknown system. A graphical representation of the automatic quality control method proposed here may be found in Figure 3.

As discussed in Section II-D, performing and evaluating a software risk assessment has become an integral part of conformity assessment for most measuring instruments in the EU. During such an assessment, the risks assigned to individual threats or their subgoals can be used to derive a list of critical state transitions that the evaluator deems to be in violation or facilitate violation of the essential requirements from the MID, see top-left corner of Figure 3. If necessary, the numerical risk scores for individual threats described in Section II-D could be used to rank new state transitions according to their risk level. In the heat meter example from Section III, one such critical transition would be reverting from measurement state M back to the configuration state C, potentially leading to modified measurement parameters while a device is in use. The conformity assessment procedure could also be used to perform an initial execution of the L^* algorithm in a known environment. The initially discovered transition function δ and the known remaining elements of the DFA shall together be referred to as the model M_{old}. Continuous repeated learning of the DFA (right-hand side of Figure 3) will produce potentially modified models M_{new} which can be compared against the previously learned and accepted model taking into account the identified list of critical state transitions. As long as no critical transition is identified, the learning loop could be repeated indefinitely to ensure that the system still operates within certified functional limits. The updated model representation also allows human evaluators to graphically identify the recent software changes and determine their potential effect. Of course, model comparison only allows inspection of the internal DFAs of measuring instruments, neglecting to address the measurement function itself. However, this approach is also used in many conformity assessment bodies in the EU where software examination (focussed on the IT security of examined prototypes) and metrological examination of the measurement functionality itself (addressing measurement uncertainty, reproducibility etc.) are two separate tasks usually conducted by two separate examiners. Therefore, it appears justifiable to monitor changes to the protection and security measures, e.g., the order of transitions within internal DFAs, separately from the measurement function itself.

If a critical modification is detected (if-then-statement in the lower right corner of Figure 3), a manual intervention is needed. In order to revert to a certified state, conformity assessment for such a modified instrument must be repeated. If problems are identified with the modified instrument during reassessment, potential corrective actions regarding improper use of non-conformant measuring instruments may be required. The workflow of the procedure will be illustrated by a detailed example in Section V. Depending on the complexity of the automaton, learning its representation can be computationally expensive. Since measuring instruments usually possess rather simple DFAs and devices like taximeters are usually inactive for longer periods on a daily basis, applying the L^* algorithm to such instruments still appears feasible.

It should be noted that neither L^* nor L^*_M work in actual black-box scenarios. Instead, they require the existence of a teacher T who has full access to the SUT and can answer membership and equivalence queries accordingly, see Section II-A. To this end, it is envisioned that such a teacher T might be developed by the instrument manufacturer and evaluated during initial conformity assessment. Subsequently, the teacher T could then act as a test interface for market surveillance and inspectors, enabling them to continually monitor individual device in the field remotely until the need for manual intervention arises.

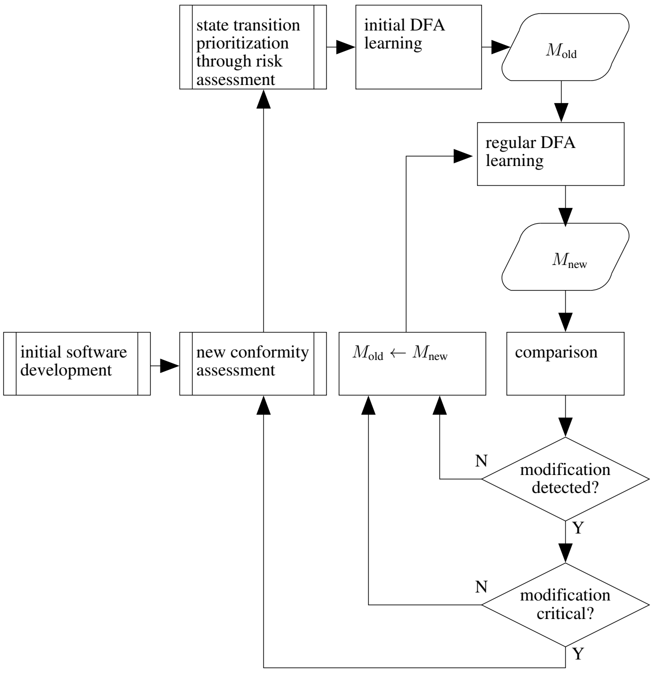

state transition prioritization through risk assessment
initial DFA learning
M_{old}
regular DFA learning
M_{new}
initial software development
new conformity assessment
$M_{\text{old}} \leftarrow M_{\text{new}}$
comparison
N
modification detected?
Y
N
modification critical?
Y

Fig. 3. Anticipated workflow of the risk-based testing approach. The initially learned model M_{old} is continually compared with newly learned models M_{new}, unless the comparison between both models identifies a critical state transition.

V. Exemplary Evaluation

To illustrate the usage of the proposed risk-based ACQC workflow for measuring instruments, a real-world exemplary instrument will be examined in detail in Section V-A, followed by an investigation into different types of new state transitions in Section V-B and a discussion regarding discovery of unknown states in Section V-C. An analysis of the example that also identifies open issues of the approach will be provided in Section V-D.

A. Taximeter as a complex DFA

A taximeter (as defined in Annex IX of the MID [4]) is a ”device [that] measures duration, calculates distance on the basis of a signal delivered by the distance signal generator. Additionally, it calculates and displays the fare to be paid for a trip on the basis of the calculated distance and/or the measured duration of the trip.“ Therefore, the sensor is not part of this type of measuring instrument and it solely performs processing operations on the received digital distance data. This makes

TABLE II
TRANSITION FUNCTION δ FOR THE TAXIMETER EXAMPLE IN FIG 4. STATE F REPRESENTS A FREE VEHICLE WITH NO PASSENGER, O AN OCCUPIED VEHICLE AND M AN ONGOING MEASUREMENT. I REPRESENTS RETRIEVAL OF FISCAL DATA AND U A SOFTWARE UPDATE. SYMBOL s SIGNIFIES THE START OF A MEASUREMENT, e SIGNIFIES EXITING A STATE, i INITIALIZES A FISCAL REVIEW AND u CORRESPONDS TO A SOFTWARE UPDATE PACKAGE.

		symbol			
		s	**e**	**u**	**i**
state	**F**	O	F	U	I
	O	M	F	O	I
	M	M	F	M	M
	I	I	F	U	I
	U	U	F	U	U

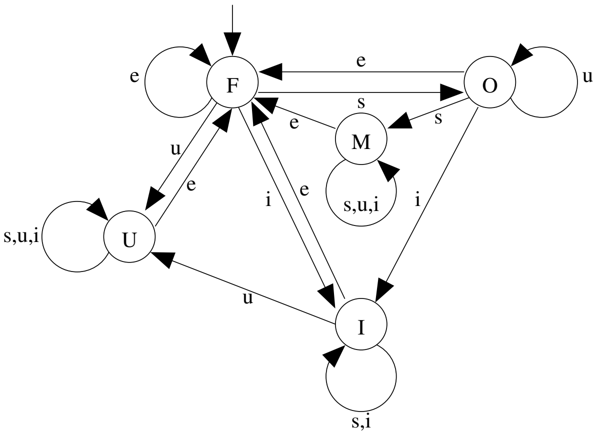


Fig. 4. DFA representing the different states of a taximeter and the state transitions. The taximeter states are F for a free vehicle with no passenger, O for an occupied vehicle, M for an ongoing measurement, I for retrieval of fiscal data and U for a software update. The input alphabet consists of the symbol s to start a measurement, e to exit a state, i to initialize fiscal review and u for a software update package. In the original model of the DFA shown here, the instrument will always return to the default state F after completing a software update in state U.

taximeters especially suitable as a test case for the proposed method, see Section III.

Since taxis have frequently changing customers, they usually possess DFAs that mirror the process of a customer entering and leaving a vehicle as well as the starting and stopping of the measurement itself. Subsequently, said DFAs contain a state F that represents a free vehicle, whereas the DFA enters the state O to signal that the taxi is now occupied. This could either be triggered through a button on the device or by means of a seat contact. For the sake of a simple example, it shall be assumed that the price per travelled kilometer is fixed. It should be noted, however, that some EU member states have complex tariff structures that take current time, number of passengers etc. into account. If the occupied vehicle starts travelling, the internal DFA then enters the measuring state M. To leave the state, the customer must first pay the price, after which the driver pushes the corresponding button to exit the measurement state. In addition, most taximeters also possess an option to retrieve fiscal data, such as the overall total of calculated fares and the complete travelled distance. Both are needed to perform tax audits for taxi companies. The corresponding fiscal inspection state I can be entered if no measurement is running and it should not be possible to start a measurement from this state. Finally, some taximeters possess a functionality to perform software updates. This functionality shall be represented by a state U. A use case oriented input alphatbet would then consist of the symbol s to start a measurement or transition from the free state F to the occupied state O. The symbol e correspondingly signals the exiting of the current state and return to the default free state F. Input symbols i for fiscal inspection and u for a software update indicate the command to either perform an inspection or trigger a remote update. The corresponding graphical representation of the complete DFA may be found in Figure 4. The corresponding transition function δ, which maps a current state to the next state given a specific input symbol, is shown in Table II. As indicated in Section II-D, all measuring instruments must be subjected to a risk assessment as part of the necessary conformity assessment procedure before putting such instruments on the common European market. Figure 1 shows the attack probability tree as one outcome of the risk assessment procedure for a taximeter's software. When comparing the attack probability tree with the example described above, it should become clear that child node B (modification of a taximeter's parameters) cannot be linked to the transition function δ in Table II since there is no corresponding state that enables parameter changes. Child node C (replacing the software of a taximeter), however, could be enabled by inadmissible transitions to and from the update state U. In fact, node E (installing new software) addresses specifically the functionality behind the update state. In this context, one should keep in mind that breaking and replacing of the seal (represented by child nodes F and G) do not necessarily have to address physical hardware seals. So-called electronic seals realized as protected logbooks are equally common in Legal Metrology [5]. Subsequently, all additional transitions to and from the update state U (represented by the detected state su in Table III) would be classified as critical during conformity assessment since such transitions could interfere with normal processing of updates and damage the continuous audit trail of logged software modifications.

TABLE III
TRANSITION FUNCTION δ OBTAINED BY L^* ALGORITHM FOR THE ORIGINAL TAXIMETER EXAMPLE FROM FIGURE 4. STATES ARE GIVEN IN THE REPRESENTATION OBTAINED BY THE ALGORITHM, E.G., ssu, TOGETHER WITH THEIR CLEARTEXT REPRESENTATION, E.G., M.

		symbol			
		s	**e**	**u**	**i**
state	**s/F**	ss	s	su	si
	ss/O	sss	s	ss	si
	sss/M	sss	s	sss	sss
	si/I	si	s	su	si
	su/U	su	s	su	su

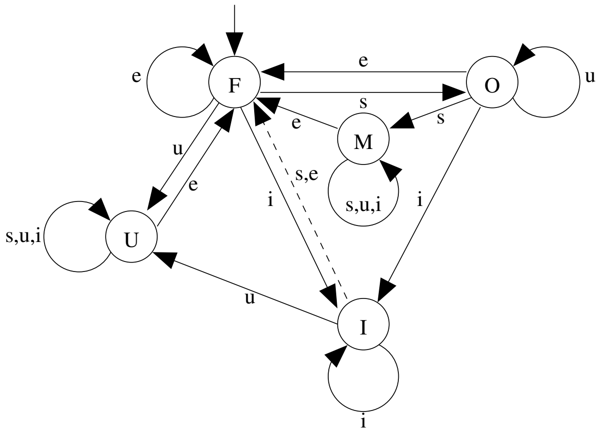


Fig. 5. DFA representing the different states of a taximeter and the state transitions after addition of an non-critical state change from fiscal inspection I to the free state F (dashed arrow). While originally only the input symbol e for exiting triggered that change, symbol s now has the same effect. In the orignal example, symbol s had no effect on the automaton when in state I.

As explained in Section II-B the L^* algorithm produces a transition function δ that references the internal states of the examined DFA by their corresponding input sequences. To improve readability of the example, the first column of Table III contains both the cleartext representation of the states as well as their representations obtained by the learning algorithm, which correspond to the symbol sequences needed to transition to a certain state. Since s is the first symbol tested by thevused algorithm, it denotes the default state F also with that symbol. Consequently, all other state representations start with that symbol, too. Section V-C will address how representation variations may affect the interpretation of the algorithm output and how this effect can be mitigated.

B. Non-critical and critical state changes

To test the proposed automatic detection method, the DFA of the taximeter shall now be modified by adding another transition from state I for fiscal inspection to the free state F triggered by the input symbol s (originally only triggered by symbol e), see Figure 5. Although this transition no longer matches the original assignment linked to that input symbol, it does not constitute a critical modification from the point of view of conformity assessment. Following the learning cycle proposed in Figure 3, the L^* algorithm is applied to the modified DFA resulting in a new version of the transition function δ, see Table IV. As can be seen from the table, the state representation obtained by the L^* algorithm remains the same, e.g., state M is still represented by the input symbol sequence sss. The only difference between the original transition function (see Table III) and the updated version in Table IV may be found in the row for transitions from state si/I, where the input symbol s now triggers a return to state s/F. Since an added transition to this state was deemed uncritical during conformity assessment, monitoring of the system can be continued without the need for human intervention.

TABLE IV
TRANSITION FUNCTION δ OBTAINED BY APPLICATION OF THE L^* ALGORITHM TO THE TAXIMETER EXAMPLE FROM FIGURE 5 WITH A MODIFICATION THAT ENABLES A SECOND TRANSITION FROM I TO F. THE CORRESPONDING NEW STATE TRANSITION IS UNDERLINED.

		symbol			
		s	**e**	**u**	**i**
state	**s/F**	ss	s	su	si
	ss/O	sss	s	ss	si
	sss/M	sss	s	sss	sss
	si/I	$\underline{\text{s}}$	s	su	si
	su/U	su	s	su	su

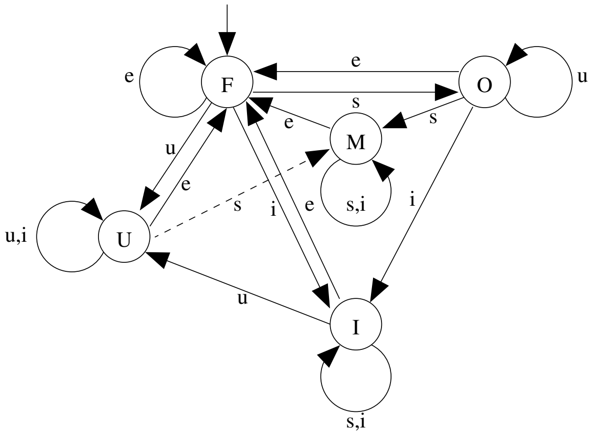


Fig. 6. DFA representing the different states of a taximeter and the state transitions after addition of a critical state change directly from the update state U to the measurement M (dashed arrow) if the input symbol s is received.

As a second test case, the original taximeter DFA shall now be modified by adding a state change between update state U and measurement state M, see Figure 6. Such a transition was deemed critical during initial assessment of the measuring instrument and should trigger an automatic response. The corresponding function δ learned after application of the L^* algorithm to the modified example is given in Table V. Again, the linking between cleartext names of the states and the representations found by the algorithm appears to be unchanged. However, as can be seen from Table V, the transition function δ now also reflects the intended additional transition from the

TABLE V
TRANSITION FUNCTION δ OBTAINED BY APPLICATION OF THE L^* ALGORITHM TO THE TAXIMETER EXAMPLE FROM FIGURE 6 WITH A MODIFICATION THAT ALLOWS SWITCHING TO MEASUREMENT STATE M IMMEDIATELY AFTER A SOFTWARE UPDATE (REPRESENTED BY STATE U).

		symbol			
		s	**e**	**u**	**i**
state	**s/F**	ss	s	su	si
	ss/O	sss	s	ss	si
	sss/M	sss	s	sss	sss
	si/I	si	s	su	si
	su/U	$\underline{\text{sss}}$	s	su	su

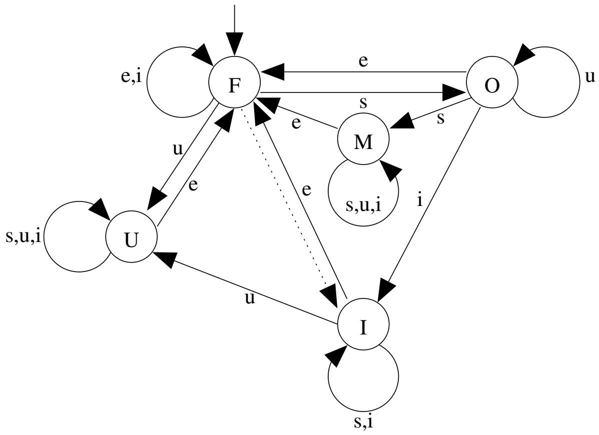


Fig. 7. DFA representing the different states of a taximeter and the state transitions after removing a state transition from free state F to the fiscal inspection state I (dotted arrow). Instead, the DFA remains in state F if an input symbol i is received in that state.

update state U (represented by the symbol sequence su in the table) to the measurement state M (represented by the symbol sequence sss). Since any additional transition to and from the update state was classified as critical during the original risk assessment (see Section V-A), the algorithm now issues a warning that triggers a repetition of the conformity assessment procedure to check whether the modified instrument still complies with legal regulations. As part of the repeated assessment, the risk analysis would also be performed and evaluated again. During this step, the classification of critical state changes might have a different outcome because of additional information not available during original assessment. If the modified software were deemed acceptable, the proposed quality assurance algorithm would be supplied with a new list of critical state changes and the L^* algorithm would be started again. If not, manual withdrawal of all affected taximeters in the field would become necessary.

C. Necessary discovery of state correspondences

As indicated in Sections II-B and V-A, the state representations by their corresponding input symbol sequences within the transition function δ obtained by the L^* algorithm depend on the order in which states are discovered. To illustrate this fact, a modified version of the original taximeter DFA shall be used, where the state transition from free state F to the fiscal inspection state I has been removed, see Figure 7. The corresponding state transitions identified by the L^* algorithm may be found in Table VI. Due to the different order of state discovery, the fiscal inspection state I is now no longer referenced as si but rather as ssi in the table. Since such an assignment of a different label could potentially affect more than one state, it becomes necessary to add a matching step to the comparison step between consecutive learned models $M_{\text{old}}, M_{\text{new}}$ included in the proposed workflow in Figure 3. For the sake of simplicity, the matching step shall consist of checking all possible assignments between cleartext representations of DFA states and corresponding symbolic state representations from Table II. The one assignment that minimizes the number of new or modified state transitions compared to the original DFA shall then be assumed to be correct and the identified transitions shall be evaluated against the list of critical state changes from the risk assessment. If there is more than one assignment that minimizes the number of new or modified state transitions, the state assignment is no longer unambiguous and the modification will be assumed to be critical by default. This approach will only fail under two conditions: If the overall number of discovered states does not match the original DFA or if sufficiently many state changes have been implemented by the manufacturer so that the learned transition function matches the original one, even if the underlying functionality is different. Both cases will be revisited in Section V-D.

TABLE VI
TRANSITION FUNCTION δ OBTAINED BY APPLICATION OF THE L^* ALGORITHM TO THE TAXIMETER EXAMPLE FROM FIGURE 7 AFTER DELETING ONE OF THE ORIGINAL STATE TRANSITIONS FROM F TO I.

		symbol			
		s	**e**	**u**	**i**
state	**s/F**	ss	s	su	s
	ss/O	sss	s	ss	ssi
	sss/M	sss	s	sss	sss
	ssi/I	ssi	s	ss	ssi
	su/U	su	s	su	su

D. Analysis of the Example

When comparing the transition functions obtained by the L^* algorithm for the non-critical and critical modifications of the taximeter DFA (see Tables IV and V respectively), it can be seen that the proposed risk-based quality control approach can effectively identify and deal with both types of modifications. Manual intervention as the result of a detected assumed critical change will likely be able to assess the actual impact of the modifications and ensure compliance of all serial devices in the field. The monitoring approach might fail, however, if several state transitions are modified or added iteratively so that they are only examined individually by the L^* algorithm. Even if the combination of modifications or additions produces effects that are in violation of legal requirements, the current implementation would not be able to detect these effects. However, this scenario is implicitly already covered by today's practice of performing periodic reverifications of measuring instruments in use. As outlined in Section IV, the manufacturer of the measuring instrument would need to implement a teacher in the form of a test interface for the proposed approach to work. Of course, it cannot be guaranteed that such an interface actually interacts with the internal DFA of the measuring instrument. Instead, a dummy DFA could be implemented to hide software modifications from the automatic quality checker. During the above-mentioned reverifications, however, it would be possible to also practically check whether the implemented teacher T

correctly abstracts the measuring instrument's DFA for the external Learner L^*, thereby mitigating such a threat. As illustrated in the example in Section V-C, reproducibility of the L^* algorithm's output depends on the context-based interpretation of learned state labels. The proposed brute-force matching algorithm to identify correspondences between cleartext state representations and learned state identifiers has several shortcomings which shall be addressed here:

- While a brute-force approach, matching all DFA states against all possible representations, is guaranteed to find one or more optimal matches, the approach might become computationally complex if large DFAs are monitored. Breadth-first search algorithms should be able to provide quicker solutions without missing any transition modifications.
- As discussed in Section V-C, the number of discovered states does not necessarily have to match the number of states in the original DFA, even after application of the DFA minimization algorithm. In such a case, the currently investigated approach would always classify the modification as critical, even if a state has been removed that is not legally regulated.
- It is theoretically possible to implement sufficiently many state changes simultaneously that cannot be detected because the learned transition function δ contains the same number of states and matching state transitions as the original transition function.

It should be noted again that the current approach only focuses on simple state transitions within DFAs while the behavior of more complex instruments than heat meters or taximeters will likely be better characterized by the more general Mealy automata, see Section III. Using Mealy automata would enable checking of input and ouput behavior of such systems, thus ensuring a wider range of useful application scenarios. Investigation into an approach using the adapted L^*_M algorithm will, therefore, form the basis for further work. Similarly, machine learning algorithms such as the one described by Yan, Tang, Luo, Fu, and Zhang in [16] are already able to perform anomaly detection for complex IT systems. Due to the similarities between such systems and measuring instruments, similar approaches might also be able to model and monitor the software of measuring instruments to some extent, while potentially bridging the gap between automata models and mathematical models for measurements themselves. Once more elaborate quality control approaches for software in measuring instruments are available and have proven their reliability, it might be possible to replace mandatory periodic reverifications with risk-based reverifications based on the detected behavior of individual devices. If proven useful, such quality control approaches could be added as an acceptable solution for dealing with software modifications in the currently established technical interpretation of the MID, namely the WELMEC 7.2 Software Guide [5]. Such an acceptable solution could facilitate the uptake of the method and harmonize the approach across the EU if needed.

VI. Summary

In this paper, a new risk-based quality control approach for measuring instruments in legal metrology was proposed as a high-level attempt to realize functional identification for software of such systems. The approach is based on work published in [2] as well as [3] and uses the L^* algorithm to monitor changes in the DFAs of measuring instruments in the field. To this end, the outcome of the mandatory risk assessment procedure for regulated measuring instruments is used to identify critical state transitions to be checked if software changes occur. Based on an example for a DFA in a taximeter, the approach was evaluated regarding the detection of non-critical and critical state changes, even in light of varying conditions like modified state representations. To mitigate potential effects of varying state representations, a brute-force matching algorithm was added to the proposed method that can effectively reduce the number of falsely identified critical state transitions. This proof of concept has shown that automatic quality control of measuring instruments is indeed possible if the SUT fulfills certain preconditions, such as a clear separation between measurement function and internal DFA. Manual intervention in case of doubt and periodic reverifications are still necessary to cover all eventualities. While the method requires instrument manufacturers to implement a test interface in their devices, they would benefit from the possiblity of issuing bugfixes to their software without having to go through conformity assessment by default. Similarly, conformity assessment bodies would have to check said interfaces initially, but would benefit when updates are deemed to be in line with the originally certified instrument functionality, thus avoiding repetition of software examinations. Finally, market surveillance authorities and inspectors in Legal Metrology could use the data provided by the L^* algorithm to assess modifications in devices in the field to a certain extent without the need to be on site. It is envisioned that the approach would work in any industry sector where software systems are used whose compliance with specific requirements must be checked by external authorities in the field. Further work will focus on validating the current approach with additional, more realistic practical test cases (also outside legal metrology) and optimizing the matching algorithm between learned and known state representations. Extending the approach from DFAs to more general Mealy automata will hopefully pave the way towards an actual functional identication mechanism for software in measuring instruments since it would also encompass the output language of devices in the field rather than simply monitor state transitions.

References

[1] M. Jang, *Linux Patch Management: Keeping Linux Systems Up To Date*, 1st ed. Prentice Hall, Jan. 2006. ISBN 978-0132366755

[2] S. Windmüller, J. Neubauer, B. Steffen, F. Howar, and O. Bauer, "Active continuous quality control," in *Proceedings of the International Symposium on Component-Based Software Engineering*. ACM, Jun. 2013. doi: 10.1145/2465449.2465469 pp. 111–120.

[3] J. Neubauer, S. Windmüller, and B. Steffen, "Risk-based testing via active continuous quality control," *International Journal on Software Tools for Technology Transfer*, vol. 16, pp. 569–591, 2014. doi: 10.1007/s10009-014-0321-6

[4] EC, "Directive 2014/32/EU of the European Parliament and of the Council of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of measuring instruments," European Union, Council of the European Union; European Parliament, Directive, February 2014.

[5] "WELMEC 7.2 Software Guide," European cooperation in legal metrology, WELMEC Secretariat, Braunschweig, Standard, Mar. 2022.

[6] M. Sipser, *Introduction to the theory of computation*, 2nd ed. Boston, Massachusetts: Thomson, 2006. ISBN 0-534-95097-3

[7] G. H. Mealy, "A method for synthesizing sequential circuits," *The Bell System Technical Journal*, vol. 34, no. 5, pp. 1045–1079, 1955. doi: 10.1002/j.1538-7305.1955.tb03788.x

[8] D. Angluin, "Learning regular sets from queries and counterexamples," *Information and Computation*, vol. 75, no. 2, pp. 87–106, 1987. doi: 10.1016/0890-5401(87)90052-6

[9] M. Shahbaz and R. Groz, "Inferring mealy machines," in *FM 2009: Formal Methods*, A. Cavalcanti and D. R. Dams, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-05089-3_14. ISBN 978-3-642-05089-3 pp. 207–222.

[10] M. S. Lund, B. Solhaug, and K. Stølen, *Model-Driven Risk Analysis - The CORAS Approach*. 0314 Oslo, Norway: Springer, 2011. ISBN 978-3-642-12323-8

[11] M. Esche, F. Grasso Toro, and F. Thiel, "Representation of attacker motivation in software risk assessment using attack probability trees," in *Proceedings of the Federated Conference on Computer Science and Information Systems*, Prague, Czech Republic, September 2017. doi: 10.15439/2017F112 pp. 763–771.

[12] ISO/IEC, "ISO/IEC 27005:2011(e) Information technology - Security techniques - Information security risk management," International Organization for Standardization, Geneva, CH, Standard, June 2011.

[13] ——, "ISO/IEC 18045:2008 Common Methodology for Information Technology Security Evaluation," International Organization for Standardization, Geneva, CH, Standard, September 2008, Version 3.1 Revision 4.

[14] M. Esche and F. Grasso Toro, "Developing defense strategies from attack probability trees in software risk assessment," in *Proceedings of the Conference on Computer Science and Information Systems*, 2020. doi: 10.15439/2020F21 pp. 527–536.

[15] "Guide to the expression of uncertainty in measurement - part 6: Developing and using measurement models," Joint Committee for Guides in Metrology (JCGM), BIPM, Sèvres Cedex FRANCE, techreport, Mar. 2020.

[16] S. Yan, B. Tang, J. Luo, X. Fu, and X. Zhang, "Unsupervised anomaly detection with variational auto-encoder and local outliers factor for kpis," in *2021 IEEE Intl. Conf. on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking*. IEEE, 2021, pp. 476–483.

DOI: 10.15439/2023F6694
ISSN 2300-5963 ACSIS, Vol. 35

Classifying Industrial Sectors from German Textual Data with a Domain Adapted Transformer

Richard Fechner*†, Jens Dörpinghaus*‡, Anja Firll*
* Federal Institute for Vocational Education and Training (BIBB), Bonn, Germany,
Correspondence: richard.fechner.nr@gmail.com,
jens.doerpinghaus@bibb.de, https://orcid.org/0000-0003-0245-7752, anja.firll@bibb.de
† University of Tübingen, Germany
‡ University Koblenz, Koblenz, Germany

***Abstract*—For economics and sociological research, lists of industries and their branches are widely used in research to categorize data and get an overview on different types of industries. However, many different taxonomies and ordering schema exist, due to different research focus but also due to different national scenarios and interests. In this paper, we will focus without loss of generality on regional data from Germany. Manual annotation of textual data is time-consuming and tedious, naturally giving rise to our initial research question, also highly inspired by questions from computational social sciences: How can we automatically categorize textual data, e.g. job advertisements or business profiles, by industrial sectors? We will present an approach towards classification using a pre-trained domain-adapted Transformer model. We find that domain-adapted models generalize better and outperform state of the art non domain-adapted Transformer models on Out-Of-Distribution data. Additionally, we open source two novel data-sets mapping textual data to WZ2008 sections and divisions, enabling further research.**

I. Introduction

FOR economics and sociological research, lists of industries and their branches are widely used in research to categorize data and get an overview on different types of industries. However, many different taxonomies and ordering schema exist, due to different research focus but also due to different national scenarios and interests.

The manual annotation of a diverse set of textual data may not only require an equally diverse set of human experts, but also homogeneity in the ruling of annotation with respect to the underlying taxonomies thereof. Additionally, the process of manual annotation can be time consuming and cumbersome, requiring constant calibration of the annotators ruling. Naturally, one may desire a method to automatically annotate a diverse set of textual data. In this paper, we will focus without loss of generality on annotating German textual data with their respective WZ2008 key (a multi-class classification problem) and provide further details in Section III.

The applications of automated industrial sector recognition are many and varied:

- The categorization of companies. Here, the most important question is on which data a classification should operate. In this paper we will focus on textual data, but other data (e.g. economic data) are also available and could help to improve automated methods.
- The categorization of advertisements, e.g. job advertisements or training advertisements. However, the main question here is whether we want to classify the job position (e.g. a miner under “Mining and Quarrying”) or the occupation being recruited. Obviously the two approaches are not interchangeable.
- However, we can also apply this to other textual data: Which industries are mentioned in political speeches or in newspapers? Approaches to literature are even more challenging.

Very limited work has been carried out in this field as we will discuss in the next section. According to our knowledge, no work on German texts has been carried out. Data on companies is usually collected and sold by commercial providers like statista.

This paper is divided into eight sections. The first section provides an introduction and gives a brief overview of the background of the research question, the second section presents related work. Section three presents the data and an overview about existing resources. In section four, we will introduce the methods used to answer the research question. The fifth section is dedicated to experimental results and the evaluation of these methods. In section six, we discuss our findings and give a detailed interpretation of the results. After we briefly discuss possible bias in the penultimate section, our conclusions and outlook onto further research are drawn in the final section.

The contributions of this paper include the following:

(i) We fine-tune and compare an openly available domain-adapted BERT model with a standard BERT model. We find that the domain-adapted model shows an increased ability to generalize over the vanilla model on Out-Of-Distribution data.

(ii) We evaluate the models on two novel data-sets, one mapping Wikipedia paragraphs to WZ2008 keys, the other mapping job ads to WZ2008 keys. Both data-sets are open sourced for further research [1].

(iii) We discuss shortcomings of our approach and gain insight on how to improve the current methods. We conjecture that a more diverse mixture of training data will drastically improve a domain-adapted models ability to generalize.

II. Related Work

Very little work has been done in this area. There are several applications for the given research question: For example, Pejic et al. state the need to analyse Industry 4.0 skills, but do not present a generic categorization approach, but rather pre-select job advertisements according to their needs [2]. Chaisricharoen et al. noted the importance of industrial sectors for legal categories. However, their work is limited to industry-standard keywords [3]. For the generic categorization of English texts, some work has been done by McCallum [4] and Kibriya et al. [5]. However, the data and industrial sectors are mainly for marketing purposes and cannot be used in economic and sociological research. Several other works rely on these data-sets, see for example [6], [7], which underlines the general need for publicly available training and evaluation data.

Text mining on labor market data is a widely considered topic. For an automated analysis of labor-market related texts, the situation in German-speaking countries like Germany, Austria and Switzerland is not much different to English-speaking countries: “Catalogs play a valuable role in providing a standardized language for the activities that people perform in the labor market” [8]. However, while these catalogs are widely used for creating and computing statical values, for managing labor market and educational needs or for recommending trainings and jobs, there is no single ground truth. According to Rodrigues et al., one reason for this could be the fact that labor market concepts are modeled by multiple disciplines, each with a different perspective on the labor market [9]. For German texts, in particular job advertisements, Gnehm et al.[10] introduced transfer learning and domain adaptation approaches with jobBERT-de and jobGBERT. This model was also used for the detection of skill requirements in German job advertisements [11], [12].

For regional data, especially in German-speaking countries, industrial sectors are widely used as a basis for economic and labour market research, see for example [13], [14], [15], they are particularly important for future skills and qualifications [16]. Although classification is a key issue for industrial sectors, see [17], little research has been carried out using computational methods. Examples are mainly limited to regional industries [18] or agriculture and green economy [19].

To our knowledge, no work has been done on German texts. Company data are usually collected and sold by commercial providers such as statista. There is also an online guide from the Federal Office of Economic Affairs and Export Control (BAFA) (“Merkblatt Kurzanleitung Wirtschaftszweigklassifikation”[1]), but this is only a short version of the data available from the Federal Statistical Office. Therefore, we will now discuss the available data.

III. Data

As discussed above, several classifications of industrial sectors exist. In our case, we rely on the official German statistics using WZ08. We will describe this taxonomy in the first subsection. However, also a rich variety of possible applications exists. Thus, in the next subsection we will describe several textual data for training and evaluation.

A. Classification of industrial sectors

The so-called “Klassifikation der Wirtschaftszweige” (Classification of branches of industry, short: WZ) is used in Germany, in particular for official statistics by the “Statistische Bundesamt” (Federal Statistical Office), to classify economic activities of employers. The most recent version is WZ 2008, making WZ 2003 and 1993 deprecated. It is compatible to the European “Nomenclature statistique des activités économiques dans la Communauté européenne” (NACE) Rev. 2, but adds more detailed data. For more details we refer to [20]. All data is available in English and German at https://www.destatis.de/DE/Methoden/Klassifikationen/Gueter-Wirtschaftsklassifikationen/Downloads/klassifikation-wz-2008-englisch.html. In this text, for the description of examples we usually rely on the official English translation, while the work itself is carried out on German data.

Similar to NACE, WZ 2008 provides several hierarchical levels. A first level describes 21 sections (letters A-U), a second divisions, a third groups, a fourth classes. In contrast to NACE, WZ 2008 adds subgroups as fifth level, which is, however, only added to particular classes. See Figure 1 for an illustration of a particular hierarchy in sector C. Thus, with examples we find the following hierarchical elements:

- Sections (21), A-U, e.g., “B MINING AND QUARRYING”
- Divisions (88), 01-99, e.g., “05 Mining of coal and lignite”
- Groups (272), 01.1-99.0, e.g., “05.1 Mining of hard coal”
- Classes (615), 01.11-99.00, e.g., “05.10 Mining of hard coal”
- Sub-classes (839), 01.11.0-99.00.0, e.g., “05.10.0 Mining of hard coal”

While sectors are very broad and specific, for example A (Agriculture, Forestry and Fishing) and B (Mining and Quarrying), others are not clearly defined at this level, for example S (Other Service Activities). On the other hand, classes and groups often do not differ and the naming of divisions and groups usually does not provide much more information (e.g. 77 “Rental and leasing activities” towards 77.1 “Renting and leasing of motor vehicles”). In addition, a company might well belong to two or even more industrial sectors, e.g. to several manufacturing divisions. However, the official guidelines recommend to label the most dominant sector. Thus, while the taxonomy of industrial sectors is well-defined by WZ08, we rely on external data to train and evaluate our approaches. In addition, we need to discuss on

[1] See https://www.bafa.de/SharedDocs/Downloads/DE/Wirtschaft/unb_kurzanleitung_wirtschaftszweigklassifikation.pdf.

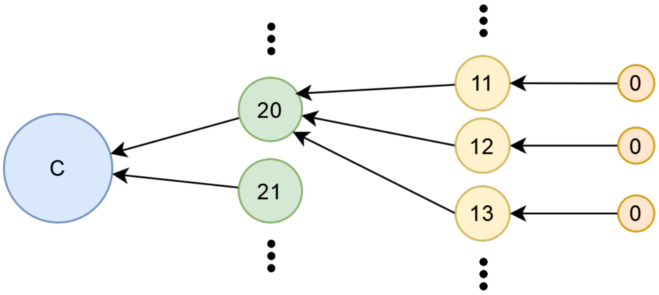
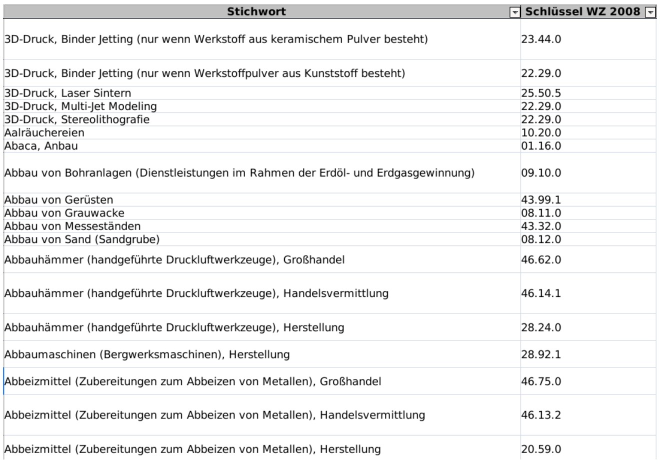


Fig. 1. An example subset of WZ08: Sector C (Manufacturing), division 20 (Manufacture of chemicals and chemical products), group 20.12 and class 20.12.0 (Manufacture of dyes and pigments). Note that for context, other groups (20.11 or 20.13) are displayed as well.



Stichwort	Schlüssel WZ 2008
3D-Druck, Binder Jetting (nur wenn Werkstoff aus keramischem Pulver besteht)	23.44.0
3D-Druck, Binder Jetting (nur wenn Werkstoffpulver aus Kunststoff besteht)	22.29.0
3D-Druck, Laser Sintern	25.50.5
3D-Druck, Multi-Jet Modeling	22.29.0
3D-Druck, Stereolithografie	22.29.0
Aalräuchereien	10.20.0
Abaca, Anbau	01.16.0
Abbau von Bohranlagen (Dienstleistungen im Rahmen der Erdöl- und Erdgasgewinnung)	09.10.0
Abbau von Gerüsten	43.99.1
Abbau von Grauwacke	08.11.0
Abbau von Messeständen	43.32.0
Abbau von Sand (Sandgrube)	08.12.0
Abbauhämmer (handgeführte Druckluftwerkzeuge), Großhandel	46.62.0
Abbauhämmer (handgeführte Druckluftwerkzeuge), Handelsvermittlung	46.14.1
Abbauhämmer (handgeführte Druckluftwerkzeuge), Herstellung	28.24.0
Abbaumaschinen (Bergwerksmaschinen), Herstellung	28.92.1
Abbeizmittel (Zubereitungen zum Abbeizen von Metallen), Großhandel	46.75.0
Abbeizmittel (Zubereitungen zum Abbeizen von Metallen), Handelsvermittlung	46.13.2
Abbeizmittel (Zubereitungen zum Abbeizen von Metallen), Herstellung	20.59.0

Fig. 2. An example subset of keywords or descriptive texts for WZ08 offered by the German Federal Statistical Office.

which level, e.g. sectors or divisions, the categorization can be carried out.

B. *Training and evaluation data*

1) Official Data: The German Federal Statistical Office (see above) provides a list with 33,945 keywords or descriptive texts of up to 30 words, hence subsequently called snippets, covering all classes in WZ08[2], see Figure 2 for an illustration. It clearly separates between different industries, for example for barrel-locks (“Zylinderschloss”) we find entries for retail (47.52.1), whole sale (46.74.1), trade agency (46.15.4) and production (25.72.0). However, this underlines the complexity of this data-set containing not only single keywords but also activities and often even more information, e.g. technical information (“Zylinderschleifereien für Kraftwagen von 3,5 t und weniger”). Thus, we will carefully evaluate how and in which cases we can use this data for training.

The data-set is very imbalanced, as approx. 44% of all snippets map onto a single WZ08-group, namely G 46 Wholesale (excluding trade in motor vehicles) (“Großhandel (ohne Handel mit Kraftfahrzeugen)”). Similarly, about 66% of snippets map onto one of five out of the overall 40 groups.

2) Wikipedia: Covering free text descriptions, we collected and manually annotated 1122 entries of German Wikipedia by industrial sector, division and group. This list contains companies (e.g. “Vereinigte Margarine-Werke Nürnberg”), brands (“Whiskas”), concepts (“Tabak”, “Flachglas”) and activities or tools (“Weben”, “Hammer”, “Werkzeug”). Thus, this data highlights how broad industrial sectors are and the questions remains what really characterises them. However, having a cross section of these entities at hand might result in better accuracy.

As a first example, consider 10.9 “Manufacture of prepared animal feeds”: here the entries refer to pet food (“food”), but also to brands such as Whiskas. Other industries are even more fuzzy, such as 61.1 “Wired telecommunications activities”: Here we collected mainly technical entries (e.g. network connection, mobile phone network), as there are no entries for industries with such a limited focus in the German Wikipedia.

Similarly to the WZ08 snippets, the wikipedia data-set is imbalanced, but towards a different section, namely “C” Manufacturing Industry, containing WZ08-groups like mechanical engineering and so on. About 41 % of the text descriptions belong to this section, please refer to Figure III-B3 for more details.

3) Job Advertisement: As a third data collection, we have 635 manually annotated job advertisements. Here, it is crucial to differentiate, whether the job itself or the company ought to be categorized, as the data may contain information about both domains. This is especially challenging, as the information for and requirements about the position in question may lead to a misclassification in case the inference model cannot distinguish between company-specific or job-specific information. For the annotation process, we decided to categorize companies or businesses, because in most cases job advertisements contain a section with information on them. We excluded advertisements without this profile. In addition, we excluded all advertisements from temporary-employment agencies, because they allow no conclusion about the real company searching for the particular job profile.

We collected data from the BA’s official job search, “Jobsuche”, which also classifies advertisements by industry, see figure 3. The adverts provide a free text field describing the job and the requirements, see Figure 4 (left). There is also information about the employee, although not all the information seems to be mandatory, see Figure 4 (centre, right). Some companies add extensive promotional texts and descriptions of their profile.

The data-set is also quite imbalanced, although towards completely different sections than the other two data-sets. The most represented section is “C” (Manufacturing Industry) with about 20%, followed by “M” (Provision of freelance, scientific and technical Services) with about 12%.

We make all data-sets openly available at [1].

[2] See https://www.destatis.de/DE/Methoden/Klassifikationen/Gueter-Wirtschaftsklassifikationen/Downloads/klassifikation-wz-2008-3100100089004-aktuell.pdf.

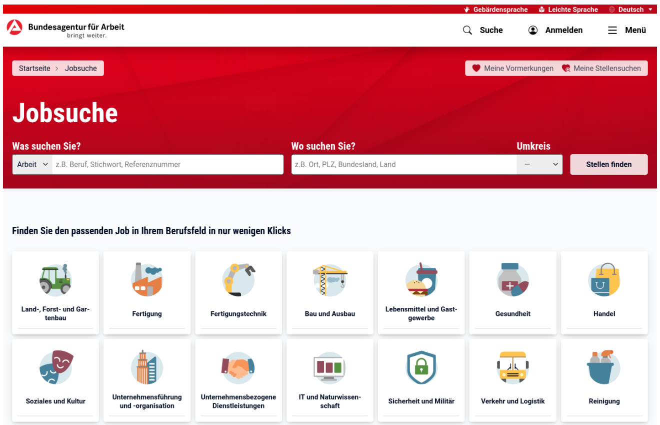


Fig. 3. The landing page of BA "Jobsuche" with job advertisements sorted by industrial sectors (bottom).

C. Manual curation

In this section, we present some information about the manual curation of the data and how we created a gold standard to evaluate the methods presented in this paper. The process was carried out by five domain experts.

During the annotation of various job advertisements from the data-set of the Federal Employment Agency, it was found that certain occupations could not be clearly assigned to one economic sector, but belonged to several industries. Thus, uniqueness was disproved. Furthermore, it can be concluded that occupations can be assigned individually to the economic sectors depending on the job description in the job advertisement and are dependent on this description. Thus, depending on the description, the industry or the frequency of economic categories changes.

The same challenges apply to Wikipedia data: Similar to companies, skills and tools may belong to different industries, leading to either ambiguity in the assignments or missing data. Therefore, we decided to annotate the data only up to the group level, choosing the most typical representations and avoiding generic lemmas.

In the near future, we will provide a more detailed qualitative evaluation of the annotation process, such as the interannotator agreement.

IV. Method

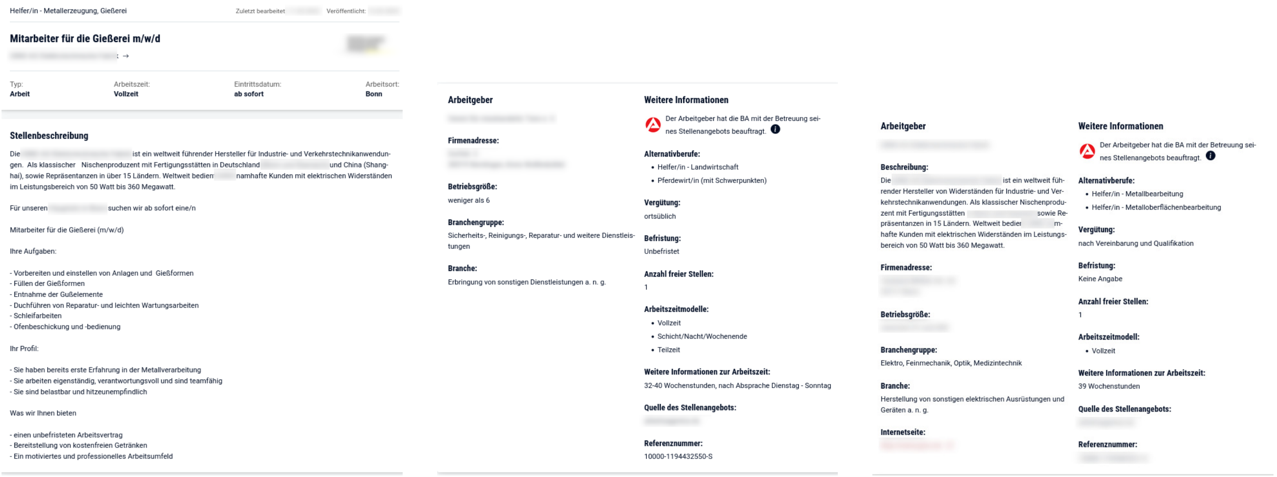
A. Preprocessing

The training subset contains data from the 10 most frequent sections. Only classes with more than 100 samples were selected. To reiterate on the issue of imbalanced data, about 44% of the data points of the training data-set map onto one class (G 46 Wholesale). As we didn't want to introduce an intrinsic bias of our classifier towards one class, we had to re-balance the data-set s.t. the samples constitute a uniform distribution over the WZ08-groups. Naturally, one may consider the threshold for samples per class to be the sample size for the lowest represented class (minority class). This approach might work, but limits the effective usage of a large portion of the training data for classes, which are over-represented (majority class). We chose to set a sample threshold of 4000, under-sample the majority classes, i.e. draw from the pool of samples for the respective class without replacement, and over-sample, i.e. sample with replacement from the respective samples, for the minority-classes to arrive at precisely 4000 samples for each of the classes. To optimize the diversity of the data of the minority classes, we made sure that all of the samples originally contained within the subset were present in the over-sampled data-set. This allowed for a balanced training data-set, trading in loss of generalization of the model on minority classes for a gain of generalization on the majority classes. In practice, we observed that for our training data-set the over- and under-sampling yielded very minor performance improvements on the evaluation data-sets.

B. Model Architecture

We used the `spacy` Python framework for Natural Language Processing (NLP) to fine-tune pre-trained transformer models to the task of text classification. Since it's great influx in popularity following the original publication by authors at Google [21], the transformer has arrived as a commonly used Neural Network Architecture. It has become the de-facto-standard for applications in NLP, given it's highly preferable ability to work on sequential data in parallel, making use of today's large amount of available compute resources as well as enabling the processing of even larger data-sets. The breakthrough included the introduction of a so called "Self-Attention-Layer", a Neural Network component (Figure 6) able to introduce the importance of the relationship between words within a sequence into an embedding used for subsequent processing. Each "Attention-Head" within the Self-Attention-Layer learns to attend to different semantic relationships during training, allowing for enough capacity to find crucial structural and semantic information in the data. We used the encoder part of the transformer to create sequence embeddings, which were then fed into a fully connected block, followed by classification head. This allowed us to fine-tune the pre-trained models on the variable length snippets. We fine-tuned two transformer models `bert-base-german-cased` and `agne/jobBERT-de`, evaluating them respectively on the evaluation data-sets described in Section (III-B). The base BERT (Bidirectional Encoder Representations from Transformers) model is a transformer based model trained by authors at Deepset, available at [22] and was pre-trained on German Wikipedia dumps, OpenLegalData dumps and news articles. The jobBERT-de model [23] is based on the previously mentioned base BERT model and adapted to the domain of job advertisements through continued in-domain pre-training on approx. 4 million German-speaking job advertisements from Switzerland from the years of 1990 to 2020.

We trained our models on a V100 GPU for approximately one hour each. We make the training configurations as well



Helfer/in - Metallerzeugung, Gießerei Zuletzt bearbeitet [redacted] Veröffentlicht: [redacted]

Mitarbeiter für die Gießerei m/w/d

[redacted] →

[redacted]

Typ:	Arbeitszeit:	Eintrittsdatum:	Arbeitsort:
Arbeit	**Vollzeit**	**ab sofort**	**Bonn**

Stellenbeschreibung

Die [redacted] ist ein weltweit führender Hersteller für Industrie- und Verkehrstechnikanwendungen. Als klassischer Nischenproduzent mit Fertigungsstätten in Deutschland [redacted] und China (Shanghai), sowie Repräsentanzen in über 15 Ländern. Weltweit bedien[redacted] namhafte Kunden mit elektrischen Widerständen im Leistungsbereich von 50 Watt bis 360 Megawatt.

Für unseren [redacted] suchen wir ab sofort eine/n

Mitarbeiter für die Gießerei (m/w/d)

Ihre Aufgaben:

- Vorbereiten und einstellen von Anlagen und Gießformen
- Füllen der Gießformen
- Entnahme der Gußelemente
- Duchführen von Reparatur- und leichten Wartungsarbeiten
- Schleifarbeiten
- Ofenbeschickung und -bedienung

Ihr Profil:

- Sie haben bereits erste Erfahrung in der Metallverarbeitung
- Sie arbeiten eigenständig, verantwortungsvoll und sind teamfähig
- Sie sind belastbar und hitzeunempfindlich

Was wir Ihnen bieten

- einen unbefristeten Arbeitsvertrag
- Bereitstellung von kostenfreien Getränken
- Ein motiviertes und professionelles Arbeitsumfeld

Arbeitgeber

[redacted]

Firmenadresse:

[redacted]

Betriebsgröße:

weniger als 6

Branchengruppe:

Sicherheits-, Reinigungs-, Reparatur- und weitere Dienstleistungen

Branche:

Erbringung von sonstigen Dienstleistungen a. n. g.

Weitere Informationen

Der Arbeitgeber hat die BA mit der Betreuung seines Stellenangebots beauftragt. ⓘ

Alternativberufe:

- Helfer/in - Landwirtschaft
- Pferdewirt/in (mit Schwerpunkten)

Vergütung:

ortsüblich

Befristung:

Unbefristet

Anzahl freier Stellen:

1

Arbeitszeitmodelle:

- Vollzeit
- Schicht/Nacht/Wochenende
- Teilzeit

Weitere Informationen zur Arbeitszeit:

32-40 Wochenstunden, nach Absprache Dienstag - Sonntag

Quelle des Stellenangebots:

[redacted]

Referenznummer:

10000-1194432550-S

Arbeitgeber

[redacted]

Beschreibung:

Die [redacted] ist ein weltweit führender Hersteller von Widerständen für Industrie- und Verkehrstechnikanwendungen. Als klassischer Nischenproduzent mit Fertigungsstätten [redacted] sowie Repräsentanzen in 15 Ländern. Weltweit bedien[redacted]mhafte Kunden mit elektrischen Widerständen im Leistungsbereich von 50 Watt bis 360 Megawatt.

Firmenadresse:

[redacted]

Betriebsgröße:

[redacted]

Branchengruppe:

Elektro, Feinmechanik, Optik, Medizintechnik

Branche:

Herstellung von sonstigen elektrischen Ausrüstungen und Geräten a. n. g.

Internetseite:

[redacted]

Weitere Informationen

Der Arbeitgeber hat die BA mit der Betreuung seines Stellenangebots beauftragt. ⓘ

Alternativberufe:

- Helfer/in - Metallbearbeitung
- Helfer/in - Metalloberflächenbearbeitung

Vergütung:

nach Vereinbarung und Qualifikation

Befristung:

Keine Angabe

Anzahl freier Stellen:

1

Arbeitszeitmodell:

- Vollzeit

Weitere Informationen zur Arbeitszeit:

39 Wochenstunden

Quelle des Stellenangebots:

[redacted]

Referenznummer:

[redacted]

Fig. 4. Several anonymized parts of job advertisements at BA "Jobsuche": A description (left), and two information on the employee.

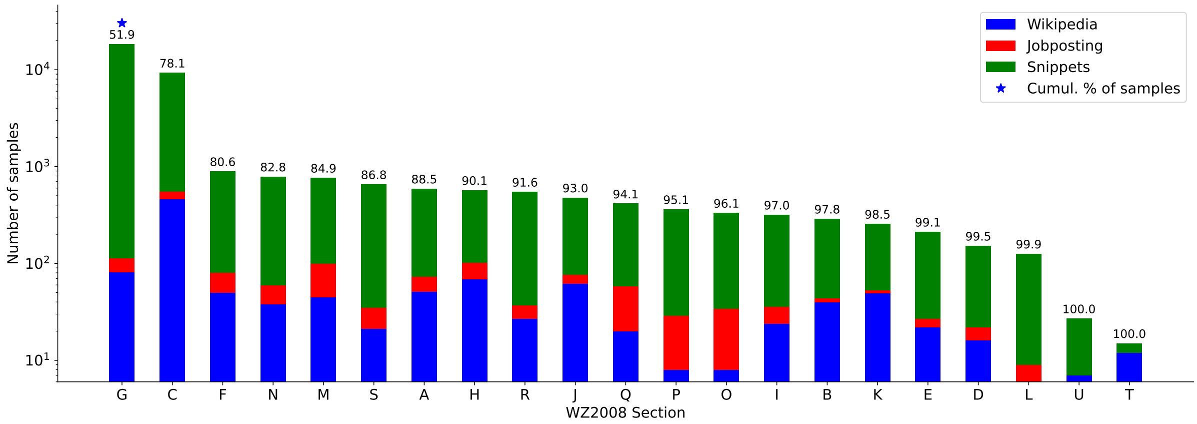


Fig. 5. The distribution of samples training and evaluation data across sections of WZ2008. Data are very unbalanced. Some sections are strongly underrepresented. Training data (green) cover only part of the evaluation data.

as all hyperparameters used in the preprocessing, training and evaluation of our experiments openly available at [1].

V. Evaluation

In the following we present the performance of the two fine-tuned models, for simplicity we will address them by the name of the pre-trained transformer model they are based on, `bert-base-german-cased` (BERT) and `agne/jobBERT-de` (jobBERT). We evaluated both models on three different data-sets. For each pre-trained model, we fine-tuned two classifiers, one to classify Sections and one to classify Divisions. As portrayed in Figure 1, a Sector may contain multiple Divisions. Naturally, the task to classify Divisions is harder, as there are more classes to be classified than in sector-classification.

A. Sections

TABLE I

(MACRO-) F_1-SCORE, PRECISION, RECALL AND TOP-5-ACCURACY FOR CLASSIFIERS TRAINED FROM BERT (a) AND JOBBERT (b) TO CLASSIFY SECTIONS (19 CLASSES).

Data-set	Wikipedia		Job-postings		Snippets	
Model	(a)	(b)	(a)	(b)	(a)	(b)
F_1	0.62	0.64	0.20	0.22	0.95	0.88
Precision	0.62	0.66	0.29	0.43	0.94	0.87
Recall	0.72	0.71	0.25	0.23	0.97	0.90
Top 5 Accuracy	0.92	**0.95**	0.58	**0.72**	0.99	**0.99**

We can see from Table I, that the BERT model has generally demonstrated a higher capability to capture information from the training set (Snippets) and archives better results on the Snippets evaluation data-set, which comes from the same data distribution as the training set. The domain-adapted jobBERT model although, demonstrates it's ability to generalize better across data distributions, as we can see that the metrics for the non-training data distributions (i) Wikipedia and (ii) Job-postings are increased in comparison to the vanilla BERT model.

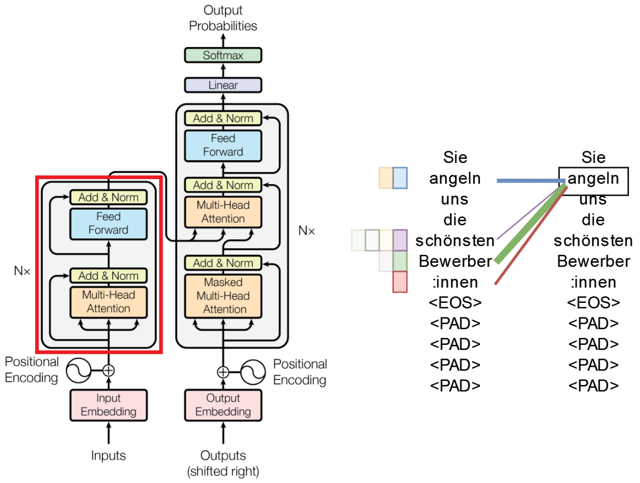


Fig. 6. *Left*: Encoder- (red box) and Decoder-components of the Transformer architecture presented in [21]. For our experiments, the encoder part of the architecture was used.
Right: The Self-Attention mechanism. Different attention-heads (here indicated by different colors) attend to different words in the sequence. Note, that for clarity only the attention for the word "angeln" is displayed. This example highlights the importance of context-sensitive methods. The word "angeln" (fishing) can have multiple meanings in different contexts.

B. Divisions

Our experiments for the classification task on divisions, which partition the sections into subsections, is in contrast to our findings for the section-classification, as we see in Table II that the difference in the predictive capability on the training set data distribution for both classifiers is very similar. The advantages of the domain-adapted jobBERT model on the Job-postings data-set is marginal and on the Wikipedia data-set, the BERT model even outperforms the jobBERT model in terms of the macro-Precision metric. Nonetheless, the jobBERT model still achieves higher top 5 accuracy on the non-training data distributed evaluation data-sets.

TABLE II
(MACRO-) F_1-SCORE, PRECISION, RECALL AND TOP-5-ACCURACY FOR CLASSIFIERS TRAINED FROM BERT (a) AND JOBBERT (b) TO CLASSIFY DIVISIONS (40 CLASSES).

Data-set	Wikipedia		Job-postings		Snippets	
Model	(a)	(b)	(a)	(b)	(a)	(b)
F_1	0.46	0.43	0.11	0.13	0.86	0.86
Precision	0.70	0.51	0.17	0.16	0.87	0.87
Recall	0.45	0.46	0.13	0.14	0.86	0.87
Top 5 Accuracy	0.80	**0.85**	0.41	**0.51**	0.98	**0.98**

VI. Discussion

With our experiments, which we presented in the previous section, we provided empirical evidence, that the domain-adapted transformer model jobBERT generalizes better on non-training data distributions whereas the vanilla BERT model outperforms jobBERT on the training data distribution for a text classification task. We think that the results presented in Table I are comparatively more representative, as the lowered granularity increased the overall training data size. The precision of these results may be improved further by collecting a more diverse data-set, containing more samples from the under-represented minority classes. In our experiments, we counteracted the imbalance of the data-set with oversampling but although this method was mainly introduced to avoid an intrinsic bias of the model and still be able to use most of the training data for the majority classes, it cannot improve predictive capability on the minority classes. If anything, it even lowers the predictive capability for minority classes as our model is prone to over-fitting. An additional concern is, that since the number of tokens the BERT models can process is limited to 512 tokens, some of the valuable information contained in the latter parts of a job-posting or Wikipedia article might be lost, as the underlying `spacy` textcat-model may truncate the input to fit its maximal token length. An approach would be to split the input sequence into sub-sequences that individually conform to the token length constraints of the model, then feed the sub-sequences to the model individually, obtaining multiple classifications from one partitioned sequence. This approach would then introduce the obvious problem, that the model cannot include potentially crucial context across sub-sequences. Another approach would be to make use of a summarization-model or repeated prompting of a Large Language Model (LLM), in order to condense or extract important information. Simple tests, which we've conducted using state of the art LLMs have shown that this approach yields underwhelming summarization or data-extraction performance for our data. However, using more powerful models or different prompting techniques like Chain-Of-Thought [24] or Tree-Of-Thought [25], one might be able to improve upon the simple pre-processing used in this work.

Most importantly though, we would like to bring attention to the dominant problem we are faced with in classifying sectors, namely the problem of mixed domains. As already outlined in the latter parts of subsection III-B, job-postings might contain information on the job itself (e.g. mechanic) *and* the company searching for said talent (e.g. an agricultural firm looking for a mechanic). It is challenging to make the context sensitive distinction between both pieces of information. Staying in the domain of job-postings, text-segmentation could be used to first partition the text into sections (e.g. Address of company, Job description, Requirements etc.), then feeding the segmented text into a subsequent network.

A point of discussion should also be the plausibility of the choice of granularity for the classification, meaning whether it is plausible to attempt to differentiate between Divisions, as for some cases the distinction of inter-section Divisions is challenging even for a human expert. It is for this reason we decided to include the Top-N Accuracy metric into our analysis. We show that even with a straight forward approach like fine-tuning domain-adapted transformer models, we are able to reach 85% Top-5 Accuracy on non-training data distributions (Wikipedia data-set, Table II), i.e. data from a distribution the model has never seen before and hence has had no chance of adapting to. To this end, our findings should be interpreted as a proof of concept, as we have demonstrated that with simple data-sets and straight forward methods we are able to generalize across different data distributions. Naturally, more work is to be done, as we currently lack big annotated data-sets from different data-distributions (one may consider Twitter data, YouTube descriptions, reddit posts etc.) in order to train a model on a balanced mixture data-set.

VII. Bias

We would like to touch on the topic of bias, which is in part introduced by the fine-tuned models themselves a priori. It is likely, that BERT has seen some collection of job-postings, for which the data distribution is unknown. Similarly, the domain-adapted jobBERT was pre-trained on a corpus of job-postings, for which we also don't know the data-distribution. Additionally, it is crucial to mention, that the job-postings on which jobBERT was pre-trained on are of Swiss origin which introduces additional bias. It remains a topic of discussion, whether a corpus of Swiss job-advertisements suffices as a pre-training data-set, if one is trying to fine-tune for classification onto German industrial sectors. As for the evaluation data-sets, the Wikipedia data-set and the Job-postings data-set may have introduced human bias of unknown form. As outlined in subsection (III-C) we will supplement the open sourced datasets provided at [1] with a detailed qualitative evaluation of the annotation process, such as the inter-annotator agreement.

VIII. Conclusion and Outlook

In this paper, we presented our novel approach to classifying general textual data onto German industry sectors using pre-trained transformer models. In the second sections, we introduced the WZ08, a taxonomy of the German industrial sectors, and subsequently discussed the imbalance of both the training- and evaluation-data. We sourced two novel data-sets (a) Wikipedia articles and (b) Job-postings mapping to WZ08 Divisions, to be included in our analysis and discussed the respective details in section III. In the evaluation, we showed that in spite of the difficult challenge of mixed domains and the imbalance of the data available, the domain-adapted transformer model jobBERT was able to generalize better across different data distributions than the regular BERT model in a text classification task. This hints that even with simple methods like fine-tuning a domain-adapted transformer model, one is able to generalize relatively well across unknown data-distributions given a good mixture of data-sets.

Our initial research question was whether one can automatically categorize textual data, such as job ads or company profiles, by industry. We presented and discussed several approaches and showed that this categorization is possible. However, its quality depends on both training and evaluation data. Thus, it also depends on the application and the research question.

All approaches failed for job advertisements. Here we need to redefine a precise research question and in particular provide more feasible information about what data is available (e.g. which metadata could help to improve the quality) and what the expected result should be.

However, our approach provides a reasonable recall of Wikipedia data. Thus, it could help to recommend and provide suggestions for manual curation and annotation on similar textual data. Further research and quality control could help to improve the model. While the presented approach works and provides meaningful results, it is far from being ready for productive use, but shows the significant impact of research in this area.

The initial research question was difficult not only because of the diversity of data and expected outcomes, but also because of the interdisciplinary nature of the research. The social sciences and the example use cases for labor market research have a different perspective on industrial sectors than, for example, economics. Thus, understanding the correct classification depends not only on the research questions, but also on the perspective of different scientific domains. It is very unlikely that a single generic solution could be developed to cover all these different needs. However, more interdisciplinary exchange could help to clarify and guide computer science research in this area.

Acknowledgments

We would like to thank our colleagues Katharina Bär, Nicolai Bör, Lisa Fournier and Jan-Philipp Schroer for helping us out with data annotation.

References

[1] R. Fechner, D. J. Dörpinghaus, and A. Firll, "FedCSIS 2023 Classifying Industrial Sectors with a Domain Adapted Transformer - Datasets and Configuration files," Jul. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8192546

[2] M. Pejic-Bach, T. Bertoncel, M. Meško, and Ž. Krstić, "Text mining of industry 4.0 job advertisements," *International journal of information management*, vol. 50, pp. 416–431, 2020.

[3] R. Chaisricharoen, W. Srimaharaj, S. Chaising, and K. Pamanee, "Classification approach for industry standards categorization," in *2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*. IEEE, 2022, pp. 308–313.

[4] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Madison, WI, 1998, pp. 41–48.

[5] A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes, "Multinomial naive bayes for text categorization revisited," in *AI 2004: Advances in Artificial Intelligence: 17th Australian Joint Conference on Artificial Intelligence, Cairns, Australia, December 4-6, 2004. Proceedings 17*. Springer, 2005, pp. 488–499.

[6] H. Hayashi and Q. Zhao, "Quick induction of nntrees for text categorization based on discriminative multiple centroid approach," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2010, pp. 705–712.

[7] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[8] C. Ospino, "Occupations: Labor market classifications, taxonomies, and ontologies in the 21st century," *Inter-American Development Bank*, 2018.

[9] M. Rodrigues, Fernández-Macías, and Enrique, Sostero, Matteo, "A unified conceptual framework of tasks, skills and competences," Seville, 2021. [Online]. Available: https://joint-research-centre.ec.europa.eu/publications/unified-conceptual-framework-tasks-skills-and-competences_en

[10] A.-S. Gnehm, E. Bühlmann, and S. Clematide, "Evaluation of transfer learning and domain adaptation for analyzing german-speaking job advertisements," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022, pp. 3892–3901.

[11] A.-S. Gnehm, E. Bühlmann, H. Buchs, and S. Clematide, "Fine-grained extraction and classification of skill requirements in german-speaking job ads." Association for Computational Linguistics, 2022.

[12] J. Büchel, J. Engler, and A. Mertens, "The demand for data skills in german companies: Evidence from online job advertisements," *How to Reconstruct Ukraine? Challenges, Plans and the Role of the EU*, p. 56, 2023.

[13] B. Gehrke, H. Legler, M. Leidmann, and K. Hippe, "Forschungs- und wissensintensive wirtschaftszweige: Produktion, wertschöpfung und beschäftigung in Deutschland sowie qualifikationserfordernisse im europäischen vergleich," Studien zum deutschen Innovationssystem, Tech. Rep., 2009.

[14] N. Gillmann and V. Hassler, "Coronabetroffenheit der wirtschaftszweige in gesamt-und ostdeutschland," *ifo Dresden berichtet*, vol. 27, no. 04, pp. 03–05, 2020.

[15] U. Kies, D. Klein, and A. Schulte, "Cluster wald und holz deutschland: Makroökonomische bedeutung, regionale zentren und strukturwandel der beschäftigung in holzbasierten wirtschaftszweigen," *Cluster in Mitteldeutschland–Strukturen, Potenziale, Förderung*, p. 103, 2012.

[16] V.-P. Niitamo, "Berufs-und qualifikationsanforderungen im ikt-bereich in europa erkennen und messen," *Schmidt, SL; Strietska-Ilina, O.; Dworschak, B*, pp. 194–201, 2005.

[17] J. Hartmann and G. Schütz, "Die klassifizierung der berufe und der wirtschaftszweige im sozio-oekonomischen panel-neuvercodung der daten 1984-2001," SOEP Survey Papers, Tech. Rep., 2017.

[18] M. Titze, M. Brachert, and A. Kubis, "The identification of regional industrial clusters using qualitative input–output analysis (qioa)," *Regional Studies*, vol. 45, no. 1, pp. 89–102, 2011.

[19] U. Kies, T. Mrosek, and A. Schulte, "Spatial analysis of regional industrial clusters in the german forest sector," *International Forestry Review*, vol. 11, no. 1, pp. 38–51, 2009.

[20] Statistisches Bundesamt, "Klassifikation der Wirtschaftszweige," Wiesbaden, 2008. [Online]. Available: https://www.destatis.de/static/DE/dokumente/klassifikation-wz-2008-3100100089004.pdf

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[22] B. Chan, T. Möller, M. Pietsch, and T. Soni. (2019) bert-base-german-cased transformer model. [Online]. Available: https://huggingface.co/bert-base-german-cased

[23] A.-S. Gnehm, E. Bühlmann, and S. Clematide, "Evaluation of transfer learning and domain adaptation for analyzing german-speaking job advertisements," in *Proceedings of the 13th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2022.

[24] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[25] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *arXiv preprint arXiv:2305.10601*, 2023.

DOI: 10.15439/2023F8906
ISSN 2300-5963 ACSIS, Vol. 35

Towards Enhancing Open Innovation Efficiency: A Method for Ontological Integration of BPMN and EMMO

Christophe Feltus, Damien Nicolas, Carlos Kavka, Djamel Khadraoui, Salim Belouttar
Luxembourg Institute of Science and Technology (LIST)
Avenue des Hauts–Fourneaux, 5,
L–4362 Esch–sur–Alzette, Luxembourg
Email: name.firstname@list.lu

Natalia Konchakova
Helmholtz-Zentrum Hereon, Institute of Surface Science
Max-Planck-Straße 1
21502 Geesthacht, Germany
Email: natalia.konchakova@hereon.de

Heinz A. Preisig
Norwegian University of Science and Technology (NTNU)
Department of Chemical Engineering,
N-7491 Trondheim, Norway
Email: heinz.a.preisig@ntnu.no

Peter Klein
Fraunhofer Institut für Techno- und Wirtschaftsmathematik
Fraunhoferplatz 1
67663 Kaiserslautern, Germany
Email: peter.klein@itwm.fraunhofer.de

***Abstract*—The process of open innovation based on advanced materials involves the collaborative sharing of knowledge, ideas, and resources among different organisations, such as academic institutions, businesses, and government agencies. It is suggested that Business Process Modelling and Notations (BPMN) and Elementary Multiperspective Material Ontology (EMMO) be closely integrated to accelerate the development of new materials and technologies and address complex material challenges. In this paper, we examine the integration of EMMO and BPMN through an initial investigation to streamline workflows, enhance communication, and improve the understanding of materials knowledge. We propose a four-step approach to integrate both ontologies, which involves ontology alignment, mapping, integration, and validation. Our approach supports faster and more cost-effective research and development processes, leading to more effective and innovative solutions.**

I. Introduction

DIGITALISATION efforts in the engineering and materials development domains are today introducing new methods for digital collaboration and open innovation, like the one proposed in VIPCOAT[1]: Development projects implementing digitalisation approaches offer a multi-sided platform to create a collaborative environment to connect modellers (software owners, academia), and translators [1], manufacturers, governmental bodies and society to initiate and implement innovation projects (see Fig.1). To assist industrial end-users in making optimal decisions about materials and process design and manufacturing based on predictive modelling, it is increasingly necessary to examine innovation through a quadruple helix approach, which addresses the need for a *Digital Single Market strategy for Open Innovation 2.0* [2].

In parallel, an enormous amount of materials, manufacturing and processing data are currently generated by high throughput experiments and computations, possessing a significant challenge in terms of data integration, sharing and interoperability. A common ontology lays the foundation for solving these issues, enabling semantic interoperability of models, experiments, software and data, which is vital for using rational development design principles and testing and manufacturing of materials in general.

The aim of this work is to contribute to the current efforts by the European Materials Modelling Council EMMC on establishing common standards for materials modelling through the Elementary Multiperspective Material Ontology (EMMO), e.g.: [3]. The basic idea is to merge Business Decision Support Systems, implemented in terms of the BPMN and DMN standards, with materials modelling workflows by using ontologies as a glue between these hitherto distinct worlds.

Given that a product or a material system is defined by a combination of its physical, chemical and other technical properties, as well as other business-related aspects, such as cost, environmental footprint, and other relevant information to the organisation and the society at large. Therefore, it is essential for companies to gather data on the properties of the materials used in their products and vice versa. For instance, the physical and chemical properties of a protective coating can have a significant impact on production time, resource utilisation, manufacturing cost, sustainability, and toxicity.

[1] https://ms.hereon.de/vipcoat/

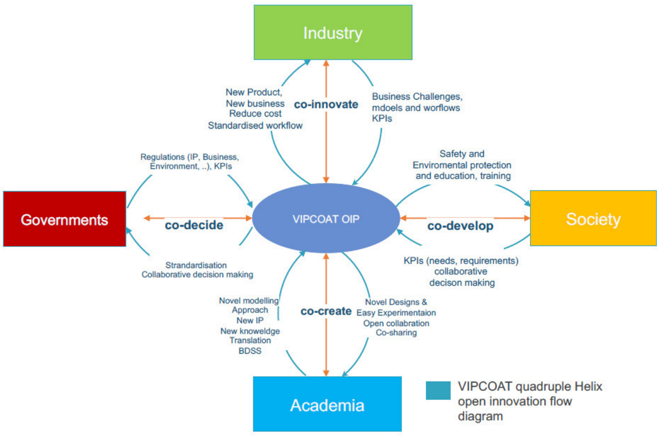


Fig. 1: Four Helix Virtual Open Innovation Framework: Industry, Society, Academia, and Governments

Hence, comprehending the properties of materials is critical to streamlining the manufacturing process, identifying appropriate machinery and equipment, and estimating relevant business indicators for informed decision-making [4]. This integration is particularly important in the context of Open Innovation, where companies collaborate to develop new products and services [5].

BPMN is a crucial tool for Open Innovation processes [6]. BPMN enables organisations to visually depict their business processes and workflows in a standardised format, which fosters more effective communication and collaboration with external stakeholders such as customers, suppliers, and partners. The standardised representation of business processes using BPMN allows for the identification of inefficiencies, redundancies, and bottlenecks in the workflow, leading to streamlined operations and increased efficiency [7]. Moreover, the use of BPMN provides a common language for discussing business processes, making it easier to share ideas and identify opportunities for improvement [5]. As a result, the ontology facilitates collaboration, accelerates innovation, and promotes the sharing of knowledge and best practices between organisations.

Elementary Multiperspective Material Ontology (EMMO), [8], is a comprehensive and versatile ontology for materials science that aims to provide a common language for describing materials and their properties. EMMO was developed by a group of European researchers as a part of the European Materials Modelling Council (EMMC)[2], which recognised the need for a unified approach to materials modelling and interoperability. EMMO is designed to be applicable to all levels of granularity, from atoms and molecules to macroscale materials, and it covers all aspects of materials science, including properties, structures, processes, and applications. EMMO is based on a multiperspective approach, which means it considers different perspectives and scales when describing materials. It provides a hierarchical structure that allows for the description of complex systems and a comprehensive set of classes and relationships for describing materials properties, including chemical composition, crystal structure, thermodynamic and mechanical properties, and more. EMMO is also designed to be extensible. Thus, it can be customised to meet the specific needs of different domains and applications. One of the key strengths of EMMO is its potential to promote interoperability between different materials modelling approaches and software tools. By providing a common language for describing materials and their properties, EMMO can facilitate the integration of models and data from different sources and the development of open standards and interfaces for materials modelling. This, in turn, can accelerate the development of new materials and improve the efficiency of materials design and testing.

In order to bridge the gap between the material science and business domains, this manuscript proposes the use of ontologies to establish a common understanding of the terminology and concepts used in both fields. The integration of the Business Process Model and Notation (BPMN) and the European Materials and Modelling Ontology (EMMO) can facilitate communication and collaboration among stakeholders, ultimately leading to the development of new materials and products. The integration of ontologies can lead to faster and more cost-effective research and development and the creation of innovative solutions to address complex material challenges. The paper aims to answer the research question of *how BPMN can be connected with EMMO* or vice versa, and proposes a concrete approach for integrating ontologies, consisting of conceptual alignments, concept mapping, concept integration, and validation. The proposed approach is applied to a preliminary analysis of integrating BPMN into EMMO. Section II provides an overview of the ontologies, extension mechanisms, and related works, while Section III describes the process of developing the integrated ontology. To know: III-A proposes processes alignment, III-B explains the concept and relationships mapping, III-C yields the integration of the concepts, and finally, III-D formally validates this integration using Incoherence Solving techniques. The paper concludes in Section IV with suggestions for future research areas.

II. Background

This section introduces BPMN and EMMO ontologies, reminds the different ontology extension mechanisms at our disposal, and presents the main related works.

A. BPMN

BPMN stands for Business Process Model and Notation [9]. It is a graphical representation for specifying business processes in a standardised way. BPMN was created by the Business Process Management Initiative (BPMI) and is now maintained by the Object Management Group (OMG)[3].

The primary purpose of BPMN is to provide a standardised notation that is readily understandable by all business stakeholders, including technical and non-technical users. This

[2] https://emmc.eu/

[3] https://www.omg.org/

notation enables clear communication and collaboration between business and technical teams when modelling and analysing processes and supports the execution of processes in a technology-agnostic manner.

BPMN provides a set of graphical elements, such as process, task, gateways, and events, that can be used to model various types of business processes. The notation also supports the modelling of more complex process flows, such as parallel and sequential execution, exception handling, and compensation.

BPMN is a widely adopted standard that helps organisations model, analyse, and improve their business processes, leading to increased efficiency and effectiveness.

B. EMMO

The Elementary Multiperspective Material Ontology (EMMO) is an ontology that provides a standardised and structured representation of the domain of materials science and engineering [10]. An ontology is a type of knowledge representation that defines a common vocabulary and formal model for describing concepts and relationships in a specific domain.

EMMO provides a comprehensive, hierarchical, and interlinked view of the concepts, classes, and relationships that are commonly used in materials science and engineering. It covers a wide range of topics, including material properties, processing techniques, and the relationships between materials and their components. EMMO aims to provide a shared understanding of the concepts and terms used in the field, making it easier for researchers, engineers, and data scientists to collaborate and exchange information.

EMMO is designed to be used as a resource for a variety of applications, including knowledge management, semantic search, and data integration in materials science and engineering. It can also help to integrate diverse data sources and support interdisciplinary research by providing a common vocabulary and conceptual framework. In this paper, we use the EMMO version 1.0.0.bata4 from github. [4]

C. Ontology extension mechanism

According to [11], the integration of two models (metamodels [12] or ontologies [13]) requires resolving three types of heterogeneity: *syntactic*, *semantic* and *structural*. For our integration, only the semantic and structural heterogeneity have been addressed. Indeed, the syntactic heterogeneity aims at analysing the difference between the serialisations of metamodel and, as explained by [14], addresses technical heterogeneity like hardware platforms and operating systems, or access methods, or it addresses the interface heterogeneity like the one which exists if different components are accessible through different access languages [15], [16]. Hence, it is not relevant in the case of this ontological integration.

Structural heterogeneity exists when the same metamodel concepts are modelled differently by each metamodel primitive. This structural heterogeneity has been addressed together with the analysis of the conceptual mapping and the definition of the integration rules. Finally, the semantic heterogeneity represents differences in the meaning of the considered metamodel's elements and must be addressed through elements mapping and integration rules. Regarding the mappings, three situations are possible: no mapping, a mapping of type 1:1, and a mapping of a type n:m (n concepts from one metamodel are mapped with m concepts from the other).

After analysing the heterogeneities, ontology extension mechanisms are applied. Ontology extension mechanisms refer to the ways in which an existing ontology can be expanded or modified to better suit the needs of a particular application or domain. There are several methods that can be used for ontology extension, including:

- Inheritance (generalisation): Inheritance is a refinement, detailing. Generalisation lifts things up. It is an additional level of abstraction. This is a common method of ontology extension in which a new class is defined that inherits properties and characteristics from an existing class. This allows new classes to be defined while reusing existing definitions and knowledge (e.g., in [17], inheritance relationships to extend OWL-S)
- Restriction (specialisation): This is a method of ontology extension in which the definition of an existing class is restricted to exclude certain individuals or objects. This can be used to refine a class's definition to better match a particular application's requirements.
- Extension (by adding axioms): This is a method of ontology extension in which new axioms or statements and rules are added to the ontology to provide additional information or, *a priory*, knowledge.
- Modules and Libraries: This is a method of ontology extension in which ontologies can be packaged as modules or libraries and can be imported or reused in other ontologies.

Each of these methods has its own strengths and limitations, and the appropriate method for a particular extension depends on the application's requirements and the design of the ontology being extended on a case-by-case basis.

D. Related Works

In [18], the proposed approach aims to integrate material modelling with business data and models to develop a Business Decision Support System (BDSS) [6] that assists in the complex decision-making process of selecting and designing polymer-matrix composites. This system combines materials modelling, business tools, and databases into a single workflow, providing a comprehensive solution supporting decision-making. In [7], the authors suggest utilising the BPMN and DMN[5] standards [19] to bridge the gap between business processes, materials science, and engineering workflows in the context of composite material modelling, which can potentially open up new horizons for industrial engineering applications. By using these standards ([20], [19]), it is possible to establish

[4] https://github.com/emmo-repo/EMMO

[5] https://www.omg.org/dmn/

a connection between the diverse domains and provide a more integrated approach to the modelling process, which could lead to improved efficiency and effectiveness in engineering applications.

In line with the previous approach, [21] extends the analysis by incorporating technical key performance indicators (KPIs) and financial KPIs, such as part costs, calculated using cost modelling applications. By including financial KPIs in the analysis, a more comprehensive understanding of the overall performance can be achieved, which can assist in the decision-making process related to product design and development. In [22], the authors discuss the development of an ontology called OSMO, which is an extension of the MODA workflow metadata standard [23], [24] used in European materials modelling projects. OSMO was created as part of the VIMMP project[6] and is connected to the larger effort of ontology engineering by the European Materials Modelling Council, with EMMO as its core. The article explains the purpose, design choices, implementation, and applications of OSMO [22], including its connections to other domain ontologies in computational engineering.

III. Integrating EMMO with BPMN

Merging two ontologies involves the integration of two separate ontologies into a single ontology that reflects the combined knowledge represented by both ontologies [25], [26]. To incorporate BPMN into EMMO, we propose a method, illustrated in Figure 2, that includes the following four steps:

- **Alignment**: This involves identifying and matching the concepts, classes, and relationships in the two ontologies that correspond to each other. This step requires a careful examination of the structure, content, and meaning of the concepts and relationships in both ontologies.
- **Mapping**: This involves creating a mapping between the concepts and relationships in the two ontologies based on the results of the alignment step. This mapping defines how the concepts and relationships in the two ontologies correspond to each other.
- **Integration**: This involves combining the two ontologies into a single ontology, using the mapping as a guide. The resulting merged ontology should reflect the combined knowledge represented by both original ontologies.
- **Validation by Incoherence Solving**: This involves checking the merged ontology to ensure that it is logically consistent and coherent and that it correctly represents the combined knowledge from both original ontologies.

A. Alignment

Conceptual alignment is the process of identifying and establishing the syntactic and structural correspondences between concepts or entities from two or more different sources or domains, should it be at the definition or at the association with other concepts level. To achieve this alignment, we listed all BPMN concepts, including their definition and association, and then we looked for correspondence with the EMMO concepts.

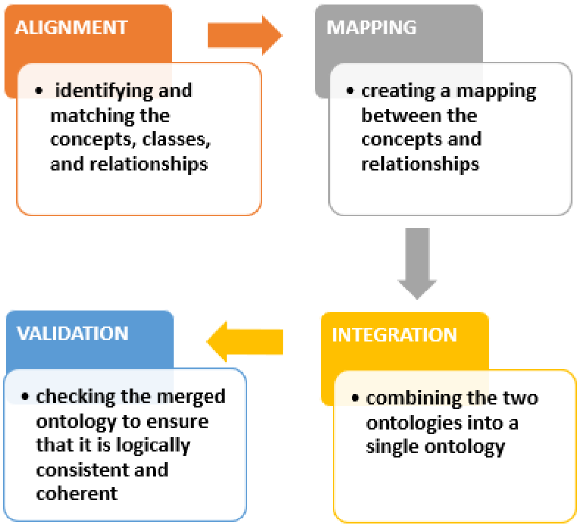

ALIGNMENT
• identifying and matching the concepts, classes, and relationships
MAPPING
• creating a mapping between the concepts and relationships
VALIDATION
• checking the merged ontology to ensure that it is logically consistent and coherent
INTEGRATION
• combining the two ontologies into a single ontology

Fig. 2: Four steps of the method used to integrate BPMN into EMMO: Alignment, Mapping, Integration and Validation

After a deep review of all BPMN concepts, we observed that eight concepts from BPMN may be aligned with nine concepts from EMMO. This alignment is possible based on analysing the concepts' names and definitions (syntactic alignment) and their associations with the other concepts (structural alignment).

1) Process vs. IntentionalProcess: The definition of **Process** from BPMN is *a Process describes a sequence or flow of Activities in an organisation with the objective of carrying out work*, although in EMMO, the **Process** is defined by *A whole that is identified according to criteria based on its temporal evolution that is satisfied throughout its time extension* and the **IntentionalProcess** extends the definition with *occurring with the active participation of an agent that drives the process according to a specific objective (intention)*. Both the **Process** and the **IntentionalProcess** are respectively part of and subClass of Process, and are associated with the **Participant**.

2) Participant (BPMN) vs. Participant (EMMO): In BPMN, a **Participant** *represents a specific PartnerEntity (e.g., a company) and/or a more general PartnerRole (e.g., a buyer, seller, or manufacturer) that are Participants in a Collaboration. A Participant is often responsible for the execution of the Process enclosed in a Pool* although in EMMO, this is *an object which is a holistic spatial part of a process. If plays an active role in the process, this is an Agent*. Both are linked to the concept of BPMN and EMMO's **Process**.

3) Activity vs. Elaboration: BPMN defines the **Activity** as a *work that is performed within a Business Process. An Activity can be atomic or non-atomic (compound)*. From the side of EMMO, an Elaboration is *the process in which an agent works with some entities according to some operative*

[6] Virtual Materials Market Place – https://cordis.europa.eu/project/id/760907

rules. Elaboration is a subClass of IntentionalProcess, and Activity is a component of Process (although not represented in BPMN metamodel from [20]). Both also have subClasses ElementaryWork, Computation, Workflow for Activity and, similarly, CallActivity, Task, SubProcess for Elaboration.

4) Task vs. ElementaryWork: The definition of **Task** in BPMN is *an atomic Activity within a Process flow. A Task is used when the work in the Process cannot be broken down to a finer level of detail. Generally, an end-user and/or applications are used to perform the Task when it is executed*. In EMMO, a **ElementyraWork** is *an elaboration that has no elaboration proper parts, according to a specific type*, which means that an ElementaryWork does not break down into smaller pieces of work. **Task** and **ElementaryWork** are respectively subClasses of **Activity** and **Elaboration**.

5) ThrowEvent vs. Status: Throwing events, following BPMN, *are triggers for catching events and are triggered by the process*, which result in **ThrowEvent** and **Status**, following EMMO, consists in *an object which is a holistic temporal part of a process*. Both concepts have no similar association with other modelling concepts.

6) InteractionNode vs. SubProcess and Stage: The alignment between both concepts from both metamodels is more arduous to establish but is real. In BPMN, the **InteractionNode** is *a type of flow object that represents a point in a process where participants interact with each other to exchange information or perform some action*, and in EMMO, the **SubProcess** is *a process which is a holistic spatial part of a process*, and the **Stage** is *a process which is a holistic temporal part of a process*. The semantic analysis of these three definitions does not make it possible to establish an indisputable alignment between the concepts. However, the analysis of associations clearly shows the similarities. Indeed, the InteractionNode is a subClass of Activity and FlowElementaryContainer, and is composed of Artifact and similarly, (1) the SubProcess has SubProcess and is SubClass of Process and (2), the stage has Stage and is SubClass of Process.

7) SequenceFlow and WorkFlow: According to BPMN, the **SequenceFlow** *is used to show the order of Flow Elements in a Process or a Choreography. Each Sequence Flow has only one source and only one target*. For EMMO, the **Workflow** is *an elaboration that has at least two elaborations as proper parts.* At the association level, the SequenceFlow is a subClass of FlowElement (abstract superclass for all elements that can appear in a Process flow), and the WorkFlow is a SubClass of Elaboration.

8) ItemAwareElement and EncodeData: The **ItemAwareElement** in BPMN refers to *several elements that are subject to store or convey items during process execution* and the **EncodedData** are in EMMO *causal object whose properties variation are encoded by an agent and that can be decoded by another agent according to a specific rule.* The ItemAwareElement concept has type DataObject, DataSTore, DataInput and DataOutput, which are type of information, and the EncodedData is a subClass of Data and has subClass Information.

B. Mapping

In order to integrate BPMN concepts and relationships within EMMO, it is necessary to analyse and select the best ontology extension mechanism (detailed in Section II-C) for each conceptual mapping achieved in Section III-A: Inheritance, Restriction, Extension, or Modules and Libraries – knowing that the last method is inappropriate to the purpose of our work.

1) IntentionalProcess: The analyse of the definitions provided in Section III-A1 demonstrates that both metamodels define the IntentionalProcess/Process based on the same arguments, to know: that a process is structured following a sequence of activities and that it aims to reach an objective. BPMN's semantics is richer than EMMO's semantics in that it associates the process to an organisation. Therefore, the preferred extension mechanism is the restriction (EMMO restricts BPMN conceptual semantics).

2) Participant: EMMO's definition of Participant is more generic than the definition of BPMN, which considers that the participant is a human, or an organisation, that is often responsible for the execution of a process. This is more specific than EMMO's point of view, which considers that an object demonstrating a holistic spacial part of the process is a participant. Accordingly, the extension mechanism that fits this alignment is inheritance. First, the BPMN's participant inherits the characteristics of EMMO's participant, and second, the EMMO's participant is extended with two possible statements: the participant is either a human or an organisation.

3) Elaboration: EMMO's definition of Elaboration is semantically a bit different than BPMN's definition of Activity. On one side, BPMN explains that the Activity may be atomic or compound, and on the other side, EMMO stresses the importance of the Elaboration to work following some operative rules. As a result, the most appropriate extension mechanism is inheritance, and the EMMO Elaboration is extended with a composition link from/to the EMMO Elaboration concept.

4) ElementaryWork: Task and ElementaryWork have the same semantics, and both refer to the smallest and indivisible piece of work composing a process. The definition of the Task from BPMN (Section III-A4) is semantically richer in that it stresses the importance of being within a traffic flow and being performed by an end-user or an application. In this case, the ontology extension mechanism used is the extension (BPMN extends EMMO conceptual semantic).

5) Status: The definition of Status in EMMO highlights that this concept stands for an object that reflects a temporal part of a process, whereas BPMN defines ThrowEvent as a trigger for catching events by the process. Although not explicitly embedded in the definition, the Status associated with a process often triggers other events in practice. Therefore, we consider that this Status may be a type of trigger and, by extension, a ThrowEvent. Therefore, the mapping between both concepts is achieved using the restriction mechanism given that EMMO restricts ThrowEvent to Status.

6) SubProcess and Stage: Both concepts represent part of the process (spacial or temporal), such as the InteractionNode from BPMN, which is described as a point in a process. The semantic heterogeneity between both BPMN and EMMO meanings is that the first specialises the finality of the concept to a place (or moment) where participants get together to achieve something or to exchange information. The description of the InteractionNode is consequently semantically more expressive, although both SubProcess and Stage refer to a spacial or a temporal dimension. As a result, the extension mechanism is the restriction since both EMMO's concepts restrict BPMN one. This situation is quite similar to the case of the IntentionalProcess, but because two concepts of EMMO are mapped to one concept of BPMN, it is not necessary to extend the concepts with a dedicated extension mechanism.

7) WorkFlow: Analysing the definitions of the WorkFlow and of the SequenceFlow, we conclude that the equivalence between both concepts is thin and limited. Both concepts are direct or indirect elements of the process that are associated with at least two flowing elements. The SequenceFlow adds a supplementary characteristic which is the existing sequence between the happening of the flowing elements. The extension mechanism preferred is, by the way, the restriction as WorkFlow restricts the SequenceFlow meaning.

8) EncodedData: The ItemAwareElement concept in BPMN represents an abstract concept that may be specialised in many types like DataObject, DataStore, DataInput and DataOutput although the EncodedData concept is well defined and refers to properties variation of an object. This definition restricts by the way the definition of the ItemAwareElement and, as a consequence, the restriction extension mechanism is the one naturally designated.

C. Integration

In the approach used in this work, all concepts from BPMN without EMMO equivalence have been introduced in the integrated EMMO ontology. The main concepts are: Gateway, Events, Artifact, InteractionNode, FlowElementContainer, FlowElement, MessageFlow, DataAssociation, DataOutputAssociation, DataInputAssociation, DataObject, DataOutput, DataInput, CallableElement. Further explanations of those concepts are available in BPMN 2.0 specifications [9].

The integration of BPMN concepts with EMMO equivalence is achieved based on the mapping performed in Section III-B and taking in hand the resolution of potential associations–related issues. This analyses, for each concept is the following:

1) IntentionalProcess: The BPMN process being semantically richer than the IntentionalProcess, we may consider that the IntentionalProcess is a subClass of the BPMN Process concept, which is represented as a **type of** relation in UML. In the integrated ontology, the IntentionalProcess is preserved. Concerning the relationships, two associations which did not exist for the EMMO concept have been added in the integrated version. It consists of (1) the IntentionalProcess **is linked to** Collaboration and (2) the IntentionalProcess **is composed of** Artefact.

2) Participant: EMMO's definition of Participant being more generic, we have maintained the EMMO's Participant concept in the integrated ontology, and we have extended it with an attribute inherited from BPMN, to know: *the Participant is an individual or an organisation that is often responsible for the execution of the Process*. Regarding the relationships, two associations which did not exist for the EMMO concept have been added in the integrated version: (1) the Participant **composes** the Collaboration (2) the Participant is a **type of** InteractionNode.

3) Elaboration: Given the small heterogeneity's existing between Elaboration and Activity and the decision to consider the inheritance extension mechanism, we have maintained the EMMO's Participant concept in the integrated ontology, and we have extended it with a composition link, as explained in Section III-B3, such as an Elaboration **composes** an Elaboration. In parallel, three additional Activity related associations from BPMN have also been included in EMMO Elaboration: (1) an Elaboration **is composed of** DataInputAssociation, (2) an Elaboration **is composed of** DataOutputAssociation, and (3) an Elaboration is a **type of** FlowNode.

4) ElementaryWork: Alike the IntentionalProcess, the ElementaryWork is less rich than the Task semantic from BPMN, and for the same reason, the extension mechanism elected during the mapping step was the extension mechanism. Accordingly, we keep the ElementaryWork in EMMO extended ontology. Concerning the associated relationships, we complete the existing ones with (1) the ElementaryWork is a **type of** InteractioNode, and (2) the ElementaryWork **has type** various kinds of tasks (i.e., ScriptTask, ServiceTask, BusinessRuleTask, ManualTask, SendTask, ReceiveTask and UserTask)

5) Status: EMMOS's definition of Status restricts BPMN's definition of ThrowEvent to a state of a temporal part of a process, and as a result, that a Status is a **type of** ThrowEvent. Accordingly, the Status process is preserved in the EMMO ontology. Concerning the relationships, four associations which previously did not exist in EMMO have been added in the integrated version. It consists of (1) Status is a **type of** Event, (2, 3 and 4) Status **has type** EndEvent, ImplicitThrowEvent and IntermediateThrowEvent.

6) SubProcess and Stage: SubProcess and Stage's definitions, as reviewed in Section III-B6, restrict the definition of InteractionNode. They are both preserved in the EMMO ontology. Moreover, to express that these concepts may correspond to points where participants get together to achieve something or to exchange information, new associations are defined between them and the participants.

7) WorkFlow: Provided the tight analogy between Wokflow from the EMMO ontology and the SequenceFlow from BPMN, our strategy was to use the restriction extension mechanism and, consequently, to preserve the concept of WorkFlow in the integrated ontology. Two associations are needed to complete the ontology integration with some workflow–related semantics coming from BPMN: (1) the WorkFlow **is source**

of and **targets** FlowNode and (2) the WorkFlow is a **type of** FlowElement.

8) EncodedData: EncodedData from EMMO has a precise meaning compared to ItemAwareElement from BPMN, which has more for the purpose of specifying a collection of data. On the opposite, the ItemAwareElement may be of various types described in [20]: DataObject, DataStore, DataOutput and DataInput. Hence, EncodedData will remain in the integrated ontology. Finally, one additional association must be integrated: EncodedData **is source** and **is target** of DataAssociation.

D. Validation by Incoherence Solving

In general, validating a single ontology involves checking whether the ontology adheres to certain principles and standards [27]. Here are the type of validations that can be encountered and applied: syntax validation (*Does the ontology follows the correct syntax and format of the ontology language?*), consistency validation (*Is the ontology internally consistent?*), completeness validation (*Does the ontology covers all the necessary concepts and relationships in the domain?*), coherence validation (*Is the ontology coherent with other ontologies and standards in the same domain?*), usability validation (*Is the ontology easy to use and understand?*), but also the validation of specific ontology criterion such as the accuracy, coverage, scalability, and maintainability.

In the case of the validation of the integration of one ontology with another (BMPN with EMMO), we may assume that the above validation types have been achieved during the design of each specific ontology and that the item left to be validated is merging part itself, to know: Checking for inconsistencies between the ontologies. In general, this can be achieved by using a reasoner that checks for logical consistency (e.g., Pellet [28]), such as whether there are unsatisfiable classes or cycles in the hierarchy.

In the integration of BPMN within EMMO (Figure 3), we illustrate the validation by discussing one type of incoherency manually discovered in the integrated ontology. This incoherence is a cycle in the hierarchy that has been introduced between in the concept of **ElementaryWork** from EMMO and **InteractionNode** from BPMN. Solving this incoherency, in this case, requires a deeper analysis of both source ontologies. Therefore, by analysing EMMO and BPMN, it can be argued that an **ElementaryWork** can be considered a type of **InteractionNode** because an elementary work is a basic process that involves the transformation of materials, energy, or information, often through the application of energy such as heat or mechanical work. This transformation typically involves some kind of interaction between two or more entities, such as a chemical, an electrical or even a nuclear reaction or a physical change in state. Moreover, an **InteractionNode** is a node representing any type of interaction between two or more entities in a business process model. This can include tasks, events, and gateways, which are used to model different types of interactions. Therefore, it can be argued that an elementary work, which represents a basic process that transforms materials, energy, or information, can also be considered an **InteractionNode** because it involves an interaction between two or more entities, even if it is a more fundamental type of interaction compared to other types of nodes. Hence, both **ElementaryWork** and **InteractionNode** represent different types of nodes in a business process model, but an **ElementaryWork** can be seen as a more fundamental type of interaction that involves the transformation of materials, energy, or information, making it a type of **InteractionNode** in a broader sense. As a consequence, the decision was made during the Validation by Incoherence Solving step to keep the link "ElementaryWork is a type of InteractionNode" in the integrated model while removing the link "InteractionNode is a type of ElementaryWork".

IV. Conclusions and future works

This paper enhances the process of open innovation within the materials industry [2]. To achieve this goal, we propose an approach never achieved before that involves integrating two ontologies to unify the innovation process with the processing of materials. Specifically, we utilise the BPMN ontology to support the open innovation process and the EMMO ontology to describe the materials. By merging these two ontologies, we can create a more comprehensive framework that can facilitate collaboration and innovation in the materials industry. This integration aims to streamline the workflow, improve communication, and enhance the understanding of materials, leading to more effective and innovative solutions.

Our proposed approach consists of four key steps: Alignment, Mapping, Integration, and Validation by Incoherence Solving. While alignment and mapping are relatively straightforward, the integration step requires more careful consideration. For instance, we have observed that when the extension mechanism takes the form of a restriction, the BPMN concept is not taken into account. On the other hand, when the extension mechanism is an inheritance, the EMMO concept is extended with the attributes inherited from BPMN. So far, we have not yet encountered a case where the extension mechanism involves a type extension by adding new axioms.

This paper presents a significant step towards achieving open innovation in the materials industry. The ontological integration of BPMN with EMMO can bring about transformative changes to the field by enabling faster and more cost-effective research and development processes and creating innovative material solutions to address complex challenges. Our approach also has the potential to improve communication and streamline workflows, which can lead to greater efficiency and productivity. Moreover, the integration of these two ontologies can enhance the understanding of materials and provide a more detailed description of materials in the innovation process. As a result, organisations can gain a competitive advantage by leveraging this approach and advancing the materials industry with novel and impactful solutions. Overall, our proposed approach has the potential to revolutionise the field of materials science and accelerate progress towards open innovation.

As future works, much must be done to improve ontology and

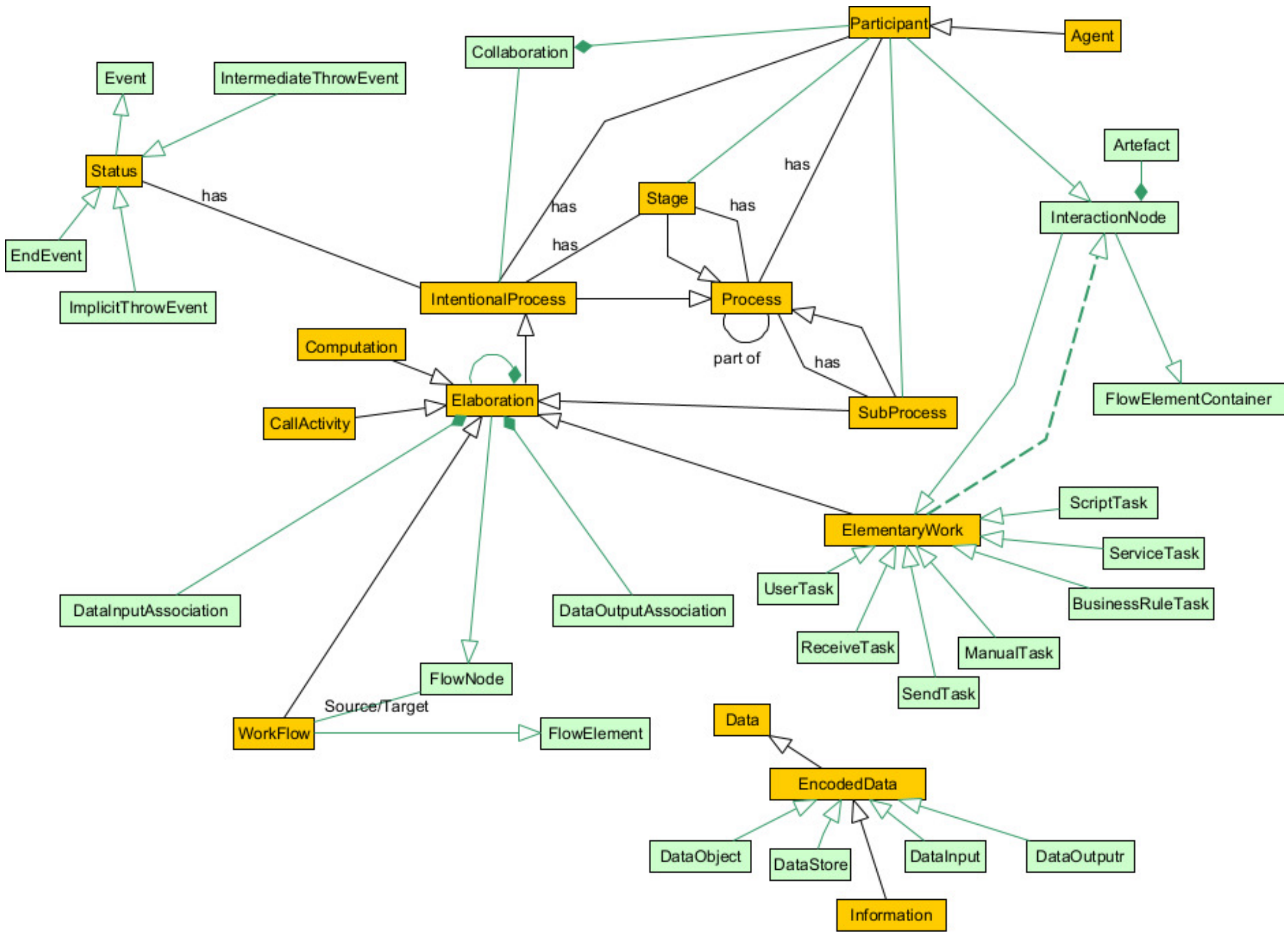


Fig. 3: Integrated BPMN's concepts within EMMO ontology. On this schema, the concepts from EMMO ontology are represented in orange and the concepts from BPMN are in green.

its application in real-world scenarios. Our first priority is to enhance the integration of the two ontologies by potentially considering other concepts that have not yet been integrated. By doing so, we aim to create a more robust and complete ontology that can capture all relevant aspects of the domain. Secondly, we must conduct further analysis to identify potential incoherencies in the integrated ontology. While we have already identified one example of a cyclic association between the ElementaryWork and the InteractionNode, additional review and analysis are necessary to ensure that no remaining incoherencies exist. Thirdly, we plan to validate the ontology by applying it to a real case and observing to what extent it is possible to consider all the dimensions of the real scenario with the integrated ontology. This will allow us to assess the advantages of using the integrated ontology in practice and identify any areas for further improvement. Finally, we aim to deploy the ontology in a software tool like *Protégé* [29] to facilitate its manipulation and representation. This will enable other researchers and practitioners to use the ontology easily and effectively in their own work. Overall, our goal is to create a more comprehensive, coherent, and useful ontology that can help researchers and practitioners better understand and navigate the complex domain. We hope our ongoing efforts will lead to further advancements in the field and contribute to developing more effective tools and applications.

ACKNOWLEDGMENT

This work was supported by the VIPCOAT project: A virtual open Innovation Platform for active protective coatings guided by modelling and optimisation. VIPCOAT project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952903.

REFERENCES

[1] P. Klein, N. Konchakova, D. G. Hristova-Bogaerds, M. Noeske, A. Simperler, G. Goldbeck, and D. Höche, "Translation in materials modelling: Process and progress," OntoTrans – FORCE, White Paper, 2021.

[2] "The open innovation publications," https://digital-strategy.ec.europa.eu/en/library/open-innovation-publications, accessed: 2023-2-24.

[3] M. T. Horsch, S. Chiacchiera, M. A. Seaton, I. T. Todorov, K. Šindelka, M. Lísal, B. Andreon, E. Bayro Kaiser, G. Mogni, G. Goldbeck *et al.*, "Ontologies for the virtual materials marketplace," *KI-Künstliche Intelligenz*, vol. 34, pp. 423–428, 2020.

[4] S. Belouettar, C. Kavka, B. Patzak, H. Koelman, G. Rauchs, G. Giunta, A. Madeo, S. Pricl, and A. Daouadji, "Integration of material and process modelling in a business decision support system: Case of composelector h2020 project," *Composite Structures*, vol. 204, pp. 778–790, 2018.

[5] N. Konchakova, H. A. Preisig, C. Kavka, M. T. Horsch, P. Klein, and S. Belouettar, "Bringing together materials and business ontologies for protective coatings," *FOMI 2022: Formal Ontologies Meet Industry*, 2022.

[6] H. Tomaskova, P. Maresova, M. Penhaker, M. Augustynek, B. Klimova, O. Fadeyi, and K. Kuca, "The business process model and notation of open innovation: The process of developing medical instrument," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 5, no. 4, p. 101, 2019.

[7] C. Kavka, D. Campagna, M. Milleri, A. Segatto, S. Belouettar, and E. Laurini, "Business decisions modelling in a multi-scale composite material selection framework," in *2018 IEEE International Systems Engineering Symposium (ISSE)*. IEEE, 2018, pp. 1–7.

[8] G. Goldbeck, E. Ghedini, A. Hashibon, G. Schmitz, and J. Friis, "A reference language and ontology for materials modelling and interoperability," 2019.

[9] "Notation (bpmn) version 2.0 howpublished =https://www.omg.org/spec/bpmn/2.0, author=OMG, publisher=PDF."

[10] E. Ghedini, A. Hashibon, J. Friis, G. Goldbeck, G. Schmitz, and A. De Baas, "Emmo the european materials modelling ontology," in *EMMC Workshop on Interoperability in Materials Modelling*. St John's Innovation Centre Cambridge, 2017.

[11] S. Zivkovic, H. Kuhn, and D. Karagiannis, "Facilitate modelling using method integration: An approach using mappings and integration rules," *ECIS 2007 Proceedings*, pp. 2038–2049, 2007.

[12] C. Feltus, E. Dubois, and M. Petit, "Alignment of remmo with rbac to manage access rights in the frame of enterprise architecture," in *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*. IEEE, 2015, pp. 262–273.

[13] C. Feltus and A. Rifaut, "An ontology for requirements analysis of managers' policies in financial institutions," in *Enterprise Interoperability II: New Challenges and Approaches*. Springer, 2007, pp. 27–38.

[14] S. Spaccapietra and C. Parent, "Database integration: the key to data interoperability," *Advances in Object-Oriented Data Modeling*, pp. 221–253, 2000.

[15] C. Feltus and E. H. Proper, "Conceptualization of an abstract language to support value co-creation," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2017, pp. 971–980.

[16] C. Feltus, E. H. Proper, and K. Haki, "Towards a language to support value cocreation: An extension to the archimate modeling framework," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2018, pp. 751–760.

[17] S. Ferndriger, A. Bernstein, J. S. Dong, Y. Feng, Y.-F. Li, and J. Hunter, "Enhancing semantic web services with inheritance," in *The Semantic Web-ISWC 2008: 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings 7*. Springer, 2008, pp. 162–177.

[18] S. Belouettar, C. Kavka, B. Patzak, H. Koelman, G. Rauchs, G. Giunta, A. Madeo, S. Pricl, and A. Daouadji, "Integration of material and process modelling in a business decision support system: Case of composelector h2020 project," *Composite Structures*, vol. 204, pp. 778–790, 2018.

[19] J. Taylor and J. Purchase, *Real-world decision modeling with DMN*. Meghan-Kiffer Press Tampa, 2016.

[20] A. Correia, "Elements of style of bpmn language," *arXiv preprint arXiv:1502.06297*, 2015.

[21] C. Kavka, D. Campagna, and H. Koelman, "A business decision support system supporting early stage composites part design," in *Advances in Computational Methods and Technologies in Aeronautics and Industry*. Springer, 2022, pp. 263–279.

[22] M. T. Horsch, D. Toti, S. Chiacchiera, M. A. Seaton, G. Goldbeck, and I. T. Todorov, "Osmo: Ontology for simulation, modelling, and optimization," 2021.

[23] M. Büschelberger, J. F. Morgado, K. Frei, C. Eichheimer, J. Boehm, A. Calzolari, and A. Hashibon, "Report on intersect developed ontologies and moda."

[24] M. T. Horsch, C. Niethammer, G. Boccardo, P. Carbone, S. Chiacchiera, M. Chiricotto, J. D. Elliott, V. Lobaskin, P. Neumann, P. Schiffels *et al.*, "Semantic interoperability and characterization of data provenance in computational molecular engineering," *Journal of Chemical & Engineering Data*, vol. 65, no. 3, pp. 1313–1329, 2019.

[25] D. Nicolas, C. Feltus, and D. Khadraoui, "Multidimensional dih4cps ontology," *21st International Conference on e-Society*, p. 20, 2023.

[26] ——, "Towards a multidimensional ontology model for dih-based organisations," *International Journal of Knowledge and Systems Science (IJKSS)*, vol. 14/1, 2023.

[27] R. Cobe and R. Wassermann, "Ontology merging and conflict resolution: Inconsistency and incoherence solving approaches," *BNC@ ECAI 2012*, p. 20, 2012.

[28] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007.

[29] "Protégé – the open-source ontology editor and framework," https://protege.stanford.edu/, accessed: 2023-2-24.

DOI: 10.15439/2023F4268
ISSN 2300-5963 ACSIS, Vol. 35

Compiler Support for Parallel Evaluation of C++ Constant Expressions

Andrew Gozillon
0000-0001-7558-7166
Advanced Micro Devices AB,
Nordenskioldsgatan 11 A, Office 233
211 19 Malmö, Sweden
Email: andrew.gozillon@amd.com

Seyed H. Haeri
0000-0002-7969-8573
University of Bergen, Norway & PLWorkz R&D, Belgium
Av. Chapelle-aux-Champs 49, Brussels, Belgium
Email: hossein@uib.no

James Riordan, Paul Keir
0000-0001-6516-9446
0000-0002-4781-9377
University of the West of Scotland
High St., Paisley PA1 2BE, Scotland, United Kingdom
Email: {james.riordan, paul.keir}@uws.ac.uk

***Abstract*—Metaprogramming, the practice of writing programs that manipulate other programs at compile-time, continues to impact software development; enabling new approaches to optimisation, static analysis, and reflection. Nevertheless, a significant challenge associated with advanced metaprogramming techniques, including the *constexpr* functionality introduced to C++ in 2011, is an increase in compilation times. This paper presents ClangOz, a novel Clang-based research compiler that addresses this issue by evaluating relevant constant expressions in parallel, thereby reducing compilation time.**

ClangOz includes a set of compiler intrinsics that allows developers to parallelise their own code and take full advantage of recent constexpr language support. By utilising parallel evaluation, ClangOz significantly reduces the compile time for metaprogramming-intensive codebases, enhancing developer productivity and iterative software development processes.

Benchmark results demonstrate the performance advantage of ClangOz over traditional compilers, including a decrease in compilation times across all benchmarks; and parallel efficiency of up to 95% in one case. The evaluation of constant expressions in parallel unlocks substantial speedups, enabling developers to leverage advanced metaprogramming techniques without sacrificing compilation efficiency.

We highlight the opportunities afforded by the constexpr functionality and emphasise the importance of compiler support for efficient metaprogramming. By introducing ClangOz, a compiler tailored for parallel evaluation of relevant constant expressions, developers can utilise modern metaprogramming while minimising compilation times parametrically.

I. Introduction

COMPILE time metaprogramming in C++ has been of interest since the discovery that C++ templates were Turing complete [1]. Exploration of compile time metaprogramming has resulted in the addition of constant expressions to the language; a concept proposed in 2003 [2]; and added in C++11 with the inclusion of the *constexpr* specifier. A constant expression is an expression which would remain constant at runtime and could thus be evaluated at compile time. The *constexpr* specifier allows functions and variable declarations to assert that they can be evaluated at compile time. With the addition of this specifier, compile time programming has become more approachable, with a syntax almost identical to runtime code.

Since the addition of constant expressions to C++, the standard library specification has begun to incorporate support for both compile time and runtime execution for its functionality. With the increasing *constexpr* support, larger program segments can now be evaluated at compile time. However, as more components are evaluated at compile time, so too do compilation times increase. Adding parallelism to a program can help increase performance when used correctly. Yet parallelism is currently only available in runtime contexts; there is no existing concept of C++ compile time parallelism.

In this article we introduce ClangOz [3], an experimental Clang [4] compiler which adds support for the parallel execution of for-loops at compile time. ClangOz seeks to give users control of parallelism through compiler intrinsics. The intrinsics are a set of functions built into the compiler, which may be utilised to convey information about the algorithm being *constexpr* parallelised to ClangOz. A higher level application programming interface (API) is also provided which builds upon the *execution policy* overloads of existing standard C++ runtime library functions such as *std::for_each*, to allow easier access to *constexpr* parallelism.

A survey of related and relevant literature is presented in Section II. Section III covers the ClangOz compiler, discussing its architecture; parallel constant expression evaluation; and intrinsics library, along with a concise implementation example of *std::for_each*. Before concluding in Section V, Section IV reports on experiments, with benchmarks implemented using the novel compile time parallelism feature, and considers performance and scaling in comparison with serial counterparts.

II. Background

Parallelism within compilers is not new, and much research has been undertaken, aiming to speed up different phases of the compiler. For example, investigation on the parallelisation of parsing [5], assembling [6], semantic analysis [7], lexical analysis [8] and code generation [9] compiler phases have been conducted. Despite this, most modern compilers avoid the additional complexity of adding parallelism for performance. The research presented in this paper differs from the prior art as it pertains to a smaller segment of the compiler; a subsection of the semantic analysis process. Nevertheless, it does adds an overhead for compiler developers; even if it is smaller in scope. This work is also distinct in placing the parallelism into the users' hands through an API, making the parallelism explicitly programmable.

The landscape of C++ compile time programming has continued to expand in recent years. In the 2020 iteration of the language (C++20 [10]) dynamic memory allocation and deallocation at compile time was added [11]. This allowed the creation of variable size containers at compile time. C++20 also introduced a feature called *Concepts* [12], allowing a user to constrain template instantiation according to composable boolean predicates. Concepts allow for increased user defined type-safety within code bases, while improving error messages. Further proposals are pending, with the most interesting possible additions being metaclasses [13] and reflection [14]. The former allows users to define a compile time function that manipulates how a class's definition is generated; for example to make member functions of a class public by default. The latter allows deeper compile time introspection of types; for instance, to check the names of class members.

Additions like these to the language specification have allowed for projects that were previously impossible. The processing of regular expressions at compile time [15], static reflection through a library rather than the language [16], big-integer computation [17] and compile time functional composition [18] are prominent examples. Such projects require a sizable amount of computation at compile time, and would benefit from acceleration by *constexpr* parallelisation.

Providing language features to allow processing at compile time is not unique to C++. Lisp [19], D [20], Rust [21], Julia [22], Elixir [23] and Circle [24] all give various compile time facilities. Lisp was the first language with Turing-complete compile time functionality; Lisp provides this feature in the form of macros which, unlike C-style macros, can perform computation as well as text substitution. The D programming language has many similarities to C++. Its compile time features are based upon it, with the intent to simplify them. The D language allows compile time programming using constructs similar to C++ templates and constant expressions, though it extends these concepts with the introduction of eponymous and nested templates. In contrast: Rust, Julia and Elixir make use of Lisp-style macros that manipulate the AST for their compile time metaprogramming. Rust and Julia also support compile time computation through constant expressions that share similarities with C++. The Circle language is interesting as it builds on-top of C++17, adding a host of new data-driven metaprogramming features including range operators and pack generators. The concepts introduced in this paper could in turn be extended to such programming languages, having similar compile time capabilities, albeit in a different guise when the capabilities are macro based.

The evaluation of constant expressions at compile time has parallels in the field of partial evaluation [25], where programs are specialised dynamically at runtime or statically at compile time to achieve better performance. Partial evaluation of programs can lead to optimisations including constant folding; code simplification; strength reduction; and control flow optimisation. All such optimisations are possible through the explicit utilisation of *constexpr* within C++ to specialise code; and we recognise that the comparison of partial evaluation and C++'s compile time features has been made before [26]. Research into partial evaluation and its applications have been ongoing for many years; and recently applied to the field of High-Performance Computing with the aim of increasing performance in a myriad of ways. For example, using static partial evaluation to optimise memory access patterns on the GPU [27]; creating domain specific languages utilising partial evaluation to facilitate high-performance libraries for accelerators [28]; and in the development of compilers and interpreters for dynamic languages that utilise partial evaluation to speculatively optimise code [29]. In a similar vein to the work in this paper for compile time evaluation, some research on parallel evaluation of partial evaluation has also been conducted. Some examples are the parallelisation of a partial evaluator utilised in the specialisation of mutually recursive procedures [30]; distributed parallelisation of partial evaluators within programming languages [31]; and parallelisation of partial evaluations within evolutionary algorithms [32].

III. Compile Time Parallelism

The ClangOz[1] compiler builds on Clang by adding parallelisation support to *for* loops in specific *constexpr* contexts. This is performed using an API of four intrinsic functions[2], used to communicate to the compiler how a loop should be parallelised. The intrinsic calls are placed within the function body containing each targeted loop; and assist with loop dependency analysis [33].

When the following constraints have been met within a *constexpr* function, ClangOz will use these intrinsics to gather the information required to partition the loop body across multiple CPU threads. There are two constraints that have to be met by a *for* loop to be *constexpr* parallelised. First, it must be within a *constexpr* function; and adjacent to the appropriate parallelisation intrinsics. Second, the enclosing function must include our new C++ execution policy class [34] within its parameter list; and be invoked with the corresponding object as an argument. An execution policy parameter allows an algorithm designer to overload a function's behaviour based on the

[1] The compiler has no connection to Mozart and the Oz language.

[2] These are not *Clang* intrinsics per se; though they perform a similar role.

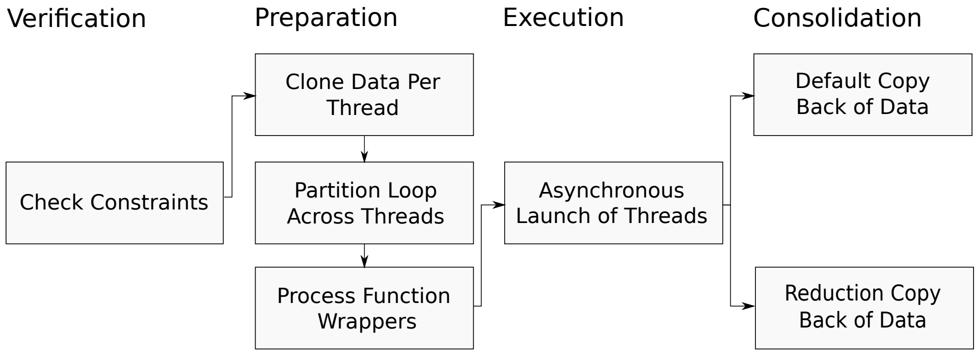

Verification
Preparation
Execution
Consolidation
Check Constraints
Clone Data Per Thread
Partition Loop Across Threads
Process Function Wrappers
Asynchronous Launch of Threads
Default Copy Back of Data
Reduction Copy Back of Data

Fig. 1. Compilation Phases involved in the Parallelisation Process

type of each distinct policy. For example, a *std::for_each* function may be passed a *std::execution::parallel_policy* which indicates that the *std::for_each* should select an overload that has a parallelised implementation. In the case of ClangOz a new *constexpr_parallel_policy* was added to ClangOz's C++ standard library (a modified version of Clang's implementation of the C++ standard library: libc++) which is used to indicate that a function should be *constexpr* parallelised if possible.

These constraints define a minimal C++ API that *constexpr* functions must meet to undergo the parallelisation process. The constraints were added as a way to limit the scope at which ClangOz would try to apply its parallelisation process.

There are some limitations of the ClangOz compiler worth noting. First, there is no support for nested parallelism; only the outer loop of a loop nest is parallelisable. Second, only a single loop is parallelisable within each function. Thirdly, only container argument types owning *contiguous* data are supported; and container objects must utilise a pointer-based iterator. An example usage of a *constexpr* parallel *std::for_each* from ClangOz's modified libc++ can be found in Listing 1. *execution::ce_par* is a simple, 1-byte, pre-constructed *constexpr_parallel_policy* object. Passing this policy into the *std::for_each* indicates to use a parallel implementation of the function; if one exists. The 4th argument, a C++ lambda function, is then executed in parallel on the elements of the *std::array*.

Listing 1 ClangOz's modified libc++ supports a new *ce_par* policy parameter, allowing users to avoid compiler intrinsics.

```
constexpr auto f() {
  std::array<int, 4> arr {};
  std::for_each(execution::ce_par,
                arr.begin(), arr.end(),
                [](int &i) { i++; });
  return arr;
}
```

The parallelisation process takes place within Clang's constant expression evaluator. The constant expression evaluation executes within the frontend, usually during the semantic analysis process; either when generating the abstract syntax tree (AST), or during later code generation.

The constant expression evaluator attempts constant folding on expressions stored in AST nodes; collapsing them into a value or values at compile time by processing the expression. The values are calculated and stored using Clang's *APValue* class. This class holds constant data of arbitrary bit-widths for several C++ value types; including *float*, *integer* and arrays.

Manipulation of *APValue* objects is pivotal to the parallelisation process, and in particular those that are *LValue*s. Generally, *LValue*s are locators for objects. An *LValue* can contain either the path from a complete object to its subobject; or a memory address offset. These are important as the majority of the parallelised standard C++ library algorithms use *iterators*: an idiomatic abstraction over the traversal strategy of each container. The parallelisation process manipulates and gathers information from these iterators; with pointers a common form.

Clang's *CallStackFrame* and *EvalInfo* classes are also integral. The former acts as a call stack for the constant expression evaluator; maintaining information for the current call stack frame, and tracking the arguments passed to the frame and the temporaries that reside within it. Alongside, a pointer to the preceding frame in the stack is also stored, to facilitate backwards traversal. The *EvalInfo* class maintains information about the expression being evaluated, including the *CallStackFrame*. These classes maintain most of the evaluator's state during evaluation.

Two Clang AST components that are useful for the parallelisation process are the *Expr* and *Decl* family of classes. The former maintains information about types of expressions; for example *CallExpr* maintains information about function invocations. The latter, tracks declarations or definitions of different language constructs; for example information on each function definition is stored within a *FunctionDecl*.

A. The Parallelisation Process

The parallelisation process consists of four phases (See Fig. 1). The first phase is *verification*, confirming that the two constraints have been satisfied. These constraints are checked whenever a *for* loop is encountered within a *constexpr* context.

TABLE I
THE CLANGOZ INTRINSIC FUNCTIONS

Intrinsic Function Code	Function and Parameters Synopsis

`template <class T, class U>`
`constexpr void`
`__BeginEndIteratorPair(`
`  T& Begin,`
`  U& End`
`)`
	Indicates the range of a *for* loop, allowing the partitioning process to split work across multiple threads. The *__BeginEndIteratorPair* or *__PartitionUsingIndex* intrinsic are required and the minimum necessary for parallelisation.
	• **Begin**, **End**: Indicates the beginning and end of the loops range.
	• **Type requirements**: *T* and *U* must be a one member iterator where the member is a pointer or a pointer.

`template <class T, class U>`
`constexpr void`
`__PartitionUsingIndex(`
`  T LHS,`
`  U RHS,`
`  RelationalType RelTy`
`)`
	Indicates the range of a *for* loop, allowing the partitioning process to split work across multiple threads. The *__BeginEndIteratorPair* or *__PartitionUsingIndex* intrinsic are required and the minimum necessary for parallelisation.
	• **LHS**, **RHS**: Indicates the beginning and end of the loops range.
	• **RelTy**: Indicates the relational operator used within the loop's condition e.g. >, <, !=, <=, >=.
	• **Type requirements**: *T* and *U* must be a numeric type, such as an integer.

`template <class T>`
`constexpr void`
`__IteratorLoopStep(`
`  T& StartIter,`
`  OperatorType OpTy,`
`  const T& BoundIter`
`)`
	States *StartIter* is bound to the loops step. Thread clones will initially be offset by invoking operator based mutation the same number of steps taken by the thread partitions loop at its start point. The operator used for mutation is indicated by *OpTy*.
	• **StartIter**: Indicates the variable that will be offset.
	• **OpTy**: Indicates the prefix or postfix operator (e.g. ++) used for mutation.
	• **BoundIter**: Indicates the boundary of *StartIter* if one exists, preventing offsetting past the boundary.
	• **Type requirements**: *T* must be a one member iterator where the member is a pointer or a pointer.

`template <class T>`
`constexpr void`
`__ReduceVariable(`
`  T Var,`
`  ReductionType RedTy,`
`  OperatorType OpTy`
`)`
	Indicates that a container or value should be reduced when the launched threads are joined. Three types of reduction are supported *PartitionedOrderedAssign*, *OrderedAssign* or *Accumulate*.
	• **Var**: Notates the variable that should be reduced on thread completion.
	• **RedTy**: Indicates the reduction method to be used by the compiler.
	• **OpTy**: States the operator, if any, used to mutate the variable in the reduction step.
	• **Type requirements**: *T* must be a vector or array iterator or numeric value.

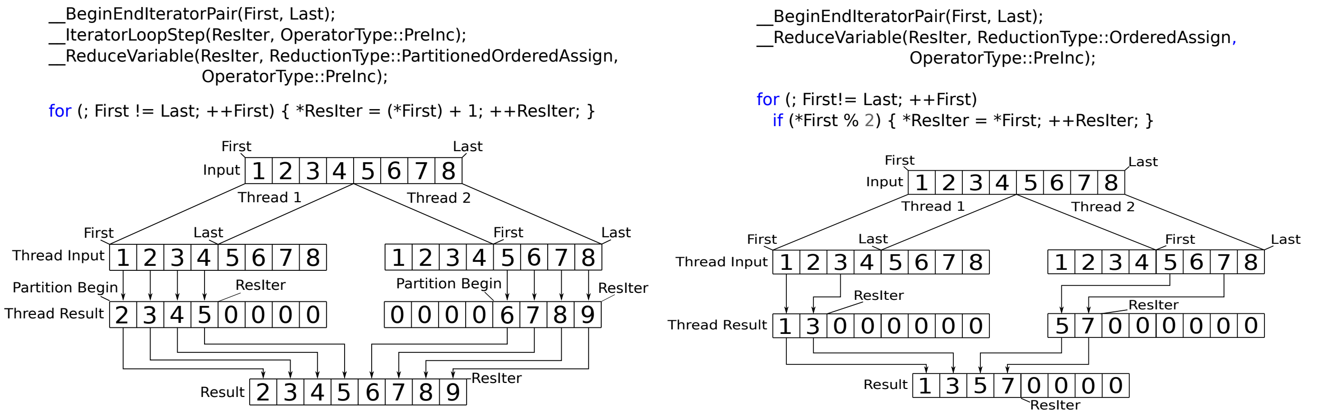
The first step checks the enclosing context is a function, before iterating over the function's parameters to detect if the function takes a *constexpr_parallel_policy* as an argument. As nested parallelism is not supported, the second check makes sure that no other *constexpr* parallel tasks are in flight.

The second phase is *preparation*, where the intrinsics are processed; local data is prepared for each thread; and the loop space is partitioned. This phase involves creating a clone of the *EvalInfo* object per thread, as well as each of the *CallStackFrame*s it contains. The *APValue*s that reside in each *CallStackFrame* are also cloned; representing both dynamically and statically allocated data.

After the data has been cloned the partitioning process begins, using static loop partitioning to divide the work across multiple threads. If the data cannot be divided evenly across threads which are maintained by a single thread pool; any excess work is given to the final thread. This partitioning process is part of the *LoopIntrinsicGatherer* class, an addition to ClangOz that implements the functionality for handling the intrinsics, cloning data and reducing data. The partitioning is reliant on the *__BeginEndIteratorPair* or *__PartitionUsingIndex* intrinsic being used by the creator of the function to specify the loop bounds. These and other intrinsics are discussed further in Section III-B.

The intrinsics are discovered prior to the partitioning process by traversing the function body containing the loop, statement by statement, checking the name of each function called against the list of intrinsic names. This is done by making the *LoopIntrinsicGatherer* a child of *Clang*'s *ConstStmtVisitor* which recursively visits a *Stmt*, breaking it into its constituent *Stmt* types. Each *Stmt* in the body of the *FunctionDecl* is then iterated over, and passed to the recursive visitor, which then searches for *CallExpr* nodes to verify and process.

Each thread has copies of the variables defining the loop



```
__BeginEndIteratorPair(First, Last);
__IteratorLoopStep(ResIter, OperatorType::PreInc);
__ReduceVariable(ResIter, ReductionType::PartitionedOrderedAssign,
                 OperatorType::PreInc);

for (; First != Last; ++First) { *ResIter = (*First) + 1; ++ResIter; }
```

```
__BeginEndIteratorPair(First, Last);
__ReduceVariable(ResIter, ReductionType::OrderedAssign,
                 OperatorType::PreInc);

for (; First!= Last; ++First)
  if (*First % 2) { *ResIter = *First; ++ResIter; }
```

Fig. 2. Example PartitionedOrderedAssign (Left) and OrderedAssign (Right) Reduction With Two Threads

bounds; provided in the initialisation, condition and iteration statements. Dividing the workload across threads requires offsetting the underlying *APValue*s of these variables; and in particular those found in the loop's condition statement, which help to define its range. In the case of the standard library algorithms, the loop conditions are equality checks; for example, comparing the start and end pointers from a container to check that they are not equal. It is possible to offset the pointer's *APValue* to point to the start and end of the loop partition for each thread, effectively segmenting the loop.

To calculate the appropriate size of each partition, and the amount required to offset the loop's start and end by, the size of the iteration space must be calculated. This distance can then be divided by the number of partitions provisioned. For loop bounds defined by integer values, this is straightforward. With pointers, the size of the container's element type is required; and memory addresses to a contiguous container are traversed using an offset based on the size of the element type. Calculating the distance requires utilising this size to convert from a memory address to an integral number representing the loop's range. This can then be used to calculate the offset for each partition, and then each pointer can be offset by the appropriate amount. Only containers of contiguous data are supported, as partitioning non-contiguous data involving arbitrary memory locations is non-trivial, and time intensive.

After the work has been distributed, the third phase begins: the *execution* phase. Tasks, encapsulated as C++ function objects (often lambda functions), are launched asynchronously, and then a *wait* for completion is issued. The parallelised task contains the *constexpr* evaluation of the body of the loop. The initialisation and destruction steps in the loop's evaluation are executed sequentially, and occur once on loop entry and exit. The task itself does not deviate from the original sequential algorithm.

The final phase after thread completion is *consolidation*. This phase focuses on synchronising thread data back into the main process's *CallStackFrame* and *EvalInfo*, allowing sequential evaluation to continue. This is done in two steps, the first copies the cloned data back into its original location. The second step involves an optional reduction, and is controlled by the *__ReduceVariable* intrinsic discussed in Section III-B. Data that is marked for reduction by *__ReduceVariable* skips the first step.

Data which has been cloned, is split into two components before being copied back. The second component is specific to array data, the first is for everything else. A primary thread is selected to copy data for the non-array component, which is dependent on an *EvalStmtResult* object returned by each parallel task. This *EvalStmtResult* is a Clang enumerator that contains different evaluation result flags for statements. Each returned *EvalStmtResult* is checked: if all return successfully, then the final thread is selected as the primary thread. As the whole loop range was iterated across, the newest values should be contained within the final partition space. In other cases, where threads complete early, perhaps due to encountering *return* or *break* statements, the first thread that signalled early completion is selected. This ensures that values in later partitions are ignored, as they would not be processed when executing the loop sequentially.

The re-synchronisation of arrays is done by determining which elements have been written to by each thread and then copying these elements from the respective clone, to the original. This does not factor in alteration of the same array element by multiple threads.

Applying reductions can be thought of as a special case of the first step which can be requested by a user through the *__ReduceVariable* intrinsic when a more complex data synchronisation method is required. There are several different types of reduction possible, and these are discussed in Section III-B.

B. The Intrinsic Functions

The compiler intrinsics are used as standard C++ functions, to communicate to the compiler how a loop is to be parallelised. They are implemented as functions rather than Clang intrinsics as it simplified modifications to the parallelisation process. This use of an API of intrinsics has much in common

```
template <class _ExecutionPolicy, class _ForwardIterator, class _Function>
constexpr __enable_if_constexpr_par_execution_policy_t<_ExecutionPolicy, void>
for_each(_ExecutionPolicy&& __exec, _ForwardIterator __first,
         _ForwardIterator __last, _Function __f)
{
    __BeginEndIteratorPair(__first, __last);
    __ReduceVariable(__first, PartitionedOrderedAssign, PreInc);

    for (; __first != __last; ++__first)
        __f(*__first);
}
```

Fig. 3. *Constexpr* parallelised libc++ *std::for_each* implementation

with OpenMP [35] and other directive-based programming paradigms.

The intrinsics required to describe the parallelisation of a loop should be placed prior to the loop within a function that meets the aforementioned constraints. There are four different intrinsic functions used for parallelisation, with descriptions listed in Table I. The intrinsics have no body and are no-ops at runtime with a trivial overhead at compile time. They are defined as function templates so that they can be used with a variety of different types. The name of the intrinsics are prefixed with double underscores to avoid conflicts with user-defined functions. The parameters of each intrinsic allow users to pass important information to the compiler.

__BeginEndIteratorPair and *__PartitionUsingIndex* indicate to partition iterations of the loop across multiple threads based on the loop bounds indicated by their arguments. The former was designed with the use of C++ standard library containers and algorithms in mind, which make use of *begin* and *end* iterators to mark the range of loops. The latter was designed with numeric loop conditions in mind and takes an extra parameter indicating the relational operator used within the loop's condition clause.

__IteratorLoopStep indicates that a pointer based index is bound to the loop's step. Clones of the index are offset by the number of loop steps taken by the loop in the thread partition at its start point. The offset is calculated by mutating the index by the number of loop steps taken by the C++ operator indicated by the *OpTy* argument. This keeps the bound value synchronised with the loop across all threads, and is used for indices not used within *__BeginEndIteratorPair*.

The intrinsic *__ReduceVariable* helps to denote how a container or value should be reduced when the launched threads are joined. Three reduction types are supported: *PartitionedOrderedAssign*, *OrderedAssign* and *Accumulate*. *Accumulate* is used in conjunction with an accumulator variable, ensuring that the local result from each thread is combined using the specified operator to obtain a final value. *PartitionedOrderedAssign* is intended for use with containers, and its operation is illustrated in Fig. 2. This reduction assigns elements to the original container in order, where each element is taken from the starting offset in each thread partition, to its final offset on thread completion. This allows appropriate collapse of data as threads are working on local copies of data rather than shared data. *OrderedAssign* (also Fig. 2) is similar to *PartitionedOrderedAssign*, although it is used with containers that have not been modified in lock step with the loop. *OrderedAssign* assigns elements to the original container in order, where each element is taken from the initial offset of each cloned container to its final offset within its partition.

C. An Example Constexpr Parallel Function

Within ClangOz's libc++ library, 30 of the functions contained inside the Algorithms and Numerics libraries have been *constexpr* parallelised. In Fig. 3 a *std::for_each* is shown as an example of how a function can be *constexpr* parallelised. The function takes an execution policy as its first parameter which will be verified by the compiler before it attempts parallelisation. In this example there is also an alias for an *std::enable_if* check, which ensures the correct execution policy is used in conjunction with this variation of *std::for_each*. Alternatively, the compiler will select a more apt function if one exists, or issue an error message.

Once the policy has been verified, each of the intrinsics is processed; and in this example there are only two. The first, *__BeginEndIteratorPair* defines the loop's range which the parallelisation process uses to partition the loop across multiple threads. In this case the range is delimited by the arguments *__first* and *__last*. The second intrinsic *__ReduceVariable* states that a *PartitionedOrderedAssign* should be performed on the data pointed to by the argument *__first*. *OperatorType::PreInc* indicates which operator to use when traversing the data, allowing the compiler to correctly reduce the data.

IV. Parallelism Benchmarking

Three *constexpr* programs based on existing benchmarks were created to test the performance of the *constexpr* parallelism implementation. The original benchmarks were modified to utilise function templates from ClangOz's C++ standard library supporting *constexpr* parallel execution. Benchmark selection was also mindful that certain language features are not available at compile time; for example assembly instructions or *goto* statements.

The *Blackscholes* benchmark is taken from The Princeton Application Repository for Shared-Memory Computers (PARSEC) [36]. The PARSEC suite contains several multi-threaded programs that explore different workloads on shared memory

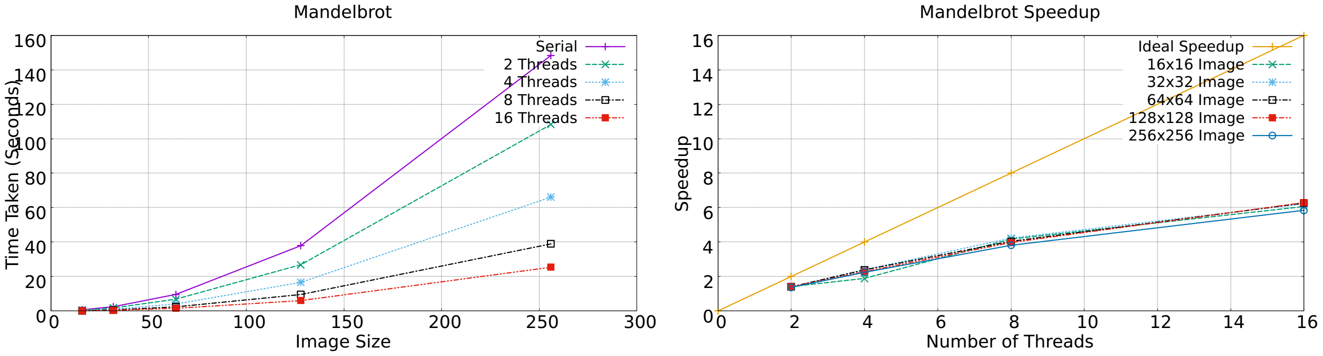


Fig. 4. Compilation Time and Speedup for the Mandelbrot Benchmark

architectures. *Blackscholes* processes financial data using a partial differential equation.

The *N-Body Problem* and *Mandelbrot* benchmarks are based on solutions provided to The Computer Language Benchmarks Game [37]. These are originally micro-benchmarks with the goal of testing performance of different programming languages as opposed to directly testing parallel performance.

All benchmarks are parallelised using static partitioning. The partition sizes are selected by the compiler based on the number of threads made available and the size of a loop's range. The parallelised loop regions are indicated by an invocation of a *std::for_each*, which has been adapted to support *constexpr* parallelisation. The *std::for_each* invokes a function on each piece of data, in this case each thread will be given a set of data to invoke the function on individually.

The performance data gathered for each benchmark is displayed using two types of graph. The first is a line graph comparing serial and parallelised execution times (during compilation). Each of the plots in these graphs is calculated by averaging six separate runs of each benchmark and data size configuration. The second type of graph displays *parallel speedup*; comparing times when using different numbers of threads against the ideal speedup on different problem sizes for the benchmark. The ideal speedup is a one to one match for the number of threads used. A cumulative speedup graph is also provided, including speedups for all of the benchmarks.

A. Timing Compile Time Performance

Performance of the *constexpr* paralleliser is measured by comparing the speed of the parallelised constant expression evaluator against the original serial implementation on each of the benchmarks. The benchmarks are tested with different numbers of threads using an Intel Core i9-12900K CPU, containing eight performance cores, and with support for 16 hardware threads via hyper-threading. The benchmarks were run under the WSL2 virtual machine on Windows 11; and executed using two, four, eight and sixteen threads.

Time is measured from the beginning of a parallel region to its end, rather than timing the length of the entire program; this is to allow us to focus on the regions of interest. The same location is measured for the serial execution. Three timing intrinsics were implemented to allow measurement of the phase within the ClangOz compiler.

It is worth noting that the same compiler is used for both the parallel and serial tests, as the intrinsics are needed for timing alongside a modified standard library implementation. This has a minor impact on the measurements for both implementations as the time measurement functionality requires checking the intrinsics' names, every time a function is considered for parallel execution. To determine if the parallel code path should be executed within the compiler, the verification phase discussed in Section III-A must be processed, which also adds an extra performance cost when evaluating the original serial implementation.

B. Mandelbrot

The Mandelbrot benchmark consists of three main areas of computation that are executed in parallel using *std::for_each*. Two initialisation steps populate a 2D array of complex values (a class containing two 64-bit floats) per pixel. Subsequently, the main Mandelbrot computation uses the naïve escape time algorithm. This algorithm loops over each complex value and performs a repeating calculation until an escape condition is met (limited to a maximum of 128 iterations). The final value generated after the escape condition has been met is the colour of the pixel which is assigned to an array of integers representing the final image.

The graphs in Fig. 4 show that the increase in data size gradually progresses towards higher polynomial growth as the number of threads diminish. With more threads, the increase in data size has less impact on compilation time. Breaking the computation into two separate parallel regions, requiring two thread group launches, has had minimal impact on this benchmark. The speedup graph shows that across all image sizes the performance improvement is similar.

C. Blackscholes

Blackscholes has two areas of computation which are parallelised, requiring two separate calls to *std::for_each*. The first is the main computation which computes the Black-Scholes equation; the second verifies the results from the

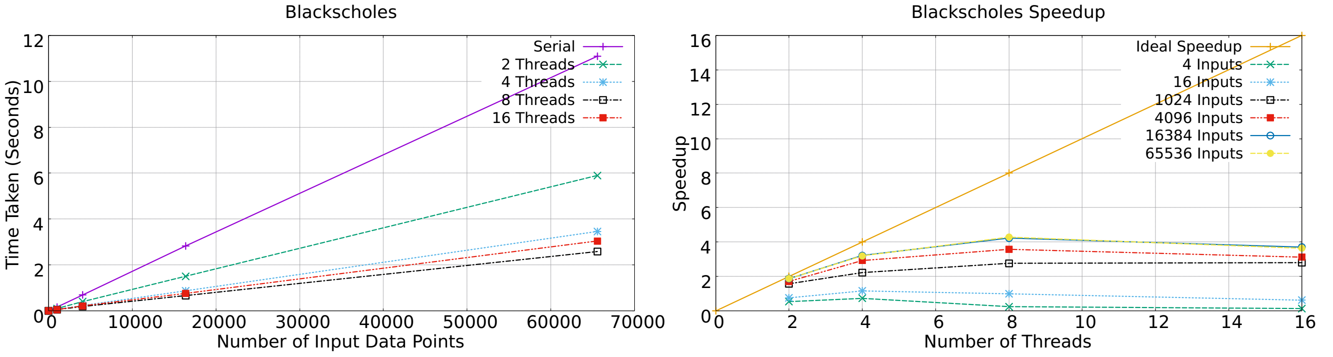


Fig. 5. Compilation Time and Speedup for the Blackscholes Benchmark

first computation. The main data that requires cloning in this benchmark is a 1D array containing a structure for each input that owns 9 literal values.

The Blackscholes data in Fig. 5 shows promising performance increases when utilising both two and four threads. The speedup graph indicates that the highest speedup occurs when larger input sets are used. Smaller data sets still yield an increase but plateau or fall slightly at four threads. The poor performance of the 4 and 16 sized data sets can be accounted for by the increased cost of preparing and launching threads outweighing the work required to process these tiny data sets.

D. N-Body

The N-Body benchmark has two parallel regions: the advance of the particle system; and the position and velocity update. This means with more iterations of the system more launches of threads occur. To allow a range of body numbers, the number of iterations is restricted to 32; while still avoiding compilation limits. The main data cloned within the benchmark are the bodies; structures containing seven 64-bit floats stored contiguously within an array. The number of bodies is the parameter that is varied within this benchmark.

The graphs in Fig. 6 show that increasing the number of threads again outperforms the serial implementation, however using two threads is the closest to the ideal speedup achieved within this benchmark, with larger body numbers also aligned with better performance. This is likely due to the cost of cloning having an adverse impact on smaller workloads. As with the Blackscholes benchmark, there is no improvement in speedup as the number of cores is increased from eight to sixteen (which utilises hyper-threading).

E. Benchmark Comparison

The speedup graph in Fig. 7 compares the most time consuming variations of each benchmark against each other and indicates which benchmarks achieved the best performance after parallelisation. The N-Body benchmark is the best performer reaching the closest to the ideal speedup across all of the benchmarks. This is due to the trivial size and simplicity of the data that requires cloning. Until eight threads, the Mandelbrot benchmark is the furthest from the ideal speedup; possibly according to the number of parallel regions, compared to the other benchmarks. Two of the parallel regions execute relatively small computations, to fill small arrays of data, while cloning a significant amount within the context of each parallel function call. The possibility of multiple thread launches causing cloning to have a negative impact is highlighted by Blackscholes performing better than Mandelbrot despite having a similar amount of data to clone per parallel region, but less parallel regions overall. With sixteen threads, hyperthreading should allow two threads to run efficiently on each of the eight performance cores, but only the Mandelbrot benchmark shows a reduction in execution time at the largest thread count.

The results indicate that the parallelisation of *for* loops during compile time evaluation can lead to notable speedups when a large portion of the program conforms reasonably to a loop. However, the cost of cloning and partitioning data comes with significant costs. It is plausible that performance could be increased by removing the need for cloning in cases where it should not be required. For example, containers like *std::array* which have no conflicting data accesses across threads should not require cloning. This could yield a performance increase in most of these benchmarks as the main input data is generally contained in an array. Data that is not required for the execution of the parallel *constexpr* function could also be elided from the cloning process, which could have a large impact on benchmarks that contain a significant amount of data unrelated to the parallel invocation. A simple form of workload balancing may also yield reasonable results in certain circumstances where the majority of the work is not perfectly divisible by the number of threads in use. Whilst this is not seen in the performance analysis within the paper, there is likely an opportunity for improvement over the current implementation that could allow a performance increase.

V. Conclusion

As the C++ standard has evolved, additional compile time language features have been added, extending the reach of compile time metaprogramming. As C++'s compile time repertoire and its use has expanded, the problem of increasing

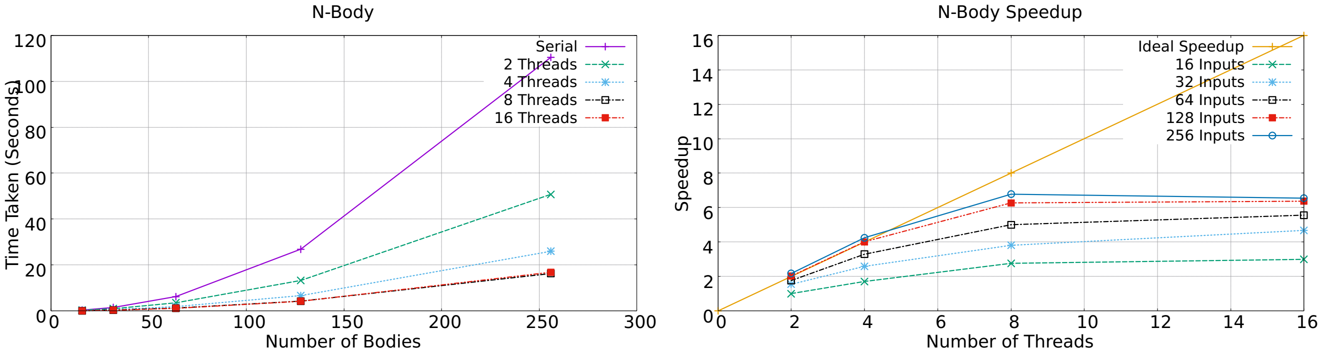


Fig. 6. Compilation Time and Speedup for the N-Body Benchmark

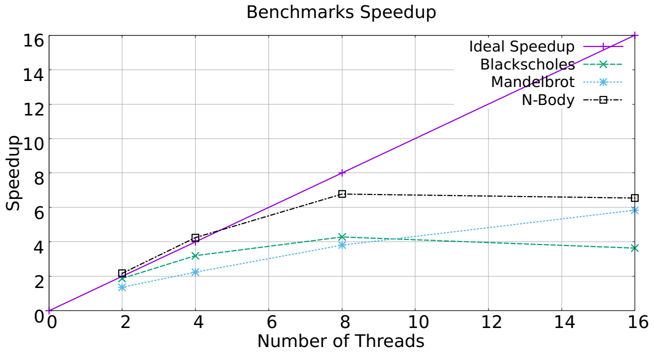


Fig. 7. Speedup Graph Comparison of all Benchmarks Compilation Time

compilation times becomes prominent, leading to adverse effects on programmer work flow. This opens up the question of how to alleviate the issue. In the project introduced here, the option of acceleration through multi-threaded data-parallelism, within the compiler is investigated. ClangOz, an extended Clang compiler for C++ is introduced that can parallelise *for* loops at compile time; including with reduction/accumulation. Therein, intrinsic functions allow users to explicitly relay information to the compiler about the loop being parallelised. This firstly allows users the flexibility to implement their own low-level compile time parallel algorithms, while understanding the intrinsics' semantics. Together, the compiler and intrinsics create a framework for accelerating constant expression evaluation.

This low-level functionality has then been utilised to provide a *high-level API*, which builds on recent C++ standard library support for parallelism to implement 30 *constexpr* parallel function templates. These functions are based on existing function template signatures within the C++ standard library, and differ only by a single argument; the policy parameter, providing access to parallelism through C++ overloading. Three compile time benchmarks were implemented that utilise a *constexpr* parallel *std::for_each* from this extended library. Through testing of these benchmarks it was shown that the ClangOz framework can have large performance benefits; with up to 95% parallel efficiency on one benchmark, and above 50% on average. These benchmarks also show that the complexity of the framework can be hidden within a library; removing users from the onus of understanding low-level intrinsics, while maintaining high performance. Benchmark results nevertheless indicate that there is still room for improvement.

The current parallelisation process has some areas that could be addressed to improve performance. One issue stems from the fact that the data copying process required when forking and joining threads can be expensive. This leads to significant startup costs, meaning that multiple sequential parallel regions for trivial amounts of computation are slower than if done sequentially. Large data dependencies can also have an impact on how much of a performance increase you will get from the parallelisation process. Optimising or removing the need for the cloning process would likely improve performance. Another issue is the lack of work load balancing in the implementation, which necessitates that users must choose their thread partitioning carefully. When data is indivisible by the thread count this can have a performance penalty as one thread will keep the others waiting as it deals with the excess data. A solution would be implementing a work load balancing algorithm within the compiler. These adjustments to the compiler could improve the parallelisation algorithm's overall performance.

Acknowledgement

The authors wish to thank the Royal Society of Edinburgh for their support through the Saltire International Collaboration Award (Grant Number 1981).

References

[1] L. V. Todd, "C++ templates are turing complete," *Available at citeseer.ist.psu.edu/581150.html*, 2003.

[2] D. R. Gabriel, B. Stroustrup, and J. Maurer, "Generalized constant expressions–revision 5," ISO SC22 WG21 TR, Tech. Rep., 2007.

[3] (2023) ClangOz. [Online]. Available: https://github.com/agozillon/ClangOz

[4] C. Lattner, "LLVM and Clang: Next generation compiler technology," in *The BSD conference*, vol. 5, 2008.

[5] H. Alblas, R. op den Akker, P. O. Luttighuis, and K. Sikkel, "A bibliography on parallel parsing," *ACM Sigplan Notices*, vol. 29, no. 1, pp. 54–65, 1994.

[6] H. P. Katseff, "Using data partitioning to implement a parallel assembler," in *Proceedings of the ACM/SIGPLAN conference on Parallel programming: experience with applications, languages and systems*, 1988, pp. 66–76.

[7] V. Seshadri, S. Weber, D. Wortman, C. Yu, and I. Small, "Semantic analysis in a concurrent compiler," in *Proceedings of the ACM SIGPLAN 1988 conference on Programming language design and implementation*, 1988, pp. 233–240.

[8] G. U. Srikanth, "Parallel lexical analyzer on the cell processor," in *2010 Fourth International Conference on Secure Software Integration and Reliability Improvement Companion*. IEEE, 2010, pp. 28–29.

[9] T. Gross, A. Sobel, and M. Zolg, "Parallel compilation for a parallel machine," in *Proceedings of the ACM SIGPLAN 1989 Conference on Programming language design and implementation*, 1989, pp. 91–100.

[10] I. J. S. 22, *ISO/IEC 14882:2020 Programming languages — C++*, 2020.

[11] P. Dimov, L. Dionne, N. Ranns, R. Smith, and D. Vandevoorde, "More constexpr containers," 2019. [Online]. Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p0784r7.html

[12] A. Sutton, "C++ extensions for concepts," 2017. [Online]. Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2017/p0734r0.pdf

[13] H. Sutter, "Metaclasses: Generative c++," 2018. [Online]. Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/p0707r3.pdf

[14] M. Chochlík, A. Naumann, and D. Sankel, "Static reflection," 2017. [Online]. Available: http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/p0707r3.pdf

[15] H. Dusíková. (2016) Compile time regular expressions. [Online]. Available: https://github.com/hanickadot/compile-time-regular-expressions

[16] M. Sánchez. (2018) tinyrefl. [Online]. Available: https://github.com/Manu343726/tinyrefl

[17] N. J. Bouman, "Multiprecision arithmetic for cryptology in c++ - compile-time computations and beating the performance of hand-optimized assembly at run-time," arXiv:1804.07236, 2018, https://arxiv.org/abs/1804.07236.

[18] B. Fahller. (2017) lift. [Online]. Available: https://github.com/rollbear/lift

[19] G. Steele, *Common LISP: the language*. Elsevier, 1990.

[20] A. Alexandrescu, *The D programming language*. Addison-Wesley Professional, 2010.

[21] N. D. Matsakis and F. S. Klock, "The rust language," *ACM SIGAda Ada Letters*, vol. 34, no. 3, pp. 103–104, 2014.

[22] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[23] C. McCord, *Metaprogramming Elixir*, 1st ed. Pragmatic Bookshelf, 2015.

[24] S. Baxter. (2020) Circle: The c++ automation language. [Online]. Available: https://benchmarksgame-team.pages.debian.net/benchmarksgame/

[25] N. D. Jones, "An introduction to partial evaluation," *ACM Computing Surveys (CSUR)*, vol. 28, no. 3, pp. 480–503, 1996.

[26] T. L. Veldhuizen, "C++ templates as partial evaluation," *arXiv preprint cs/9810010*, 1998.

[27] A. Tyurin, D. Berezun, and S. Grigorev, "Optimizing gpu programs by partial evaluation," in *Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2020, pp. 431–432.

[28] R. Leißa, K. Boesche, S. Hack, A. Pérard-Gayot, R. Membarth, P. Slusallek, A. Müller, and B. Schmidt, "Anydsl: A partial evaluation framework for programming high-performance libraries," *Proceedings of the ACM on Programming Languages*, vol. 2, no. OOPSLA, pp. 1–30, 2018.

[29] T. Würthinger, C. Wimmer, C. Humer, A. Wöß, L. Stadler, C. Seaton, G. Duboscq, D. Simon, and M. Grimmer, "Practical partial evaluation for high-performance dynamic language runtimes," in *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2017, pp. 662–676.

[30] C. Consel and O. Danvy, "Partial evaluation in parallel," *Lisp and Symbolic Computation*, vol. 5, no. 4, pp. 327–342, 1993.

[31] M. Sperber, P. Thiemann, and H. Klaeren, "Distributed partial evaluation," in *Proceedings of the second international symposium on Parallel symbolic computation*, 1997, pp. 80–87.

[32] A. Bouter, T. Alderliesten, A. Bel, C. Witteveen, and P. A. Bosman, "Large-scale parallelization of partial evaluations in evolutionary algorithms for real-world problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 1199–1206.

[33] K. Kennedy and J. R. Allen, *Optimizing compilers for modern architectures: a dependence-based approach*. Morgan Kaufmann Publishers Inc., 2001.

[34] I. J. S. 22, "Technical specification for c++ extensions for parallelism," Tech. Rep., 2018.

[35] L. Dagum and R. Menon, "Openmp: an industry standard api for shared-memory programming," *IEEE computational science and engineering*, vol. 5, no. 1, pp. 46–55, 1998.

[36] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, 2008, pp. 72–81.

[37] D. Bagley. (2001, Apr.) The computer language benchmarks game. [Online]. Available: https://benchmarksgame-team.pages.debian.net/benchmarksgame/

DOI: 10.15439/2023F4139
ISSN 2300-5963 ACSIS, Vol. 35

Explainability in RIONA Algorithm Combining Rule Induction and Instance-Based Learning

Grzegorz Góra
University of Warsaw
Banacha 2, 02-097 Warszawa
Poland
Email: ggora@mimuw.edu.pl

Andrzej Skowron
Systems Research Institute PAS
Newelska 6, 01-447 Warszawa
Poland
Email: skowron@mimuw.edu.pl

Arkadiusz Wojna
DeepSeas
12121 Scripps Summit Court
San Diego, CA 92131, USA
Email: wojna@mimuw.edu.pl

***Abstract*—The article concerns the well-known RIONA algorithm. We focus on the explainability property of this algorithm. The theoretical results, formulated and proved in the paper, show the relationships of the RIONA classifiers to both instance- and rule-based classifiers. In particular, we show the equivalence (relative to the classification) of the RIONA algorithm with the rule-based algorithm generating all consistent and maximally general rules from the neighbourhood of the test case.**

I. Introduction

IN THE paper, we focus on the learning algorithm for supervised learning [1], [2], [3]. Specifically, we focus on the well-known RIONA algorithm [4], [5], [6]. This algorithm combines two widely-used empirical approaches: rule induction and instance-based learning [7], [8], [9], [10]. Both these approaches use reasoning schemes comprehensible to a human. It is essential since, Explainable AI [11], [12], [13] is becoming more and more useful in real-life applications. The classifying system should provide for the given test object not only decision but also its user-understandable explanation. In the paper, we present theoretical results of considered algorithm that allow to meet these requirement.

A few concepts form the framework of the RIONA algorithm. First, instead of inducing an excessive number of decision rules in advance to use them during testing, it induces decision rules relevant only for the test example. This is a strategy of so-called lazy learning [14]. Second, only rules from the neighbourhood of the given test example are considered. Third, it automatically groups numerical and symbolic values of attributes by using more general than commonly-used conditions. Fourth, RIONA computes optimal size for the neighbourhoods of objects.

The properties of RIONA algorithm are worth studying as it was reported in the literature as one of the most accurate classification methods in many experimental comparisons done by various researchers (the most commonly used RIONA implementation is a classifier in the WEKA platform named RseslibKnn [15]), to name a few: Facebook content recognition [16, Chapter 1] (RIONA was the best one of 21 tested algorithms), environmental sound recognition [17] (best of 9 algorithms), metabolic pathway prediction of plant enzymes [18] (2nd of 47 algorithms), acoustic-based environment monitoring [19] (2nd of 8 algorithms), context awareness of a service robot [20] (2nd of 8 algorithms), student performance prediction [21] (5th of 47 algorithms).

The novelty of the paper is in theoretical results creating the basis for explainability of classifications returned by classifiers. The results concern the relationships of the classifiers generated by the RIONA algorithm to classifiers obtained by applying instance-based as well as rule-based approaches. In particular, it occurs that RIONA classifiers are equivalent (relative to the classification property) to classifiers produced by the rule-based algorithm based on all consistent and maximally general rules generated from the neighbourhood of the test case. Such rules are easily interpretable by humans.

The paper relates to the PhD thesis [6].

II. Basic notions

$|X|$ denotes cardinality (size) of the set X set. If $\mathbf{A}$ and $\mathbf{B}$ are algorithms then the equality $\mathbf{A}(v) = \mathbf{B}(w)$ means that the values returned by $\mathbf{A}$ on input v and by $\mathbf{B}$ on input w are the same.

By $\mathbf{X}$ we denote a set of objects, called the domain of learning and by $Atr = A \cup \{d\}$ a finite set of attributes atr, where $atr : \mathbf{X} \longrightarrow V_{atr}$. V_{atr} is called the value set of atr. Attributes from A are called *conditional attributes* and the attribute $d \notin A$ is called the *decision*. V_d is called the decision set. We assume for simplicity of notation that $V_d = \{1, \ldots, n_d\}$. Any object $x \in \mathbf{X}$ is represented relative to its signature $Inf_{A(x)}$, i.e. a set of pairs $(a, a(x))$ for each $a \in A$. We use the symbols A_{sym} and A_{num} for denoting the sets of symbolic and numerical attributes, respectively. If $a \in A_{sym}$ then V_a is a finite set. If $a \in A_{num}$ then without loss of generality, we assume that V_a is equal to an interval (l_a, u_a), where $l_a, u_a \in \mathbb{R}$ (possibly not all of the values from the interval are used).

A *decision system* is a triplet $(\mathbf{X}, A, d)$ if $\mathbf{X}$ is the set of objects, A is a set of attributes, and d is a decision function. A *pseudometric decision system* is a 4-tuple $(\mathbf{X}, A, d, \{\varrho_a\}_{a \in A})$ if $(\mathbf{X}, A, d)$ is a decision system and for any attribute $a \in A$, ϱ_a is a pseudometric on the respective value set V_a, i.e. for any $a \in A$, (V_a, ϱ_a) is a pseudometric space [22].

In the sequel by $\mathbb{D}$ is denoted a given (pseudometric) decision system of the above form.

In the paper we study a combination of two methods (learning algorithms) inducing a *classifier* from a given subdecision system $(trnSet, A, d)$, where $trnSet \subseteq \mathbf{X}$ and attributes from $A \cup \{d\}$ are restricted to $trnSet$, which for any $x \in \mathbf{X}$ computes the decision $\hat{d}(x)$ in such a way that $\hat{d}$ is *close* to d [2]. It should be noted that in practical experiments, the normalized Euclidean metric was used for numerical attributes, and the SVDM [23] pseudometric for symbolic attributes. If $a \in A_{num}$, the normalisation is obtained by using $a^{\max}$ and $a^{\min}$, which denote the maximal and the minimal values for an attribute a among training examples $trnSet$.

A. Rule-based methods

The induction of rule sets is one of the fundamental Machine Learning (ML) techniques (see e.g. [7]). Its significance stems from the fact that a human may easily comprehend knowledge representation in the form of rules. Decision rules specify the appropriate course of action in a given circumstance. Decision rules frequently have the form 'if φ then ψ', where φ denotes the premise of the rule and ψ denotes its consequence; ψ is a formula defined by the decision attribute d.

Decision rules are generated using rule induction algorithms from a training set. The premises of the rules are represented by a conjunction of *elementary conditions* and the consequences are describing the particular decision. Each elementary condition describes a collection of the attribute's values. Roughly speaking, it has the form $a \in V$, where $V \subseteq V_a$, and a is an attribute. We first determine how such sets V of values can be expressed in a formal language. Next, we define the semantics (meaning) of specified expressions from this language in the powerset of the attribute value set V_a. For simplicity of notation we do not distinguish between symbols denoting values (or intervals) and values (or intervals) themselves.

Definition II.1. *Let $\mathbb{D}$ be a (pseudometric) decision system. For symbolic attributes $a \in A_{sym}$, the* description of any elementary set *has one of the following forms:*

$$\emptyset \tag{1}$$

$$\{v\}, \text{ where } v \in V_a, \tag{2}$$

$$V_a, \tag{3}$$

$$B(c,r), \text{ when } \mathbb{D} \text{ is pseudometric decision system and where } c \in V_a,\ r \in \mathbb{R}, r \geq 0. \tag{4}$$

The description of elementary set *for the decision attribute d is of the form $\{v\}$, for $v \in V_d$.*

For numerical attributes $a \in A_{num}$, the description of elementary set *has one of the following forms:*

$$\emptyset \tag{5}$$

$$[b,e], (b,e], [b,e), (b,e), \text{ where } b,\ e \in \mathbb{R} \text{ are such that the corresponding interval between points } b \text{ and } e \text{ is included in } V_a. \tag{6}$$

The semantics $||des||_{\mathbb{D}} \subseteq V_a$ of any description des of the elementary set for attribute $a \in A \cup \{d\}$ is defined as follows:

$$\begin{aligned} ||\emptyset||_{\mathbb{D}} = \emptyset, \quad ||\{v\}||_{\mathbb{D}} = \{v\}, \quad ||V_a||_{\mathbb{D}} = V_a, \\ ||[b,e]||_{\mathbb{D}} = [b,e], ||(b,e]||_{\mathbb{D}} = (b,e], \\ ||[b,e)||_{\mathbb{D}} = [b,e), ||(b,e)||_{\mathbb{D}} = (b,e), \\ ||B(c,r)||_{\mathbb{D}} = \{w \in V_a : w \in B(c,r)\} \\ = \{w \in V_a : \varrho_a(c,w) \leq r\} \text{ (called the ball set).} \end{aligned}$$

Now, the elementary conditions expressed in a language and their semantics can be defined.

Definition II.2. *Let $\mathbb{D}$ be a (pseudometric) decision system.*

Any expression $a \in V$, where $a \in A$ and V is a description of elementary set for attribute a is called an elementary condition. *The semantics of $a \in V$ is defined by $[[a \in V]]_{\mathbb{D}} = \{x \in \mathbf{X} : a(x) \in ||V||_{\mathbb{D}}\}$.*

$[[a \in V]]_{\mathbb{D}}$ may be restricted to subsets of $\mathbf{X}$, e.g. $[[a \in V]]_{trnSet} = [[a \in V]]_{\mathbb{D}} \cap trnSet$.

An example (case) x satisfies the elementary condition $a \in V$(or, in short, $(a \in V)(x)$ is satisfied) if $x \in [[a \in V]]_{\mathbb{D}}$.

The set $||V||_{\mathbb{D}} \subseteq V_a$ for a given elementary condition $a \in V$ is equal to

- $\{v\}$ for some $v \in V_d$ for the decision attribute d (see set description 2 and its semantics),
- a proper interval for the numerical attribute (see set description 6 and its semantics), and
- $\{v\}$ for some $v \in V_a$, V_a or a ball set for the symbolic attribute a (see set descriptions 2, 3, 4, respectively and their semantics).

A given object x satisfies the elementary condition $a \in V$ if the value of a on x, i.e. $a(x)$ is in the set defined in $\mathbb{D}$ by V, i.e. $a(x) \in ||V||_{\mathbb{D}}$. Instead of $a \in \{v\}$ we write $a = v$ and instead of *trivial* elementary condition $a \in V_a$ (always true, i.e. the set of objects satisfying this condition is equal to the set $\mathbf{X}$) we write $a = *$.

Now, we introduce concepts related to syntax and semantics of decision rules.

Definition II.3. *Let $\mathbb{D}$ be a (pseudometric) decision system. A* decision rule *is an expression of the form*

$$if\ t_1 \wedge t_2 \wedge \ldots \wedge t_m\ then\ d = v,$$

where $m = |A|$, t_i is an elementary condition for an attribute a_i for $i = 1, 2..., m$, and $v \in V_d$.

The semantics of the premise $t_1 \wedge t_2 \wedge \ldots \wedge t_m$ of the rule r (in $\mathbb{D}$) is defined by

$$[[t_1 \wedge t_2 \wedge \ldots \wedge t_m]]_{\mathbb{D}} = [[t_1]]_{\mathbb{D}} \cap [[t_2]]_{\mathbb{D}} \ldots [[t_m]]_{\mathbb{D}}$$

If $x \in [[t_1 \wedge t_2 \wedge \ldots \wedge t_m]]_{\mathbb{D}}$ then we say that

- *the premise of the rule r is satisfied by example x (or x satisfies this premise),*
- *example x* matches *the rule r,*
- *r* covers *x.*

One can treat a single rule as a classifier assigning examples covered by that rule to the decision class from the rule's consequence. Ideally, we could search for rules $if\ \varphi\ then\ d = v$ such that $[[\varphi]]_{\mathbb{D}} \subseteq [[d = v]]_{\mathbb{D}}$. However, the restriction of this inclusion to $trnSet$, i.e. $[[\varphi]]_{trnSet} \subseteq [[d = v]]_{trnSet}$ is available only. Rules satisfying this last condition (for $trnSet$) are induced from $trnSet$ and an hypothesis on extension of the truth of this inclusion on $\mathbf{X}$ is made. Moreover, rules covering as many as possible examples are generated.

In description of decision rules, trivial conditions are usually omitted [1]. The typical conditions are equations $a = v$ in case of symbolic attributes and inclusions into intervals in case of numerical attributes, e.g.:

$$if\ a_2 = 3 \wedge a_4 \in [2, 5] \wedge a_7 = 3\ then\ d = 2.$$

In this paper we use for symbolic attributes more general conditions, i.e. $a \in V$ (see Definition II.2), making it possible to extend singleton sets to the ball sets. If the data set under consideration has some numerical attributes, then by applying discretisation the relevant intervals can be constructed. By applying discretisation to a given decision system a new one is obtained with new attributes being characteristic functions of induced intervals including objects from $trnSet$ labeled by the same decision (see e.g. [24]).

By $t_i(r)$, where r is a given decision rule we denote the i-th condition t_i from Definition II.3. We write $t_a(r)$ instead of $t_i(r)$ if the condition t_i from Definition II.3 concerns the attribute a.

We define three kinds of decision rules by distinguishing elementary conditions (used in Definition II.3) occurring in them. In consequence, we obtain three sets of decision rules.

Definition II.4. *Let $\mathbb{D}$ be a decision system.*

The set $SimRules$ of simple rules *is the set of all rules from Definition II.3 with the elementary conditions of the form $a = v$ for $v \in V_a$ and $a = *$ only.*

The set $CombRules$ of combined rules *is the set of all rules from Definition II.3 with elementary conditions for symbolic attributes of the form as in $SimRules$ and for numerical attributes of the form $a \in I$ (where I is a proper interval description) only.*

Definition II.5. *Let $\mathbb{D}$ be a pseudometric decision system such that for any symbolic attribute $a \in A_{sym}$ there is a distinguished specific value $c_a \in V_a$.*

The set $GenRules\left(\{(\varrho_a, c_a)\}_{a \in A_{sym}}\right)$ (or simply $GenRules$ whenever pairs (ϱ_a, c_a) are clear from the context or irrelevant due to generality) of general rules *is the set of all rules from Definition II.3 where set descriptions in elementary conditions in the premises of the rules are*

(i) *as in the definition of $CombRules$ for numerical attributes and*
(ii) *of specific form of 4, i.e. $B(c, r)$, where $c = c_a$, $r = \varrho_a(c_a, v)$, $v \in V_a$ for symbolic attributes $a \in A_{sym}$.*

More details on *general rules* presented in Section III-A will help the reader to better understand our definition.

Definition II.6. *A rule if ϕ then $d = v$ is* consistent *with a set of objects $X \subseteq \mathbf{X}$ (or* consistent *if X is clear from the context) if $d(x) = v$ for any object $x \in X$ matching this rule.*

If the rule if ϕ then $d = v$ is not consistent than it is called inconsistent.

Typically, in the above definition $trnSet$ is used as X. Any rule if ϕ then $d = v$ consistent with $trnSet$ classifies correctly all the training examples covered by that rule i.e. $[[\phi]]_{trnSet} \subseteq [[d = v]]_{trnSet}$.

For further considerations the concept of maximality of rule will be useful.

Definition II.7. *Let $\mathbb{D}$ be a (pseudometric) decision system and let $a \in A$, and let V_1, V_2 be elementary conditions for a. The condition $a \in V_2$ is* more general than *(or is* implied by*) $a \in V_1$, in symbols $a \in V_1 \Rightarrow a \in V_2$ if $||V_1||_{\mathbb{D}} \subseteq ||V_2||_{\mathbb{D}}$.*

For any two rules r_1, r_2 (over $\mathbb{D}$) with the same consequence $d = v$, we say that a rule r_2 is more general than *(or is* implied by*), in symbols $r_1 \Rightarrow r_2$ if $t_i(r_1) \Rightarrow t_i(r_2)$ for $i = 1, \ldots, m$ and $m = |A|$.*

A consistent rule r with a training set $trnSet$ is maximally general *(relative to $trnSet$ and a given set of rules $Rules$) if there is no rule in $Rules$ more general than r which is different from r and consistent with $trnSet$.*

Definition II.8. *Let $\mathbb{D}$ be a (pseudometric) decision system and let $Rules$ be a given set of admissible rules. The* set of maximally general rules *(relative to $trnSet$) $MaxRules(Rules, trnSet)$ is equal to the set of all maximally general rules $r \in Rules$ consistent with $trnSet$.*

We use the following sets of rules: $SimRules$, $CombRules$, $GenRules$ from Definitions II.4 and II.5 as $Rules$ (see Definition II.8). We write $MaxRules$ instead of $MaxRules(Rules, trnSet)$, if from the context $Rules$ and $trnSet$ are known. Hence, $MaxRules$ may denote: $MaxRules(SimRules, trnSet)$, $MaxRules(CombRules,trnSet)$, or $MaxRules(GenRules,trnSet)$.

Computing from $trnSet$ the set of all consistent and maximally general matching at least one case from $trnSet$ is important for some learning algorithms.

In the case of $MaxRules(SimRules, trnSet)$, a consistent rule is maximally general relative to $trnSet$ if it becomes inconsistent (relative to $trnSet$) after substitution of the trivial condition instead of a non-trivial one. Hence, consistent rules from $MaxRules(SimRules, trnSet)$ can be characterised as the rules with minimal length (measured by the number of non-trivial conditions in predecessors). Hence, in the considered case, the problem is to generate the complete set of consistent and minimal decision rules (see e.g. [25]). One can observe that searching for the set of minimal rules can be motivated by the minimum description length principle (MDL) (see e.g. [26]). However, the computational time complexity of

[1] In fact, in the description of rules only non-trivial conditions are used. We use trivial conditions only to make the notation simpler.

algorithms generating $MaxRules(SimRules, trnSet)$ is not feasible when the number of training objects or attributes are large. In fact, the size of the $MaxRules$ set can be exponential relative to $|trnSet|$ (see e.g. [27]). Hence, efficient heuristics are often used to overcome this drawback, especially when not necessarily complete sets of minimal rules are required (see e.g. [28]). There are also other approaches inducing a set of rules fully covering cases from $trnSet$ (see e.g. [29]). Here, we focus on the complete $MaxRules$ set.

In the case of $MaxRules(CombRules, trnSet)$, additionally we deal with numerical attributes. From $trnSet$ maximally general intervals of reals are induced. Searching for maximally general rules for numerical attributes is closely related to the problem of discretisation. A partition of reals in discretisation is consistent if each interval covers only objects with the same decision (see e.g. [24], [27]).

The discretisation problem is a complex task. For example, searching for a consistent partition with the minimal number of cuts is NP-hard (see e.g. [24]). In Subsection III-A we show how to overcome this drawback using lazy learning and focusing on a local part of $\mathbf{X}$ instead of on the whole universe. This is illustrated by the lazy rule induction algorithm Algorithm 3 or Algorithm 4.

In the case of $MaxRules(GenRules, trnSet)$, the additional search is performed for the relevant grouping of values for symbolic attributes into a partition of value sets of symbolic attributes. One can define a partition over an attribute a as any function $P_a : V_a \to \{1, \ldots, m_a\}$. It should be noted that the problem of searching for a consistent family of partitions with the minimal $\sum_{a \in A} |P_a(V_a)|$ is NP-hard (see e.g. [30]). We show in the paper how to overcome this drawback by limiting the number of possible groupings of values of any attribute (from 2^n to n^2, where n is the number of values for an attribute) and by using the lazy rule induction (see Section III-A).

The induced sets of rules from $trnSet$ are used to classify objects. First, for any test object tst there are selected all rules from the set matching this object. Next, the set of matched rules is checked. If all rules matched by tst have the same decision then this decision is assigned to tst else it should be resolved conflict between matching rules voting for different decisions (see e.g. [31]). Typically, it is selected the decision with the highest value of a selected measure used for conflict resolution. We use the commonly used measure for conflict resolution.

Definition II.9. *Let us assume that $\mathbb{D}$ – (pseudometric) decision system, $trnSet$ – training set $trnSet$, tst – test example (case), and $MaxRules$ – set of maximally general rules are given. By $supportSet(r) \subseteq trnSet$, where $r \in MaxRules$ we denote the set of all objects from $trnSet$ matching r, and by $MatchR(tst, v) \subseteq MaxRules$ (where $v \in \{1, \ldots, n_d\}$, i.e. v is a decision of d on some object from $trnSet$) the set of rules from $MaxRules$ with decision v matching the test object tst. Now, we define*

$$Strength(tst, v) = \left| \bigcup_{r \in MatchR(tst,v)} supportSet(r) \right|. \quad (7)$$

From definition it follows that by computing $Strength(tst, v)$ it is counted the number of objects from $trnSet$ covered by some maximally general rule from $MaxRules$ (i) with the decision v and (ii) covering the test example tst.

On the basis of $MaxRules$ and the defined conflict resolution strategy using $Strength$ we define the classifier assigning to a given test object tst the most frequent decision of such training examples from $trnSet$ which are covered by matched by tst rules from $MaxRules$, i.e.:

$$decision_{MaxRules}(tst) = \arg\max_{v \in V_d} Strength(tst, v). \quad (8)$$

As it was observed, the drawback of the presented approach comes from the high computational complexity of $MaxRules$ generation.

B. Lazy rule learning for symbolic attributes

The *lazy learning* (or *memory based learning*) algorithms do not require construction of sets of decision rules before classification of new objects.

kNN is the well known example of such algorithms (see Algorithm 1). For these algorithms, first for any test object tst it is defined its neighbourhood $N(tst, trnSet, k, \varrho) \subseteq trnSet$ (N, for short) with k the most similar to tst (relative to a given distance function ϱ) training examples (where k is a parameter). If more than one example has the same distance from tst as the one already added to the N under construction then all of them are added to $N(tst, trnSet, k, \varrho)$. Then the set $N(tst, trnSet, k, \varrho)$ may contain more than k examples[2].

An interesting example of lazy rule-based algorithm for $SimRules$ is presented in [32]. For a new tst it generates only the relevant for it decision rules and next tst is classified as before on the basis of such rules. The value of Eqn. 7 is computed for any tst object without computing the whole set $MaxRule$.

For given two objects tst, trn, we first define *simple local decision rule*, in symbols $s\text{-}rule(tst, trn)$. The relationship of the set of such rules with $SimRules$ will be presented in the following proposition.

Definition II.10. *Let $\mathbb{D}$ be a decision system, $trn \in trnSet$ and let tst be a test object. A* simple local decision rule *(for short* s-rule*) $s\text{-}rule(tst, trn)$ is the decision rule of the form* if $\bigwedge_{a \in A} t_a$ then $d = d(trn)$, *where conditions t_a for each symbolic attribute a are as follows*

$$t_a = \begin{cases} a = a(trn) & \text{if } a(tst) = a(trn) \\ a = * & \text{if } a(tst) \neq a(trn). \end{cases}$$

[2] This assumption is used in RIONA. Hence, we also adopt it for the kNN algorithm in the paper.

Algorithm 1: kNN(tst, $trnSet$, k, ϱ)

```
Input: a test example $tst$, training set $trnSet$, positive
       integer $k$, pseudometric $\varrho$
Output: predicted decision for $tst$
begin
    $neighbourSet = N(tst, trnSet, k, \varrho)$
    foreach decision $v \in V_d$ do
        $supportSet(v) = \emptyset$
    end
    foreach $trn \in neighbourSet$ do
        $v = d(trn)$
        $supportSet(v) = supportSet(v) \cup \{trn\}$
    end
    return $\arg\max_{v \in V_d} |supportSet(v)|$
end
```

Algorithm 2: isCons(r, $verifySet$)

```
Input: a rule $r$ : if $\alpha$ then $d = v$,
       set of examples $verifySet$
Output: true if rule $r$ is consistent with $verifySet$, false
otherwise
for all $trn \in verifySet$ do
    if $d(trn) \neq v$ and $trn$ satisfies $\alpha$ then
        return $false$
    end if
end for
return $true$
```

Observe that both objects trn and tst are matching the rule $s\text{-}rule(tst, trn)$ which is maximally specific (the number of trivial conditions is minimal; or inversely, the number of non-trivial conditions is maximal). The following crucial relation between s-rule and maximally general consistent rules from $SimRules$ holds:

Theorem II.1. *[32][3] If $trn \in trnSet$ and tst is a test object than the rule $s\text{-}rule(tst, trn)$ is consistent with $trnSet$ if and only if $MaxRules(SimRules, trnSet)$ contains a rule covering both objects tst and trn.*

Hence, for any test object tst, decision $v \in V_d$ and $MaxRules(SimRules, trnSet)$ set the value $Strength(tst, v)$ from Eqn. 7 is equal to the number of $trn \in trnSet$ having decision $d(trn) = v$ and for which the rule $s\text{-}rule(tst, trn)$ is consistent with $trnSet$. The *simple lazy rule induction algorithm for symbolic attributes* (LAZY) presented in Algorithm 3 realises this idea.

$isCons(r, verifySet)$ in Algorithm 3 verifies if r is consistent with a $verifySet$. For a given object tst and any $trn \in trnSet$, the rule $s\text{-}rule(tst, trn)$ is constructed by

Algorithm 3: LAZY(tst, $trnSet$)

```
Input: test example $tst$, training set $trnSet$
for all decision $v \in V_d$ do
    $supportSet(v) = \emptyset$
end for
for all $trn \in trnSet$ do
    $v = d(trn)$
    if $isCons(s\text{-}rule(tst, trn), trnSet)$ then
        $supportSet(v) = supportSet(v) \cup \{trn\}$
    end if
end for
return $\arg\max_{v \in V_d} |supportSet(v)|$
```

Algorithm 3. Next, Algorithm 3 is testing the consistency of the rule $s\text{-}rule(tst, trn)$ with the set $trnSet \setminus \{trn\}$, i.e. if all the training examples matching the left-hand side of $s\text{-}rule(tst, trn)$ have identical decision with trn. If the result of testing is positive than trn is added to the support set with the relevant decision. Finally, Algorithm 3 predicts the decision with the support set of the highest cardinality. From Theorem II.1 we obtain:

Corollary II.2. *The following equality holds: $LAZY(tst, trnSet) = decision_{MaxRules}(tst)$, where $trnSet$ is a training set, tst is a test object, and $decision_{MaxRules}(tst)$ is the classifier from Eqn. 8 with $MaxRules = MaxRules(SimRules, trnSet)$.*

From the above considerations it follows that LAZY takes into account only these decision rules that can be involved in the classification of a given test object.

III. RIONA DESCRIPTION

A. Extension and generalisation of lazy rule learning

We introduce an extension and generalization of the LAZY algorithm (see Algorithm 3) that was discussed in Subsection II-B. This novel algorithm permits the use of numerical attributes as well as more general conditions for symbolic attributes.

In Subsections III-A1, III-A2 we present a generalisation of rules introduced before.

For a given test object tst, training object $trn \in trnSet$ and pseudomietric decision system $\mathbb{D}$ with pseudometrics ϱ_a for $a \in A_{sym}$, in addition to simple local decision rule (in short s-rule) (see Subsection II-B) denoted by $s\text{-}rule(tst, trn)$, we consider two new types of local rules: *combined local decision rule* (in short c-rule) and *generalised local decision rule* (in short g-rule) denoted by $c\text{-}rule(tst, trn)$, $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$ (or simply $g\text{-}rule\,(tst, trn)$), respectively. In this way we obtain sets composed out of simple rules, combined rules and general rules, denoted by $SimRules$ (see Subsection II-A), $CombRules$, $GenRules$, respectively. We already demonstrated (see Subsection II-B) an important relation between any s-rule and the set of maximally general consistent rules

[3] The formulation of this proposition in [32] was different. However, in the case of $SimRules$ it is equivalent to the original proposition and makes it possible in a more direct way present the relationship between local rules as well as $MaxRules$ and algorithms based on these two types of rules.

$MaxRules(SimRules, trnSet)$. Here, we show analogous important relations between any c-rule or g-rule and sets of maximally general rules $MaxRules(CombRules, trnSet)$, $MaxRules(GenRules, trnSet)$ corresponding to sets of rules $CombRules$ and $GenRules$, respectively.

1) Extension of lazy rule learning for numerical attributes: In this section we assume that $\mathbb{D}$ is a given decision system and $trnSet \subseteq \mathbf{X}$.

Now, we extend the definition of the local decision rule to the case of symbolic and numerical attributes.

Definition III.1. *Let tst be a test object and $trn \in trnSet$. By t_a for $a \in A_{sym}$ we denote a condition as in Definition II.10 and $min_a = min(a(tst), a(trn))$, $max_a = max(a(tst), a(trn))$ for $a \in A_{num}$ We define the* combined local decision rule *(for short* c-rule*) if $\bigwedge_{a \in A} T_a$ then $d = d(trn)$, denoted by $c\text{-}rule(tst, trn)$, where conditions T_a for $a \in A$ are as follows*

$$T_a = \begin{cases} a \in [min_a, max_a] & \text{if } a \in A_{num} \\ t_a & \text{if } a \in A_{sym}. \end{cases}$$

Let us note that conditions for numerical attributes contain intervals with endpoints determined by attribute values of objects tst and trn.

The relationship of the set $MaxRules(CombRules, trnSet)$ and $c\text{-}rule(tst, trn)$ is analogous to the relation between $MaxRules(SimRules, trnSet)$ and $s\text{-}rule(tst, trn)$ as the following lemma states.

Lemma III.1. *Any rule $r \in MaxRules(CombRules, trnSet)$ covering the given test object tst and training object trn is implied by the rule $c\text{-}rule(tst, trn)$.*

Proof. From $r \in MaxRules(CombRules, trnSet)$ we have, in particular that r is consistent with $trnSet$. Hence, the postcondition of r is $d = d(trn)$, i.e. the decision of r is the same as $c\text{-}rule(tst, trn)$.

Since r covers tst and trn, $t_a(r)(trn)$ and $t_a(r)(tst)$ are satisfied for each $a \in A$, i.e. $trn \in [[t_a(r)]]_{\mathbb{D}}$ and $tst \in [[t_a(r)]]_{\mathbb{D}}$.

It is enough to show that for any $a \in A$ the implication $t_a(c\text{-}rule(tst, trn)) \Rightarrow t_a(r)$ holds, i.e. $V_a(c\text{-}rule(tst, trn)) \subseteq V_a(r)$, where for any rule r' if $t_a(r')$ is of the form $a \in V$ then $V_a(r')$ is defined as $V_a(r') = ||V||_{\mathbb{D}}$.

Let us first assume that $a \in A_{sym}$. If $t_a(r)$ is of the form $a \in V_a$, then the implication obviously holds (trivial condition is implied by any condition, because for any elementary condition $a \in V$ for attribute a, $||V||_{\mathbb{D}} \subseteq ||V_a||_{\mathbb{D}}$. If $t_a(r)$ is for some $v \in V_a$ of the form $a = v$ (i.e. $a \in \{v\}$) then $v = a(trn) = a(tst)$. The last equalities hold because we already concluded that $t_a(r)(trn)$ and $t_a(r)(tst)$ are both satisfied. Hence, we have $trn \in [[a = v]]_{\mathbb{D}}$ and $tst \in [[a = v]]_{\mathbb{D}}$, i.e. $a(trn) \in \{v\}$ and $a(tst) \in \{v\}$. It means that in the considered case the equality $t_a(r) = t_a(c\text{-}rule(tst, trn))$ holds (see Definition III.1 and Definition II.10).

If $a \in A_{num}$ is numerical then $t_a(r)$ is of the form $a \in I$, where I is the description of interval corresponding to the numerical attribute a of rule r. Because $t_a(r)(trn)$ and $t_a(r)(tst)$ are both satisfied then $a(trn) \in ||I||_{\mathbb{D}}$ and $a(tst) \in ||I||_{\mathbb{D}}$. Thus $\{a(trn), a(tst)\} \subseteq ||I||_{\mathbb{D}}$. Hence, all points between $a(trn)$ and $a(tst)$ are also in $||I||_{\mathbb{D}}$. In consequence, $[min_a, max_a] \subseteq ||I||_{\mathbb{D}}$, where $min_a = min(a(tst), a(trn))$, $max_a = max(a(tst), a(trn))$ what ends the proof of inclusion $V_a(c\text{-}rule(tst, trn)) \subseteq V_a(r)$ (see Definition III.1). □

Theorem III.2. *The rule $c\text{-}rule(tst, trn)$ for the test object tst and the training object trn is consistent with the training set $trnSet$ if and only if there exists a rule $r \in MaxRules(CombRules, trnSet)$ covering objects tst and trn.*

Proof. We start from a proof of the following fact: if $c\text{-}rule(tst, trn)$ is consistent with $trnSet$ then it can be extended to a rule from $MaxRules(CombRules, trnSet)$. Such a rule can be constructed inductively. From assumption we have that $r_0 = c\text{-}rule(tst, trn) \in CombRules$ is consistent with $trnSet$. In the induction step to define each next rule r_i, for $i = 1, 2, \ldots, m$, where $m = |A|$, we assume that r_{i-1} is consistent with $trnSet$ and conditions $t_j(r_{i-1})$ for all $j = 1, 2, \ldots, i-1$ are maximally general, i.e. if we replace any condition t_j with a more general t (i.e. $t_j \Rightarrow t$) preserving consistency, then $t_j = t$.

$t_i(r_i)$, in i-th induction step, is defined as the maximal generalisation of $t_i(r_{i-1}) = \ldots = t_i(r_0) = t_i(c\text{-}rule(tst, trn))$ preserving consistency with $trnSet$. All others conditions and the decision of the rule are not changed, i.e. $t_j(r_i) = t_j(r_{i-1})$ for $j \neq i$; $d(r_i) = d(r_{i-1})$. Hene, in i-th induction step we simply maximally generalise condition for attribute a_i.

If $a_i \in A_{sym}$ and $t_i(r_{i-1})$ is the trivial condition, then we put $r_i = r_{i-1}$. If $t_i(r_{i-1})$ is non-trivial, it is substituted by $a \in V_a$ if the consistency of the rule is preserved; otherwise, we put $r_i = r_{i-1}$.

If $a_i \in A_{num}$ then $t_i(r_{i-1})$ is of the form $a_i \in [min, max]$. Let us denote by $rule_i(r, t)$ the result of replacement in r of i-th condition by a condition t. Now, we define a set of values of attribute a by $a(Inc) = \{a(trn) : trn \in Inc\}$, where $Inc = \{trn \in trnSet : d(trn) \neq d(r_0) \wedge rule_i(r_{i-1}, a_i = *)$ covers $trn\}$, i.e. Inc contains objects which may violate the consistency of the rule under the maximal possible extension of the condition $t_i(r_{i-1})$. From the inductive assumption, r_{i-1} is consistent with Inc because $Inc \subseteq trnSet$. Hence, $a(Inc) \cap [min, max] = \emptyset$. Now we define $newmax = min\{v \in a(Inc) : v > max\}$. This minimum exists because Inc and also $a(Inc)$ are finite sets. If the set $\{v \in a(Inc) : v > max\}$ is empty we take $newmax = u_{a_i}$ (i.e. maximal possible extension of the right end of the interval). Analogously, we define $newmin = max\{v \in a(Inc) : v < min\}$. If $\{v \in a(Inc) : v < min\}$ is empty we put $newmin = l_{a_i}$ (i.e. maximal possible extension of the left end of the interval). Finally, we define $t_i(r_i)$ by $a \in (newmin, newmax)$. From the definition, r_i is consistent with $trnSet$ and is also maximal because other ends of the interval $(newmin, newmax)$ even if extended by one point to a closed interval will cause inconsistency (in case $newmin = l_{a_i}$ this end of the interval cannot be extended; analogously in case $newmax = u_{a_i}$).

It is easy to prove that all other conditions $t_j(r_i)$ for $j < i$ remain still maximal. To prove this, let us assume that for some $j < i$ $t_j(r_i)$ could be extended to t with preserving consistency, i.e. $rule_j(r_i, t)$ is consistent. We also have $rule_j(r_{i-1}, t) \Rightarrow rule_j(r_i, t)$. Therefore $rule_j(r_{i-1}, t)$ is consistent with $trnSet$. From the inductive assumption t is identical with $t_j(r_{i-1})$. Because $t_j(r_{i-1})$ is the same as $t_j(r_i)$ for $(j < i)$, then t is the same as $t_j(r_i)$. It means that $t_j(r_i)$ is maximally general.

Our inductive reasoning leads to a conclusion that the last rule r_m is consistent with $trnSet$ and maximally general.

We have $c\text{-}rule(tst, trn) \Rightarrow r$ for any rule $r \in MaxRules(CombRules, trnSet)$ covering objects tst and trn (see Lemma III.1). Hence, inconsistency of $c\text{-}rule(tst, trn)$ implies inconsistency of all rules $r \in MaxRules(CombRules, trnSet)$ covering tst and trn. □

2) Generalisation of lazy rule learning for symbolic attributes:
In the section by $\mathbb{D}$ we denote a given pseudometric decision system and $trnSet \subseteq \mathbf{X}$ is a given training set.

In the previous Definitions II.10 and III.1, the trivial condition $a \in V_a$ for a symbolic attribute a is introduced. This condition represents the specific grouping of all possible values of an attribute and is satisfied by any object. However, a proper subset of V_a may be more relevant for the classification. Grouping of values can be obtained by applying a given pseudometric ρ_a for a. Here, now we formulate the following generalisation of Definition III.1, related to a grouping of values for symbolic attributes:

Definition III.2. *Let tst be a test object, $trn \in trnSet$, and $min_a = min(a(tst), a(trn))$, $max_a = max(a(tst), a(trn))$ for $a \in A_{num}$. We also use the following notation: (i) $r_a = \varrho_a(a(tst), a(trn))$ for radius, (ii) $B(c, R)$ for closed pseudometric ball of radius R centred at point c defined by the pseudometric ϱ_a. Now, we define the* generalised local decision rule *(for short* g-rule*) if $\bigwedge_{a\in A} t_a$ then $d = d(trn)$, denoted by g-rule $\left(tst, trn, \{\varrho_a\}_{a\in A_{sym}}\right)$ or simply g-rule (tst, trn) (if parameters $\{\varrho_a\}_{a\in A_{sym}}$ are clear from the context or irrelevant due to generality of considerations), where:*

$$t_a = \begin{cases} a \in [min_a, max_a] & \text{if } a \text{ is numerical} \\ a \in B\left(a(tst), r_a\right) & \text{if } a \text{ is symbolic.} \end{cases}$$

Now, we prove that an analogous relationship of the set $MaxRules(GenRules, trnSet)$ and $g\text{-}rule\,(tst, trn)$ (g-rule) to the relation between $MaxRules(SimRules, trnSet)$ and $s\text{-}rule(tst, trn)$ holds.

Lemma III.3. *Let tst be any test object and $trn \in trnSet$. Let GenRules be defined by parameters ϱ_a (from given pseudometric decision system) and $c_a = a(tst)$ for $a \in A_{sym}$ (see Definition II.5), i.e. $GenRules = GenRules\left(\{(\varrho_a, a(tst))\}_{a\in A_{sym}}\right)$. Then g-rule $\left(tst, trn, \{\varrho_a\}_{a\in A_{sym}}\right) \Rightarrow r$ for any $r \in MaxRules(GenRules, trnSet)$ covering objects tst and trn.*

Proof. The proof is an extension of proof of Lemma III.1. For numerical attributes, the proof is the same as before. For symbolic attributes, it is enough to change the part of the proof of Lemma III.1 by the following one. Let $a \in A_{sym}$. Then $t_a(r)$ is of the form $a \in B(a(tst), R_a)$, where $R_a = \varrho_a(a(tst), v)$, for some $v \in V_a$. Hence, $R_a \geq \varrho_a(a(tst), a(trn))$ because $t_a(r)(trn)$ is satisfied. So, we obtain $B\left(a(tst), r_a\right) \subseteq B(a(tst), R_a)$, where $r_a = \varrho_a(a(tst), a(trn))$. Hence, we have $t_a(g\text{-}rule\,(tst, trn)) \Rightarrow t_a(r)$. □

Theorem III.4. *Under the assumptions of Lemma III.3, the rule g-rule $\left(tst, trn, \{\varrho_a\}_{a\in A_{sym}}\right)$ is consistent with $trnSet$ if and only if there exists a rule $r \in MaxRules(GenRules, trnSet)$ such that r covers tst and trn.*

Proof. The proof can be obtained by a modification of the proof of Theorem III.2.

It is enough to modify the inductive step of the proof of Theorem III.2 for $a_i \in A_{sym}$ as follows. If $a_i \in A_{sym}$ then $t_i(r_{i-1})$ is of the form $a \in B(a(tst), r_a)$, where $r_a = \varrho_a(a(tst), v)$, for some $v \in V_a$. Now, let us consider possible extensions of $a \in B(a(tst), r_a)$ by $a \in B(a(tst), R_a)$, where $R_a = \varrho_a(a(tst), w)$, for some $w \in V_a$ and $R_a \geq r_a$ preserving consistency (with $trnSet$) of the rule. In the finite set of such possible extensions (due to the fact that V_a is finite) we select the one with the maximal value of R_a. The selected extension is maximally general.

If $a_i \in A_{num}$ then we extend the formula as in Theorem III.2.

One can conclude that the last rule r_m is consistent with $trnSet$ and maximally general by performing analogous reasoning as in Theorem III.2.

Also, in analogous way as in Theorem III.2 with the use of Lemma III.3, we obtain that the following implication holds: if $g\text{-}rule\,(tst, trn)$ is inconsistent with $trnSet$, then in $MaxRules(GenRules, trnSet)$ there is no rule covering tst and trn. □

Let us note that the set $MaxRules(GenRules, trnSet)$ is defined for the given values c_a for $a \in A_{sym}$ (in the testing procedure we assume $c_a = a(tst)$). The idea behind construction of $MaxRules(SimRules, trnSet)$ was to compute all maximally general rules in advance for the later use in the classification process. In order to construct $MaxRules(GenRules, trnSet)$, this should be done for all possible combinations of all possible values for all symbolic attributes. It would increase the number of generated rules by the factor no more than b^k, where b is the maximal cardinality of $|V_a|$ for $a \in A_{sym}$ and k is the number of symbolic attributes.

From Theorem III.4 it follows that it is sufficient to generate g-rules for all training examples and then check their consistency with $trnSet$ (instead of computing the support sets for rules from $MaxRules(GenRules, trnSet)$ covering a new test case). The lazy Rule Induction Algorithm (RIA) realises this idea.

```
Algorithm 4: RIA($tst$, $trnSet$, $\{\varrho_a\}_{a\in A_{sym}}$)
Input: test example $tst$, training set $trnSet$, family of
       pseudometrics for symbolic attributes
       $\{\varrho_a\}_{a\in A_{sym}}$
Output: predicted decision for $tst$
1  begin
2      foreach decision $v \in V_d$ do
3          $supportSet(v) = \emptyset$
4      end
5      foreach $trn \in trnSet$ do
6          $v = d(trn)$
7          if $isCons\left(g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a\in A_{sym}}\right), trnSet\right)$ then
8              $supportSet(v) = supportSet(v) \cup \{trn\}$
9          end
10     end
11     return $\arg\max_{v\in V_d} |supportSet(v)|$
12 end
```

$isCons(r, verifySet)$ is defined in Algorithm 2 (see Subsection II-B). The RIA (see Algorithm 4) and LAZY (see Algorithm 3) algorithms differ only in line 7, namely in Algorithm 3 the rule $s\text{-}rule(tst, trn)$ is used and in Algorithm 4 this is the rule $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a\in A_{sym}}\right)$.

RIA computes the measure $Strength$ for $MaxRules = MaxRules(GenRules, trnSet)$ what directly follows from Theorem III.4. Hence, the results of RIA are equivalent to the results of the algorithm based on calculating $MaxRules$ with $Strength$ as a strategy for conflict resolution. In this way, we obtain the corollary analogous to Corollary II.2 (see Subsection II-B).

Corollary III.5. *For any test object tst, and the classifier* $decision_{MaxRules}(tst)$ *(see Eqn. 8), where*

$$MaxRules = MaxRules(GenRules, trnSet)$$

and

$$GenRules = GenRules\left(\{(\varrho_a, a(tst))\}_{a\in A_{sym}}\right),$$

we have

$$RIA(tst, trnSet, \{\varrho_a\}_{a\in A_{sym}}) = decision_{MaxRules}(tst).$$

B. Combining instance-based learning and rule methods – RIONA

In this section, we additionally assume that Agr is an aggregation function defined as sum of individual metrics.

Let us recall that RIONA is based on a combination of instance-based learning and rule-based methods. The primary observation used in the development of RIONA concerns the property of the widely used kNN method. kNN has quite good performance, usually for small values of k. Hence, one may expect that that only training examples close to a given test case are important in the process of inducing (inferring) the final decision. The intuition supporting this claim is that the training examples which are far from a given test object are less relevant for classification than the closer ones. Contrary to this, in the case of rule-based methods, in general, all training examples are used in the process of rule generation. Hence, instead of considering all training examples in constructing the support set in the case of rule-based approach, like in the RIA algorithm, one can bound it to a certain neighbourhood of a test example. In the case of RIONA algorithm, the classification of a given test case is based on training objects from a neighbourhood of this example.

Our approach to inducing of decision for a given test case is basing on a combination of instance-based learning and lazy rule learning (see Section III-A). The core idea concerns the strategy for conflict resolution based on $Strength$ measure (see Eqn. 7) slightly modified by bounding it to the neighbourhood of the test case:

$$LocStrength(tst, v, k, \varrho) = \\ = \left| \bigcup_{r\in MatchR(tst,v)} locSuppSet(r) \right|, \tag{9}$$

where most notation is as in Eqn. 7; additionally $\varrho = Agr(\{\varrho_a\}_{a\in A})$ is the aggregated pseudometric and k is the number indicating the size of the neighbourhood, and $locSuppSet(r) = supportSet(r) \cap N(tst, trnSet, k, \varrho)$. One can observe that the change from $supportSet(r)$ to $locSuppSet(r)$ causes that only those examples covered by the rules matched by a test object that are in a specified neighbourhood of the test example are considered. The predicted decision based on $LocStrength$ is analogous to the previous one (see Eqn. 8):

$$decLoc_{MaxRules}(tst, k, \varrho) = \\ \arg\max_{v\in V_d} LocStrength(tst, v, k, \varrho). \tag{10}$$

Let us note that the size k of the neighbourhood is optimised in the learning phase (see [5]) while in the classification process, we assume that number k for the neighbourhood $N(tst, k)$ is set to this optimal value.

The above measures can be calculated for a given $MaxRules$ by bounding the support sets of the rules from $MaxRules$ covering a test example to the specified neighbourhood of a given test example. Hence, the algorithm based on maximally general rules with $LocStrength$ can be used here.

It is worthwhile mentioning that $LocStrength$ can also be calculated using the lazy rule learning methodology. This can be done analogously to computing by RIA the measure $Strength$ (see Corollary III.5). For this purpose we modified Algorithm 4 as follows (i) in line 5 of the algorithm, only examples $trn \in N(tst, k)$ should be considered, (ii) it is not necessary to consider all the examples from the training set to check the consistency of the $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$ (see line 7 of Algorithm 4). This follows from the next proposition.

Proposition III.6. *Suppose that ϱ_a (for $a \in A_{num}$) in a given pseudometric decision system are defined as normalised Euclean metric, $\varrho = Agr(\{\varrho_a\}_{a \in A})$ and Agr is defined either by sum of metrics or weighted sum of metrics. If $trn' \in trnSet$ satisfies g-rule $\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$, then $\varrho(tst, trn') \leq \varrho(tst, trn)$.*

Proof. If trn' satisfies $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$, then we have (see Definition III.2 of g-rule):

- for any $a \in A_{sym}$ we have $a(trn') \in B\left(a(tst), r_a\right)$, where $r_a = \varrho_a(a(tst), a(trn))$. Hence, from definition of the closed ball it follows that $\varrho_a(a(tst), a(trn')) \leq \varrho_a(a(tst), a(trn))$.
- for any $a \in A_{num}$ we have $a(trn') \in [min_a, max_a]$, where $min_a = min(a(tst), a(trn))$, $max_a = max(a(tst), a(trn))$. Hence $|a(tst) - a(trn')| \leq |a(tst) - a(trn)|$. Thus, using definiton of metric for numerical attributes (normalised Euclidean metric) we have $\varrho_a(a(tst), a(trn')) = \frac{|a(tst) - a(trn')|}{a^{\max} - a^{\min}} \leq \frac{|a(tst) - a(trn)|}{a^{\max} - a^{\min}} = \varrho_a(a(tst), a(trn))$.

Hence, for any $a \in A$ we have $\varrho_a(a(tst), a(trn')) \leq \varrho_a(a(tst), a(trn))$. In consequence, we obtain the following inequality between the global distances[4] for Agr defined by sum of metrics $\varrho(tst, trn') = \sum_{a \in A} \varrho_a(a(tst), a(trn')) \leq \sum_{a \in A} \varrho_a(a(tst), a(trn)) = \varrho(tst, trn)$.

One can observe that we have the same result also for aggregation function defined by weighted sum of metrics. This is because adding weights for each attribute preserves the above inequality.

□

From the above considerations it follows that the examples distanced from tst more than the training example trn cannot cause inconsistency of $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$. Hence, one can use $N(tst, trnSet, k, \varrho)$ instead of $trnSet$ in line 7 of Algorithm 4.

The description of classification algorithm RIONA is presented in Algorithm 5. Later on we prove that Algorithm 5 computes $LocStrength$ (see Theorem IV.2). Algorithm 5 returns the most common class corresponding to decisions on the training examples covered by the rules satisfied by tst and belonging to the specified neighbourhood. One should note that all pseudometrics in the argument of Algorithm 5 are given (used for computation of the final pseudometric). However, in g-rule only pseudometrics for symbolic attributes are used (see Definition III.2 and note after it).

Algorithm 5: RIONA-classify(tst,$trnSet$,k,$\{\varrho_a\}_{a \in A}$)

```
Input: test example $tst$, training set $trnSet$, positive
       integer $k$, family of pseudometrics for attributes
       $\{\varrho_a\}_{a \in A}$
Output: predicted decision for $tst$
1  begin
2      $\varrho = Agr(\{\varrho_a\}_{a \in A})$
3      $nSet = N(tst, trnSet, k, \varrho)$
4      foreach decision $v \in V_d$ do
5          $supportSet(v) = \emptyset$
6      end
7      foreach $trn \in nSet$ do
8          $v = d(trn)$
9          if
             $isCons\left(g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right), nSet\right)$
           then
10             $supportSet(v) = supportSet(v) \cup \{trn\}$
11         end
12     end
13     return $\arg\max_{v \in V_d} |supportSet(v)|$
14 end
```

For every decision value, RIONA computes the support set restricted to the neighbourhood $N(tst, k)$ rather than the whole support set of the maximally general rules covering tst (as in the case of RIA algorithm). This is done as follows. For any $trn \in trnSet$ from $N(tst, k)$ RIONA constructs the rule $g\text{-}rule\left(tst, trn, \{\varrho_a\}_{a \in A_{sym}}\right)$ based on trn and tst. Next, RIONA is testing whether g-rule is consistent with the examples from the neighbourhood $N(tst, k)$. If g-rule is consistent with $N(tst, k)$ then the support set of the decision $d(trn)$ is extended by trn. Finally, RIONA returns the decision value with the support set of the highest cardinality.

IV. Relationships of RIONA to other approaches

A specific combination of kNN approach and lazy rule induction allowed us to develop the algorithm RIONA. One can observe that only the line 9 of RIONA, where is examined the consistency of the rule determined by the training and testing example, differs from Algorithm 1 (kNN).

The relationships between RIONA, RIA and kNN for $k = 1$ are as follows.

Proposition IV.1. *Let us assume that* 1NN *is the nearest neighbour algorithm for $k = 1$ with a distance defined by pseudometric $\varrho = Agr(\{\varrho_a\}_{a \in A})$. Then for any test object tst we have*

$$RIONA(tst, trnSet, k, \{\varrho_a\}_{a \in A}) =$$
$$\begin{cases} RIA(tst, trnSet, \{\varrho_a\}_{a \in A_{sym}}) & for\ k \geq |trnSet| \\ 1NN(tst, trnSet, \varrho)\ for\ k = 1, |N(tst, trnSet, k, \varrho)| = 1. \end{cases}$$

[4] We assume that pseudometrics used for grouping symbolic attributes are the same as the pseudometrics composing the aggregated pseudometric used for measuring distance between examples. The analogous assumption is used for numerical attributes: real values are grouped using interval contained in the ball $B(a(tst), \varrho_a(a(tst), a(trn)))$ determined by the Euclidean metric. The same Euclidean metric (however normalised) is used for components of the final pseudometric.

Proof. If $k \geq |trnSet|$ then the $neighbourSet = trnSet$, where $neighbourSet$ is defined in the RIONA algorithm (see Algorithm 5). Hence, RIONA works exactly as the RIA algorithm (see Algorithm 4).

If $k = 1$ and $|N(tst, trnSet, 1, \varrho)| = 1$[5] then the $neighbourSet$ in the RIONA algorithm is a singleton set and consistency checking can be omitted. Hence, RIONA works exactly as 1NN (see Algorithm 1). □

RIONA behaves like the RIA algorithm for the maximal neighborhood (and from the Corollary III.5 as the algorithm based on the maximally general rules with the $Strength$ strategy for conflict resolution). By taking a neighbourhood based on the one nearest training example, the nearest neighbour algorithm is obtained. RIONA is positioned between the rule-based classifier based on the maximally general rules and the nearest neighbour classifier. If a small neighborhood is chosen then it acts more like a kNN classifier and if a large neighbourhood is chosen it works more like a rule-based classifier based on inducing maximally general rules. Selection of a neighbourhood that is not the maximal may be interpreted as taking more specific rules instead of maximally general rules consistent with the training examples.

Below, we present more properties of RIONA.

Theorem IV.2. *The following equality holds:*
$RIONA(tst, trnSet, k, \{\varrho_a\}_{a\in A}) =$
$decLoc_{MaxRules}(tst, k, \varrho)$,
where

(i) *tst is any test object,*
(ii) $decLoc_{MaxRules}(tst, k, \varrho)$ *is the output of classifier from Eqn. 10 with*
- $MaxRules = MaxRules(GenRules, trnSet)$,
- $GenRules = GenRules\left(\{(\varrho_a, a(tst))\}_{a\in A_{sym}}\right)$,
- $\varrho = Agr(\{\varrho_a\}_{a\in A})$.

Proof. We have $RIA(tst, trnSet, \{\varrho_a\}_{a\in A_{sym}})$ $= decision_{MaxRules}(tst)$ (see Corollary III.5). This is based on the following fact: RIA computes measure $Strength$ for $MaxRules = MaxRules(GenRules, trnSet)$, i.e. for each $v \in V_d$ $supportSet(v)$ (in line 11 of Algorithm 4) $=$ $Strength(v)$ (see Theorem III.4). RIONA works only on training examples from $N(tst, trnSet, k, \varrho)$ and examples consistent with $N(tst, trnSet, k, \varrho)$ are also consistent with the whole training set, $trnSet$ (see Proposition III.6). In computing of the measure $LocStrength(tst, v, k, \varrho)$ are used only training examples from the neighbourhood $N(tst, trnSet, k, \varrho)$. Hence, $supportSet(v)$ (see line 13 of Algorithm 5) $=$ $LocStrength(tst, v, k, \varrho)$ for any $v \in V_d$. From this the equation of the theorem follows. □

Theorem IV.3. *The following equality holds:*
$RIONA(tst, trnSet, k, \{\varrho_a\}_{a\in A}) =$
$decLoc_{MaxRules}(tst, k, \varrho)$,
where

(i) *tst is any test object,*
(ii) $decLoc_{MaxRules}(tst, k, \varrho)$ *is the output of classifier from Eqn. 10 with*
- $MaxRules =$ $MaxRules(GenRules, N(tst, trnSet, k, \varrho))$,
- $GenRules = GenRules\left(\{(\varrho_a, a(tst))\}_{a\in A_{sym}}\right)$,
- $\varrho = Agr(\{\varrho_a\}_{a\in A})$.

Proof. The following equality holds (see Theorem IV.2):
$RIONA(tst, N(tst, trnSet, k, \varrho), k, f) =$
$decLoc_{MaxRules}(tst, k, \varrho)$,
where $trnSet$ is substituted by a new training set $N(tst, trnSet, k, \varrho))$,
and
$f = \{\varrho_a\}_{a\in A}$,
$MaxRules = MaxRules(GenRules, N(tst, trnSet, k, \varrho))$.
We also have
$RIONA(tst, N(tst, trnSet, k, \varrho), k, f) =$
$RIONA(tst, trnSet, k, f)$
what ends the proof. □

From the last two theorems the following interesting corollary follows.

Corollary IV.4. *Let us assume that there are given* $\{\varrho_a\}_{a\in A}$, $\varrho = Agr(\{\varrho_a\}_{a\in A})$, $trnSet$,
$MaxRules = MaxRules(GenRules, trnSet)$,
$MaxLocalRules =$
$MaxRules(GenRules, N(tst, trnSet, k, \varrho))$,
where $GenRules = GenRules\left(\{(\varrho_a, a(tst))\}_{a\in A_{sym}}\right)$.
Then the outputs returned by the following classifiers are the same for any tst *object:*

1. $RIONA(tst, trnSet, k, \{\varrho_a\}_{a\in A})$,
2. $decLoc_{MaxRules}(tst, k, \varrho)$,
3. $decLoc_{MaxLocalRules}(tst, k, \varrho)$,
4. $decision_{MaxLocalRules}(tst)$ *with a new training set* $trnSet' = N(tst, trnSet, k, \varrho)$.

Proof. From Theorems IV.2 and IV.3 it follows the equivalence of the first three classifiers. To obtain the equivalence of the third and fourth classifiers let us observe that
$trnSet' = N(tst, trnSet, k, \varrho) =$
$N(tst, N(tst, trnSet, k, \varrho), k, \varrho) = N(tst, trnSet', k, \varrho)$.
We also have $supportSet(r) \subseteq trnSet'$. Hence, from the previous equation we obtain $supportSet(r) \subseteq N(tst, trnSet', k, \varrho)$. Then $locSuppSet(r) =$ $supportSet(r) \cap N(tst, trnSet', k, \varrho) = supportSet(r)$. Now one can see that Eqn. 9 becomes Eqn. 7, what implies that Eqn. 10 becomes Eqn. 8. □

Summarising, the conclusions are the following: (i) RIONA calculates the $LocalStrength$ measure (see Eqn. 9). (ii) The $LocStrength$ measure is the $Strength$ measure with $N(tst, k)$ as the local training set (fourth algorithm). (iii) The algorithm presented in Eqn. 8 after substitution of a

[5] This assumption can't be omitted. If $|N| > 1$ then it may happen (even for consistent training set) that N has (equally distanced from test example) two cases with different decisions leading to inconsistency.

new training set $trnSet' = N(tst, trnSet, k, \varrho)$ instead of $trnSet$ becomes the fourth algorithm. (iv) One can consider the RIONA algorithm as an algorithm for computing all maximally general, consistent rules locally and using (locally) $Strength$ for conflict resolution.

In Table I and Table II is presented comparison of these algorithms (the third algorithm is omitted because it is very similar to the fourth).

Table I
A GENERAL COMPARISON OF THREE ALGORITHMS FROM COROLLARY IV.4: ALGORITHM (1) RIONA, ALGORITHM (2) BASED ON THE MEASURE $LocStrength$ AND ALGORITHM (4) BASED ON THE MEASURE $Strength$ COUNTED LOCALLY.

RIONA	algorithm (2) based on the measure $LocStrength$	algorithm (4) based on the measure $Strength$ counted locally
counting rules		
no need to count rules explicitly	counts $MaxRules$ globally once at the beginning	counts $MaxRules$ locally for each test case
counting support		
counts support using lazy local rules	counts support locally	counts support locally

Table II
A COMPARISON SCHEME OF THREE ALGORITHMS FROM COROLLARY IV.4: ALGORITHM (1) RIONA, ALGORITHM (2) BASED ON THE MEASURE $LocStrength$ AND ALGORITHM (4) BASED ON THE MEASURE $Strength$ COUNTED LOCALLY.

RIONA	algorithm (2) based on the measure $LocStrength$	algorithm (4) based on the measure $Strength$ counted locally
Global input: $trnSet$, $k \in \mathbb{N}$		
1.	count $MaxRules$ for $trnSet$	
Input: test case tst		
2. $nSet = N(tst, k)$		
3.	$RuleBase = MaxRules$	count (locally) $MaxRules(nSet)$ $RuleBase = MaxRules(nSet)$
4.	consider rules from $RuleBase$ with premise satisfied by tst	
5. for each decision d		
6. consider $trn \in nSet$ with decision d	consider rules from step 4 with decision d	
7. count the number of trn from step 6 forming consistent rules with tst	count the number of $trn \in nSet$ supporting rules from step 6	
8. choose the decision with the maximal count (maximally supported)		

A. RIONA and rules

Some important properties of instance-based classifiers and rule-based classifiers are inherited by the RIONA algorithm. Even though rule-based classifiers produce less accurate classifications, there are several features of them that users prefer over instance-based classifiers. The ability for a human, non-computer science professional, to interpret rules is one of these crucial features. He or she can check to see if the information found in such rules is non-trivial, accurate, and revealing brand-new features of the considered case. A rule includes an explanation for making the specific decision that is simple enough for a human to comprehend.

Here, we assume that the RIONA algorithm's parameter k is fixed (potentially learnt [5]). Let's now concentrate on algorithm (4) from Sect. III. Because the local complete set of consistent and maximally general decision rules must be computed for each test case tst, the direct computation of $MaxLocalRules$ may initially appear to be highly expensive and impractical. However, if we assume that the size of N is k, then the size of the local training sample is much smaller than the size of the entire training sample being reduced from $n = |trnSet|$ to k. As a result, the total cost of computing $MaxRules$ (globally or locally) is decreased from $O(2^n)$ to $O(m \cdot 2^k)$, where m is the number of test cases. We don't just present this strategy from a theoretical standpoint only. When a classifier's decision needs to be explained, this kind of method might be useful. In this way, the RIONA algorithm shares characteristics with rule algorithms and quick lazy learning algorithms, i.e. its parameters can be converted into rules.

Additionally, algorithm (4) might be extended to construct all rules globally once at the beginning, analogously to algorithm (2) from Corollary IV.4, except that the rules would be based on the local neighborhood only. Such rules would mimic the RIONA algorithm's behavior. The use of such a strategy has several benefits. First, a set of rules could be immediately provided to explain the predicted decision on a particular test object. Second, the usefulness of the knowledge acquired might be tested against all potential rules generated at the beginning.

The approach for construction of these rules is analogous as in algorithm 4 from Corollary IV.4. One could just construct $MaxRules$ locally for each training case and use each training example as a test example. It could be seen as a computation of specific local reducts i.e. reducts constructed during generation of maximally general rules for a given object (see e.g. [33], [34], [35], [32]). Usually, in construction of local reducts one should be aware to keep discernibility for objects with various decisions. Only objects with different decisions and at a distance of no more than determined by k would be required to be discernible in this case.

V. CONCLUSION

The presented findings indicate some important relationships of classifiers generated by the RIONA learning algorithm with instance- and rule-based classifiers. For example, it is proved the relative to classification equivalence of the RIONA algorithm to the algorithm generating all consistent and maximally general rules from a training set including the close training cases to a given test case. As a result, the classification by RIONA classifier can be performed by a relatively small set of rules that are simple for a person to comprehend. It might be applied in circumstances where it's crucial to provide an explanation for the decision that was reached by classifier. Finally, it is worthwhile to mention that the RIONA algorithm, based on hybridization of instance- and rule-based techniques, has the following properties (i) it is efficient as well as effective from the point of view of classification, (ii) it can be used as a high quality tool in the process of explanation of the predicted decisions.

REFERENCES

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ: Pearson Education, 2021.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2009. [Online]. Available: https://doi.org/10.1007/978-0-387-84858-7

[3] T. M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.

[4] G. Góra and A. Wojna, "RIONA: A New Classification System Combining Rule Induction and Instance-Based Learning," *Fundamenta Informaticae*, vol. 51, no. 4, pp. 369–390, 2002. [Online]. Available: https://doi.org/10.5555/2371138.2371141

[5] ——, "RIONA: A Classifier Combining Rule Induction and K-nn Method with Automated Selection of Optimal Neighbourhood," in *Proceedings of the 13th European Conference on Machine Learning (ECML 2002)*. Heidelberg: Springer-Verlag, 2002, pp. 111–123. [Online]. Available: https://doi.org/10.1007/3-540-36755-1_10

[6] G. Góra, "Combining instance-based learning and rule-based methods for imbalanced data," Ph.D. dissertation, University of Warsaw, Warsaw, 2022, [Online]. Available: https://www.mimuw.edu.pl/sites/default/files/gora_grzegorz_rozprawa_doktorska.pdf.

[7] J. Fürnkranz, D. Gamberger, and N. Lavrac, *Foundations of Rule Learning*, ser. Cognitive Technologies. Heidelberg: Springer, 2012. [Online]. Available: https://doi.org/10.1007/978-3-540-75197-7

[8] A. Skowron and D. Ślęzak, "Rough sets turn 40: From information systems to intelligent systems," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30, 2022, pp. 23–34. [Online]. Available: https://doi.org/10.15439/2022F310

[9] C. C. Aggarwal, "Instance-Based Learning: A Survey," in *Data Classification: Algorithms and Applications*, 1st ed., C. C. Aggarwal, Ed. New York: Chapman & Hall/CRC, 2014, pp. 157–186. [Online]. Available: https://doi.org/10.1201/b17320

[10] C. Cornelis, "Hybridization of fuzzy sets and rough sets: Achievements and opportunities," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30, 2022, pp. 7–14. [Online]. Available: https://doi.org/10.15439/2022F302

[11] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2870052

[12] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. Croatian Society MIPRO, 2018, pp. 0210–0215. [Online]. Available: https://doi.org/10.23919/MIPRO.2018.8400040

[13] H. Hagras, "Toward Human-Understandable, Explainable AI," *Computer*, vol. 51, no. 9, pp. 28–36, 2018. [Online]. Available: https://doi.org/10.1109/MC.2018.3620965

[14] D. W. Aha, Ed., *Lazy Learning*, 1st ed. Dordrecht: Springer, 1997. [Online]. Available: https://doi.org/10.1007/978-94-017-2053-3

[15] A. Wojna and R. Latkowski, "Rseslib 3: Library of Rough Set and Machine Learning Methods with Extensible Architecture," in *Transactions on Rough Sets XXI*, J. F. Peters and A. Skowron, Eds. Berlin, Heidelberg: Springer, 2019, pp. 301–323.

[16] N. Dey, S. Borah, R. Babo, and A. S. Ashour, Eds., *Social Network Analytics: Computational Research Methods and Techniques*, 1st ed. London: Academic Press, 2019. [Online]. Available: https://doi.org/10.1016/C2017-0-02844-6

[17] L. Grama and C. Rusu, "Choosing an accurate number of mel frequency cepstral coefficients for audio classification purpose," in *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (ISPA 2017)*, 2017, pp. 225–230. [Online]. Available: https://doi.org/10.1109/ISPA.2017.8073600

[18] R. de Oliveira Almeida and G. T. Valente, "Predicting metabolic pathways of plant enzymes without using sequence similarity: Models from machine learning," *The Plant Genome*, vol. 13, no. 3, p. e20043, 2020. [Online]. Available: https://doi.org/10.1002/tpg2.20043

[19] C. Rusu and L. Grama, "Recent developments in acoustical signal classification for monitoring," in *2017 5th International Symposium on Electrical and Electronics Engineering (ISEEE)*, 2017, pp. 1–10. [Online]. Available: https://doi.org/10.1109/ISEEE.2017.8170705

[20] L. Grama and C. Rusu, "Adding audio capabilities to TIAGo service robot," in *2018 International Symposium on Electronics and Telecommunications (ISETC)*, 2018, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ISETC.2018.858389

[21] A. Almasri, E. Celebi, and R. S. Alkhawaldeh, "EMT: Ensemble Meta-Based Tree Model for Predicting Student Performance," *Scientific Programming*, vol. 2019, pp. Article No. 3 610 248, 1–13, 2019. [Online]. Available: https://doi.org/10.1155/2019/3610248

[22] N. R. Howes, *Modern Analysis and Topology*. New York: Springer Science+Business Media, 1995. [Online]. Available: https://doi.org/10.1007/978-1-4612-0833-4

[23] P. Domingos, "Unifying instance-based and rule-based induction," *Machine Learning*, vol. 24, no. 2, pp. 141–168, 1996. [Online]. Available: https://doi.org/10.1007/BF00058656

[24] H. S. Nguyen and A. Skowron, "Quantization of Real Value Attributes – Rough Set and Boolean Reasoning Approach," in *Proceedings of the 2nd Joint Annual Conference on Information Sciences (JCIS 1995)*, 1995, pp. 34–37.

[25] A. Skowron and C. Rauszer, "The Discernibility Matrices and Functions in Information Systems," in *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, R. Słowiński, Ed. Dordrecht: Springer, 1992, pp. 331–362. [Online]. Available: https://doi.org/10.1007/978-94-015-7975-9_21

[26] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning and Data Mining*, 2nd ed. USA: Springer, 2017. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1

[27] H. S. Nguyen, "Approximate Boolean Reasoning: Foundations and Applications in Data Mining," in *Transactions on Rough Sets V*, J. F. Peters and A. Skowron, Eds. Heidelberg: Springer, 2006, pp. 334–506. [Online]. Available: https://doi.org/10.1007/11847465_16

[28] J. G. Bazan and M. Szczuka, "RSES and RSESlib – A Collection of Tools for Rough Set Computations," in *Rough Sets and Current Trends in Computing (RSCTC 2001)*. Heidelberg: Springer, 2001, pp. 106–113. [Online]. Available: https://doi.org/10.1007/3-540-45554-X_12

[29] J. W. Grzymala-Busse, "LERS-A System for Learning from Examples Based on Rough Sets," in *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, R. Słowiński, Ed. Dordrecht: Springer, 1992, pp. 3–18. [Online]. Available: https://doi.org/10.1007/978-94-015-7975-9_1

[30] S. H. Nguyen, "Regularity Analysis and its Applications in Data Mining," in *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, L. Polkowski, S. Tsumoto, and T. Y. Lin, Eds. Heidelberg: Physica-Verlag, 2000, pp. 289–378. [Online]. Available: https://doi.org/10.1007/978-3-7908-1840-6_7

[31] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, "The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains," in *Proceedings of the 5th AAAI National Conference on Artificial Intelligence*. AAAI Press, 1986, pp. 1041–1045.

[32] J. G. Bazan, "Discovery of Decision Rules by Matching New Objects Against Data Tables," in *Rough Sets and Current Trends in Computing (RSCTC 1998)*. Heidelberg: Springer, 1998, pp. 521–528. [Online]. Available: https://doi.org/10.1007/3-540-69115-4_72

[33] Z. Pawlak and A. Skowron, "A Rough Set Approach to Decision Rules Generation," in *Proceedings of the Workshop W12: The Management of Uncertainty at the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993)*. Chambéry: Morgan Kaufmann, 1993, pp. 114–119.

[34] J. Wróblewski, "Covering with Reducts – A Fast Algorithm for Rule Generation," in *Rough Sets and Current Trends in Computing (RSCTC 1998)*. Heidelberg: Springer, 1998, pp. 402–407. [Online]. Available: https://doi.org/10.1007/3-540-69115-4_72

[35] J. G. Bazan, H. S. Nguyen, S. H. Nguyen, P. Synak, and J. Wróblewski, "Rough Set Algorithms in Classification Problem," in *Rough Set Methods and Applications: New Developments in Knowledge Discovery in Information Systems*, L. Polkowski, S. Tsumoto, and T. Y. Lin, Eds. Heidelberg: Physica-Verlag, 2000, pp. 49–88. [Online]. Available: https://doi.org/10.1007/978-3-7908-1840-6_3

DOI: 10.15439/2023F3560
ISSN 2300-5963 ACSIS, Vol. 35

Estimation of absolute distance and height of people based on monocular view and deep neural networks for edge devices operating in the visible and thermal spectra

Jan Gąsienica-Józkowy*†, Bogusław Cyganek *†, Mateusz Knapik*†, Szymon Głogowski† and Łukasz Przebinda†
*Faculty of Computer Science, Electronics and Telecommunication,
Email: *cyganek@agh.edu.pl*
†MyLED Inc.
Email: *m.knapik@myled.pl*
Ul. W. Łokietka 14/2, 30–016 Kraków, Poland

***Abstract*—Accurate estimation of absolute distance and height of objects in open area conditions is a significant challenge. In this paper, we address these problems and we propose a novel approach that combines classical computer vision algorithms with modern neural network-based solutions. Our method integrates object detection, monocular depth estimation, and homography-based mapping to achieve precise and efficient estimations of absolute height and distance. The solution is implemented on the edge device, which enables real-time data processing using both visual and thermography data sources. Experimental evaluation on a height estimation dataset prepared by us demonstrates an accuracy of 97.06% and validates the effectiveness of our approach.**

I. Introduction

ACCURATE estimation of spatial positions and parameters of objects, such as their localization on a bird's-eye view map, absolute distance, and absolute height, is an important computer vision task with wide practical implications. In this paper, we propose a novel solution for absolute distance and height estimation that combines homography-based mapping algorithms with state-of-the-art deep learning techniques. Our approach harnesses the strengths of both classical and modern solutions to achieve highly accurate and efficient estimations under various conditions.

The proposed method integrates several key components to provide a comprehensive solution for absolute height estimation. Firstly, we capture video frames from visual and thermography cameras and input them into an object detector, specifically the YOLOv5 model [1]. This model enables robust identification and localization of objects in the monocular view. To estimate the relative depth information, we utilize a transformer-based monocular depth estimation model called DPT Levit 224 [2], [3]. This model learns to infer depth information from a single image, allowing us to determine the relative distances between objects in the scene. Additionally, we incorporate homography-based mapping techniques to establish correspondences between points in different images or views. By leveraging homography projection, we can accurately map objects from the video frame plane to the bird's-eye view 2D map, enabling easy estimation of their distance from the camera. The final stage of our approach involves polynomial regression-based estimations to compute the absolute distance and height of objects.

The proposed solution is implemented on the Arabox III-A edge device, which is based on the Jetson Nano board and offers real-time data processing capabilities. Arabox is specifically designed for fully anonymous data acquisition and is commonly used in the Digital Out-of-Home (DOOH) advertising industry.

In our experimental evaluation, we demonstrate the effectiveness of our approach and its evaluation on the prepared by us absolute height estimation dataset. The obtained results show accuracy equal to 97.06% in real-time performance, emphasizing the usefulness of our solution in a wide range of applications requiring precise absolute distance and height estimation.

The rest of this paper is organized as follows: Section 2 II provides a detailed overview of related works, including object detection, homography-based mapping, monocular depth estimation, and absolute height estimation techniques. Section 3 III presents a comprehensive description of the architecture of our solution, along with information about the necessary configuration and calibration process. Section 4 IV presents the experimental results, which are divided into indoor and outdoor experiments, accompanied by a description of the dataset used and the methodology employed. Finally, section 5 V discusses the implications of our findings and identifies potential areas for future improvement.

II. Related works

In this chapter, we provide an overview of the existing research and advancements in the field of computer vision, with a special focus on object detection, homography-based mapping, monocular depth estimation, and absolute height estimation techniques.

A. Object detection

The task of object detection is widely used in computer vision and has a wide range of applications [4]–[6]. Currently, the best object detectors are based on convolutional neural networks (CNN). The initial success of CNN-based object detectors came with two-stage detectors like the region-based convolutional neural network (R-CNN) proposed by Girshick et al. [7] which has shown remarkable performance. This led to further advancements such as Fast R-CNN [8] and Faster R-CNN [9] - improved two-stage detectors, faster and with better accuracy. Another approach that gained popularity are one-stage detectors, exemplified by groundbreaking architectures like You Only Look Once (YOLO) [10] and Single-Shot Detector (SSD) [11]. They are faster, part of them can even work in real-time on edge devices, and currently have comparable accuracy to two-stage detectors [12].

The leading one-stage detection architecture YOLO has undergone significant improvements over time. Namely, its improved versions YOLOv2 [13] and YOLOv3 [14] introduced deeper convolutional backends, residual skip connections, residual blocks, and upsampling, resulting in one of the fastest object detection techniques while maintaining respectable accuracy. Bochkovskiy et al. presented YOLOv4 [15], which brought further enhancements to the training process, including data augmentation methods like CutMix, regularization techniques such as DropBlock, and architectural changes like the CSPDarknet53 backend network and path aggregation network with spatial attention blocks.

More recently, Jocher et al. presented YOLOv5 [1], which refreshed the YOLO architecture and improved its performance. The YOLO-based architecture remains a state-of-the-art object detector, with subsequent versions continually being developed and published under different names.

These advancements in object detection have significantly improved the accuracy and speed of detecting objects in various applications.

B. Homography-based mapping

Homography-based mapping is a widely used technique in computer vision that establishes correspondences between points in different images or views. It relies on the concept of a homography, which is a projective transformation that maps points from one plane to another. This mapping has numerous applications, including image stitching, augmented reality, camera calibration, and object tracking.

Works by Hartley and Zisserman [16], as well as by Cyganek and Siebert [17] provide a comprehensive overview of homography estimation algorithms. Additionally, the work of Szeliski [18] presents techniques for the robust estimation of homographies in the presence of outliers and noise. These studies serve as foundational knowledge for our use of homography-based mapping in height estimation.

C. Monocular Depth Estimation

Monocular depth estimation aims to recover depth information from a single image. This task is challenging due to the inherent ambiguity in monocular vision. Nevertheless, it plays a crucial role in various applications such as 3D reconstruction, scene understanding, and autonomous navigation.

Over the years, significant progress has been made in monocular depth estimation techniques. Early approaches were focused on hand-crafted features, superpixelation, and traditional computer vision algorithms [19]–[21]. However, with advancements in deep learning, convolutional neural networks have emerged as powerful tools for monocular depth estimation.

One notable work in this field is the pioneering study by Eigen et al. [22] where they introduced a CNN-based approach for monocular depth prediction. This work paved the way for subsequent research in deep learning-based depth estimation. Another significant contribution is the work of Laina et al. [23], who proposed a faster and lighter solution by training a fully convolutional residual network based on ResNet-50 [24]. They replaced the fully connected layers with up-convolutional blocks and modified the loss function.

Subsequently, the development of CNN-based solutions accelerated, leading to the creation of numerous works addressing this area. A few noteworthy contributions deserving special attention are listed below. Lee et al. [25] proposed a solution based on the relative depths between objects in the image. Ranftl et al. [26] presented a tool for mixing multiple datasets during monocular depth estimation training, even when their annotations were incompatible. This tool has facilitated future advancements in this field. Additionally, Ranftl et al. [2] proposed a dense vision transformer-based depth estimation architecture with a transformer backbone. Their architecture produces more fine-grained and globally coherent predictions compared to fully-convolutional architectures.

D. Absolute height estimation

Absolute height estimation is an intriguing computer vision task, but less popular than those mentioned above. To address this problem, various approaches based on image depth estimation [27], convolutional neural networks [27]–[29], and convolutional-deconvolutional deep neural networks (CDNNs) [30] have been proposed.

A notable work in this domain is the study conducted by Yin et al. [27], where they developed a four-stage estimator based on multiple CNN networks operating on a single-depth image. Their approach achieved impressive accuracy in height estimation, reaching as high as 99.1%. It is worth mentioning that their solution was limited to a controlled laboratory environment, where measurements were conducted on individuals positioned approximately 2 meters away from the camera. Despite this limitation, the achieved result is truly remarkable.

The field of absolute height estimation is relatively specialized, and fewer studies have been conducted compared to

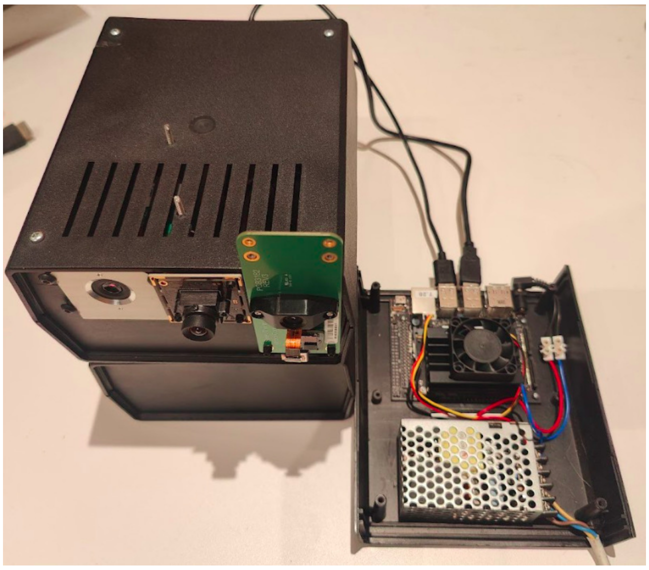


Figure 1: Arabox device - version with normal and termovision cameras and Jetson Nano board.

other computer vision tasks. Therefore, we believe that it is an interesting area for further research and development.

III. System Architecture

In this chapter, we present the architecture of our method for estimating the absolute height and distance of objects, which is implemented on an edge device called Arabox III-A, developed by MyLED [31]. Our approach for height estimation relies on two sources of data: a video signal and a thermal image signal. The system flow consists of integrating object detection using the YOLOv5 neural network, monocular depth estimation transformer, homography-based mapping, and polynomial regression-based estimations. We provide a detailed explanation of our method in section III-B.

A. Arabox device

Arabox, shown in Figure 1, is a device developed for fully anonymous data acquisition in the retail industry. It can be used in both stationary stores (including those operating in the omnichannel model) and the outdoor advertising industry (particularly in digitized form, known as DOOH). The device's key component is an embedded system that includes a GPU, such as the Jetson Nano, which is responsible for encrypting and processing data from the connected cameras. Arabox also includes a carrier board, power supply, fans, and a special case. Arabox has many use cases, but below we will focus on its height estimation functionality.

B. Height estimation pipeline

The main contribution of our paper is the absolute height estimation pipeline presented in Figure 2. It utilizes two data sources: a video signal and a thermal image signal. Both signals are initially processed by the YOLOv5 object detector before being fed into separate flows. In the first flow, we project the video signal onto a bird's-eye view using homography-based mapping. This enables us to estimate the spatial position of objects, as well as their distance from the camera and height (based on initial configuration, polynomial regressions, and the height of bounding boxes estimated by YOLOv5). We perform the same process for the thermal image signal, but with a different homography matrix.

The second flow is only performed for the video signal and is based on DPT Levit 224 monocular depth estimation model. This network estimates the relative depth of the image, and its output is combined with YOLOv5 detections to calculate the average depth value for the detected bounding boxes. Using a polynomial regression model from the device configuration and the relative depths of the detected objects, we can estimate their absolute distance from the camera, as well as their absolute height, in the same way as in the first flow. Finally, we average the results from all three flows to estimate the final height of the objects. More details on each of the pipeline steps are provided in the following subsections.

1) YOLOv5 detections: This part of our pipeline comprises two YOLOv5 models, which have been trained on two datasets that we prepared - one based on visual and the other on thermal images. These datasets contain approximately 20,000 annotated photos captured in urban environments. The YOLOv5 models output class of an object (such as human, car, or bus), its anchor location represented by two coordinates, and the height and width of the bounding box. All expressed in local coordinates associated with an image plane. This information is then used in the subsequent steps of our system, i.e. in the homography-based mapping and monocular depth estimation.

2) Homography-based mapping: The goal of this step is to project the location of the detected object from a 3D photo to a 2D "map" presented from a bird's-eye view. This projection enables us to estimate the distance and height of the object in the next step. To accomplish this, we first calculate the homography matrix for a given location during the device calibration process III-C. Subsequently, we use this matrix to transform the YOLOv5 detections and project them onto a 2D plane. By knowing their positions on the 2D plane and the scale of the plane saved during the configuration process, we can accurately estimate their distance from the camera, as well as their height, using polynomial regression.

3) Monocular depth estimation: In this step, we utilize the monocular depth estimation neural network called DPT Levit 224 [2], [3]. Given an input image, the network returns a map of relative depth estimates, where lower pixel values correspond to objects that are further from the camera. To improve the accuracy of our estimations, we first crop the image to remove any visible sidewall or casing fragments before passing it to the neural network.

The DPT Levit 224 model we use was trained using a publicly available tool for mixing monocular depth estimation datasets [26]. By using this pre-trained model, we can estimate the relative depth of objects in the scene with high accuracy, even in cases where the objects are partially occluded or have

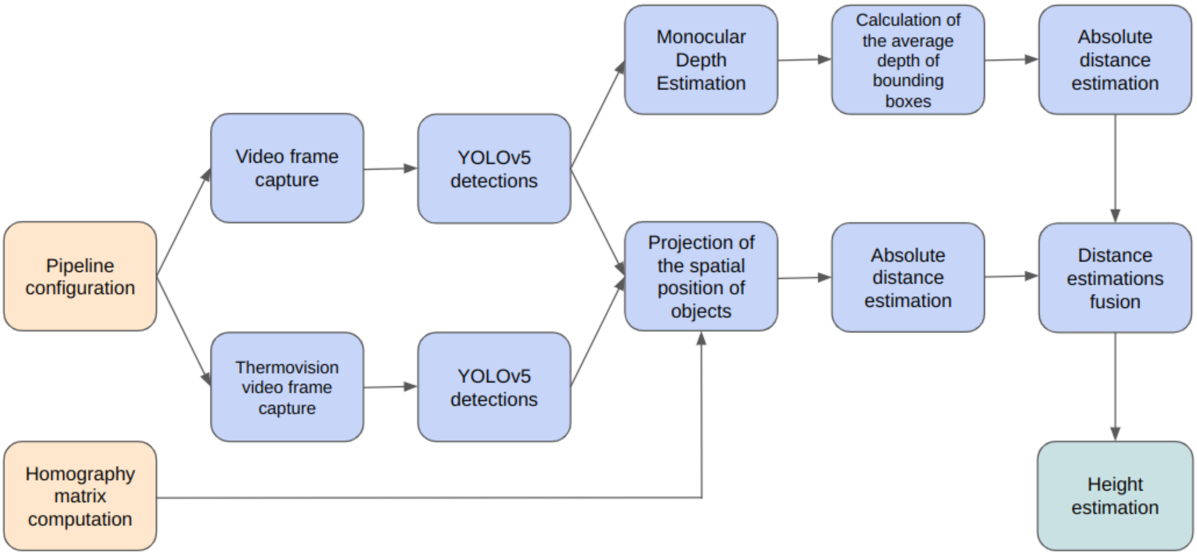

Pipeline configuration
Video frame capture
YOLOv5 detections
Thermovision video frame capture
YOLOv5 detections
Homography matrix computation
Monocular Depth Estimation
Calculation of the average depth of bounding boxes
Absolute distance estimation
Projection of the spatial position of objects
Absolute distance estimation
Distance estimations fusion
Height estimation

Figure 2: Block diagram of our height estimation system.

complex geometries.

4) Objects' absolute distance estimation: The estimation of object distance from the camera is performed using two methods, depending on the results of the previous step. If we estimate the distance based on the spatial position of the object obtained through homography-based mapping, the calculation is straightforward. We simply multiply the object's distance from the camera (expressed in pixels) by the scale factor included in the device configuration III-C.

On the other hand, if we estimate distance using monocular depth estimation, the process is more complex. In this case, we first calculate the average depth value for each bounding box returned by YOLOv5, and then substitute these values into a polynomial regression formula contained in the device configuration. This formula expresses the relationship between depth values and distance at a given location, enabling us to estimate absolute distance accurately. Details on how to calculate the coefficients of said polynomials are contained in the configuration section III-C.

5) Objects' absolute height estimation: After we have obtained the absolute distance of the object from the camera, we can estimate its height. However, for the estimate to be accurate, we need to calibrate the device to a specific location beforehand and calculate the coefficients of the 3rd-order polynomial accurately. This polynomial regression formula is used to determine the relationship between the distance of the object from the camera and the ratio of its height in pixels to the height in the real world. The process of calibration and calculating these coefficients is further explained in the configuration section III-C.

Once we have the coefficients and the distance value, we substitute them into the polynomial formula to obtain a height ratio. We then multiply this ratio by the height of the object in pixels obtained from the YOLOv5 detector. This calculation allows us to obtain an accurate estimate of the absolute height of the object.

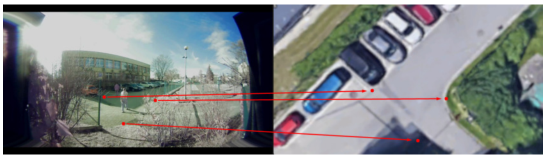


Figure 3: A photo showing the process of calculating the homography matrix. The person responsible for the configurations marks the points on the image from the camera and the corresponding 2D map.

C. Configuration and calibration process

To ensure the proper functioning of the methods described, it is necessary to configure and calibrate the system. The most important parameters that we need to configure for each location where the device is to be used are: homography matrices, coefficients of the third-order polynomials used in the polynomial regression of distance and height, and the scale factor of pixels to meters.

Homography matrices are calibrated for a specific location using a simple script that requires marking several points on the original image and the 2D image, as shown in Figure 3. If the points are marked accurately, the script will return a homography matrix that allows for the projection of objects' locations from the camera's perspective onto a bird's-eye view. A separate matrix should be calculated for each camera (i.e. visual and thermal), since their parameters are different.

The polynomial regression is employed to determine the relationship between the distance of an object from the camera and the ratio of its height in pixels to its height in the real world. To compute the third-order polynomial coefficients used

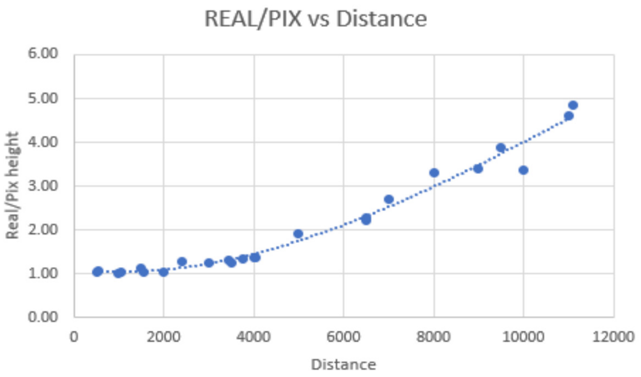
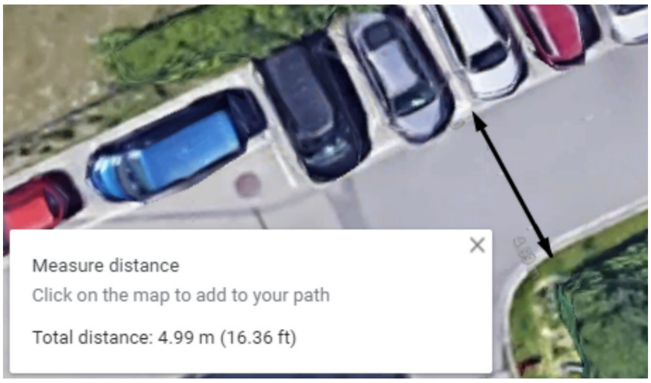


Figure 4: Regression curve calculated in the calibration phase for mapping the height of objects in pixels to the height of objects in meters.


Measure distance

Click on the map to add to your path

Total distance: 4.99 m (16.36 ft)

Figure 5: An example of distance measurement for device calibration using Google Maps.

for this regression, the following steps are proposed: Firstly, several YOLOv5 detections of a person with a known height should be made at different distances from the camera, and the height returned by YOLOv5 in pixels should be recorded. Next, a plot similar to the one shown in Figure 4 should be created, and a third-order polynomial regression should be computed on it. The obtained coefficients should then be saved in the device configuration. The height estimation module will then multiply the distance of the object from the camera by these coefficients and then by the height of the YOLOv5 prediction. This will provide an estimate of the height of the given object. A similar process should also be performed for the monocular depth estimation module and its depth-to-distance regression.

The final parameter needed to calibrate the device to a specific location is the scale that determines how many centimeters in the real world correspond to one pixel on the 2D map. This parameter can be easily calculated by measuring the distance between two characteristic objects on a Google Maps and then checking how many pixels on our 2D map they correspond to, as shown in Figure 5. Once all the parameters have been calibrated and configured, the device is ready for use in estimating the absolute height and distance of objects for chosen location. In the future we want to improve and automate the configuration process.

IV. Experimental part

To validate the effectiveness of our method, we conducted an experiment using a small dataset comprising videos of 11 individuals with known heights. The videos were captured in two distinct locations: one in an open environment and the other inside the building. By utilizing this dataset, we evaluated the performance of our system following the methodology outlined in section IV-B and achieved an estimation accuracy of 97.06%.

The experiment aimed to assess the system's ability to accurately estimate the height of individuals in different environmental conditions and validate the effectiveness of our proposed approach. In the following sections, we will discuss the details of the gathered dataset, our methodology and we will present the results obtained from our evaluation.

A. Dataset

The dataset comprises 10 recordings, each featuring a different individual with a known height. The recordings were captured using two types of cameras: a regular vision camera (model ELP-USB500W05G-FD100) and a thermal imaging camera (model SEEK Thermal MS202SP Micro Core).

The dataset includes videos from two distinct locations: indoors, specifically in an office space, with a total of three recordings, and outdoors, in a parking lot, with a total of seven recordings. The individuals participating in the recordings had heights ranging from 160cm to 185cm. While the dataset may not be extensive, we believe it provides sufficient variety to validate and confirm the effectiveness of our absolute growth estimation method. Sample frames from videos used in our dataset are presented in Figure 6.

B. Methodology

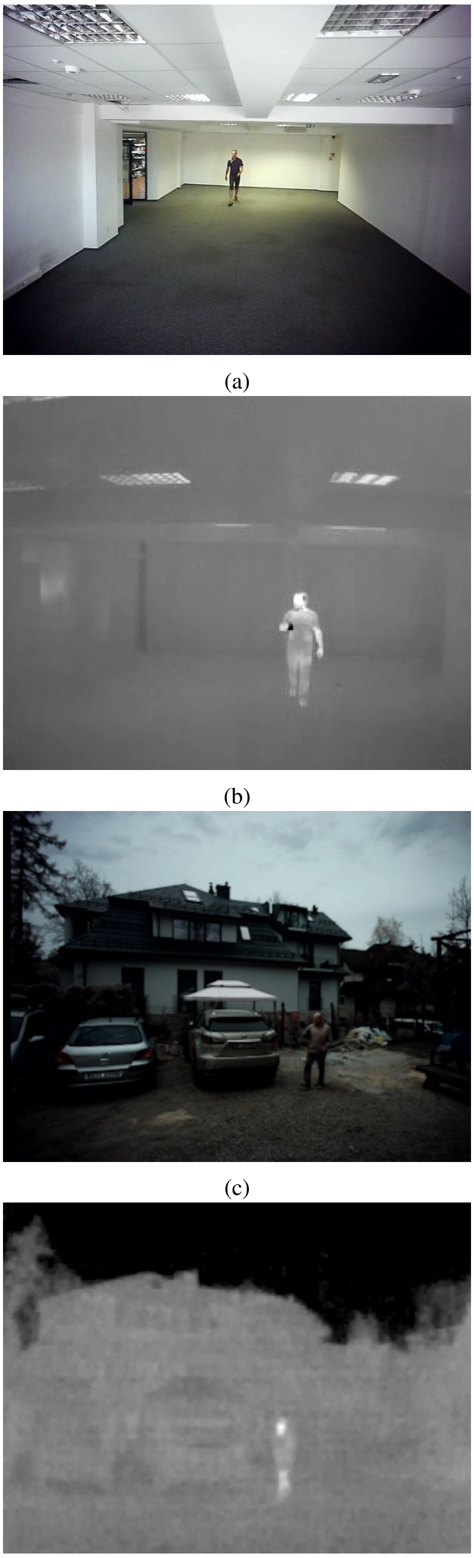
To validate the performance of our method for estimating the absolute height of individuals, we employed the following methodology.

For each video in our dataset, our model conducted height estimations on every frame in which the YOLOv5 detection model detected a person. The estimated heights were stored in a temporary table, and the measurements were averaged at the end of the video, using the Formula 1. Where, h_a represents the averaged height measurement result, h_i represents the result from a single frame, and N is the number of frames in which the person was measured.

$$h_a = \frac{\sum\limits_{i=1}^{N} h_i}{N} \tag{1}$$

These estimates were then compared against the known actual heights of the individuals (h_e) to calculate the percentage errors using Formula 2.

$$\delta = |\frac{h_a - h_e}{h_e}| * 100\% \tag{2}$$


(a)

(b)

(c)

(d)

Figure 6: Sample images from the dataset

To provide a comprehensive evaluation, we stored all the results, as well as results from every module of our system in Table I and Table II, respectively. These tables serve as a consolidated record of the estimated heights, actual heights, and corresponding absolute errors for each video. Additionally, they contain the estimated heights from each component of the pipeline, namely results from the homography-based mapping using the vision data (HBM vision), homography-based mapping using the thermal data (HBM thermo), monocular depth estimation (MDE), and the fusion module. The fusion module results are calculated as presented in formula 3.

$$Fusion = \frac{\frac{HBE_{vision}+HBE_{thermovision}}{2}+MDE}{2} \tag{3}$$

As can be observed, the fusion formula is not a simple arithmetic average, as the module based on monocular depth estimation carries the greatest weight. This is because the homography-based mapping modules provide similar information, whereas the monocular depth estimation module offers distinct and additional insights. By utilizing this methodology, we can quantitatively assess the accuracy and reliability of our height estimation method across the entire dataset. The percentage error values obtained will allow us to analyze the performance of our system and identify areas for improvement.

In the subsequent sections, we will present the detailed results obtained from our evaluation and discuss the implications of these findings for the effectiveness of our proposed method.

C. Results

The results of our experiments are presented in three separate tables. Table I displays the measurements conducted indoors for three individuals, while Table II showcases the measurements carried out in an open area for eight individuals. Finally, Table III provides a weighted average summary of the results obtained from all experiments.

The average accuracy achieved in each experiment is as follows: 97.73% for Experiment 1, 96.77% for Experiment 2, with an overall weighted average accuracy of 97.06%. These accuracy percentages represent the degree of agreement between the estimated heights and the actual heights of the individuals. A more detailed description of the experiments and their results is provided below.

1) Experiment 1 - indoor area: In the first experiment conducted indoors, specifically in an office space; we recorded the heights of three individuals ranging from 173 cm to 186 cm; the maximum distance from the camera in which they could walk was around 12 meters. The system performed around 370 measurements for each person and then averaged them to obtain the final results presented in Table I. The average accuracy of the absolute height estimations obtained in this experiment was 97.73%. The best-performing module is based on homography mapping with a signal from the video camera with an error of only 1.14%. On the other hand, the worst-performing module is also homography-based mapping, but with a signal from the thermal camera - with a percentage

Table I: Indoor experiment results

	HBM vision	HBM thermo	MDE	Fusion	Ground Truth	Number of frames
Person 1	185cm	180cm	188cm	185cm	186cm	346
Error 1	0.54%	3.23%	1.08%	0.54%	-	346
Person 2	179cm	167cm	170cm	172cm	178cm	350
Error 2	0.56%	6.18%	4.49%	3.37%	-	350
Person 3	177cm	167cm	164cm	168cm	173cm	412
Error 3	2.31%	3.47%	5.20%	2.89%	-	412
Avg. Error	1.14%	4.29%	3.59%	2.27%	-	-

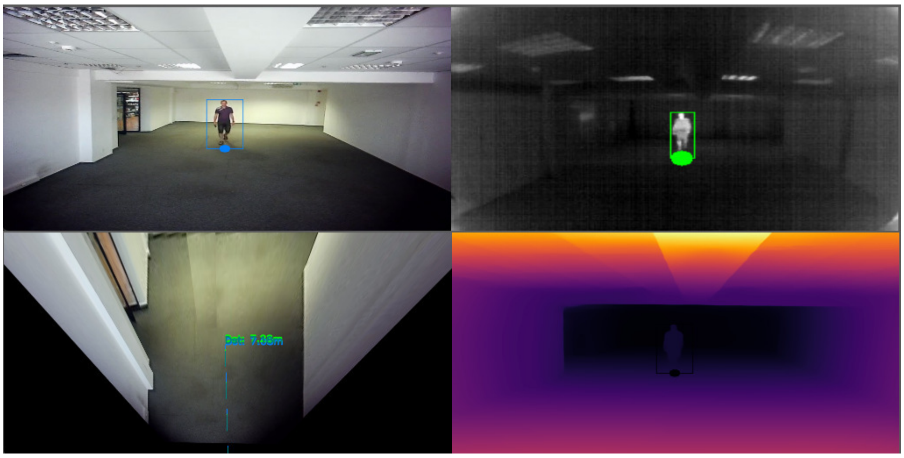


Figure 7: Visualization of the described method: Upper left - detection on a video signal, upper right - detection on a thermal image, bottom left - projection of a person's detections into bird's eye view, bottom right - monocular depth estimation output

error equal to 4.29%. A sample visualization of our method's work on data from the indoor experiment was presented in Figure 7.

2) Experiment 2 - outdoor area: In the second experiment, we conducted measurements in an open area, specifically a small parking lot. Seven individuals with heights ranging from 160 cm to 185 cm participated in this experiment; the maximum distance from the camera in which they could walk was around 20 meters. For each person, a varying number of measurements, ranging from 865 to 1968, were conducted and averaged. Final results are presented in Table II. The average accuracy of the estimations obtained in the second experiment was 96.77%. The best-performing module in this experiment was the monocular depth estimation model, with an average percentage error equal to 3.03%, whereas the worst-performing method was once again thermovision homography mapping with an error equal to 4.39%.

3) Results summary: Summarizing the results of the aforementioned experiments, we achieved a weighted average accuracy of 97.06%. Among the different modules used, the height estimation module based on homography projecting yielded the highest accuracy of 97.35%. The other modules, namely the monocular depth estimation module and homography-based mapping working on thermal imaging, achieved slightly lower accuracies of 95.89% and 95.64%, respectively.

Looking for reasons for such results, the lower accuracy of the model working on thermal imaging data can be attributed to the less accurate detections of the YOLOv5 on the thermal images. The thermal images dataset, on which the YOLOv5 model was trained, was smaller than the traditional dataset, which can correspond to weaker results. Notably, the detections from the thermal-based model were often 10-15% higher in the vertical axis, which was not observed in normal data.

Regarding monocular depth estimation, certain challenges were encountered due to the background conditions. For instance, if a person passed in front of a car, the model believed that person to be closer than if they were at the same distance but there was no car in the background. Despite this limitation, the results achieved in this experiment were considered very good, taking into account the difficulty of the scenery.

Experiment no. 2 presented slightly weaker results due to the more complex scene and higher maximal distance in which individuals could walk. Particularly, beyond 15 meters, the system encountered significant challenges in accurately estimating the distances and therefore absolute heights.

Moving forward, we aim to expand our dataset and conduct experiments in a larger number of testing locations with a more diverse group of individuals. This will further validate and enhance the proposed method. Additionally, we will focus on improving other aspects of our method, which will be discussed in detail in the following section.

V. Conclusion and future works

Presented in this paper a method for absolute distance and height estimation that incorporates a combination of visual and

Table II: Outdoor experiment results

	HBM vision	HBM thermo	MDE	Fusion	Ground Truth	Number of frames
Person 4	187cm	188cm	185cm	186cm	185cm	865
Error 4	1.08%	1.62%	0%	5.40%	-	865
Person 5	171cm	187cm	169cm	174cm	179cm	1044
Error 5	4.47%	4.47%	5.56%	2.79%	-	1044
Person 6	173cm	189cm	179cm	180cm	174cm	937
Error 6	0.57%	7.94%	2.87%	3.45%	-	937
Person 7	167cm	180cm	172cm	173cm	170cm	1255
Error 7	1.76%	5.88%	1.18%	1.76%	-	1255
Person 8	157cm	179cm	171cm	170cm	168cm	1968
Error 8	6.55%	6.55%	1.79%	1.19%	-	1968
Person 9	172cm	171cm	173cm	172cm	167cm	1080
Error 9	2.99%	2.40%	3.59%	2.99%	-	1080
Person 10	151cm	157cm	150cm	152cm	160cm	1015
Error 10	5.63%	1.88%	6.25%	5.00%	-	1015
Avg. Error	3.29%	4.39%	3.03%	3.23%	-	-

Table III: Results summary

	HBM vision	HBM fusion	MDE	Fusion
Avg. Error	2.65%	4.36%	4.11%	2.94%

thermal imaging data, and which employs advanced technologies such as object detection, homography-based mapping, and monocular depth estimation, constitutes a significant scientific contribution to the field of spacial position estimation in real conditions.

With an accuracy of 97.06%, our method demonstrates promising results, making it suitable for applications on edge devices. However, we acknowledge that there is room for improvements. In our future endeavors, we aim to enhance the accuracy of our method and streamline the configuration and calibration processes.

Moving on to future works, one of our primary objectives is to expand our dataset by incorporating additional locations and involving a more diverse range of participants. This expansion would provide valuable insights into the performance of different modules of our height estimation method and their effectiveness in various environmental conditions. By evaluating our method on a more diverse dataset, we can identify areas for improvement and optimize its performance accordingly.

Another improvement of the proposed method will be streamlining the configuration and calibration process. At the moment, it takes an experienced operator about 30 minutes to configure the device for a new location. We would like to streamline this process and automate it further, especially the part related to the calculation of the polynomial regression coefficients of the distance and height estimation modules.

Additionally, we plan to extend our method with new modules. These could include methods such as monocular depth estimation based on thermal imaging, a human pose estimation [32] module, and the utilization of the object segmentation [33] methods for obtaining more accurate data for calculating the average depth of objects with monocular depth estimation module. By incorporating these new modules, we aim to enhance the capabilities and versatility of our method in estimating absolute distance and height.

In conclusion, our article shows that the accurate estimation of absolute distance and height from the monocular view is possible with high accuracy by using a hybrid solution based on object detection, homography-based mapping, and monocular depth estimation. Furthermore, we recognize the potential for further development and propose future improvements in this task.

Acknowledgments

This work was supported by the National Centre for Research and Development, Poland, under the grant no. POIR.01.01.01-00-1116/20.

References

[1] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[2] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," 2021.

[3] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference," *CoRR*, vol. abs/2104.01136, 2021. [Online]. Available: https://arxiv.org/abs/2104.01136

[4] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. N. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *CoRR*, vol. abs/2104.11892, 2021. [Online]. Available: https://arxiv.org/abs/2104.11892

[5] J. Gąsienica-Józkowy, M. Knapik, and B. Cyganek, "An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance," *Integrated Computer-Aided Engineering*, vol. 28, pp. 221–235, 2021, 3.

[6] M. Knapik and B. Cyganek, "Driver's fatigue recognition based on yawn detection in thermal images," *Neurocomputing*, vol. 338, pp. 274–292, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219302280

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. [Online]. Available: https://doi.org/10.1007%2F978-3-319-46448-0_2

[12] M. Knapik and B. Cyganek, "Fast eyes detection in thermal images," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3601–3621, Jan 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-09403-6

[13] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2017.

[14] ——, "Yolov3: An incremental improvement," *ArXiv*, vol. abs/1804.02767, 2018.

[15] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[16] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[17] B. Cyganek and J. Siebert, "An introduction to 3d computer vision techniques and algorithms," pp. 459–474, 01 2009.

[18] R. Szeliski, "Image alignment and stitching: A tutorial," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 1, p. 1–104, jan 2006. [Online]. Available: https://doi.org/10.1561/0600000009

[19] J. Michels, A. Saxena, and A. Y. Ng, "High speed obstacle avoidance using monocular vision and reinforcement learning," in *Proceedings of the 22nd International Conference on Machine Learning*, ser. ICML '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 593–600. [Online]. Available: https://doi.org/10.1145/1102351.1102426

[20] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," *Advances in neural information processing systems*, vol. 18, 2005.

[21] D. Hoiem, A. A. Efros, and M. Hebert, "Automatic photo pop-up," *ACM Trans. Graph.*, vol. 24, pp. 577–584, 2005.

[22] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," 2014.

[23] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," 2016.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[25] J.-H. Lee and C.-S. Kim, "Single-image depth estimation using relative depths," *Journal of Visual Communication and Image Representation*, vol. 84, p. 103459, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320322000190

[26] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," 2020.

[27] F. Yin and S. Zhou, "Accurate estimation of body height from a single depth image via a four-stage developing network," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8264–8273.

[28] D.-s. Lee, J.-s. Kim, S. C. Jeong, and S.-k. Kwon, "Human height estimation by color deep learning and depth 3d conversion," *Applied Sciences*, vol. 10, no. 16, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/16/5531

[29] P. Alphonse and K. Sriharsha, "Depth estimation from a single rgb image using target foreground and background scene variations," *Computers & Electrical Engineering*, vol. 94, p. 107349, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790621003207

[30] L. Mou and X. X. Zhu, "Im2height: Height estimation from single monocular imagery via fully residual convolutional-deconvolutional network," 2018.

[31] M. sp. z o.o., "Myled sp. z o.o." 2021, accessed on 05-22-2023. [Online]. Available: https://myled.pl/

[32] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," 2022.

[33] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *International Journal of Multimedia Information Retrieval*, vol. 9, no. 3, pp. 171–189, jul 2020. [Online]. Available: https://doi.org/10.1007%2Fs13735-020-00195-x

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 513–522
DOI: 10.15439/2023F6718
ISSN 2300-5963 ACSIS, Vol. 35

The reception of holidays in social networks: A case study on Twitter

Bettina K. Hakius*
* Biblisch-Theologische Akademie Wiedenest, Bergneustadt, Germany
E-Mail: hakius@wiedenest.de

***Abstract*—This study aims to push the boundaries of research in practical theology by applying methods from computational social science to identify the reception of holidays in online social networks in German tweets. Can we identify how people talk about holidays and especially Christian holidays on Twitter? As a subquestion, we try to find relevant information for interreligious topics, especially between Christians and Jews: Can we see how Christian holidays are related to or embedded in their Jewish counterparts? Is there an awareness of the Jewish roots of certain Christian holidays? While there is a growing awareness of these issues, there are still a number of unanswered questions. In addition to analysing and discussing these questions, we will also discuss methodological issues. First, we will discuss how computational methods fit in with common research in practical theology. Secondly, we will discuss the challenges of working with digital data beyond quantitative and qualitative research.**

I. Introduction

MUCH research in practical theology is devoted to religious practices and their interaction with society. It tends to take an empirical approach similar to that of the social sciences. Digital methods and the analysis of digital texts and communications are emerging themes. There is an ongoing theological and cultural discussion about the connection between Jewish and Christian holidays, and we want to find out how they are perceived in online social networks.

The research of this project tries to identify the reception of holidays in online social networks in German tweets and to evaluate the lack of awareness of the relationship between Christianity and Judaism in the context of a post-Christian society in German-speaking countries. While not our primary concern, we will also present a discussion on tweets with anti-Semitic content[1]. Our main research questions are

- Can we identify how people talk about holidays and especially Christian holidays on Twitter? (RQ1)
- If we find relevant information: Can we see how Christian holidays are related to or embedded in their Jewish counterparts? Is there an awareness of the Jewish roots of certain Christian holidays? (RQ2)

We will provide an overview of related work, leading to a careful reflection on the methodological background of the computer science methods used. While these methods – using an API, searching, clustering, analysis – are not new, they are not directly related to traditional empirical methods. We reflect on the validity of the chosen mix of methods – combining qualitative and quantitative research – and evaluate our methodological approach. We also provide a brief introduction to the religious and theological background of the material studied.

This paper is divided into five sections. The first section provides an introduction, the second a brief overview of related work. The third section describes the methodological background, the data, the pipeline and the matching approaches. The fourth section is devoted to the analysis of the results. This includes a religious-historical and theological pre-location of the festivals studied and a discussion of the relevance of these results for Christian churches. Conclusions and implications are drawn in the final section.

II. Related Work

The proposed approach aims to analyse social networks, in this case Twitter, to gain insights into religious phenomena [5], [6]. The analysis of online social networks is an emerging trend in research within the last decade [7]. For a general overview of Twitter data analysis, see [6], [8]. The applications and research questions are numerous, it has been used for health analysis [9], IoT [10], politics, recommender systems or emergency situations [11]. In addition to technical issues, ethical issues are also discussed [12].

First, we consider social network analysis of Twitter data in theological and religious research. This is more or less a niche topic, see [13], [14], [14]. Research questions focus on how religious leaders use social media [15], uncovering hidden church populations [16] or the relationship between sentiment of tweets and church growth [17]. In addition to metadata such as geographic data, word frequencies are widely used [18]. Secondly, clustering approaches are widely used to organise data [19], [20]. This is either related to identifying topics [22] or grouping similar tweets, for example using word similarities [21]. However, there is no one-size-fits-all solution and the choice of the best technology also depends on the data being processed, e.g. hashtags [22].

Thirdly, while we focus on German tweets, the technologies used are largely identical to other languages [23]. Basic statistical information can be extracted [24], other researchers focus on hashtag and word frequencies [25].

In summary, we can rely on many existing solutions, and especially the focus on German tweets does not remain a technical challenge. However, for interdisciplinary research,

[1] For further details on hate speech, racism and anti-Semitism we refer to [1], [2], [3], [4]. Especially hate-speech detection is under constant research. However, in this work we will only collect evidence for further research.

especially for religious and theological research, there are several methodological issues that have received more attention in recent years: While the analysis of online social networks is based on methods from computer and data science, which lead to quantitative data, the research usually also includes a qualitative analysis of these data [26]. While the use of qualitative analysis was widely perceived as inappropriate [27], the situation has changed [28], [29]. Salvatore and Bianchi propose a mixed methods approach, while Dongo et al. summarise that “the differences in terms of accuracy and efficiency of both extraction methods [give] relevance to much more problems related to this area to pursue true transparency and legitimacy of information on the Web.” As this issue is particularly important for religious and theological research, which often focuses on qualitative research, we will also focus on a critical interdisciplinary reflection of a mixed-methods approach

III. Method

While there are numerous features that can be used to analyse Twitter data, such as network measures, finding related information, text measures, word frequency patterns, sentiment analysis and local information, our approach mainly aims at answering the question “what people say when they tweet about holidays”, similar to the approach of Vidal et al. [30]. We propose a mostly generic analysis workflow, which we present in the first subsection. We then describe the clustering approach and the data generated and analysed.

With the digital methods presented here, we have created a text corpus focusing on aspects of two Jewish-Christian festivals. The available quantitative data will be analysed in terms of content and interpreted with available qualitative methods (especially the concordant search function) and with methods of qualitative social research, and only on the side we consider the findings of corpus linguistics [31].

In general, tweets were retrieved using Python and the Tweepy library, which accesses the Twitter API itself. This allows us to search for Twitter users, tweets and collect data. We also used spaCy and sklearn.

In order to preserve the anonymity of the Twitter users, we provide a paraphrasing translation of the German tweets into English. For an in-depth discussion of these and other ethical issues, we refer to [32].

A. Analysis Workflow

The analysis workflow consists of four steps: querying Twitter (using Twitter API v2), data processing, data storage and data analysis, see Figure 1. Querying Twitter is done using predefined queries, and data processing consists of two simple tasks: Extraction and handling of metadata, e.g. tweet ids, datetime, retweets and likes, and a basic NLP approach to extract hashtags.

Due to the relatively small number of tweets, we used a SQLite database to store the tweets for further analysis. We used a clustering approach based on hashtags, which is described in the next subsection.

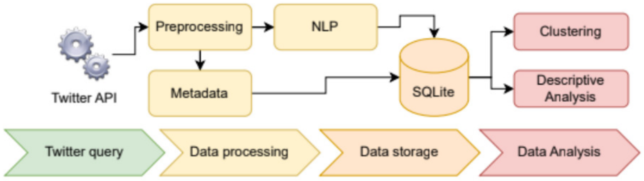


Figure 1. The proposed workflow queries the Twitter API and uses basic NLP techniques to extract hashtags and stores all data and metadata in a SQLite database. Data analysis is performed using descriptive analysis and clustering approaches.

B. Clustering

To cluster tweets by hashtags, we rely on K-means as a classic hard clustering approach. Similarity was computed as semantic relatedness according to GermaNet [33], similar methods exist for other data and languages [34]. However, for some small datasets no meaningful clustering of the data can be obtained. For all other datasets, we worked with different values of K that gave the best performance in distinguishing four different clusters of interest:

- Church related tweets.
- Pandemic related tweets.
- Tweets related to holidays.
- Tweets related to holidays and travel in general.

For two datasets, we will describe more details on these steps in the next section.

As mentioned above, we were not able to apply clustering to all datasets. Therefore, we will now briefly present the data retrieved from Twitter.

C. Data

In order to query Twitter for the four different holidays, we prepared special queries. As there are hardly any differences in the naming, spelling and usage of these holidays in German, these queries are quite simple, see Table I.

In summary, we used population data have retrieved around 65,000 unique German tweets about different holidays. As we can see, Easter is by far the most present, while Shavuot receives very little attention. The overlap between Jewish and Christian holidays is also comparatively small.

We now move on to a more detailed analysis.

IV. Evaluation

For the analysis, the results for the individual feasts are discussed and then the texts in which the Jewish and Christian feasts are mentioned together are specifically discussed. In the qualitative textual analysis, the quantitative clustering (see Section II B) was further differentiated. The new clustering is based on a content categorization of individual tweets, especially in the festival months and the month before and after. Here, the method of conversation analysis [35], [36] and hermeneutic principles were applied.

This also depends on knowledge of Christianity and Judaism, especially basic knowledge of the Hebrew language and

Table I
DIFFERENT DATASETS USED FOR THE ANALYSIS AND THEIR SIZE.

Dataset	Query	Years	Tweets
Easter	“Ostern”	2018-2022	32,081
Pentecost	“Pfingsten”	2018-2022	22,254
Passah	“Passah” or “Pessach”	2018-2022	11,035
Shavuot	“Schawuot”	2018-2022	510
Pentacost / Schavuot	“Schawuot” and “Pfingsten”	2018-2022	92
Easter / Passah	“Ostern” and (“Passah” or “Pessach”)	2018-2022	817

knowledge of Jewish customs, mores and traditions. Therefore, a brief historical and theological context of the festivals is given first. Two research questions guided the study: (1) How do people talk about Christian holidays on Twitter; (2) How are Christian holidays related to or embedded in their Jewish counterparts? Is there an awareness of the Jewish roots of certain Christian holidays?

We also took a look at the other hashtags that appeared for each holiday (e.g. Easter 2018 hashtag). Here, randomly selected years were analyzed to identify peculiarities or patterns. In the next step, the results were summarized and theses, further hypotheses and questions were derived.

Finally, we reflected on the validity of the chosen mix of methods and evaluated our methodological approach. We articulate findings for further research, for Christian and Jewish communities, and provide suggestions for social actors seeking to prevent anti-Semitism.

A. Religious-historical and theological pre-location of the studied festivals

The two Christian feasts under consideration here, Easter and Pentecost, were originally Jewish feasts that were confirmed by Jesus Christ and his disciples on the one hand (aspects of continuity) and given new meanings on the other (aspects of discontinuity).

What Christians celebrate today at Easter is the resurrection of Jesus from the dead on the “first day of the week”, as mentioned by the evangelists (Bible: Mark 16:2, 9; Luke 24:1; John 20:1), after having been crucified according to Roman law (Good Friday).

According to Jewish reckoning, the first day of the week is the first day of creation. These events occurred exactly at the time of the Jewish festival of Passover, which most scholars believe took place in April 33 CE [37]. The Jewish festival of Passover commemorates the Exodus from Egypt: the journey from slavery to freedom.

As part of the Jewish festival of Passover (15th-21st of Nissan), the counting of the Omer (16th of Nissan, Reschith) begins, which lasts until the day after the seventh Shabbat, the 50th day, and ends with the festival of Shavuot. Other names for Shavuot include Pentecost (πεντηκοστὴ ἡμέρα, “fiftieth day”), Feast of Weeks, Feast of Harvest, Mattan Torah (Hebrew, “Gift of the Law”) in commemoration of the gift of the 10 Commandments and the Mosaic Law. Since the Jewish feast of Pentecost is a pilgrimage, the city of Jerusalem was traditionally filled with pilgrims. The disciples were in Jerusalem at Jesus’ command, awaiting “power from on high” (Luke 24:49), and on this feast of the commandments they now received the additional gift of the Holy Spirit. Apart from very different receptions in the past and present, believing Christians of the Eastern and Western Churches (Orthodox, Catholics, Protestants in churches and free churches) have always held to the bodily and historical evidence of the resurrection of Jesus, which followed the crucifixion. Since the Enlightenment, however, historical-critical theology has contributed to the secularization of churches and Western societies by subjectivizing, relativizing, and psychologizing the resurrection event. As a result, the theological and spiritual meaning of this great event was lost in many churches and thus in society.

Similarly, the Feast of Pentecost has been celebrated by Christians for over 2000 years, and Christians remember and ask again for the outpouring of God’s Spirit.

However, since the 4th century with the Constantinian turn, the Christian festivals were deliberately decoupled historically, theologically and also calendrically by imperial and episcopal legislative decisions of the emperor. The reasons for this were clearly anti-Semitic. The result was that Christians lost the Jewish references to the feasts until today, although since the Shoah a new awareness of the Jewish roots of the Christian faith has slowly emerged [38], [39], [40]. How deep is the understanding of these festivals and the connections between them – and how is this expressed in communication – is what we want to find out.

B. Easter

Figure 2 shows the total number of tweets related to “Easter” between 2018 and 2022. We see a clear increase of tweets with the start of the pandemic measures Easter 2020. An analysis of tweets shows an increased need for communication for several reasons. First, due to organizational issues. New information on restrictions for events, the contact restrictions, the date postponements and date changes, new information about alternative online offers (links) in terms of replacement events. We also see behavioral issues. Uncertainties and questions regarding travel behavior and celebration options and religious customs. In addition, we find emotional expressions of approval and disapproval, understanding and lack of understanding of policy restrictions (rules, measures, and punishments), fears and anger regarding pandemic relaxations or further restrictions, empathy for those affected by the lockdown of other religious communities

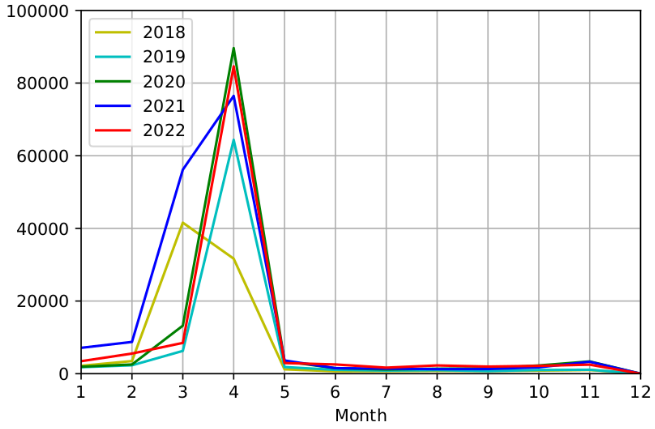


Figure 2. Total number of tweets related to “Easter” between 2018 and 2022. The peak clearly relates to the date of Easter in that year.

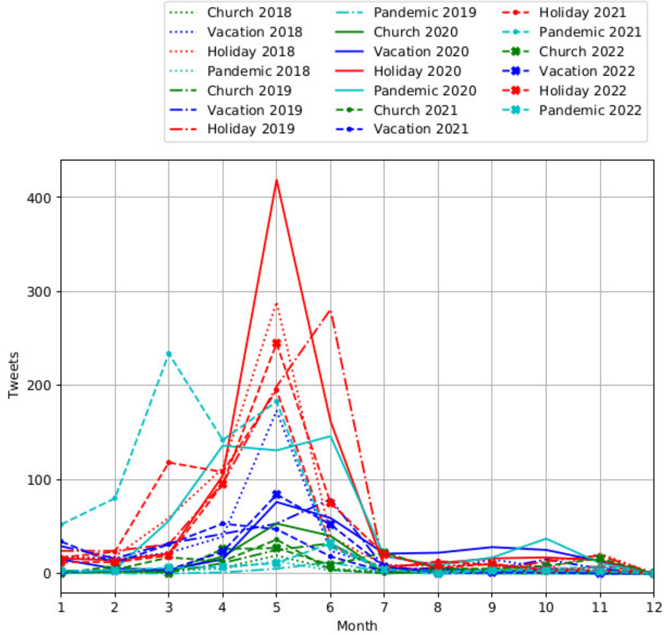


Figure 3. Total number of tweets related to “Easter” and showing the four clusters of interest during five years: Church, Vacation, Holiday, Pandemic. between 2018 and 2022. Interestingly, these topics are not aligned with the actual date of Easter in one particular year.

(e.g., Muslims Ramadan, Jews Passover). There is also an increased need for community and closeness in the phase of lockdowns, expressed through wishes, greetings, words of blessing, expressions of friendship (“I miss you ...”). However, we also see an increased interest in understanding theological and historical background of Easter celebration – often also interest (questions – answers) in context or comparison to other Christian celebrations or Jewish backgrounds. The increasing digitalization and the increased interest in Easter also in print media (e.g. newspapers) coincides with the above observations. An example is the analysis of the DWDS (Digital Dictionary of the German Language), which evaluates the text corpus of German-language newspapers. In 2020, the word course curve of the lexeme “Ostern” (Easter) showed a frequency of 20.69 per 1 million tokens[2] . We can provide some examples for explanatory, understanding, reflective tweets:

> 2018-04: ‘My thesis is that more people do not feel they belong to any religion, however, they believe in a higher power. This starts with the fact that many don’t even know the history behind Christmas, Easter, and Pentecost.’
>
> 2022-04: ‘so resurrection (Easter) is that after the crucifixion (Good Friday), then a few weeks later he goes again (Ascension) and still a little later the Holy Spirit comes (Pentecost).’

Other tweets are distant, derogatory (negative) or even faith-denying:

> 2019-04: ‘Easter is over ¿ soon comes Pentecost.... Religious people (Christians now using the example) really have something to celebrate all year round.... (We #atheists / #agnostics, on the other hand, look for substitute events to look forward to - they’re still not #substitutegods, are they?)’
>
> 2021-04:‘14 days after Ramadan: the numbers are finally receding. But then comes Pentecost with mass baptisms and singing in the free churches. We have already had experience with Easter and Christmas. Religion as a superspreader. Now we finally know what it (religion) is really good for.’

However, we also find examples for politicization of terms and language used in faiths:

> 2021-04: ‘50 days. This is also the period between Easter and Pentecost. The decisive MPC at the beginning of March could have been the resurrection and the day today could have been the outpouring of the Holy Spirit, the realization that it can work if you listen to science.’
>
> 2022-04: ‘@Karl_Lauterbach @haintz_markus Also happy Easter #LauterbachRuecktritt ! Now only the Holy Spirit has to enlighten them on Pentecost and convince them to resign’

As discussed above, several tweets are also concerning the explanation of culture change or showing incomprehension:

> 2019-04: ‘On the other hand, I would also like to see those who say that on Easter, Christmas and Ascension Day as well as Pentecost should be worked. If we limit the privileges of the churches, then we also limit our privileges that result from the church.’
>
> 2022-04:‘Why don’t Jews and Muslims just move their holidays to Easter, Pentecost, etc.? Wouldn’t that be the easiest solution?’

[2] See https://www.dwds.de/r/plot/?view=1&corpus=zeitungenxl&norm=date\%2Bclass&smooth=spline&genres=0&grand=1&slice=1&prune=0&window=3&wbase=0&logavg=0&logscale=0&xrange=1946\%3A2022&q1=Ostern.

> 2022-04: ‘The holidays already make no sense because no one knows what is actually the reason for the holiday. What percentage of the population can say what happened on Corpus Christi, Pentecost? And the main holidays are also just Santa Claus and Easter Bunny.’

So there is basically explanatory talk about the festivals, but there is also negative and derisive talk, especially because the church and faith itself are no longer perceived as credible. In addition, there is always the interreligious aspect, not only towards Judaism, but also towards Muslim festivals. But there are also secular and rejectionist tendencies, such as the shifting of festivals.

In Figure 3 we show the results of the cluster analysis described in the previous section. The events around Corona strongly influence the communication at Easter and dominate 2020 and 2021, which is not surprising since the Corona-related changes (restrictions) at the festivals were massive. However, we also see a religious consciousness that still dominates Easter. In the first year of the pandemic, 2020, Easter is the dominant factor, with only a small increase in attention to the church. According to the tweets, a marginalization of the church can be observed before 2020, while from 2020 onwards, the festivals enjoy a high level of attention, but the church as an institution does not correlate with this. As with Pentecost, the holiday plays a very large role in 2018, while it plays a rather subordinate role in 2019 (even before the pandemic).

C. *Pentecost*

As discussed above, we analyze tweets related to a particular holiday by year and month. For a first overview of the total number of tweets per month, we refer to Figure 4. This clearly shows that the tweets are related to the date of Pentecost, underlining that there is little background noise in the data.

The K-means clustering approach was performed on $K = 7$ clusters, see Figure 6 for a PCA plot of these clusters. We will present some more details on some of the clusters obtained. In Figure 7 we show a word cloud for cluster 7, which is mostly related to holidays. However, we still see “Corona” as a central theme, while other words refer to traffic jams, traffic and holiday destinations. This underlines that some topics can be identified quite well. Other topics are more difficult: In Figure 8 we show the word cloud for a cluster related to “Church”. However, there are also other topics in this cluster: the pandemic and holidays. Therefore, a closer look at the tweets themselves was necessary to identify the underlying topic, and clustering could not be used without supervision.

For a more detailed analysis, we group tweets with hashtags into different categories related to church, vacation, holidays and pandemic, see figure 5 for details. Clearly, the number of tweets is still related to the date of Pentecost. However, few tweets use hashtags and are therefore available for categorization. While holidays are important in 2018 and 2022, the pandemic is the most prominent topic in 2020 and 2021. This again shows that the data is not too noisy. What is striking, however, is that the term is usually not mentioned in connection with churches or religious issues.

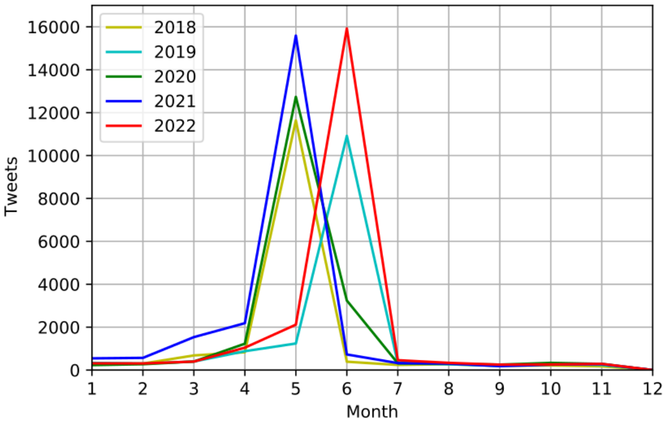


Figure 4. Total number of tweets related to “Pentecost” between 2018 and 2022.

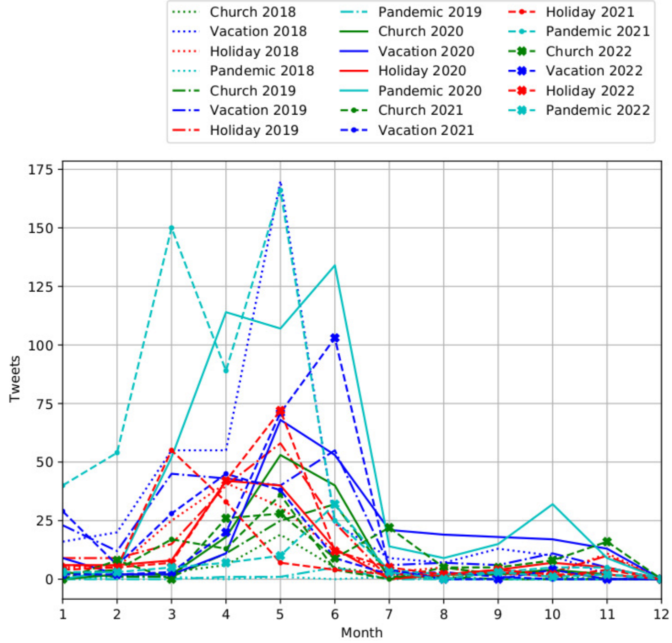


Figure 5. Total number of tweets related to “Pentecost” and showing the four clusters of interest during five years: Church, Vacation, Holiday, Pandemic. between 2018 and 2022.

There are a number of important observations. Pandemic topics dominate around the time of the Pentecost holidays in 2020 and 2021, which is understandable since the restrictions strongly influenced or limited holiday behavior. Tweets mentioning churches spike in 2020 - the first pandemic year with lockdowns and massive restrictions on worship. However, given the number of church restrictions during this time, the curve is rather flat, which may indicate the writers’ comparatively low interest in church. There is an increasing interest in holidays over the study period. Can we infer an increased awareness of the religious significance of these holidays? In some tweets there seems to be a basic understanding that

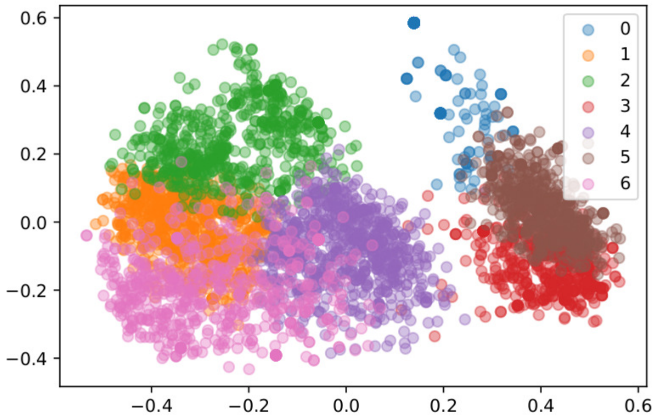


Figure 6. PCA plot of K-means clustering for tweets related to "Pentecost" with $K = 7$.

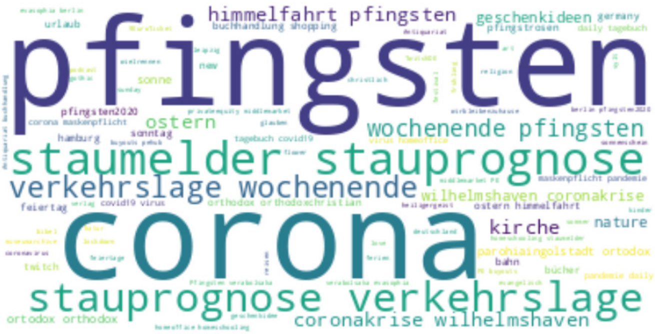


Figure 7. Wordcloud for Cluster 7, mostly related to vacation.

Pentecost has something to do with the "Spirit", even if the hit rate for "Holy Spirit" is almost zero:

> 2022-06 'I still don't know what is celebrated on Pentecost, Ascension Day and Corpus Christi. Something about a holy spirit or something? I do not know.'

However, the comments, explanations and questions about the content are rather low in 2018 and 2019. Similar to our previous analysis we find several negative tweets:

> 2021-05: 'Question at the Pentecost Walk: "Are Christians celebrating on Pentecost that they invented something as 'Spirit-rich' as the Church?" — "No, they celebrate that God's Spirit is still there. In spite of this church.'

As with Easter, it is striking that the Spirit is spoken of primarily as a metaphor in a political and social context. Spirit stands as a symbol for "having the right insights, making the right decisions, doing the right things. These metaphors are used almost exclusively in the negative sense, meaning the absence of spirit. Criticism of people or parties, organizations or groups is usually made with a sarcastic undertone and is emotionally charged.

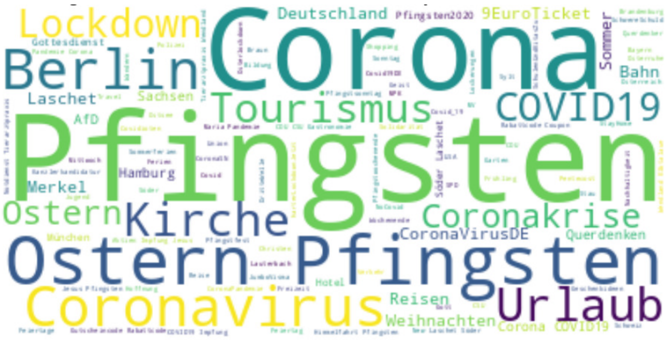


Figure 8. Wordcloud for Cluster 2, related to church but also covering different topics like the pandemic and vacation.

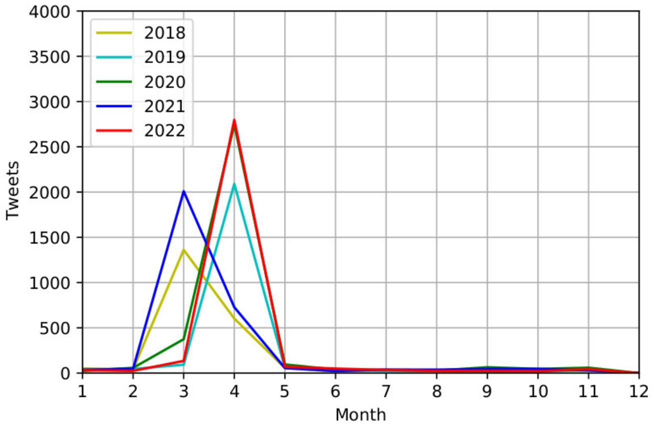


Figure 9. Total number of tweets related to "Passah" between 2018 and 2022.

D. Passah

For a first overview of the total number of tweets per month, we refer to Figure 9. Again, this shows that the tweets are related to the date of Passah, underlining that there is little background noise in the data. We find a significant higher number of tweets related to Passah compared to Shavuot, which underlines the importance of this festival compared with other Jewish festivals within the German-speaking part of Twitter. The total number of almost 3,000 German Tweets is high – compared to the Jewish population in German-speaking countries (about 0,1%).

However, a more detailed cluster analysis, was not possible. A detailed analysis of tweets showed that they are mostly connected to religious topics. For example, 69 tweets were connected to Ramadan. Having both festivals at the same time is a challenge for Jews, as the Ramadan period is known for unrest, especially in Israel and Jerusalem. Corresponding comments and prayers for peace/wishes for political unrest, e.g. on the Temple Mount. In addition, wishes for a happy holiday are the most common, but Easter and Ramadan are often mentioned as well. Another block is information about events and tweets that actively deal with customs, e.g. 'Are you allowed to drink beer on Passover or do you have to dispose

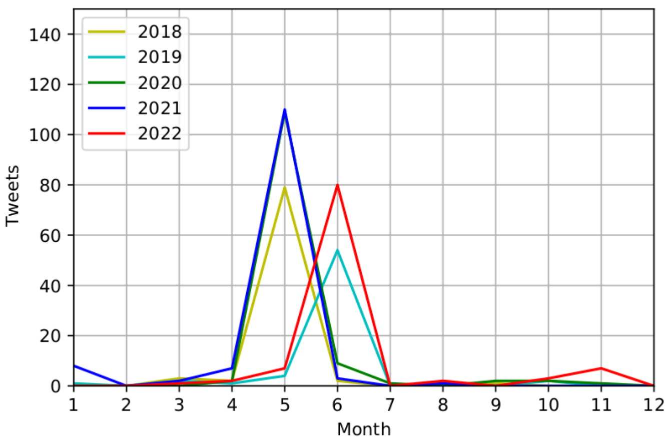


Figure 10. Total number of tweets related to “Shavuot” between 2018 and 2022.

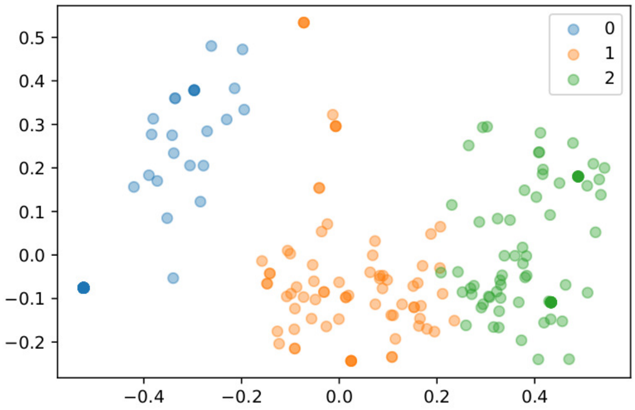


Figure 11. PCA plot of K-means clustering for tweets related to “Shavuot” with $K = 3$.

of it beforehand’ (2022-02 12/10).

E. Shavuot

For a first overview of the total number of tweets per month, we refer to Figure 10. Again, this shows that the tweets are related to the date of Shavuot, underlining that there is little background noise in the data. Since the pandemic, there have been significantly more mentions of the festival. All of the tweets are highly religiously charged and almost all of the terms have a reference to spiritual, religious themes related to the festival. This clearly anchors Shawuot as a spiritual festival for individuals.

However, a more detailed cluster analysis was not possible. In Figure 11 we show a PCA plot for K-means clustering for tweets related to “Shavuot” with $K = 3$. It is almost impossible to distinguish between the predefined topics and it shows again that clustering cannot be used without supervision. It also shows that this approach is difficult to apply to small datasets.

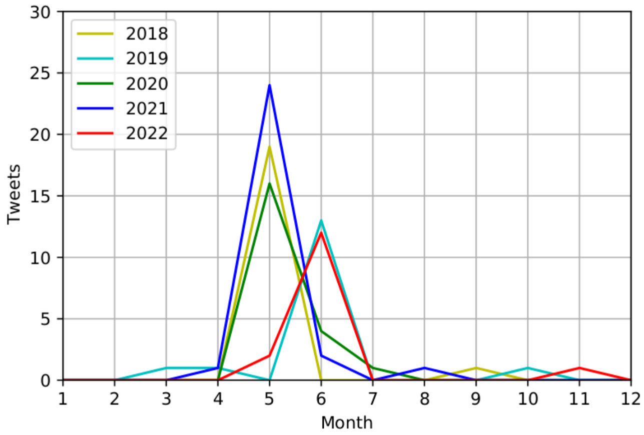


Figure 12. Total number of tweets related to “Pentacost / Schavuot” between 2018 and 2022.

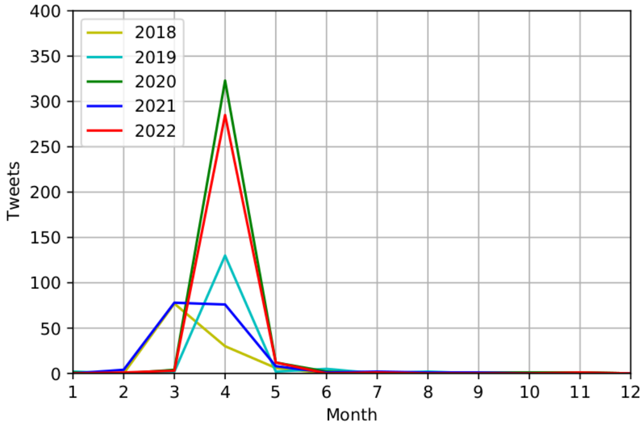


Figure 13. Total number of tweets related to “Easter / Passah” between 2018 and 2022.

While tweets referring to Passah showed that they are mostly connected to religious topics, for Shavuot we find much more information on events, links to webpages and several tweets explaining cultural issues. For example:

> 2022-06 77: ‘On Shavuot, Jews celebrate receiving the Torah at Mount Sinai. This is the most important event in Jewish history, but the holiday is rather unknown. And what is it about the cheesecake?’

F. Connection to Jewish holidays

In Figure 12 we show the total number of tweets related to “Pentacost / Schavuot” between 2018 and 2022.

In 2018, 2019, and 2022, the Christian and Jewish holidays of Pentecost/Shavuot fell on the same calendar days. However, the influence on the dates is not obvious. What is evident, however, is a significant increase in 2018, 2020 and 2021, when both are referenced.

In Figure 13 we show the total number of tweets related to “Easter / Passah” between 2018 and 2022. Again, we see

Table II
SUMMARY OF PARTICULAR TOPICS IDENTIFIED WITHIN DIFFERENT HOLIDAYS. ALTHOUGH ALL TOPICS WERE FOUND, SOME WERE MORE IMPORTANT THAN OTHERS. THESE ARE MARKED WITH AN "X".

#	Topic	Easter	Pentecost	Passah	Shavuot	Pentacost / Schavuot	Easter / Passah
1	Explanatory	X	X	X	X	X	X
2	Faith-based					X	X
3	Distance	X	X				
4	Critics	X	X				
5	Metaphorical/ political use	X	X	X	X		
6	Incomprehension	X	X	X	X	X	X
7	Hate speech						
8	Pandemic	X	X	X	X	X	X

an increasing interest in the combination of both festivals in 2020 and 2022.

Comparing these results to our previous analysis clearly shows that there are less references to Jewish Holidays, and very little references to them and the corresponding Christian Holidays.

In spite of all this, we can take two questions as an interim conclusion for further discussion: First, it seems that the Corona period made people more curious and interested in the connections between Judaism and Christianity – maybe according to the motto: what I'm not allowed to do, makes me curious.. In addition, the tweets seem to indicate that there is currently more awareness of religious festivals, after participation was significantly hampered during the pandemic. But we have also to consider a possible artifact here, because we have to take into account the Jewish Orthodox customs: When the feast-day is a Shabbat, many pious Jews do not unse a mobile phone or electronic equipment.

G. Relevance for Christian Churches

In Table II we summarise our findings from the previous sections. The cluster analysis was helpful in those cases where it could be carried out. Horizontally, however, we were able to identify eight themes that were particularly helpful in answering our question. Although all themes were found, some were more important than others, as highlighted in this table. Most interestingly, we found very little evidence of hate speech. However we do not know, if Twitter may have already deleted those tweets, violating their rules and regulations. Critical and distant remarks are found especially for Christian holidays. In addition, we find both explanatory and non-explanatory tweets in all cases.

Several points are relevant for Christian churches. First, there is a lack of knowledge about the biblical background and customs of the festivals. But also the connection to the Jewish roots of the festivals is rarely recognized. Thus, there is a lack of both theology and practice. In order to promote interreligious dialogue and understanding of their own festivals, churches need to reawaken awareness of the Jewish roots and clearly anchor them in doctrine, liturgy and customs. Where the reference to Judaism is clear (e.g. through the use of Jewish festival names), theological interest and knowledge is deeper and greater – and so is interreligious interest and understanding – or the desire and curiosity to understand.

Where Jewish references are clearly recognizable, references to Islam (Ramadan wishes) or understanding of the problems of the various religious festivals in corona measures and pandemic-related restrictions are also often evident. Of fundamental interest, as discussed above, is the phase of pandemic restrictions. This has led to increased communication about festivals, e.g. 2020. However, this must be seen in the context of the increased church departures in the 2nd and 3rd years. There are broader questions to be answered here, but they are not the focus of this paper.

Often, however, the use of core theological terms is merely a stepping stone into metaphorical language, usually for cynical or sarcastic remarks about the state or the church. Another question is whether there is a danger that in a secularized world theological content will have only a symbolic or linguistic effect. Is this a consequence of a liberal theology that no longer regards many biblical events as historical and thus demands a reduction to the metaphorical level?

V. CONCLUSIONS AND OUTLOOK

Our two main research questions were:

- Can we identify how people talk about holidays and especially Christian holidays on Twitter? (RQ1)
- If we find relevant information: Can we see how Christian holidays are related to or embedded in their Jewish counterparts? Is there an awareness of the Jewish roots of certain Christian holidays? (RQ2)

In this work, we could answer RQ1: Yes, we can identify how people talk about holidays. However, distinguishing between church-related and secular topics was not at hand. We could show that there is a lack of knowledge about the biblical and theological background and customs of the festivals. But also the connection to the Jewish roots of the festivals is rarely recognized. Thus, we could also answer RQ2: Where we could identify the reference to Judaism – which was mainly done through the use of Jewish festival names – theological interest and knowledge is deeper and greater. We find at least a small interest in interreligious exchange and understanding. It was interesting, that, where Jewish references are clearly recognizable, references to Islam or understanding of the problems of the various religious festivals in corona measures and pandemic-related restrictions are also often evident.

There were a number of methodological challenges. First, there is the large amount of data. Classical qualitative approaches provide a first orientation, but also show the need for quantitative work. In this step, the work could and had to be done classically by keyword searches on selected terms such as Acts, disciples, speaking in tongues, Holy Spirit, Spirit, glossalia, and so on. Some aspects had to be counted by hand, because terms have to be assigned and interpreted, e.g. with different spellings or with spelling mistakes. While K-means in Twitter data gains some insight, in another iteration of our research we plan to make use of the full text body for deriving alternative clusters.

Another aspect that arises is the question of the authors of the tweets, which, unlike the qualitative work, was not considered in this work. But who writes the tweets? Apart from private individuals, interreligious organization organisations such as the "Israel Network" or ecumenical organisations ("House of One") should be mentioned. Further analysis would have to provide more clarity here. However, this would require in-depth knowledge of Christian churches, free churches and Jewish and Islamic organisations. Obviously, the use of computational social science methods in practical theology poses new challenges, but also makes new data available for research. In particular, the methodological overlap between classical quantitative and qualitative research is small, and thus our work is also a plea for more interdisciplinary exchange. And beyond this: Actors of churches and culture should promote an awareness of the historical, theological and cultural contexts of the faith communities, especially the festivals. Because this creates a bond between people, which can be a small contribution to peace in this divided world.

References

[1] A. Matamoros-Fernández and J. Farkas, "Racism, hate speech, and social media: A systematic review and critique," *Television & New Media*, vol. 22, no. 2, pp. 205–224, 2021.

[2] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the international AAAI conference on web and social media*, vol. 11, no. 1, 2017, pp. 512–515.

[3] I. Kalmar, C. Stevens, and N. Worby, "Twitter, gab, and racism: The case of the soros myth," in *proceedings of the 9th International Conference on Social Media and Society*, 2018, pp. 330–334.

[4] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[5] S. Kumar, F. Morstatter, and H. Liu, *Twitter data analytics*. Springer, 2014.

[6] H. Anber, A. Salah, and A. Abd El-Aziz, "A literature review on twitter data analysis," *International Journal of Computer and Electrical Engineering*, vol. 8, no. 3, p. 241, 2016.

[7] E. Bokányi, D. Kondor, L. Dobos, T. Sebők, J. Stéger, I. Csabai, and G. Vattay, "Race, religion and the city: twitter word frequency patterns reveal dominant demographic dimensions in the united states," *Palgrave Communications*, vol. 2, no. 1, pp. 1–9, 2016.

[8] A. Karami, M. Lundy, F. Webb, and Y. K. Dwivedi, "Twitter and research: A systematic literature review through text mining," *IEEE access*, vol. 8, pp. 67 698–67 717, 2020.

[9] L. Sinnenberg, A. M. Buttenheim, K. Padrez, C. Mancheno, L. Ungar, and R. M. Merchant, "Twitter as a tool for health research: a systematic review," *American journal of public health*, vol. 107, no. 1, pp. e1–e8, 2017.

[10] N. Joseph, A. K. Kar, P. V. Ilavarasan, and S. Ganesh, "Review of discussions on internet of things (iot): insights from twitter analytics," *Journal of Global Information Management (JGIM)*, vol. 25, no. 2, pp. 38–51, 2017.

[11] M. Martínez-Rojas, M. del Carmen Pardo-Ferreira, and J. C. Rubio-Romero, "Twitter as a tool for the management and analysis of emergency situations: A systematic literature review," *International Journal of Information Management*, vol. 43, pp. 196–208, 2018.

[12] A.-K. Jung, S. Clausen, A. S. Franzke, J. Marx *et al.*, "'cambridge moralica'-towards an ethical framework for social media analytics," *Australasian Journal of Information Systems*, vol. 26, 2022.

[13] A. Sulfikar, P. Kerkhof, and M. Tanis, "Tweeting for religion: How indonesian islamic fundamentalist organizations use twitter," *Journal of Media and Religion*, vol. 22, no. 1, pp. 1–16, 2023.

[14] A.-P. Cooper, E. A. Kolog, and E. Sutinen, "Exploring the use of machine learning to automate the qualitative coding of church-related tweets," *Fieldwork in Religion*, vol. 14, no. 2, pp. 140–159, 2019.

[15] S. Woodward and R. Kimmons, "Religious implications of social media in education," *Religion & Education*, vol. 46, no. 2, pp. 271–293, 2019.

[16] A.-P. Cooper, "Using geotagged twitter data to uncover hidden church populations," in *The Desecularisation of the City*. Routledge, 2018, pp. 134–147.

[17] ——, "Assessing the possible relationship between the sentiment of church-related tweets and church growth," *Studies in Religion/Sciences Religieuses*, vol. 46, no. 1, pp. 37–49, 2017.

[18] M. Aeschbach and D. Lüddeckens, "Religion on twitter: Communalization in event-based hashtag discourses," *Online Heidelberg Journal of Religions on the Internet*, vol. 14, pp. 108–130, 2019.

[19] K. Crockett, D. Mclean, A. Latham, and N. Alnajran, "Cluster analysis of twitter data: A review of algorithms," in *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*, vol. 2. Science and Technology Publications (SCITEPRESS)/Springer Books, 2017, pp. 239–249.

[20] S. Ahuja and G. Dubey, "Clustering and sentiment analysis on twitter data," in *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*. IEEE, 2017, pp. 1–5.

[21] E. Baralis, T. Cerquitelli, S. Chiusano, L. Grimaudo, and X. Xiao, "Analysis of twitter data using a multiple-level clustering strategy," in *Model and Data Engineering: Third International Conference, MEDI 2013, Amantea, Italy, September 25-27, 2013. Proceedings 3*. Springer, 2013, pp. 13–24.

[22] V. Gupta and R. Hewett, "Real-time tweet analytics using hybrid hashtags on twitter big data streams," *Information*, vol. 11, no. 7, p. 341, 2020.

[23] J. Pflugmacher, S. Escher, J. Reubold, and T. Strufe, "The german-speaking twitter community reference data set," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 1172–1177.

[24] B. Witzenberger and J. Pfeffer, "Gender dynamics of german journalists on twitter," in *2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2022, pp. 226–230.

[25] L. Neudert, B. Kollanyi, and P. Howard, "Junk news and bots during the german parliamentary election: What are german voters sharing over twitter?" 2017.

[26] M. J. Cumbraos-Sánchez, R. Hermoso, D. Iñiguez, J. R. Paño-Pardo, M. Á. A. Bandres, and M. P. L. Martinez, "Qualitative and quantitative evaluation of the use of twitter as a tool of antimicrobial stewardship," *International journal of medical informatics*, vol. 131, p. 103955, 2019.

[27] J. Einspänner, M. Dang-Anh, and C. Thimm, "Computer-assisted content analysis of twitter data," 2014.

[28] C. Salvatore, S. Biffignandi, and A. Bianchi, "Social media and twitter data quality for new social indicators," *Social indicators research*, vol. 156, pp. 601–630, 2021.

[29] I. Dongo, Y. Cardinale, A. Aguilera, F. Martinez, Y. Quintero, G. Robayo, and D. Cabeza, "A qualitative and quantitative comparison between web scraping and api methods for twitter credibility analysis," *International Journal of Web Information Systems*, vol. 17, no. 6, pp. 580–606, 2021.

[30] L. Vidal, G. Ares, L. Machín, and S. R. Jaeger, "Using twitter data for food-related consumer research: A case study on "what people say when tweeting about different eating situations"," *Food Quality and Preference*, vol. 45, pp. 58–69, 2015.

[31] H. Hirschmann, *Korpuslinguistik: Eine Einführung*. J.B. Metzler, 2019.

[32] S. Mason and L. Singh, "Reporting and discoverability of "tweets" quoted in published scholarship: current practice and ethical implications," *Research Ethics*, vol. 18, no. 2, pp. 93–113, 2022.

[33] I. Gurevych and H. Niederlich, “Computing semantic relatedness of germanet concepts,” in *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen: Proceedings of Workshop” Applications of GermaNet II” at GLDV*, 2005, pp. 462–474.

[34] Y. Feng, E. Bagheri, F. Ensan, and J. Jovanovic, “The state of the art in semantic relatedness: a framework for comparison,” *The Knowledge Engineering Review*, vol. 32, p. e10, 2017.

[35] P. Ten Have, “Doing conversation analysis,” *Doing Conversation Analysis*, pp. 1–264, 2007.

[36] J. Bergmann, “Das konzept der konversationsanalyse,” *Text-und Gesprächslinguistik*, vol. 2, pp. 919–926, 2001.

[37] D. A. Carson and D. J. Moo, *An introduction to the New Testament*. Zondervan Academic, 2009.

[38] G. Höver and S. Mosès, *In Verantwortung vor der Geschichte: Besinnung auf die jüdischen Wurzeln des Christentums*. Borengässer, 1999.

[39] G. Baltes, *Die verborgene Theologie der Evangelien: Die jüdischen Feste als Schlüssel zur Botschaft Jesu*. Francke, 2020.

[40] A. Gerdmar, *Roots of Theological Anti-Semitism (paperback): German Biblical Interpretation and the Jews, from Herder and Semler to Kittel and Bultmann*. Brill, 2008, vol. 20.

DOI: 10.15439/2023F695
ISSN 2300-5963 ACSIS, Vol. 35

Is Homomorphic Encryption Feasible for Smart Mobility?

Anika Hannemann
Dept. of Computer Science, Leipzig University
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany
Email: anika.hannemann@informatik.uni-leipzig.de

Erik Buchmann
Dept. of Computer Science, Leipzig University
Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany
Email: buchmann@informatik.uni-leipzig.de

***Abstract*—Smart mobility is a promising approach to meet urban transport needs in an environmentally and and user-friendly way. Smart mobility computes itineraries with multiple means of transportation, e.g., trams, rental bikes or electric scooters, according to customer preferences. A mobility platform cares for reservations, connecting transports, invoicing and billing. This requires sharing sensible personal data with multiple parties, and puts data privacy at risk.**

In this paper, we investigate if fully homomorphic encryption (FHE) can be applied in practice to mitigate such privacy issues. FHE allows to calculate on encrypted data, without having to decrypt it first. We implemented three typical distributed computations in a smart mobility scenario with SEAL, a recent programming library for FHE. With this implementation, we have measured memory consumption and execution times for three variants of distributed transactions, that are representative for a wide range of smart mobility tasks. Our evaluation shows, that FHE is indeed applicable to smart mobility: With today's processing capabilities, state-of-the-art FHE increases a smart mobility transaction by about 100 milliseconds and less than 3 microcents.

I. Introduction

GROWING cities, urban sprawls and environmental concerns demand for mobility concepts [1], [2] that go beyond individual cars [3]. A promising approach is smart mobility, also known as multi-modal mobility or intelligent mobility. It refers to the integration of advanced technologies and intelligent systems into transportation networks to improve efficiency, safety, and sustainability, to lower emissions and to enhance the overall quality of urban life. To this end, smart mobility encompasses solutions from the area of cloud computing, machine learning and artificial intelligence, that optimize the movement of people and goods within urban areas.

With smart mobility, customers can specify a mobility demand and travel preferences [4], e.g., the shortest, fastest, most inexpensive or eco-friendliest route from a starting point to a destination, or the route with the least transfers between the means of transportation. A cloud-based mobility platform [5] then lets the customer select an itinerary among some alternatives [6], [7], [8], [9], and connects to all services needed to process and settle the trip.

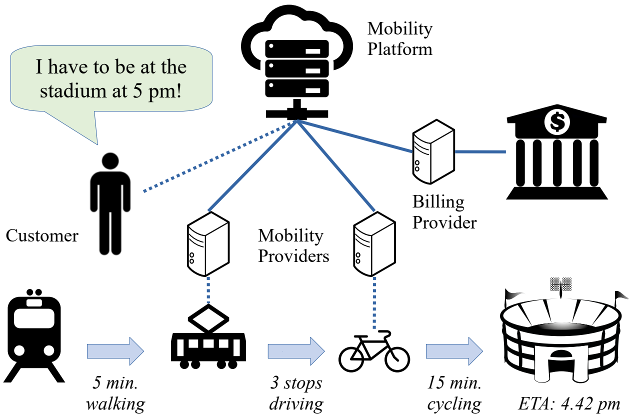


Fig. 1. Smart Mobility Scenario

***Example**: A customer might want to go from a train station to a stadium and must arrive at the beginning of a game. The mobility platform queries the databases of the connected mobility providers, and suggests three options that allow the customer to reach the destination in time: Based on an assessment of environmental impact, the most eco-friendly option is to use a tram to travel three stops to a rental bike station, as illustrated in Figure 1. As an alternative, a customer could opt to take a bus for six stops and use an electric scooter for the rest of the way. The most comfortable but expensive option would be hiring a cab. Once the customer has selected one option, the platform makes reservations. It collects the recorded distances and stations traveled. After the trip, the platform connects to a billing provider that handles invoicing and payment.*

Smart mobility approaches raise numerous privacy concerns [10], [11], [12]: Start and end of a route can reveal personal needs, e.g., if it is a church, hospital or event location. It might be possible to identify an individual by recurring ways home from work and vice versa. If customers frequently travel together, this might indicate personal relationships. If a rental sports car is preferred over a suburb train, this might tell about preferences and wealth. Travel times and frequent routes reveal habits, employment status or daily routines. What makes privacy issues even more challenging is that personal data is distributed among many parties [13], such as providers for mobility and infrastructure, and various services for payment, demand forecasting, parking space management, etc.

Existing privacy-aware smart mobility approaches make use of anonymization, e.g. by using differential privacy [14], [15], [16] or by reducing the data resolution [17], [18], [19]. Alternatively, secure multi-party computation can be used [20], [21]. This induces noise to the data and/or requires multiple rounds of computation among several parties, i.e., it reduces accuracy, efficiency and, therefore, user experience.

We investigate if fully homomorphic encryption [22] (FHE) can be an alternative. FHE allows to calculate sums or products on encrypted data without having to decrypt the data first. The decryption then provides the exact, noise-free result of the calculation. This makes FHE a natural choice for calculations with privacy-related data. However, some calculations increase the size of the data and/or make encryption, calculations or decryption computationally expensive. Some years ago, this limited the practical applicability of FHE in any real setting.

However, vast advances in FHE programming libraries such as SEAL [23], HElib [24] or OpenFHE [25], new paradigms such as edge computing, and a huge computing power available at little costs both at edge nodes and cloud services give reasons for new analyses. For example, in May 2023, a cloud instance with 128 XEON CPU cores at 3.5 GHz and 512 GiB RAM and 50 Gbps network bandwith costs only 8 USD per hour. Our concern is to find out whether the overhead of FHE for typical smart mobility transactions is reasonable to support privacy-compliant business models in this application domain.

In this paper, we make the following contributions:

- We identify three distributed transactions that are representative for a smart mobility scenario and benefit from FHE, i.e., require noise-free results and cannot be readily secured by simpler means such as one-time pseudonyms [18].
- We implement a prototype based on Microsoft SEAL [23], which uses the state-of-the-art FHE schemes BGV [26], BFV [27] and CKKS [28].
- We measure memory consumption and execution times, and we compare them with the resources available in the cloud or on a smartphone.

Our evaluation shows that with CKKS, encrypted transactions add approx. 100 ms to the CPU time of unencrypted transactions. This does not impact user experience [29]. With parallel processing, this time can be reduced, and it costs less than 3 microcents on a current cloud instance. Thus, we have confirmed that FHE is indeed feasible for smart mobility business models, where such transaction fees are several orders of magnitude smaller than the billing amount on the customer's invoice, but privacy is an important factor.

Paper structure: The next section reviews related work. In Section III, we derive our smart mobility transactions with fully homomorphic encryption. Section IV contains an experimental evaluation, and Section V concludes.

II. Related Work

In this section, we explain the state-of-the-art in (fully) homomorphic encryption, and we briefly review smart mobility approaches.

A. Homomorphic Encryption

Homomorphic Encryption (HE) is a well-established technique that enables third-party computation on encrypted data, without requiring the data to be decrypted beforehand. HE allows for data to be encrypted while keeping the features of the function and format of the encrypted data, supporting privacy-preserving data processing. Although this property of HE is known already for over than 30 years, the first plausible FHE approach was proposed by Gentry et al. in 2009 [30]. However, HE is costly in terms of computation and is, therefore, still subject of ongoing research [22].

The homomorphic property of HE allows certain operations to be computed over the encrypted data, with the resulting values also being encrypted. For two messages $\forall m_1, m_2 \in \mathcal{M}$ of a message space $\mathcal{M}$, Eq. 1 shows an HE scheme that supports any operation on their respective ciphers c_1, c_2. In the context of a public-key cryptosystem, the public key is denoted as k_e, the private key as k_d, and the encryption and decryption functions as Enc and Dec, respectively.

$$\begin{aligned} c_1 = Enc(k_e, m_1), \quad c_2 = Enc(k_e, m_2) \\ m_1 \oplus m_2 = Dec(k_d, c_1 \oplus c_2) \end{aligned} \tag{1}$$

Formally, a homomorphic encryption scheme is a quadrupel HE=($KeyGen$, Enc, Dec, $Eval$) with $KeyGen$, Enc, Dec and $Eval$ being probabilistic polynomial time algorithms:

KeyGen generates a public key k_e, a private key k_d and an evaluation key k_{eval} given some security parameter λ for the asymmetric version of HE: $KeyGen(1^{\lambda}) \rightarrow (k_e, k_d, k_{eval})$
For the symmetric version, only a secret key k_d and an evaluation key k_{eval} are created.

Enc encrypts a plaintext message $m \in \mathbb{Z}_n$ to a ciphertext c using the public key, which is shared: $Enc(k_e, m) \rightarrow c$

Dec uses the private key, which is kept secret, to decrypt a ciphertext c to a plaintext message m: $Dec(k_d, c) \rightarrow m$

Eval applies an operation $f : \mathbb{Z}_n^l \rightarrow \mathbb{Z}_n$ to a given ciphertext $c_1, ..., c_l$ and outputs a ciphertext c_f using the evaluation key k_{eval}: $Eval(k_{eval}, f, c_1, \cdots, c_l) \rightarrow c_f$
k_{eval} is generated uniquely for every computation and, therefore, does not pose a privacy threat. With $Eval$, homomorphism of the scheme can be proven.

B. Fully Homomorphic Encryption

Depending on the support of the operations applied in the $Eval$ function, HE can be categorized into **fully homomorphic encryption** (FHE), **partially homomorphic encryption** (PHE) and **somewhat homomorphic encryption** (SHE), each of them with different limitations and capabilities. PHE allows $Eval$ for one operation $\oplus$, either addition of multiplication, for an unlimited number of times. SHE allows both addition and multiplication, but with a limited number of operations due to the increasing size of the ciphertext. FHE allows an unlimited number of operations $\oplus$ for an unlimited number of times. Addition and multiplication operations as well as comparison and branching are supported. Therefore, FHE is the most powerful approach of HE and, therefore, implemented in this work.

FHE is a form of ring homomorphism with structure preserving characteristics [30]. This allows for arbitrary computations to be performed, as the homomorphic properties of the ring ensure that the results of the computations can be obtained without requiring decryption. Which operation is allowed depends on the FHE scheme; in this work the well known Brakerski-Fan-Vercauteren (BFV) [27], Brakerski-Gentry-Vaikuntanathan (BGV) [26] and Cheon-Kim-Kim-Song (CKKS) [28] schemes are implemented and evaluated. They are based on on the hardness of the (Ring) Learning With Errors (RLWE) problem. Learning with Errors is considered to be one of the hardest, post-quantum problems to solve in polynomial time: Given (x, y) where $y = f(x)$ for some linear function f, f can be easily learned. Now, when adding errors to the algorithm's input such that $y \neq f(x)$ for a small probability, it is assumed that the problem can not be solved in polynomial time and is, therefore, hard [31].

A subproblem is the Ring Learning with Errors (RLWE), an extension of the LWE problem for polynomial rings over finite fields. A major advantage of RLWE is the key size: While the private and public keys of LWE-based cryptography can become large, RLWE-based keys are roughly the square root of LWE [32].

For FHE, there are methods for maintaining the ciphertext, without modifying the message, such as bootstrapping and relinearization. In bootstrapping, the evaluation key k_{eval} is used to control noise. Thus, any number of FHE operations can now be computed without noise becoming uncontrollable. In this context, noise refers to a measure to prevent unauthorized decryption of encrypted data using the secret key, and it does not affect the precision of the computation outcome. Relinearization handles a common problem of RLWE-based FHE, whose ciphertext sizes increases with every homomorphic multiplication. During homomorphic evaluation, relinearization limits the expansion of the ciphertext to prevent high computation costs.

The variety of supported operations allow for a wide range of computations to be performed on encrypted data, making FHE powerful and versatile and applicable in multiple settings. In Cloud Computing, FHE is used to protect the client's data privacy to process them on an external party [33], [34]. The line of FHE works on Machine Learning aim to protect the training data's privacy in either a collaborative setting [35], [36] or a federated learning setting [37], [38]. Another relevant application specifically for this work is private fog computing for the Internet-of-Things (IoT) which enables multiple users to authenticate and aggregate data collected with edge devices [39], [40]. Another work at the intersection of the previously mentioned areas is the work of Zhang et al. with an approach to privacy preserving federated learning with IoT-enabled healthcare system [41].

C. Smart Mobility

The use of IoT technologies has proven to be an appropriate response to the growth of cities and the associated impact on traffic and transportation. It has brought up the concept of smart mobility, which refers to the optimal combination of various modes of transportation, including e-bikes, e-rollers, buses, shared cars, tramways, and trains, as well as infrastructure components such as roads, bridges, airports, and train stations. As transportation modes continue to grow and become more interconnected, the resulting complexity can make it increasingly challenging to efficiently use and combine available options. To tackle this problem, relevant related work has been done concerning urban mobility and multi-modal routing planning. [6], [7], [8], [9], [42] proposed a mobile recommender system for personalized multi-modal routes by utilizing a hybrid approach combining various IoT devices, primarily targeted for private cabs and taxis. The focus of a contribution of Al-Rahamneh et a. is on creating an multi-modal urban data platform with context-awareness [2].

[1] provides an analysis on the potentials of multi-modal travel support, but does not provide a framework or architecture. A mathematical model for preference-aware transport matching is contributed by [4]. The European Platforms Initiative project BIG IoT has been initiated to implement smart mobility services and applications for Barcelona, Piedmont, and Berlin/Wolfsburg. It aims to solve the interoperability gap by defining a generic, unified Web API for smart object platforms [5].

D. Security and Privacy Issues in Smart Mobility

Smart mobility offers many benefits for urban areas, users and the environment. However, there are also many privacy concerns. Smart mobility approaches manage and share both sensed and user-generated data to a large extent, which are associated with user identities, spatial information and temporal information [11], [43], [44]. This might allow to infer sensitive personal information based on location homogeneity, location distribution, probability distributions of locations, and background knowledge, even if the location data is anonymized [45]. Statistical analyses show that even the distribution of locations where users stay for some time, e.g., to switch from one vehicle to another one, is a sensitive information [12]. In addition to that, smart mobility approaches depend on a complex IT ecosystem with many different parties, which increases the likelihood of security incidents. Finally, the IoT technologies used enable new kinds of cyber-physical attacks [10].

Secure and privacy-aware smart mobility is of interest to both the research community and society. Nevertheless, there are are only a few studies targeting smart mobility with Homomorphic Encryption. For example, [46], [47] provide a privacy preserving solution for mobile cloud computing using IoT devices with HE. However, they do not propose a framework or analysis of the application on smart mobility.

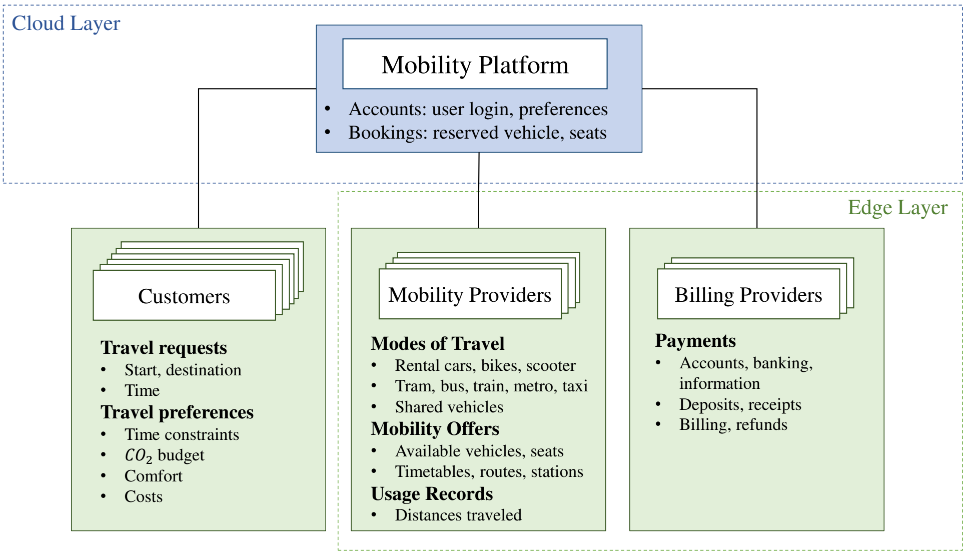


Fig. 2. Smart Mobility Architecture

III. Smart Mobility with Fully Homomorphic Encryption

In this section, we will introduce our system architecture, and we derive privacy requirements. Furthermore, we describe transactions that are representative for smart mobility and can be implemented with fully homomorphic encryption.

A. *Smart Mobility Architecture*

To find out if FHE is applicable to smart mobility, we use a generalized architecture model, as shown in Figure 2. This model is in line with existing work [5].

Customers issue travel requests via resource-constrained devices such as smart phones. Travel requests have a start point, a destination and a start/end time. A travel request can be constrained by the customer's budget and preferences regarding speed, comfort, eco-friendliness, etc.

The **mobility platform** manages user accounts, connects all parties with each other, and provides platform services such as identifying potential mobility providers for a travel request and booking seats or vehicles at the mobility providers selected by the customers. The mobility platform is part of the *Cloud Layer* of our architecture model. Thus, the mobility platform has extensive processing resources. This includes a high network throughput, plenty of primary and secondary storage, and a high processing capacity.

Mobility providers deploy various means of transportation, e.g., rental/shared vehicles or seats in a public transportation service. Furthermore, mobility providers log the actual distances traveled with each vehicle or on each seat. Mobility providers are located in the *Edge Layer*, i.e., they run services on a cloud instance that is one order of magnitude smaller than the mobility platform.

Finally, **billing providers** invoice the trips made by each means of transportation. Similarly to mobility providers, the billing providers are part of the *Edge Layer*.

Observe, that in this architecture only the customers are natural persons in the sense of the GDPR [48]. All other parties are institutions that are not covered by the data protection regulations. Thus, only customer data needs to be protected.

B. *Privacy requirements*

The components of our architecture model process five distinct categories of data:

Insensitive data cannot be related to a person, and does not carry sensible information. Any information from an institutional party such as the mobility platform or a mobility provider is insensitive data, e.g, the public keys of those parties. We also consider the aggregated values calculated with FHE to be insensitive, e.g., total travel costs, duration of a trip or CO_2 budget.

Identifiers such as a name or a bank account reveal the identity of a person. A trip can be an identifier, if it ends at the customer's home.

Pseudonyms such as a login name can be changed easily. We assume that a pseudonym only allows to recognize a person during one transaction. The public key of a customer is also a pseudonym.

Sensitive data, refers to personal information, e.g., habits, social life (persons traveling together), mobility preferences or travel costs. Note that sensitive data is not necessarily identifying.

Finally, **secret data** includes all information that must not be shared, e.g., the private keys of the various parties. From this categorization, we derive three privacy requirements:

R1 The mobility platform connects customers with two kinds of providers. Therefore, it needs to maintain user pseudonyms and transaction IDs. The platform does not need to learn identifiers or sensitive data, e.g., travel data forwarded to mobility providers.

R2 Mobility providers must learn which vehicles or seats are booked for which periods of time, and where a rented vehicle is left at the end of the trip. To create an invoice, the actual usage must be recorded. This requires pseudonymous information and sensitive data. It must be impossible to join sensitive data from multiple transactions or across multiple mobility providers.

R3 A billing provider needs to know identifiers (names, addresses, bank accounts) and invoice amounts. It is also acceptable if the billing provider learns pseudonyms. Except from that, it should learn only insensitive data.

Our privacy requirements are summarized in Table I.

Data Categories	*Mobility Platform* (R1)	*Mobility Provider* (R2)	*Billing Provider* (R3)
Insensitive Data	✓	✓	✓
Identifiers	✗	✗	✓
Pseudonyms	✓	✓	✓
Sensitive Data	✗	✓	✓
Secret Data	✗	✗	✗

TABLE I
PRIVACY REQUIREMENTS

We assume that a combination of one-time pseudonyms and traditional encryption helps to mitigate any privacy problem that relates to data-management transactions, e.g., marking a certain seat in a database as “reserved”, searching for available modes of transportation at the last stop of a tram, or recording the time a rental bike has been used.

This leaves open privacy issues related to calculations. For example, consider the billing process. In traditional smart mobility scenarios, the cloud platform might calculate the invoice total by asking each mobility provider, that was involved in the trip of a certain customer. By doing so, the smart mobility platform learns the exact movement patterns of each customer.

FHE might be able to execute such calculations without revealing personal details. The advantage of using FHE over alternatives from the realm of secure multiparty computation is that FHE does not depend on privacy models where multiple semi-honest parties execute protocols in multiple rounds of communication. If each transaction is secured with its own pair of one-time keys, the security and privacy of the approach only depend on the formal guarantees of the FHE schemes used. We also do not need to make assumptions about colluding parties.

C. Experiment Design

As we explore the applicability of FHE for smart mobility, we rule out transactions without sensitive/pseudonymous data or where encryption, decryption and computations take place on a cloud instance. An application of the privacy requirements (Table I) on our architecture model (Figure 2) has shown that the billing provider has similar properties as the mobility provider: It learns identifiers instead of pseudonyms, and it has comparable computational means and data flows. Thus, we also leave aside experiments that specifically address billing providers. We experiment with three transactions T1–T3. Each transaction contains a small number of additions and multiplications. This is typical for business transactions that compute arrival times, discounts or usage fees. With FHE, such a transaction requires one relinearization operation.

a) T1: Centralized calculations: This transaction is representative for operations where encryption and decryption takes place at the customer’s smartphone, while the calculation is executed at the mobility platform. For example, the mobility platform might add up encrypted prices, travel times, CO_2 budgets, etc., whose summands stem from the mobility providers. It might also multiply discounts or subtract bonuses. Then the mobility platform sends the encrypted result to the customer for decryption. The mobility providers might not want to reveal bonus schemes, mutual price agreements or internal calculations. Thus, it is not an option to send plain values to the customer and let the smartphone do any calculation. Instead, FHE can be applied. Figure 3 illustrates T1. Thick lines in the figure refer to encrypted data.

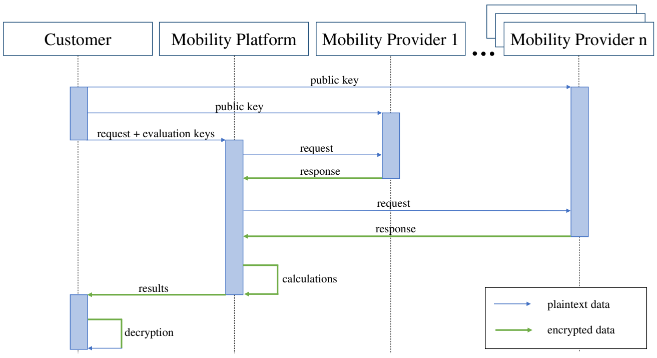


Fig. 3. T1: Centralized Calculations

With this transaction, numerous parallel transactions must not overload the computational resources of the mobility platform, and the encryption/decryption of a single transaction must be feasible on the customer’s smartphone. T1 requires the mobility providers to know the public key of the customer. This can be an one-time key, and it is a pseudonym. Each mobility provider only learns which of its own means of transport is part of the transaction. Thus, R2 is met. The mobility platform does not learn the customer’s public key, because relinearization requires evaluation keys that are used only once. To manage travels across multiple mobility providers, the mobility platform needs a transaction id, that is an one-time pseudonym. Thus, R1 is also fulfilled.

b) T2: Decentralized calculations: This transaction lets the mobility providers do the calculations. The mobility platform transfers encrypted intermediate results to different parties, e.g., to a rental car provider or a public transport provider, and orchestrates distributed computations on encrypted data there. Such parties have smaller computational resources than the mobility platform, but are also loaded with a smaller

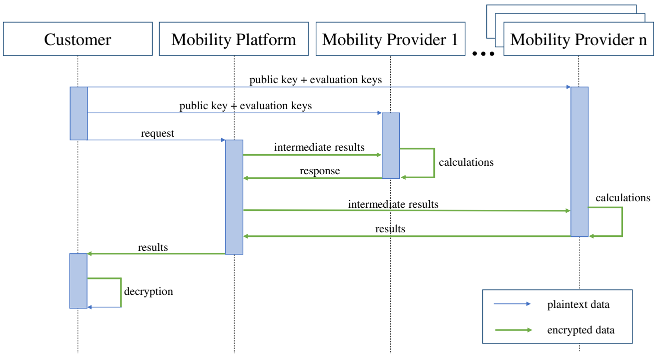


Fig. 4. T2: Decentralized Calculations

number of parallel transactions. Encryption/decryption takes place at the customer's smartphone. Figure 4 illustrates this.

Similarly to T1, the reason for using FHE is that the mobility providers might not want to share internal agreements and calculations. Again, the mobility providers need the customer's public key. Thus, the privacy properties of T2 are identical to T1, but the resources needed at the different parties are different.

c) T3: Customer-side calculations: Our third transaction performs calculations at the customer's smartphone, while the mobility providers contribute encrypted parameters, and the mobility platform is responsible for decryption (cf. Figure 5). Thus, the reason for requiring FHE is similar to T1 and T2. Such a transaction could calculate with telemetry data: Forecasting demand, costs, CO_2 per hour etc. require usage-dependent calculations with data from multiple mobility providers. But it might be sufficient for each mobility provider to learn the aggregated numbers.

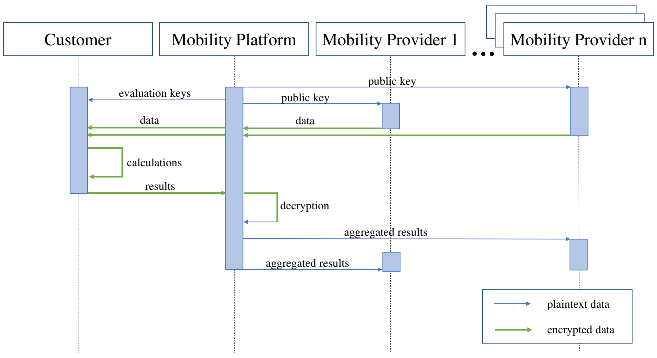


Fig. 5. T3: Customer-side Calculations

With T3, the calculations must not overload a smartphone, and decrypting many results in parallel must be feasible for a mobility platform. T3 means that the mobility providers need the public key of the mobility platform. Because it is an institutional party, this is insensitive data (cf. Table I). The mobility platform cannot learn which parameters from which mobility provider contribute to the aggregated results. Similarly, mobility providers only learn aggregated results, which is insensitive data. Thus, privacy requirements R1 and R2 are fulfilled. Our three transactions are representative for a wide number of typical real-world calculations in smart mobility scenarios. Therefore, we refrain from realizing other transactions that have the same structure and do not provide further insights. For example, in the billing process the invoice amounts could be encrypted by the mobility providers, calculated at the platform and decrypted by a billing provider, which has the same structure as T1.

IV. Experimental Evaluation

In this section, we define the computational and financial overhead we deem acceptable for smart mobility. Furthermore, we describe our prototypical implementation of our transaction, and we evaluate it with a series of experiments.

A. Resources and Costs

To find out if FHE can be applied to smart mobility scenarios, we need an understanding of the resources available and the costs involved. The *mobility platform* is a cloud service with plenty of computational resources for massive parallel processing. However, it must handle a very large number of requests at the same time. Furthermore, some FHE operations cannot be parallelized. An Amazon AWS instance "m6i.32xlarge" [49] serves as a reference for the computing costs of a large cloud instance. It is equipped with 128 XEON CPU cores at 3.5 GHz, 512 GiB RAM, 50 Gbps network bandwith. In July 2023, a m6i.32xlarge instance costs approx. 8 USD per hour, i.e., one second of one core 0,017 millicents.

The *mobility providers* are part of the edge layer of our architecture model. Thus, such providers would operate its services on a cloud instance that is one order of magnitude smaller and less expensive than those of the mobility platform. As multiple mobility providers exist, each of them has a much smaller individual load of parallel transactions than the mobility platform. A "m6g.8xlarge" cloud instance with 32 XEON CPU cores at 3.5 GHz and 128 GiB RAM might be suitable for the edge layer. Such an instance costs approx. 1.5 USD per hour.

Customers connect via smartphone to the mobility platform. A smartphone has comparatively scarce computing resources. However, as it is the customer's property, it does not need to execute multiple transactions in parallel. As a reference for computing times on a current mid-range smartphone, consider a "Fairphone 4" [50]. It has a CPU with 8 cores at 2.2 GHz and 8 GiB RAM and 128 GiB internal storage. We assume that it is acceptable for a customer and any other party if a transaction takes at most two seconds to complete. This time is comparable to starting an app on a smartphone, i.e., it does not impact the user experience. For comparison, humans do not perceive a reaction time below 400 ms as an interruption, and cannot sense delays below 100 ms [29]. Furthermore, a few seconds computing time on a small number of cores of a large cloud instance does not contribute much to the total travel costs of the customer.

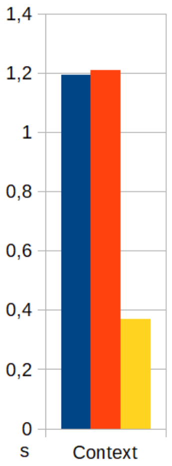


Fig. 6. Runtimes of the Context Creation

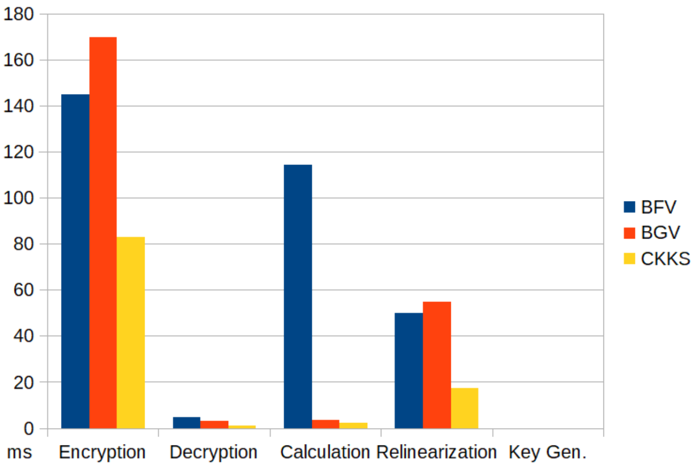


Fig. 7. Runtimes of the Operations

B. Implementation

For each of our three distributed transactions, we have implemented a FHE-encrypted variant, and a non-encrypted one for comparison. We decided to implement our transactions in C++ with Microsoft SEAL [23], because it is the most advanced implementation of the three state-of-the-art schemes BGV [26], BFV [27] and CKKS [28]. While the first two schemes compute with integers, the last one supports float-point operations. The length of the integers and the precision of float-point operations depend on the size of the modulus degree. The modulus determines how much noise can be accumulated during computations, before a relinearization operation with a pre-calculated evaluation key is needed. The noise is an internal measure to avoid that encrypted data can be decoded without knowing the secret key, i.e., it has nothing to do with the accuracy of the computation result. We used a polynomial modulus degree of 16384 and a plain modulus degree of 1024, which is the recommended setting in SEAL. For performance reasons, we disabled the debug mode and enabled batch processing, i.e., the SEAL library did not encrypt or decrypt any value individually.

We executed our experiments on a host with a 2.8 GHz Intel i7 CPU with 8 cores and 32 GiB RAM. Thus, one CPU core of our experimental host is approx. 25% slower than a core of an “m6i.32xlarge” instance and 30% faster than a core of a “Fairphone 4”. We have started one customer, one mobility platform and two mobility providers as separate processes, and we have repeated each experiment 500 times and computed the averages. We want to measure the processing time and the size of the encrypted data at each party separately. This allows us to find out if a FHE scheme exceeds the time budget of 2-3 seconds in total, or if one of the parties might be potentially drained of resources, when handling many customers in parallel.

In order to execute the measurements, we have implemented our experiments as test cases with the DOCtest framework [51]. This allows to implement experiments as a batch, and to verify that the computed results are correct. We used log4cplus [52] to monitor the execution, and we measured with the Google Benchmark v1.7.1 [53] microbenchmarking framework. Google Benchmark ensures that the compiler does not change the execution, e.g., to optimize 500 repetitions of the same execution. It delivers the values measured in a JSON format. We also evaluated the individual method calls with Intels VTUne profiler [54].

C. Evaluation Results

First we analyze the performance of BGV, BFV and CKKS. For comparability, we have measured the runtimes as CPU time on a single core. After that, we measure the memory consumptions.

a) Runtime Performance: Context creation takes only once at startup of a service or application. The purpose of this operation is to initialize and configure the SEAL library with the appropriate credentials, seeds, buffers etc. for the respective FHE scheme. As Figure 6 shows, we have measured an average context-creation time of up to 1.2 s. Thus, it is mandatory for any application not to shut down and start up the FHE library for each operation, but to preserve its state. Note that context creation can be executed in parallel with the normal launch of an application. Since even today’s smartphones have multiple CPU cores, this overhead does not necessarily increase the application’s startup time.

Figure 7 shows the runtimes in milliseconds of the FHE operations. Encryption and decryption refer to the respective cryptographic operations. With calculation, we denote to a basic mathematical operation consisting of a few additions, multiplications and subtractions. Surprisingly, this was a time-consuming operation with BFV, which took approx. 115 ms on our 2.8 GHz Intel i7 CPU. The other schemes required 4 ms and 2.3 ms. Relinearization is needed after some calculations to ensure that the decryption produces correct results. The relinearization requires an one-time evaluation key, whose creation time is depicted in the last column of Figure 7.

As the figure shows, CKKS consumes approximately half the CPU time of the two other schemes, and the most expensive operations are encryption and relinearization. The runtimes for computations on encrypted data are orders of magnitude higher than on plain-text values. However, humans do not perceive reaction times below 400 ms as annoying, and do not recognize a delay below 100 ms at all [29]. Note that our measures only consider the runtimes of the FHE operations, i.e., we leave aside context creation, operating system, start-up times of applications and network delays.

We want to find out if those runtimes add up to a disruptive amount for our three transactions T1, T2 and T3. Therefore, we measured and aggregated the CPU times for any operation on any party, again for each of the schemes BFV, BGV and CKKS. We have left aside the context creation.

To avoid confounding the effects of parallelization with the CPU times needed at the various parties, we have structured our experiment so that all mobility providers operate independently in parallel, while the customer and the mobility platform wait for all other parties. For the same reason, we do not measure network delays, effects of the operation system, etc. In a real setting, each party would start encrypting, decrypting or calculating values as soon as the first chunk of data has arrived, i.e., the total runtimes would be smaller.

To foster comparability among the transactions, we also ensured that T1-T3 used the same set of operations, and differed only in the place where each encryption, decryption, calculation etc. took place. The set of operations contained a number of additions and calculations that was large enough to require one relinearization. Having said this, the BFV scheme required a total of 314 ms CPU time on average to complete a transaction. BGV needed an average of 232 ms, and CKKS had the best runtime performance with only 104 ms on average.

Figures 8-10 show for centralized, decentralized and customer-side calculations at which party how much CPU time is consumed. For comparability, each party uses the same CPU and is limited to one core. Figures 8 and 9 confirm that centralized and decentralized calculations do not burden the smartphone of the customer. With such transactions, the customer is responsible for generating evaluation keys and decrypting values, which are fast operations (Figure 7).

Figure 8 corresponds to a centralized transaction where any computation and relinearization takes place at the mobility platform. Encrypting values must be performed at the data sources. With the mobility providers as data sources, this corresponds to an edge computing scenario, where all time-consuming operations are executed on a centralized or decentralized cloud instance.

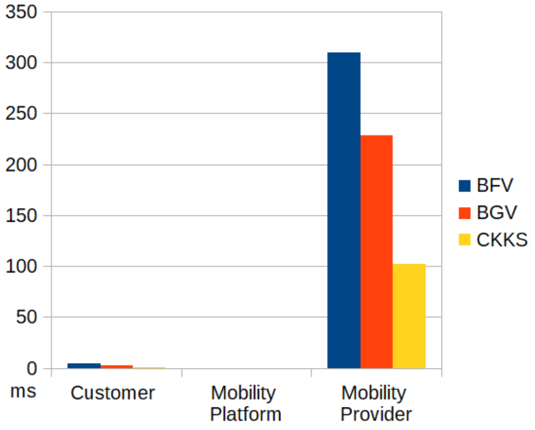


Fig. 9. T2: Decentralized Calculations

Figure 9 transfers any computationally expensive operation in the domain of the mobility providers. If we take a m6i.32xlarge cloud instance for comparison, each transaction costs each mobility provider less than 0,003 millicents, even with the slowest FHE scheme. Figure 10 confirms that even customer-side calculations incur negligible overhead on a smartphone. Similarly to T1 and T2, with T3 the encryption of the data takes place at the mobility providers.

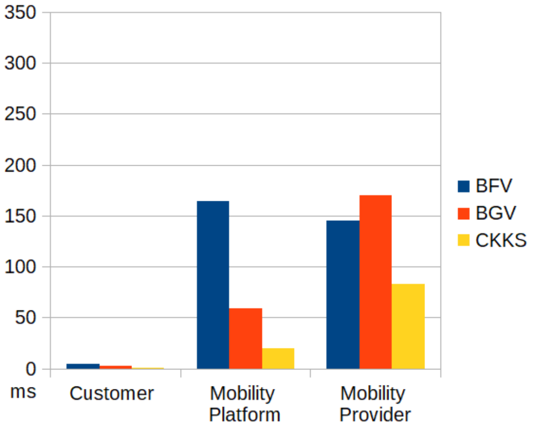


Fig. 8. T1: Centralized Calculations

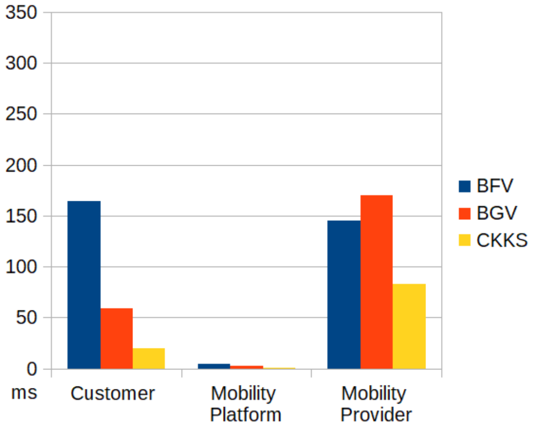


Fig. 10. T3: Customer-side Calculations

Note that for to calculations with unencrypted values at the cost of privacy, the runtimes and the costs of a single calculation are below measurement accuracy, and virtually zero. Thus, FHE is not suitable for any big-data problem, or for scenarios where numerous transactions must be executed to the smallest possible costs. However, in the field of smart mobility, such transaction fees are several orders of magnitude smaller than the billing amount on the customer's invoice, but privacy is an important factor. Thus, we have confirmed that FHE schemes are feasible for business models in the field of smart mobility.

b) Approximated Memory Comsumption: In order to have a practical estimate of the memory consumption incurred by FHE, we have implemented each customer, mobility platform and mobility provider as an individual application. Thus, we have measured the total amount of memory of the application, libraries, runtime variables and the buffers where encrypted values are stored. The isolated increase in buffer sizes needed to store encrypted intermediate results can be found in [27](BFV), [26](BGV) and [28](CKKS). Table II summarizes this.

FHE Scheme	*Memory Consumption*
Plain Text	76 – 104 MB
BFV	207 – 321 MB
BGV	216 – 306 MB
CKKS	146 – 188 MB

TABLE II
APPROXIMATED MEMORY CONSUMPTION

With our experiments, we measured a memory consumption between 207 MB and 321 MB for BFV. We measured between 216 MB and 306 MB for BGV, and 146 MB to 188 MB for CKKS. A large memory consumption corresponds to more expensive operations (encryption and calculations including relinearization). This was because such operations require many runtime variables and, therefore, a large and deep stack. In comparison, an execution on unencrypted values resulted in applications with a memory footprint between 76 MB and 104 MB. Thus, none of the FHE schemes utilized memory resources that exceeded even the capacity of a smartphone.

V. Conclusion

In the upcoming years, the planning of cities and transportation logistics for moving people and goods will undergo significant changes. The conventional concept of mobility using individual transportation modes, such as a car, is not longer useful due to environmental reasons and growing cities. The demand for multi-modal transport solutions that allow users to move flexible and eco-friendly is high. However, the implementation of this approach requires the sharing of sensitive personal data with various parties, creating potential privacy risks.

This paper explored the potential use of fully homomorphic encryption as an efficient and noise-free solution for data privacy concerns in the implementation of smart mobility. Initially, privacy requirements for such a smart mobility approach were formulated, based on which three multi-party computations were identified that benefit from FHE. An implementation was provided using state-of-the-art FHE schemes BGV [26], BFV [27] and CKKS [28] based on Microsoft SEAL [23]. Finally, memory consumption and execution times were measured, evaluated and compared with a non-encrypted benchmark. To provide optimal experimental results, a benchmark framework was used to monitor memory consumption and execution times. To test the applicability of FHE in a real-life smart mobility scenario, the ressources used in the implementation were analyzed and compared to available resources on smartphones and cloud instances.

Based on the experiments conducted, encrypting transactions with FHE increases CPU time by approximately 100 milliseconds compared to unencrypted transactions. However, this additional processing time does not adversely affect the user experience [29]. The use of parallel processing can significantly reduce this time, and the cost of such encryption on a current cloud instance is less than 3 microcents. We conclude that FHE is a cost-effective means of ensuring privacy, and a viable option for a smart mobility business model.

For future research, it would be beneficial to scale the implementation to a real-life scenario involving multiple smartphones functioning as edge devices, and leveraging cloud instances for both the mobility platform and service providers. Hence, experiments could be extended to measure actual runtimes including side effects of operating systems, and delays of a virtualization environment. Also, delays of network connection could be reported. Furthermore, it would be worthwhile to evaluate and compare other libraries such as OpenFHE [55] and other state-of-the-art FHE schemes.

Acknowledgment

We would like to thank Yasin Akbas for his excellent implementation and evaluation of the smart mobility model with fully homomorphic encryption.

References

[1] Å. Jevinger and J. A. Persson, "Potentials of context-aware travel support during unplanned public transport disturbances," *Sustainability*, vol. 11, no. 6, p. 1649, 2019.

[2] A. Al-Rahamneh *et al.*, "Enabling customizable services for multimodal smart mobility with city-platforms," *IEEE Access*, vol. 9, pp. 41 628–41 646, 2021.

[3] J. Schuppan, S. Kettner, A. Delatte, and O. Schwedes, "Urban multimodal travel behaviour: Towards mobility without a private car," *Transportation Research Procedia*, vol. 4, pp. 553–556, 2014.

[4] M. S. Chowdhury, M. A. Osman, and M. M. Rahman, "Preference-aware public transport matching," in *International Conference on Innovation in Engineering and Technology (ICIET)*. IEEE, 2018, pp. 1–6.

[5] A. Bröring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic, and E. Teniente, "Enabling iot ecosystems through platform interoperability," *IEEE software*, vol. 34, no. 1, pp. 54–61, 2017.

[6] Y. Li *et al.*, "Pare: A system for personalized route guidance," in *Conference on World Wide Web*, 2017.

[7] D. Herzog, H. Massoud, and W. Wörndl, “Routeme: A mobile recommender system for personalized, multi-modal route planning,” in *Conference on User Modeling, Adaptation and Personalization*, 2017.

[8] P. Campigotto, C. Rudloff, M. Leodolter, and D. Bauer, “Personalized and situation-aware multimodal route recommendations: the favour algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 92–102, 2016.

[9] O. Moran, R. Gilmore, R. Ordóñez-Hurtado, and R. Shorten, “Hybrid urban navigation for smart cities,” in *20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[10] S. Paiva *et al.*, “Privacy and security challenges in smart and sustainable mobility,” *SN Applied Sciences*, vol. 2, pp. 1–10, 2020.

[11] T. Borchers *et al.*, “Privacy concerns on the mobility of smart cities,” in *Brazilian Technology Symposium (BTSym’21)*, 2021.

[12] E. P. de Mattos *et al.*, “The impact of mobility on location privacy,” *IEEE Systems Journal*, vol. 16, no. 4, pp. 5509–5520, 2022.

[13] D. Eckhoff and I. Wagner, “Privacy in the smart city,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 489–516, 2017.

[14] P. Zhao, G. Zhang, S. Wan, G. Liu, and T. Umer, “A survey of local differential privacy for securing internet of vehicles,” *The Journal of Supercomputing*, vol. 76, pp. 8391–8412, 2020.

[15] A. A. Khaliq, A. Anjum, A. B. Ajmal, J. L. Webber, A. Mehbodniya, and S. Khan, “A secure and privacy preserved parking recommender system using elliptic curve cryptography and local differential privacy,” *IEEE Access*, vol. 10, pp. 56 410–56 426, 2022.

[16] G. Qin, S. Deng, Q. Luo, J. Sun, and H. Kerivin, “Toward privacy-aware multimodal transportation: Convergence to network equilibrium under differential privacy,” *Available at SSRN 4244002*, 2022.

[17] P. Shanthi and S. Balasundaram, “An efficient clique cloak algorithm for defending location-dependent attacks in location based services,” in *Conference on Information and Communication Technology for Competitive Strategies*, 2014.

[18] I. Memon, L. Chen, Q. A. Arain, H. Memon, and G. Chen, “Pseudonym changing strategy with multiple mix zones for trajectory privacy protection in road networks,” *International Journal of Communication Systems*, vol. 31, no. 1, p. e3437, 2018.

[19] F. Martelli, M. E. Renda, and J. Zhao, “The price of privacy control in mobility sharing,” in *Sustainable Smart City Transitions*. Routledge, 2022, pp. 233–258.

[20] T. Li, L. Lin, and S. Gong, “Autompc: Efficient multi-party computation for secure and privacy-preserving cooperative control of connected autonomous vehicles.” in *SafeAI@ AAAI*, 2019.

[21] G. Raja *et al.*, “Ai-powered blockchain-a decentralized secure multi-party computation protocol for iov,” in *IEEE Conference on Computer Communications*, 2020.

[22] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A survey on homomorphic encryption schemes: Theory and implementation,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–35, 2018.

[23] Microsoft Research, Redmond, WA., “Microsoft SEAL (release 4.1),” https://github.com/Microsoft/SEAL, 2023, accessed Feb. 20th, 2023.

[24] S. Halevi *et al.*, “HElib 2.2.2, December 2022,” https://github.com/homenc/HElib, 2023, accessed Feb. 20th, 2023.

[25] OpenFHE., “OpenFHE,” https://www.openfhe.org/, 2023, accessed Feb. 20th, 2023.

[26] J. Fan and F. Vercauteren, “Somewhat practical fully homomorphic encryption,” Cryptology ePrint Archive, Paper 2012/144, 2012. [Online]. Available: https://eprint.iacr.org/2012/144

[27] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “Fully homomorphic encryption without bootstrapping,” Cryptology ePrint Archive, Paper 2011/277, 2011. [Online]. Available: https://eprint.iacr.org/2011/277

[28] J. H. Cheon, A. Kim, M. Kim, and Y. Song, “Homomorphic encryption for arithmetic of approximate numbers,” in *Conference on the Theory and Applications of Cryptology and Information Security*, 2017.

[29] J. Nielsen and R. Budiu, *Mobile usability*. MITP-Verlags GmbH & Co. KG, 2013.

[30] C. Gentry, *A fully homomorphic encryption scheme*. Stanford university, 2009.

[31] O. Regev, “Lattice-based cryptography,” in *Advances in Cryptology-CRYPTO 2006: 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006. Proceedings 26*. Springer, 2006, pp. 131–141.

[32] V. Lyubashevsky, C. Peikert, and O. Regev, “On ideal lattices and learning with errors over rings,” *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–35, 2013.

[33] S. Behera and J. R. Prathuri, “Design of novel hardware architecture for fully homomorphic encryption algorithms in fpga for real-time data in cloud computing,” *IEEE Access*, vol. 10, pp. 131 406–131 418, 2022.

[34] S. Gupta *et al.*, “Memfhe: End-to-end computing with fully homomorphic encryption in memory,” *ACM Transactions on Embedded Computing Systems*, 2022.

[35] J.-W. Lee, H. Kang, Y. Lee, W. Choi, J. Eom, M. Deryabin, E. Lee, J. Lee, D. Yoo, Y.-S. Kim *et al.*, “Privacy-preserving machine learning with fully homomorphic encryption for deep neural network,” *IEEE Access*, vol. 10, pp. 30 039–30 054, 2022.

[36] J. Chen, K. Li, and S. Y. Philip, “Privacy-preserving deep learning model for decentralized vanets using fully homomorphic encryption and blockchain,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 633–11 642, 2021.

[37] F. Wibawa *et al.*, “Homomorphic encryption and federated learning based privacy-preserving cnn training: Covid-19 detection use-case,” in *European Interdisciplinary Cybersecurity Conference*, 2022.

[38] D. Stripelis *et al.*, “Secure neuroimaging analysis using federated learning with homomorphic encryption,” in *Symposium on Medical Information Processing and Analysis*, vol. 12088, 2021, pp. 351–359.

[39] L. Zhu *et al.*, “Privacy-preserving authentication and data aggregation for fog-based smart grid,” *IEEE Communications Magazine*, vol. 57, no. 6, pp. 80–85, 2019.

[40] M. Goudarzi, M. Palaniswami, and R. Buyya, “A distributed application placement and migration management techniques for edge and fog computing environments,” in *16th Conference on Computer Science and Intelligence Systems*. IEEE, 2021, p. 37–56.

[41] L. Zhang, J. Xu, P. Vijayakumar, P. K. Sharma, and U. Ghosh, “Homomorphic encryption-based privacy-preserving federated learning in iot-enabled healthcare system,” *IEEE Transactions on Network Science and Engineering*, 2022.

[42] A. Morelli, L. Campioni, N. Fontana, N. Suri, and M. Tortonesi, “A federated platform to support iot discovery in smart cities and hadr scenarios,” in *Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020, pp. 511–519. [Online]. Available: http://dx.doi.org/10.15439/2020KM48

[43] M. Jarosz, K. Wrona, and Z. Zieliński, “Formal verification of security properties of the lightweight authentication and key exchange protocol for federated iot devices,” in *17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 617–625.

[44] K. Kanciak, K. Wrona, and M. Jarosz, “Secure onboarding and key management in federated iot environments,” in *17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 627–634.

[45] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, “A classification of location privacy attacks and approaches,” *Personal and ubiquitous computing*, vol. 18, pp. 163–175, 2014.

[46] W. Ren *et al.*, “Privacy-preserving using homomorphic encryption in mobile iot systems,” *Computer Communications*, vol. 165, pp. 105–111, 2021.

[47] M. R. Baharon *et al.*, “A new lightweight homomorphic encryption scheme for mobile cloud computing,” in *IEEE Computer and Information Technology*, 2015.

[48] Council of the European Union, “Regulation (eu) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data,” OJ L 119, 4.5.2016, p. 1–88, 2016.

[49] Amazon Web Services, Inc., “Amazon EC2 M6i Instances,” 2023, accessed Feb. 20th, 2023. [Online]. Available: https://aws.amazon.com/de/ec2/instance-types/

[50] Fairphone B.V, “Fairphone 4,” 2022, accessed Feb. 20th, 2023. [Online]. Available: https://www.fairphone.com/

[51] V. Kirilov *et al.*, “Doctest v2.4.9,” 2022, accessed Feb. 20th, 2023. [Online]. Available: https://github.com/doctest/doctest

[52] T. E. Smith *et al.*, “log4cplus v2.1.0,” 2023, accessed Feb. 20th, 2023. [Online]. Available: https://github.com/log4cplus

[53] Google Inc., “google/benchmark v1.7.1,” 2022, accessed Feb. 20th, 2023. [Online]. Available: https://github.com/google/benchmark

[54] Intel Corporation, “Intel VTune Profiler,” https://www.intel.com, 2023, accessed Mar. 07th, 2023.

[55] A. Badawi *et al.*, “Openfhe: Open-source fully homomorphic encryption library,” in *Encrypted Computing & Applied Homomorphic Cryptography*, 2022.

Proceedings of the 18th Conference on Computer
Science and Intelligence Systems pp. 533–543

DOI: 10.15439/2023F5158
ISSN 2300-5963 ACSIS, Vol. 35

Efficient exact A* algorithm for the single plant Hydro Unit Commitment problem

Alexandre Heintzmann*† , Christian Artigues*, Pascale Bendotti†, Sandra Ulrich Ngueveu* and Cécile Rottner†
*LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France
Email: {alexandre.heintzmann, christian.artigues, sandra.ulrich.ngueveu}@laas.fr
†EDF Lab Paris-Saclay, 7 Bd. Gaspard Monge, 91120 Palaiseau, France
Email: {alexandre.heintzmann, pascale.bendotti, cecile.rottner}@edf.fr

***Abstract*—The Hydro Unit Commitment problem (HUC) specific to hydroelectric plants is part of the electricity production planning problem, called Unit Commitment Problem (UCP). More specifically, the studied case is that of the HUC with a single plant, denoted 1-HUC. The plant is located between two reservoirs. The horizon is discretized in time periods. The plant operates at a finite number of points defined as pairs of the generated power and the corresponding water flow. Several constraints are considered. Each reservoir has an initial volume, as well as window resource constraints, defined by a minimum and maximum volume per time period. At each time period, there is an additional positive, negative or zero intake of water in the reservoirs. The case of a price-taker revenue maximization problem is considered. An efficient exact A* variant, so called HA*, is proposed to solve the 1-HUC accounting for window constraints, with a reduced search space and a dedicated optimistic heuristic. This variant is compared to a classical Resource Constrained Shortest Path Problem (RCSPP) algorithm and a Mixed Integer Linear Programming formulation solved with CPLEX. Results show that the proposed algorithm outperforms both concurrent alternatives in terms of computational time in average on a set of realistic instances, meaning that HA* exhibits a more stable behavior with a larger number of instances solved.**

I. Introduction

AN ELECTRICITY producer aims at meeting the demand at any time. This is because electricity can hardly be stored, meaning that any excess is lost. Scheduling the short-term production in order to meet the demand defines the Unit Commitment Problem (UCP), which is solved for the day ahead. At Electricité de France (EDF), large-scale UCP instances are solved by Lagrangian decomposition [1], yielding subproblems of the same nature (thermal, hydraulic, solar,...). As each subproblem has different constraints, specific approaches are developed for each of them. Among these subproblems, the Hydro Unit Commitment (HUC) has received a lot of attention, due to the large size of its instances. Indeed, instances of the HUC involve valleys, which can be constituted of up to twenty plants, linked with reservoirs.

At EDF, the HUC is modeled as a Mixed Integer Linear Program (MILP) and solved with CPLEX [2]. This MILP considers operating points, which are pairs (water flow, corresponding generated power). In practice, the HUC is not solved to optimality within the time limit set, as such a representation induces an exponential number of solutions and large computational times. More recent work [3] have pointed out the interest of solving the HUC using a Lagrangian relaxation algorithm, where subproblems are single plant HUC (1-HUC). The main benefit of this relaxation is that the 1-HUC can be solved with dynamic programming, while the master problem handles the coupling constraints. It is shown that such a relaxation can lead to overall better results than solving the HUC as an MILP, which emphasizes the relevance of an efficient algorithm to solve the 1-HUC.

In this paper, we consider the 1-HUC with a single plant located between two reservoirs. A diagram, taken from [4] and shown in Figure 1, is sketching the 1-HUC. The principle of hydroelectric production is the following: the water from the upstream reservoir flows into the downstream reservoir through the units of the plant, thus driving the turbines of the units, which in turn power the generator to produce electricity. When operating in reverse, the pumps of the units can move the water from the downstream reservoir to the upstream reservoir, which consumes electricity. The plant operates on M turbining points, N pumping points and an idle operating point. With $I = \{-N, \dots, 0, \dots, M\}$, each operating point $i \in I$ is defined as a pair formed by a water flow D_i and a generated power P_i. Both D_i and P_i are positive (resp. negative) for turbining (resp. pumping) operating points, i.e., with $i > 0$ (resp. $i < 0$), and are 0 for the idle operating point $i = 0$. The operating points are defined in a cumulative fashion meaning that if a plant is at turbining (resp. pumping) operating point i, then order constraints apply, involving all points $1 \leq j < i$ (resp. $-1 \geq j > i$) to also be operated. At each time period, the plant cannot turbine and pump simultaneously. The time horizon is discretized into T time periods. At each time period t, the plant turbines (resp. pumps) a water flow and produces (consumes) an amount of energy that is considered to be constant for the duration of the time period. Resource window constraints state that the volume of each reservoir $n \in \{1, 2\}$ lies between a lower bound $\underline{V}_t^n$ and an upper bound $\overline{V}_t^n$ that are time-dependent. At each time period, the reservoir n receives an additional intake of water A_t^n. The additional intake can be positive to represent rain, melting snow etc., or negative to represent the use of water for local agriculture etc.

The revenues take into account the unit value Φ^n of the water in each reservoir n at the end of the time horizon, and the value of the energy produced or consumed at a time-dependent unit value Λ_t. The problem is to maximize the total revenue,

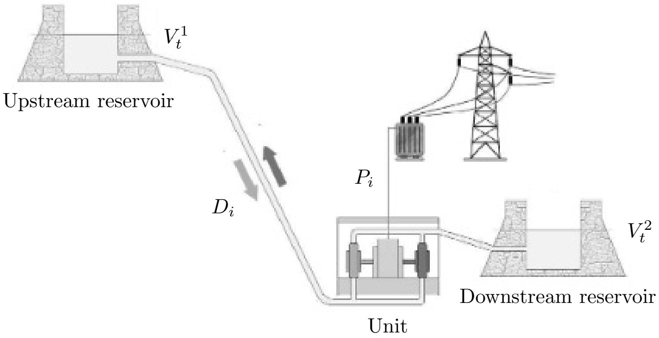


Fig. 1: Diagram of the 1-HUC

while satisfying the reservoir capacities at each time period.

In practice, water management policies require that reservoirs should meet target volumes ($\underline{V}_t^n = \overline{V}_t^n - \epsilon$) at the end of the time horizon. Due to these target volumes and the finite set of operating points, the 1-HUC may not have any feasible solution. However, it is possible to adjust [5] or relax [6] the target volumes in order to obtain feasible instances. Hence, we consider in this paper only 1-HUC instances admitting feasible solutions.

In this paper the aim is to propose a dynamic programming algorithm dedicated to the 1-HUC, modeled as a Resource Constrained Shortest Path Problem (RCSPP) with resource window constraints. The algorithm we propose is an exact variant of the A* algorithm [7] used to compute the shortest path in a graph. To obtain an efficient algorithm, the bounds of the windows are first tightened by propagating them from any time period to another, and a dedicated heuristic is developed. As we propose an exact algorithm, we compare it with two other exact methods, namely an MILP solved by default CPLEX as currently done at EDF and a classical RCSPP algorithm. The corresponding evaluation of performance is done on various realistic instances. The numerical results show that our algorithm yields smaller computational time variations compared to both the MILP formulation solved by CPLEX and the RCSPP algorithm.

The remainder of the paper is organized as follows. In Section II, a literature review of dynamic programming algorithms for related problems is reported. In Section III, a formulation of the 1-HUC is proposed. In Section IV, graph representations of the 1-HUC are provided. In Section V, the bound tightening procedure and the exact A* variant are described. In Section VI, numerical results are presented. In Section VII, concluding remarks and perspectives for further research are drawn.

II. State of the art

In this section, we first present a literature review of dynamic programming algorithms developed for the UCP and for the HUC. As the HUC can be represented as an RCSPP, we also include in this review dynamic programming algorithms for the RCSPP.

A. Dynamic programming for the Unit Commitment Problem

A dynamic programming algorithm for a single Unit Commitment (1-UC) with ramp and min up/down constraints is presented in [8]. The algorithm is based on a graph with a source vertex and several groups of T vertices. For each even (resp. odd) group, vertex t indicates that the unit is turned off (resp. on) at time period t. The arcs connect the vertices of a group to the next groups, from a time period t to a time period $t' > t$. Finding a path in this graph allows one to find an on-off schedule for the unit. The difference between the 1-UC and the 1-HUC is that there is no resource in the 1-UC, while in the 1-HUC, the presence of reservoirs with minimum and maximum volumes requires to account for the water resource.

B. Dynamic programming for the Hydro Unit Commitment

In this part we focus on dynamic programming algorithms for the HUC, most of them being cited in the survey [9].

In [1] the author presents a two phase approach to solve the HUC, solving an LP for the first phase and using a dynamic programming algorithm for the second phase. More precisely, the second phase consists of solving the 1-HUC for each plant of the valley with dynamic programming, aiming to get the closest solution to the LP solution while taking into account constraints omitted in the LP. For this phase, the considered underlying graph is as follows. The reservoir volume is discretized, yielding hundreds of possible volume values for a reservoir. Then, a vertex is defined for each volume value and each time period, thus leading to hundreds of vertices per time period. A Bellman-Ford algorithm [10] is used to find a path in this graph. Such a discretization discards a lot of realistic states. In this paper, we consider all possible states with respect to the operating points, which can be exponential for instances with a large number of time periods. With an exponential number of states, the Bellman-Ford algorithm becomes far less efficient.

In [6] a method for solving a non-linear 1-HUC with a target volume is described. To solve this problem with dynamic programming, a state diagram is constructed. In a similar fashion as in [1], evenly discretized volumes are considered, yielding a limited number of states per time periods. In order to have feasible solutions, the target volume is relaxed to match this discretization. The state diagram is constructed by generating the possibilities to reach the target volume from the initial volume, satisfying the upper and lower bounds on the volume at each time period. Starting from the state at the end of the time horizon, the dynamic programming algorithm maximizes the value of the generated power. As we consider 1-HUC instances with and without target volumes, a backward algorithm may not be practical with large volumes and no target volume.

In [3], a decomposition method for solving the HUC with shortest paths is described. The considered HUC is a valley where each plant has a finite number of operating points. The topology of the valley is not restricted to a chain, as each plant (resp. reservoir) can have a set of upstream and downstream reservoirs (resp. plants). There are additional ramping constraints, namely the flow variation is limited from one time period to another. This HUC also takes into account a target volume for each reservoir, at the last time period. Note that the latter target volume is a minimal bound, meaning there is no equality constraint. The solution approach decomposes this HUC into multiple 1-HUC. Some 1-HUC without resource constraints are solved by a shortest path algorithm, while others with resource constraints are solved by a labeling algorithm defined in [11]. The latter algorithm is adapted from a classical RCSPP algorithm [12] to take into account a minimum bound for the resource. It is mentioned that this labeling algorithm loses its dominance properties between two labels if one of them does not verify the minimum bound on the resource. Such a case is more frequent when the target volume is considered with equality constraints, making this algorithm less efficient.

There are also other problems solved by dynamic programming, related to the 1-HUC. In [13] a dynamic programming approach is described to solve an HUC on instances of the Itaipù plant (Brazil, Paraguay). This problem differs from ours as the only constraint is to satisfy the minimum and maximum number of turbines running at each time period. No volume is considered, therefore there are neither bounds on the volume, nor a target volume. In [14] the Hydro Unit Load Dispatch problem (HULD) is presented. This problem differs from the HUC, as the water flow is known, and solving the HULD is to provide the most economic distribution of the water through the different turbines, while verifying the flow capacity of the turbines at each time period.

C. Shortest path with resource constraints

As the HUC can be seen as a shortest path problem with a water resource bounded both from below and above, we are interested in the solution methods for the RCSPP (Resource Constrained Shortest Path Problem).

There are works on the RCSPP to solve the thermal problem on EDF instances [15]. In that paper it is indicated that the resource has an upper bound but no lower bound. However, as specified in [3], the difficulty of the HUC comes from the lower bound on the volume, which prevents the use of dominance rules.

In the survey [16], a state of the art review of different shortest path variants is described. More specifically, it is indicated that there is little work on the RCSPP with equality constraints, or window constraints (such as in the HUC). Three papers are cited, namely [17] describing a heuristic, [18] presenting an integer formulation and [19] proposing a dynamic programming algorithm. As we look for an exact algorithm, we will focus on the three-phase algorithm described in [19]. The presented algorithm solves the RCSPP with window constraints on acyclic graphs. The main idea, further detailed in [20], is to extend the graph, such that if multiple paths lead to the same vertex from the source, a new vertex is created for each of these paths. Once the graph has been extended in this way, the problem is solved with a pseudo-polynomial time algorithm. Such a graph extension seems difficult to apply to the 1-HUC. Indeed, graph representations of the 1-HUC (see Section IV) solely have arcs from time period t to time period $t+1$. Hence, the extension can lead up to M^T vertices at time period T. Even a pseudo-polynomial algorithm on the extended graph could be impractical in the case of the 1-HUC.

III. Integer linear programming

With $x_{t,i}$ the binary variable indicating whether the plant is at least at operating point $i \in I$ at time period $t \leq T$, we obtain the following formulation:

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T} \sum_{i \in I} \Lambda_t P_i x_{t,i} + \Phi^1 \Big(\sum_{t=1}^{T} (A_t^1 - \sum_{i \in I} D_i x_{t,i}) \Big) \\
& + \Phi^2 \Big(\sum_{t=1}^{T} (A_t^2 + \sum_{i \in I} D_i x_{t,i}) \Big) \\
\text{s.c.} \quad & V_0^1 + \sum_{t=1}^{t'} (A_t^1 - \sum_{i \in I} D_i x_{t,i}) \leq \overline{V}_{t'}^1, \quad \forall t' \leq T \qquad \text{(a1)} \\
& V_0^1 + \sum_{t=1}^{t'} (A_t^1 - \sum_{i \in I} D_i x_{t,i}) \geq \underline{V}_{t'}^1, \quad \forall t' \leq T \qquad \text{(b1)} \\
& V_0^2 + \sum_{t=1}^{t'} (A_t^2 + \sum_{i \in I} D_i x_{t,i}) \leq \overline{V}_{t'}^2, \quad \forall t' \leq T \qquad \text{(a2)} \\
& V_0^2 + \sum_{t=1}^{t'} (A_t^2 + \sum_{i \in I} D_i x_{t,i}) \geq \underline{V}_{t'}^2, \quad \forall t' \leq T \qquad \text{(b2)} \\
& x_{t,i} \geq x_{t,i+1}, \quad \forall t \leq T, \forall i \in \{1, \dots, M-1\} \qquad \text{(c)} \\
& x_{t,i} \geq x_{t,i-1}, \quad \forall t \leq T, \forall i \in \{-1, \dots, -N+1\} \qquad \text{(d)} \\
& x_{t,1} + x_{t,-1} \leq 1, \quad \forall t \leq T \qquad \text{(e)} \\
& x_{t,i} \in \{0,1\}, \quad \forall t \leq T, \forall i \in I \qquad \text{(f)}
\end{aligned}$$

In this formulation, the objective function maximizes the total revenue. Constraints (a1) to (b2) ensure that the minimum/maximum bounds on the volume are verified for both the upstream and downstream reservoirs at each time period. Constraints (c) and (d) correspond to the order of the operating points. Constraints (e) prevent the plant from turbining and pumping simultaneously. Lastly, constraints (f) indicate that all variables $x_{t,i}$ are binary.

A. Shifting all operating points

In the following, it will be convenient to only have operating points with non-negative power and flow. In [3] a modification on the flows and the volume bounds is considered to only have such operating points. First, each turbining operating point i, for $i \in \{1, .., M\}$ is renumbered $i+N$, yielding operating point (D'_{i+N}, P'_{i+N}) with $D'_{i+N} = D_i$ and $P'_{i+N} = P_N$. For each pumping operating point $i \in \{-N, \dots, -1\}$, a turbining operating point numbered $N+i+1$ is created, yielding operating point (D'_{N+i+1}, P'_{N+i+1}) with $D'_{N+i+1} = |D_i|$ and $P'_{N+i+1} = |P_N|$. Operating point 0 remains unchanged. As such, all the operating points have non-negative cumulated flows. For a given $t \leq T$, the bounds on the volume $\overline{V}_t^1$

and $\underline{V}^1_t$ (resp. $\overline{V}^2_t$ and $\underline{V}^2_t$) are shifted in order to keep the same feasible solutions: $\overline{V'}^1_t = \overline{V}^1_t - t\sum_{i=-1}^{-N}|D_i|$ and $\underline{V'}^1_t = \underline{V}^1_t - t\sum_{i=-1}^{-N}|D_i|$ (resp. $\overline{V'}^2_t = \overline{V}^2_t + t\sum_{i=-1}^{-N}|D_i|$ and $\underline{V'}^2_t = \underline{V}^2_t + t\sum_{i=-1}^{-N}|D_i|$).

Example 1. Consider an instance of the 1-HUC with $M = 3$ turbining operating points $(D_1 = 3, P_1 = 3)$, $(D_2 = 4, P_2 = 3)$, $(D_3 = 2, P_3 = 2)$ and $N = 2$ pumping operating points $(D_{-1} = -3, P_{-1} = -5)$, $(D_{-2} = -2, P_{-2} = -4)$. The initial volumes are $V^1_0 = 50$, $V^2_0 = 30$. With $T = 3$, upstream reservoir bounds are $\overline{V}^1 = [100, 100, 20]$ and $\underline{V}^1 = [0, 0, 20]$, and downstream reservoir bounds are $\overline{V}^2 = [60, 60, 60]$ and $\underline{V}^2 = [0, 0, 0]$.

The renumbering is such that the 3 turbining operating points, of index 1 to 3 are now of index $1 + N$ to $3 + N$, i.e., $(D'_3 = 3, P'_3 = 3)$, $(D'_4 = 4, P'_4 = 3)$ and $(D'_5 = 2, P'_5 = 2)$ From pumping operating point -1 a turbining operating point $N - 1 + 1 = 2$ is created, and similarly from the pumping operating point -2 operating point $N - 2 + 1 = 1$ is created, i.e., $(D'_1 = 2, P'_1 = 4)$ and $(D'_2 = 3, P'_2 = 5)$.

The upstream reservoir upper bounds are modified as follows $\overline{V'}^1_1 = \overline{V}^1_1 - 1\cdot(2+3) = 95$, $\overline{V'}^1_2 = \overline{V}^1_2 - 2\cdot(2+3) = 90$, $\overline{V'}^1_3 = \overline{V}^1_3 - 3\cdot(2+3) = 5$. Similarly, we obtain the following upstream reservoir lower bounds $\underline{V'}^1 = [-5, -10, 5]$. The shift is done in the opposite way for the downstream reservoir, $\overline{V'}^2 = [65, 70, 75]$ and $\underline{V'}^2 = [5, 10, 15]$.

In the following, we only refer to the 1-HUC with operating points of non-negative flow and power, i.e., with $I = \{0, \ldots, M\}$. In this case, only constraints (a1) to (b2), (c) and (f) are necessary to model the constraints of the 1-HUC, and no renumbering is required.

B. Rewriting the formulation

The formulation defined previously is a classical MILP model for the 1-HUC. We rewrite the formulation in order for the window constraints to appear more clearly. Constraints (a1), (a2), (b1) and (b2) can be rewritten as follows, considering $\underline{V'}^1_t$ $\overline{V'}^1_t$, $\underline{V'}^2_t$ and $\overline{V'}^2_t$:

$$\sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \geq V^1_0 + \sum_{t=1}^{t'} A^1_t - \overline{V'}^1_{t'}, \quad \forall t' \leq T \qquad \text{(a1')}$$

$$\sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \leq V^1_0 + \sum_{t=1}^{t'} A^1_t - \underline{V'}^1_{t'}, \quad \forall t' \leq T \qquad \text{(b1')}$$

$$\sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \leq \overline{V'}^2_{t'} - V^2_0 - \sum_{t=1}^{t'} A^2_t, \quad \forall t' \leq T \qquad \text{(a2')}$$

$$\sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \geq \underline{V'}^2_{t'} - V^2_0 - \sum_{t=1}^{t'} A^2_t, \quad \forall t' \leq T \qquad \text{(b2')}$$

There are redundancies between (a1') and (b2'), and between (a2') and (b1'). Let us introduce bounds $\beta_{t'}$ and $\alpha_{t'}$ in the following way with $t' \leq T$:

$$\beta_{t'} = \max(V^1_0 + \sum_{t=1}^{t'} A^1_t - \overline{V'}^1_{t'}, \underline{V'}^2_{t'} - V^2_0 - \sum_{t=1}^{t'} A^2_t)$$

$$\alpha_{t'} = \min(V^1_0 + \sum_{t=1}^{t'} A^1_t - \underline{V'}^1_{t'}, \overline{V'}^2_{t'} - V^2_0 - \sum_{t=1}^{t'} A^2_t)$$

By using $\beta_{t'}$ and $\alpha_{t'}$, and rewriting the objective function, we obtain the formulation F_{1HUC} defined as follows:

$$\begin{aligned}
\max \quad & \sum_{t=1}^{T}\sum_{i=1}^{M}(\Lambda_t P_i - \Phi^1 D_i + \Phi^2 D_i)x_{t,i} \\
& + \Phi^1 \sum_{t=1}^{T} A^1_t + \Phi^2 \sum_{t=1}^{T} A^2_t \\
\text{s.c.} \quad & \sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \geq \beta_{t'} \quad \forall t' \leq T && \text{(a)} \\
& \sum_{t=1}^{t'}\sum_{i=1}^{M} D_i x_{t,i} \leq \alpha_{t'} \quad \forall t' \leq T && \text{(b)} \\
& x_{t,i} \geq x_{t,i+1} \quad \forall t \leq T, \forall i \leq M-1 && \text{(c)} \\
& x_{t,i} \in \{0, 1\} \quad \forall t \leq T, \forall i \leq M && \text{(f)}
\end{aligned}$$

With this formulation, the objective function is to maximize the value of each active operating point, plus a constant. Also, one can see (a) and (b) as resource window constraints, or (a) as nested cover constraints and (b) as nested knapsack constraints. This formulation can also be improved by tightening the window constraints, as shown at the end of Section V-A.

IV. Graph representation

The 1-HUC can be represented graphically in two ways, namely in a fashion similar either to the knapsack problem, or to the RCSPP. In the following we describe both representations and their dominance rules. We first introduce the cumulated flows between two time periods.

Definition 1 (Cumulated flow $\mathcal{D}_{t',t}$)**.** The cumulated flow $\mathcal{D}_{t',t}$ is the sum of the flows from time periods t' to t:

$$\mathcal{D}_{t',t} = \sum_{t''=t'}^{t}\sum_{i=1}^{M} D_i x_{t'',i}$$

A. Representation as a knapsack problem

Let $G_{KP} = (V_{KP}, A_{KP})$ defined as follows. Each vertex $u \in V_{KP}$ is defined as a pair $u = (t, d)$ with t the time period, and d the cumulated flow $\mathcal{D}_{1,t}$ with variables x associated to a path that reaches u. Without loss of generality, $u = (t, d)$ is considered only if $d \in [\beta_t; \alpha_t]$. The source vertex s is defined as $s = (0, 0)$. For each vertex $u = (t, d)$ and $v = (t + 1, d + \sum_{j=0}^{i} D_j)$ with $t < T$ and $i \in \{0, \ldots, M\}$ there is an arc $(u, v) \in A_{KP}$, of value $\sum_{j=0}^{i} \Lambda_t P_j - \Phi^1 D_j + \Phi^2 D_j$.

The downside of such a graph is its exponential number of vertices. However, by tightening the bounds as shown in Section V-A, it is possible to drastically reduce the number of vertices in G_{KP}. Furthermore, we can use classical dominance rules for the longest path:

Definition 2 (Dominance rule 1)**.** Let p and q be two paths from s to a vertex u. By induction the path with the lowest value is dominated, as it cannot lead to an optimal solution.

Definition 3 (Dominance rule 2)**.** Let p be a path from s to u going through a vertex v, and q be a path from s to v. Let $p_{s,v}$ be the subpath of p, from s to v and $p_{v,u}$ the subpath of p from v to u. If the value of $p_{s,v}$ is larger than that of q, then q is dominated. If the value of q is larger than that of $p_{s,v}$, then p is dominated by the path concatenating q and $p_{v,u}$.

Example 2. Consider an instance of the 1-HUC with $T = 3$, $M = 3$. The operating points are $(4, 4)$, $(6, 5)$ and $(8, 6)$. The bounds are $\beta{=}[0, 0, 5]$ and $\alpha{=}[12, 12, 10]$. Fig. 2a represents the graph G_{KP} associated to this instance.

B. Representation as an RCSPP

Let $G_R = (V_R, A_R)$ be defined as follows. Each vertex $u \in V_R$ is defined as a pair $u = (t, i)$, with t the time period and i the operating point. The source s is defined as $s = (0, 0)$. From each vertex $u = (t, i)$, with $t < T$, and $v = (t + 1, i')$ with $i' \in \{0, \ldots, M\}$ there is an arc $(u, v) \in A_R$. Arc a towards v has value $\sum_{j=0}^{i'} \Lambda_t P_j - \Phi^1 D_j + \Phi^2 D_j$ and consumes $\sum_{j=0}^{i'} D_j$ amount of the resource.

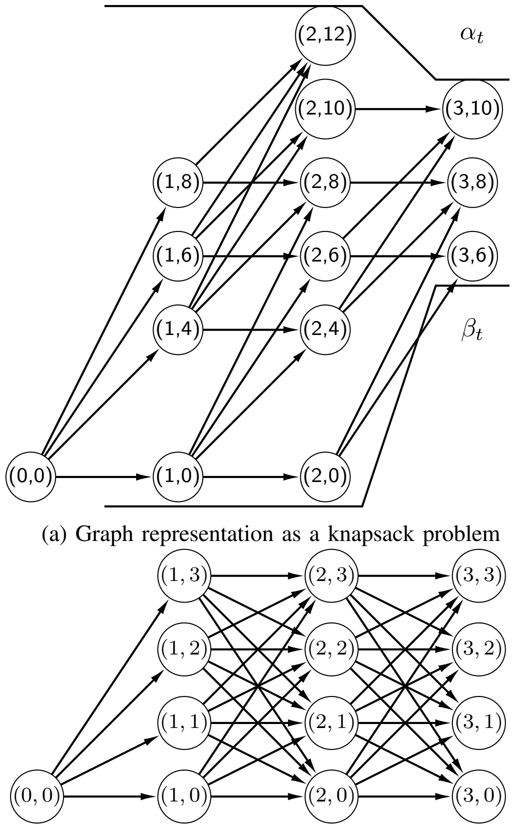
The downside of G_R, is that there are paths in this graph which do not verify the resource constraints, hence one needs to verify the resource constraints for each path. Note that the resource in this case is $\mathcal{D}_{1,t}$ for variables x associated to the path. On the positive side, the number of vertices is polynomial, and it is possible to use a classical dominance rule for the RCSPP as defined in [3]:

Definition 4 (Dominance rule 3)**.** If there are two partial paths from s to the same vertex u, by induction if a path has a lower value and uses more resource than another one, then this path is dominated provided both partial paths use sufficient resource to verify all lower bounds on the resource.

Note that the condition on the resource usage to ensure that the lower bounding constraints (a) are satisfied seriously weakens the dominance rules when these constraints are active.

Example 3. Consider the instance of Example 2. Fig. 2b represents the graph G_R associated to this instance.

There are two other graph representations for the 1-HUC described in the literature. In [20], the original graph defined for the RCSPP is similar to the one depicted in Fig. 2b. However, as mentioned in Section II-C, this graph is extended such that if multiple paths exist from vertex s to a vertex u, then u is duplicated such that only a single path leads to each vertex. The extended graph would be larger than the one depicted in Fig. 2a, as even with an exponential number of vertices, there are still multiple paths between s and most of the vertices. In [6], as described in Section II-B, the volume is evenly discretized. The resulting graph has similar vertices as in Fig. 2a, but at each time period, vertices only exist for volumes within the set of discretized volumes. In the experimental results of [6], it is stated that the discretization is ranging from 0.3% to 0.5% of the difference between the minimum and maximum volume of the reservoir, which represents about 300 vertices at each time period. Such a graph heavily differs from the graph we consider, as there can be a very large number of vertices at each time period.



(a) Graph representation as a knapsack problem

(b) Graph representation as an RCSPP

Fig. 2: Graph representations of the 1-HUC

V. Exact A* variant for the 1-HUC

In this section, we describe the new algorithm proposed to solve the 1-HUC. The aim of this algorithm is to find the longest path in graph G_{KP} as described in Section IV-A. A difficulty of this graph is the exponential number of vertices, which is why we resort to a variant of the A* algorithm [7]. The A* algorithm is particularly efficient when the number of vertices is large as it involves an optimistic heuristic to guide the search, and to discard sub-optimal partial solutions. We denote the proposed exact variant of the A* algorithm for the 1-HUC by HA*. This algorithm involves a dedicated optimistic heuristic for the 1-HUC. To further improve the performance of this algorithm, we also present a pre-processing bound tightening procedure in order to reduce the number of vertices in G_{KP}.

In the following, we first present the bound tightening procedure, then the optimistic heuristic, i.e., an upper bound on the optimal value and finally the HA* algorithm.

A. Tightening the bounds

The bounds α_t and β_t from inequalities (a) and (b) can directly be used to prune infeasible vertices. Thus, tightening these bounds may lead to a smaller number of vertices, thus reducing the search space.

In order to tighten the bounds, we use the cumulated flow $\mathcal{D}_{t',t}$ as previously defined. At each time period, the flow is between 0 and $\sum_{i=1}^{M} D_i$. For any pair of time periods (t', t), $t' < t$, the lower bound $\underline{\mathcal{D}}_{t',t} = 0$ and the upper bound $\overline{\mathcal{D}}_{t',t} = (t - t' + 1)\sum_{i=1}^{M} D_i$ will never be violated by a feasible solution. Note that α_t and β_t are not bounds on the flow at time period t, but rather bounds on $\mathcal{D}_{1,t}$. If the gap between β_t and α_t is large, then one may develop a large number of vertices, sometimes leading to intractable instances. For example, in the case of the HUC, it is very common to have upper bounds α_t very large compared to the flows, and negative lower bounds β_t. We can therefore introduce bounds $\hat{\alpha}_t$ and $\hat{\beta}_t$ in the following way:

$$\hat{\alpha}_t = \min(\alpha_t, \overline{\mathcal{D}}_{1,t})$$
$$\hat{\beta}_t = \max(\beta_t, \underline{\mathcal{D}}_{1,t})$$

By using bounds $\hat{\alpha}_t$ and $\hat{\beta}_t$, we can drastically reduce the number of vertices. However, it is still possible to further reduce it. Suppose that at time period t the bounds are such that $\hat{\beta}_t > \hat{\beta}_{t+1}$. As the water flows are all non-negative, any vertex $u = (t+1, d)$ with $d < \hat{\beta}_t + \underline{\mathcal{D}}_{t+1,t+1}$ cannot be part of a feasible solution. Similarly, if $\hat{\beta}_{t-1} < \hat{\beta}_t$, any vertex $u = (t-1, d)$ with $d < \hat{\beta}_t - \overline{\mathcal{D}}_{t,t}$ cannot be part of a feasible solution. Extending this logic to the upper bounds on d, we can tighten the bounds of any time period from the bounds of any other time period, following the rules below. Let a pair of time periods (t', t) with $t' < t$. Then, $\mathcal{D}_{1,t}$ must stay in $[\hat{\beta}_{t'} + \underline{\mathcal{D}}_{t'+1,t}; \hat{\alpha}_{t'} + \overline{\mathcal{D}}_{t'+1,t}]$ and $\mathcal{D}_{1,t'}$ lies in $[\hat{\beta}_t - \overline{\mathcal{D}}_{t'+1,t}; \hat{\alpha}_t - \underline{\mathcal{D}}_{t'+1,t}]$.

Let us define $\tilde{\alpha}_t$ and $\tilde{\beta}_t$ as follows:

$$\tilde{\alpha}_t = \min(\min_{t'<t}(\hat{\alpha}_{t'} + \overline{\mathcal{D}}_{t'+1,t}), \min_{t'>t}(\hat{\alpha}_{t'} - \underline{\mathcal{D}}_{t+1,t'}))$$
$$\tilde{\beta}_t = \max(\max_{t'<t}(\hat{\beta}_{t'} + \underline{\mathcal{D}}_{t'+1,t}), \max_{t'>t}(\hat{\beta}_{t'} - \overline{\mathcal{D}}_{t+1,t'})$$

Tight bounds α^*_t and β^*_t are calculated as follows:

$$\alpha^*_t = \min(\hat{\alpha}_t, \tilde{\alpha}_t)$$
$$\beta^*_t = \max(\hat{\beta}_t, \tilde{\beta}_t)$$

Note that computing all bounds β^*_t is of complexity T^2. Indeed, for a given t, computing β_t as well as $\tilde{\beta}_t$ both require one comparison, computing $\tilde{\beta}_t$ needs T comparisons and computing β^*_t requires one comparison. Hence, computing all β^*_t is of complexity T^2. We obtain a similar complexity for upper bounds α^*_t.

TABLE I: Reducing the search space using bounds on the flows

(a) Table with bounds α_t and β_t

t	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
0															
1	X														X
2	X														X
3	X	X	X	X	X		X	X	X	X	X	X	X	X	X
4	X														X
5	X														X
6	X	X	X	X	X	X	X	X		X	X	X	X	X	X

(b) Table with bounds $\hat{\alpha}_t$ and $\hat{\beta}_t$

t	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
0															
1	X	**X**	**X**				**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**	X
2	X	**X**	**X**						**X**	**X**	**X**	**X**	**X**	**X**	X
3	X	X	X	X	X		X	X	X	X	X	X	X	X	X
4	X	**X**	**X**										**X**	**X**	X
5	X	**X**	**X**												X
6	X	X	X	X	X	X	X	X		X	X	X	X	X	X

(c) Table with bounds α^*_t and β^*_t

t	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11
0															
1	X	X	X				X	X	X	X	X	X	X	X	X
2	X	X	X				**X**	**X**	X	X	X	X	X	X	X
3	X	X	X	X	X		X	X	X	X	X	X	X	X	X
4	X	X	X	**X**	**X**				**X**	**X**	**X**	**X**	X	X	X
5	X	X	X	**X**	**X**	**X**				**X**	**X**	**X**	**X**	**X**	X
6	X	X	X	X	X	X	X	X		X	X	X	X	X	X

Example 4. Let us define an instance of the 1-HUC with $T = 6$. Bounds are $\beta_3 = 2$, $\beta_6 = 5$, $\alpha_3 = 2$, $\alpha_6 = 5$, and $\beta_t = -2$, $\alpha_t = 10$ for t in $\{1, 2, 4, 5\}$. The operating points are such that at each time period, the maximum flow is 2. By applying tighter bounds we can see that we drastically reduce the possibilities, thus the number of vertices potentially developed by dynamic programming. In Table Ia the invalid values for the total flow at each time period, with respect to bounds β_t, are marked with a cross. Table Ib is similar to Table Ia with tighter bounds $\hat{\alpha}_t$ and $\hat{\beta}_t$, the crosses being in bold to emphasize the tightening of the bounds. Table Ic follows the same representation with the tightest bounds β^*_t and α^*_t.

As previously mentioned, formulation F_{1HUC} can be improved, considering bounds β^*_t and α^*_t instead of β_t and α_t in constraints (a) and (b). We denote by F^+_{1HUC} the formulation F_{1HUC} with the bound tightening.

In the following, we consider the graph G_{KP} with the tightest bounds β^*_t and α^*_t, denoted by G^*_{KP}.

B. Optimistic heuristic

In the case of the 1-HUC, an optimistic heuristic overestimates the value of the objective function because we are solving a maximization problem. Let p be a path from time period 1 to t representing already taken decisions. The heuristic aims at computing an optimistic cost from time period $t+1$ to T. The idea of the proposed optimistic heuristic is to compute an improved linear relaxation on time periods $t+1$ to T. To do so, we define quadruplets $(t, i, val, flow)$, with t a time period, i an operating point, val the value of any arc $\big((t-1, d), (t, d+\sum_{j=1}^{i} D_j)\big)$ in G^*_{KP}, and $flow$ the value $\sum_{j=1}^{i} D_j$. The aim is to progressively increase the values of variables $x_{t,i}$ depending on their profitability, being $val/flow$.

Algorithm 1 describes how to compute a linear relaxation on time periods $t+1$ to T from a partial path in graph G^*_{KP}, as detailed in the following four steps.

Step 1: Initialize a fractional solution with $x_{t',i} = 1$ if the partial solution represented by path p requires operating point $i \leq M$ at time period $t' \leq t$, $x_{t',i} = 0$ otherwise. This step is represented by lines 1 to 8 of Algorithm 1

Step 2: Initialize a list with all the quadruplets at time period $t' \in [t+1; T]$ by decreasing profitability $val/flow$. This step is represented by lines 9 to 14 of Algorithm 1

Feasibility step 3: This step is repeated as long as lower bounding constraints (a) are not verified (the while loop at line 15 of Algorithm 1). The algorithm looks for the smallest time period t' such that (a) is not satisfied, and for $x_{t'',i}$ with $t'' \in [t+1; t']$ maximizing profitability $val/flow$. The variable $x_{t'',i}$ is increased depending on its profitability:

- If the profitability is positive, fractionally increase $x_{t'',i}$ as much as possible provided all upper bounds α are satisfied.
- Otherwise, fractionally increase $x_{t'',i}$ as little as possible provided all upper bounds and the lower bound $\beta_{t'}$ are satisfied.

All variables $x_{t'',i'} < x_{t'',i}$ with $i' < i$ must be set to the same value as $x_{t'',i}$ in order to satisfy the order constraints. Also, for any $i' > i$, quadruplets (t'',i',val',$flow'$) are updated as (t'',i',$val' - val$,$flow' - flow$). This is because as $x_{t'',i} > x_{t'',i'}$, one can increase $x_{t'',i'}$ without increasing $x_{t'',i}$ while still verifying order constraints. All variables that cannot be further increased due to the upper bounds are removed from the list.

Optimality step 4: This step is repeated as long as there is a variable of positive profitability in the list of variables (again the while loop at line 15 of Algorithm 1). Select the first variable of the list, and fractionally increase its value as much as possible provided all upper bounds are satisfied. Remove from the list all variables that cannot be further increased due to the upper bound.

Property 1. *The fractional solution returned by Algorithm 1 verifies order constraints.*

Proof. Consider two quadruplets (t,i,val,$flow$) and (t,i',val',$flow'$), with t a time period considered in the heuristic, and $i' > i$. At the start of the algorithm, $x_{t,i} = x_{t,i'} = 0$. If $x_{t,i'}$ is increased, then $x_{t,i}$ is increased by the same amount, hence order constraints are verified. If $x_{t,i}$ is increased, it means that $val/flow > val'/flow'$. Hence, $val/flow > (val' - val)/(flow' - flow)$. Consequently, the algorithm will increase $x_{t,i'}$ only if $x_{t,i} = 1$, hence order constraints are verified. □

Theorem 1. *Algorithm 1 defines an optimistic heuristic.*

Proof. Let s be an integer solution for the 1-HUC, for which variables $x_{t,i}$ verify constraints (a) (b) (c) and (f). Let $\hat{s}$ be a fractional solution for the 1-HUC obtained with Algorithm 1, for which $\hat{x}_{t,i}$ verify all constraints (a), (b) (c), and constraint (f) only for time periods 1 to $t \leq T$. Consider $\hat{s}$ and s to be identical for time periods 1 to t.

Let $\mathcal{X}$ (resp. $\mathcal{Y}$) be the variables such that $x_{t',i} < \hat{x}_{t',i}$ $\forall x_{t',i} \in \mathcal{X}$ (resp. $x_{t',i} > \hat{x}_{t',i}$ $\forall x_{t',i} \in \mathcal{Y}$).

Clearly, for each variable $x_{t',i} \in \mathcal{X}$ of positive profitability, a fractional value for $x_{t',i}$ increases the value of the objective function compared to $x_{t',i} = 0$. Similarly, for each variable in $\mathcal{Y}$ of negative profitability, a fractional value for $x_{t',i}$ increases the value of the objective function compared to $x_{t',i} = 1$.

Let $\mathcal{X}^- \subseteq \mathcal{X}$ and $\mathcal{Y}^- \subseteq \mathcal{Y}$ be the variables with negative profitability. Suppose $|\mathcal{X}^-| > 0$. By construction of $\hat{s}$, the variables of $\mathcal{X}^-$ have value greater than 0 only in order to yield a feasible solution, with respect to a lower bound $\beta_{t'}$. In such a case the total flow of $\hat{s}$ from time period 1 to t' is exactly $\beta_{t'}$ by construction of $\hat{s}$. As solution s also verifies all lower bounds, we deduce $|\mathcal{Y}^-| > 0$. Otherwise there is either a contradiction with the construction of $\hat{s}$, or s does not verify all lower bounds. Hence, the total flow of s from time period 1 to t' is at least $\beta_{t'}$. By definition, s and $\hat{s}$ are identical from time period 1 to t, consequently the only difference is on time periods $t+1$ to t'. By construction of $\hat{s}$ the variables of $\mathcal{X}^-$ are the most profitable and have fractional value in $\hat{s}$. Consequently, their weighted value in the objective function must be higher than those of $\mathcal{Y}^-$ in the integer solution s.

A similar proof can be made for variables in $\mathcal{Y}^+ \subseteq \mathcal{Y}$ and $\mathcal{X}^+ \subseteq \mathcal{X}$ the variables with positive profitability.

The value of $\hat{s}$ is then greater or equal to the value of s. □

It is possible to tighten the fractional solution returned by Algorithm 1 while keeping it optimistic. Clearly, an integer solution is necessarily of a total flow which is a combination of the flows from the operating points. Therefore the flow of an integer solution is necessarily a multiple of the greatest common divisor (GCD) of the operating points' flows. When the heuristic increases the value of a variable, we can increase or reduce this value so that the total flow of the returned solution remains a multiple of the GCD relative to the operating points' flows. Note that since the flows are identical from one time period to another, we can quickly compute the GCD by considering only the flows of a single time period.

C. HA* algorithm

For a 1-HUC with an objective function to maximize, the principle of HA* is the following. Consider a pool of partial solutions evaluated with the heuristic. At each iteration, the partial solution with the highest heuristic value is considered and removed from the pool. From the partial solution considered, we complement it by adding neighbors relative to its last vertex. Once a solution is found, its value is used as a bound to remove some more partial solutions from the pool. Indeed, if the solution's value is higher than a partial solution's heuristic

Algorithm 1 Algorithm OptimisticHeuristic

```
Require: A path $p$ from time period 1 to $t$, a graph $G^*_{KP}$, $GCD$ the GCD of the flow
Initialize a fractional solution $\hat{x}$ with all variables to 0
for $t' \in [1;T]$ and $i \in [1;M]$ do
    if $t' \leq t$ AND $p$ requires operating point $i$ at time period $t'$ then
        $\hat{x}_{t',i} = 1$
    else
        $\hat{x}_{t',i} = 0$
    end if
end for
Initialize a list $L = []$
for $t' \in [t+1;T]$ and $i \in [1;M]$ do
    $flow \leftarrow \sum_{i'=1}^{i} D_{i'}$
    $val$ the value of an arc towards $(t', i)$ in $G^*_{KP}$
    add $(t', i, val, flow)$ in $L$, sorted by decreasing $val/flow$
end for
while $\exists t' \in [t+1, T]$ such that $\hat{x}$ does not verify $\beta_{t'}$ OR exists quadruplet in $L$ with $val > 0$ do
    $(t'', i, val, flow) \leftarrow$ first in $L$ with $t'' \leq t'$
    if $val \leq 0$ then
        set $\hat{x}_{t'',i}$ to minimum such that $\beta_{t'}$ verified and $x_{t'',i}{\cdot}flow \mod GCD = 0$; if $\beta_{t'}$ cannot be verified $\hat{x}_{t'',i} \leftarrow 1$
    else
        set $\hat{x}_{t'',i}$ to maximum such that all upper bounds are verified and $x_{t'',i} \cdot flow\%GCD = 0$
    end if
    for $i' \in [1;i]$ do
        $\hat{x}_{t'',i'} = \max(\hat{x}_{t'',i'}, \hat{x}_{t'',i})$
    end for
    for $(t'', i', val', flow') \in L$ with $i' \in ]i;M]$ do
        $val' \leftarrow val' - val$
        $flow' \leftarrow flow' - flow$
    end for
    for $(t'', i, val, flow) \in L$ such that $\hat{x}_{t'',i}$ cannot be increased do
        remove $(t'', i, val, flow)$ from $L$
    end for
end while
return the value of $\hat{x}$
```

value, then the partial solution can be removed from the pool. Once the pool of partial solutions is empty, the algorithm stops and the best solution found is the optimal solution.

We underline the need of a tight optimistic heuristic. If the heuristic is not optimistic, or optimistic but too loose, there are fewer cases where one can prune partial solutions while guaranteeing optimality. Hence, more vertices are developed which can exponentially increases the computational time.

For readability purposes, we introduce three structures:

Definition 5 (Path structure)**.** The path structure has three attributes: $vertices$ the list of vertices of the path; val the value of the path with respect to the objective function; $heur$ the optimistic heuristic value.

Definition 6 (Vertex structure)**.** A vertex structure has two attributes: t the time period and d the cumulated flow $\mathcal{D}_{1,t}$, as defined for the vertices of G^*_{KP} in Section IV.

Definition 7 (Arc structure)**.** The arc structure has one attribute: val the value as defined for the arcs of G^*_{KP} in Section IV.

Algorithm 2 presents the pseudo-code of HA*, using the three previously described structures as well as $OptimisticHeuristic$. The considered graph is G^*_{KP}, which is G_{KP} as illustrated in Fig. 2a with bounds β^*_t and α^*_t. The dominance rules used in Algorithm 2 are the dominance rules 1 and 2.

Algorithm 2 Algorithm HA*

```
Require: A graph $G^*_{KP}$
Initialize a path $p$ as follows: $p.vertices = \{(0,0)\}$, $p.val = 0.0$, $p.heur = OptimisticHeuristic(p)$
Initialize a list of paths $L_p = [p]$
Initialize the value of the best solution $bestVal = -\infty$
while $L_p$ not empty do
    $p \leftarrow$ first path in $L_p$
    remove $p$ from $L_p$
    $v \leftarrow$ last vertex of $p.vertices$
    for arc $a$ from $v$ to $u$ do
        $q.vertices = p.vertices \cup u$, $q.val = p.val + a.val$, $q.heur = OptimisticHeuristic(q)$
        if $|q.vertices| = T + 1$ then
            $bestVal \leftarrow \max(bestVal, q.val)$
            remove $q' \in L_p$ with $q'.val + q'.heur \leq bestVal$
        else
            $dom \leftarrow FALSE$
            for $q' \in L_p$ do
                if $q'$ dominates $q$ then
                    $dom \leftarrow TRUE$
                end if
                if $q$ dominates $q'$ then
                    remove $q'$ from $L_p$
                end if
            end for
            if $dom = FALSE$ and $q.val + q.heur > bestVal$ then
                add $q$ in $L_p$ by keeping $L_p$ sorted by decreasing $q.val + q.heur$
            end if
        end if
    end for
end while
return the solution of value $bestVal$
```

VI. Experimental results

Results are computed on a single thread of an Intel(R) Core(TM) i7-9850H CPU @ 2.60GHz processor, with 2 CPUs of 8 cores, with Linux as operating system. All algorithms are developed with C++. Version 12.8 of CPLEX with default setting is used to solve the MILP formulation F_{1HUC}.

A. Instances

From a large set of realistic instances derived from a real EDF plant, a first set A of 13 instances is obtained. These 13 instances are retained as preliminary results have shown that formulation F_{1HUC} is not trivially solved. This emphasizes the need of an efficient alternative in these cases. Table II depicts for each instance the main characteristics, namely the number of time periods, the number of operating points and the presence of a constraining minimum (resp. maximum) bound on the volume at the last time period. The instances cover three cases, namely when there is only an upper bound, only a lower bound, or a target volume with a constraining upper and lower bound. In the case of an equality constraint, the instance becomes infeasible, which is out of the scope of the instances considered in this paper. Hence, the target volumes are not equality constraints, but rather tight window constraint. For instances with a target volume, more resource window constraints are obtained by propagating the bounds β^* and α^* from the bounds at the last time period to the previous ones.

The water flows D and powers P are in the order of 10^3 to 10^4, with volumes in the order of 10^7. For target volumes, the difference between the upper and lower bound is in the order of 10^3, which is small enough with respect to the flows to yield very few vertices at time period T in G^*_{KP}.

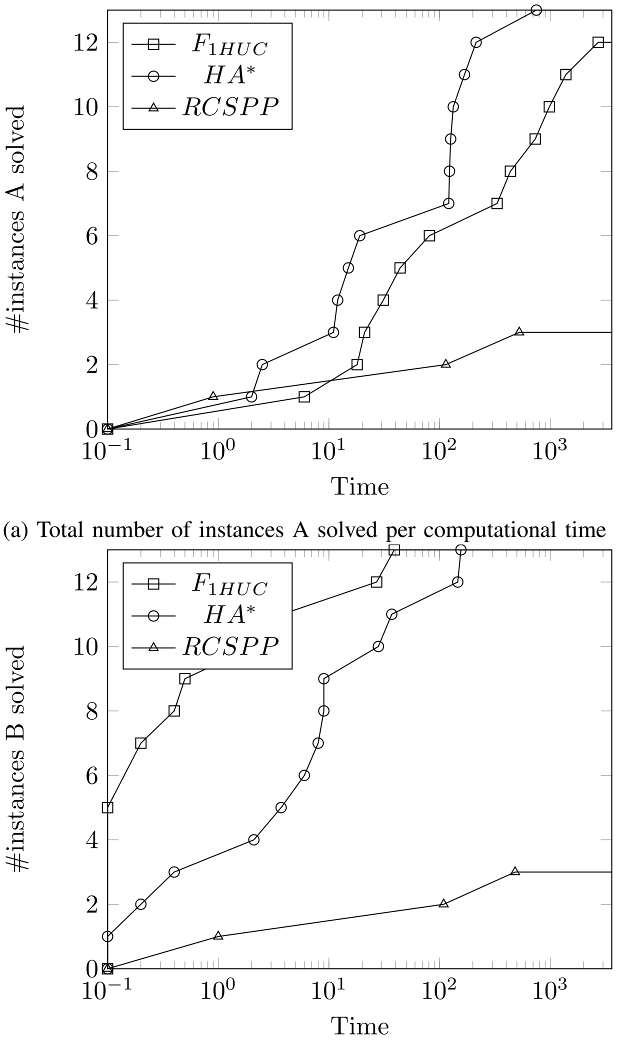
TABLE II: Main characteristics of instances set A

instance	T	M	minimum volume	maximum volume
1	96	4	✗	✓
2	96	4	✓	✗
3	96	4	✓	✓
4	96	7	✗	✓
5	96	7	✓	✗
6	96	7	✓	✓
7	96	8	✗	✓
8	96	8	✓	✓
9	96	15	✗	✓
10	96	17	✗	✓
11	96	18	✓	✗
12	96	21	✗	✓
13	96	18	✗	✓

A second set B of 13 instances, similar to the first set, is also constructed. The only differences are bounds β_T and α_T that are shifted as follows. Let an instance with bounds β^A_T and α^A_T be in set A. A random value $k \in [-9999; -1000] \cup [1000; 9999]$ is chosen. Bounds of an instance for set B are $\beta^B_T = \beta^A_T + k$ and $\alpha^B_T = \alpha^A_T + k$. Note that this shift is very small for these instances. Indeed, the water flows can be in the order of 10^4, there are nearly 100 time periods and the cumulated flow $\mathcal{D}_{1,T}$ is in the order of 10^6. The shift k is at most 1% of $\overline{\mathcal{D}}_{1,T}$. As shown in the following results, slightly modifying an instance can drastically impact the computational time.

B. Experimental results

In order to benchmark HA*, all instances are solved with HA* as well as with two alternative methods. The first alternative is a classical RCSPP algorithm [12] adapted to the 1-HUC [11]. The second alternative is to use CPLEX to solve F_{1HUC} described in Section III. A third alternative is solving F^+_{1HUC}, which is formulation F_{1HUC} described in Section III with tighter bounds described in Section V-A. However, solving F^+_{1HUC} leads to very similar results to F_{1HUC}, hence results for F^+_{1HUC} are not included in the following. All algorithms use a single thread, with a time limit of one hour.



(a) Total number of instances A solved per computational time

(b) Total number of instances B solved per computational time

Fig. 3: Total number of instances A and B solved per computational time

Figures 3a and 3b represent the number of instances solved by each algorithm with respect to the computational time.

Clearly, for instance set A, HA* is the most efficient alternative. Indeed, it solves every instance and requires less time than the other alternatives. For instance set B, solving F_{1HUC} is the most efficient alternative. Note that the difference between the solving times of F_{1HUC} and HA* is only of a few seconds for instance set B, as most instances are solved in less than 10 seconds with HA*. Solving F_{1HUC} is the least robust alternative when it comes to computational times. Indeed, when comparing the results between instance sets A and B, the computational times are drastically different with F_{1HUC}, whereas for HA* there is a smaller difference, and for the RCSPP algorithm the results are the same. The RCSPP algorithm fails to solve 10 out of 13 instances, for both instance sets A and B. This shows that the RCSPP algorithm is inefficient at solving the 1-HUC. We further explain the results in the following, by introducing Tables III and IV with detailed results.

Tables III and IV give, for each instance, the value of the objective function and the computational times obtained for all algorithms, as well as the optimality gap and the number of Branch & Bound nodes returned by the MILP solver. If the MILP solver proves optimality, the gap is noted "opt". When the time limit is reached, the time is noted "-". Results are presented for each instance individually, as well as the average (avg) and the standard deviation (sd) for the solved instances. When the time limit is reached, a computational time of 3600 seconds is accounted for in the average and the standard deviation. The computational time of the most efficient algorithm is emphasized in bold for each instance, as well as the average and the standard deviation for each set of instances.

There is a clear difference in computational time between set A and B. Indeed, F_{1HUC} is solved for 12 out of the 13 instances of set A, and needs between 6 and 2713 seconds, while it is solved for all instances of set B, most of them instantaneously. Similarly, HA* solves all instances of set A and needs between 2 and 748 seconds, while it solves all instances of set B in less than 156 seconds. Note however that there is no noticeable computational time difference between solving instance set A and B with the RCSPP algorithm.

The RCSPP algorithm is only able to solve instances 2, 5 and 11, for both sets. This is because for any other instance, the maximum volume of the upstream reservoir is constrained at the last time period. In this case, the value β_T becomes positive, meaning that there is a minimum bound on the resource. As mentioned in [3], when there is a minimum bound on the resource, the dominance properties of the RCSPP algorithm cannot always be applied, thus leading to large computational times. Clearly, HA* outperforms the RCSPP algorithm. Even when there is no minimum bound on the resource, the RCSPP algorithm yields larger computational times for all instances of set B and instance 5 of set A.

When comparing HA* algorithm to solving F_{1HUC} on the most difficult instances, the former outperforms the latter in terms of computational time and number of instances solved. On the one hand, HA* is more stable with respect to the computational times. Indeed, HA* only requires more than 10 minutes once (instance 12 of set A), while solving F_{1HUC} requires more than 10 minutes for 5 of the 26 instances (instances 1, 3, 10, 11 and 12 of set A). Moreover, F_{1HUC} is not solved to optimality for instance 12 of set A, and the best value found is not the optimal value obtained with HA*. On the other hand, solving F_{1HUC} appears to be more efficient on easier instances than HA*. In this case, there are numerous instances where the difference between the two approaches is within a few seconds (instances 1, 2, 3, 4 and 6 of set B).

The stability with respect to the computational time is noticeable on the average and standard deviation. Indeed, the average time difference between set A and B is much smaller for HA* than for the MILP solver. The standard deviation for HA* is much smaller on set A, and slightly higher for set B when compared to solving F_{1HUC}. Besides, one can compute the total average and total deviation of the resolution time for all 26 instances. The total average time and standard deviation are 402.1 seconds and 873.7 seconds for the MILP solver, and 80.3 seconds and 149.0 seconds for HA*.

VII. Conclusion

In this paper, the HA* algorithm is proposed as an exact variant of the A* algorithm dedicated to the 1-HUC. It has been adapted through a dedicated optimistic heuristic and a bound tightening pre-processing, propagating lower and upper bounds from any time period to another. On a set of realistic instances, algorithm HA* is shown to be both more stable and more efficient on average in terms of computational times compared to solving F_{1HUC}. Also, HA* outperforms the standard labeling algorithm for the RCSPP.

A natural direction for a future work would be to extend the proposed algorithm in order to take into account additional constraints of the 1-HUC. A promising perspective is to include the HA* algorithm in a decomposition of a hydraulic valley with multiple plants. Beyond the HUC, another interesting perspective could also be to try and generalize such an algorithm for other problems with window resource constraints.

References

[1] A. Renaud, “Daily generation management at Electricité de France: from planning towards real time,” *IEEE Transactions on Automatic Control*, vol. 38, no. 7, pp. 1080–1093, 1993.

[2] G. Hechme-Doukopoulos, S. Brignol-Charousset, J. Malick, and C. Lemaréchal, “The short-term electricity production management problem at EDF,” *Optima Newsletter*, vol. 84, pp. 2–6, 2010.

[3] W. van Ackooij, C. d'Ambrosio, D. Thomopulos, and R. S. Trindade, “Decomposition and shortest path problem formulation for solving the hydro unit commitment and scheduling in a hydro valley,” *European Journal of Operational Research*, vol. 291, no. 3, pp. 935–943, 2021.

[4] G. Ardizzon, G. Cavazzini, and G. Pavesi, “A new generation of small hydro and pumped-hydro power plants: Advances and future challenges,” *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 746–761, 2014.

[5] Y. Sahraoui, P. Bendotti, and C. d'Ambrosio, “Real-world hydro-power unit-commitment: Dealing with numerical errors and feasibility issues,” *Energy*, vol. 184, pp. 91–104, 2019.

TABLE III: Performance of solving F_{1HUC}, the RCSPP algorithm and HA* on instance set A

	F_{1HUC}				RCSPP		HA*	
instance	value	#nodes	gap	time	value	time	value	time
1	$-25\,428.80$	10 232 857	opt	2713.4	-	-	$-25\,428.80$	**2.5**
2	43 010.90	2 591 995	opt	437.0	43 010.9	**0.9**	43 010.90	2.3
3	3556.54	5 034 351	opt	1384.8	-	-	3556.54	**12.0**
4	2462.90	167 490	opt	44.1	-	-	2462.90	**11.1**
5	111 115.00	101 570	opt	**17.8**	111 115.0	525.2	111 115.00	18.8
6	-1706.62	888 735	opt	331.4	-	-	-1706.62	**14.9**
7	5692.65	115 626	opt	**6.2**	-	-	5692.65	120.6
8	16 581.10	487 218	opt	**30.9**	-	-	16 581.10	126.0
9	$-71\,645.90$	176 123	opt	**21.4**	-	-	$-71\,645.90$	133.0
10	$-525\,446.00$	901 533	opt	732.1	-	-	$-525\,446.00$	**168.4**
11	44 421.20	1 535 139	opt	982.0	44 421.2	**114.1**	44 421.20	213.9
12	$-20\,329.90$	1 624 614	0.58	-	-	-	$-20\,324.00$	**748.8**
13	$-103\,435.00$	365 253	opt	**80.7**	-	-	$-103\,435.00$	122.8
avg	-	1 865 577	0.58	798.6	-	1431.0	-	**130.4**
sd	-	2 755 062	0.0	1101.1	-	2818.4	-	**191.7**

TABLE IV: Performance of solving F_{1HUC}, the RCSPP algorithm and HA* on instance set B

	F_{1HUC}				RCSPP		HA*	
instance	value	#nodes	gap	time	value	time	value	time
1	$-25\,430.30$	873	opt	**0.2**	-	-	$-25\,430.30$	0.4
2	42 993.00	0	opt	**0.0**	42 993.00	1.0	42 993.00	**0.0**
3	3700.23	11	opt	**0.0**	-	-	3700.23	0.2
4	2462.90	16 330	opt	**2.1**	-	-	2462.90	3.7
5	111 143.00	34 982	opt	**3.8**	111 143.00	481.5	111 143.00	8.4
6	-1460.33	2519	opt	**0.4**	-	-	-1460.33	2.1
7	5936.31	522 284	opt	**39.2**	-	-	5936.31	155.7
8	16 730.40	6403	opt	**0.5**	-	-	16 730.40	8.6
9	$-132\,290.00$	0	opt	**0.0**	-	-	$-132\,290.00$	9.2
10	$-525\,246.00$	46 425	opt	**27.1**	-	-	$-525\,246.00$	37.2
11	44 646.80	0	opt	**0.0**	44 646.80	109.1	44 646.80	6.2
12	$-20\,153.10$	1586	opt	**0.1**	-	-	$-20\,153.10$	146.3
13	$-103\,339.00$	2314	opt	**0.2**	-	-	$-103\,339.00$	27.7
avg	-	48 748	-	**5.7**	-	1437.1	-	31.2
sd	-	137 446	-	**12.0**	-	2814.7	-	52.2

[6] J. I. Pérez-Díaz, J. R. Wilhelmi, and L. A. Arévalo, "Optimal short-term operation schedule of a hydropower plant in a competitive electricity market," *Energy Conversion and Management*, vol. 51, no. 12, pp. 2955–2966, 2010.

[7] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[8] W. Fan, X. Guan, and Q. Zhai, "A new method for unit commitment with ramping constraints," *Electric Power Systems Research*, vol. 62, no. 3, pp. 215–224, 2002.

[9] R. Taktak and C. D'Ambrosio, "An overview on mathematical programming approaches for the deterministic unit commitment problem in hydro valleys," *Energy Systems*, vol. 8, no. 1, pp. 57–79, 2017.

[10] R. Bellman, "On a routing problem," *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[11] W. van Ackooij, C. d'Ambrosio, L. Liberti, R. Taktak, D. Thomopulos, and S. Toubaline, "Shortest path problem variants for the hydro unit commitment problem," *Electronic Notes in Discrete Mathematics*, vol. 69, pp. 309–316, 2018.

[12] C. Barrett, K. Bisset, M. Holzer, G. Konjevod, M. Marathe, and D. Wagner, "Engineering label-constrained shortest-path algorithms," in *Algorithmic Aspects in Information and Management: 4th International Conference, AAIM 2008, Shanghai, China, June 23-25, 2008. Proceedings 4.* Springer, 2008, pp. 27–37.

[13] A. Arce, T. Ohishi, and S. Soares, "Optimal dispatch of generating units of the Itaipú hydroelectric plant," *IEEE Transactions on power systems*, vol. 17, no. 1, pp. 154–158, 2002.

[14] C.-T. Cheng, S.-L. Liao, Z.-T. Tang, and M.-Y. Zhao, "Comparison of particle swarm optimization and dynamic programming for large scale hydro unit load dispatch," *Energy Conversion and Management*, vol. 50, no. 12, pp. 3007–3014, 2009.

[15] M. Kruber, A. Parmentier, and P. Benchimol, "Resource constrained shortest path algorithm for EDF short-term thermal production planning problem," 2018. [Online]. Available: https://arxiv.org/abs/1809.00548

[16] L. Turner, "Variants of the shortest path problem," *Algorithmic Operations Research*, vol. 6, no. 2, pp. 91–104, 2011.

[17] C. C. Ribeiro and M. Minoux, "A heuristic approach to hard constrained shortest path problems," *Discrete Applied Mathematics*, vol. 10, no. 2, pp. 125–137, 1985.

[18] J. E. Beasley and N. Christofides, "An algorithm for the resource constrained shortest path problem," *Networks*, vol. 19, no. 4, pp. 379–394, 1989.

[19] X. Zhu and W. E. Wilhelm, "Three-stage approaches for optimizing some variations of the resource constrained shortest-path sub-problem in a column generation context," *European journal of operational research*, vol. 183, no. 2, pp. 564–577, 2007.

[20] ——, "A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation," *Computers & Operations Research*, vol. 39, no. 2, pp. 164–178, 2012.

DOI: 10.15439/2023F9521
ISSN 2300-5963 ACSIS, Vol. 35

Clustering Corticosteroids Responsiveness in Sepsis Patients using Game-Theoretic Rough Sets

Rahma Hellali
Université Paris-Saclay,
UVSQ, DAVID, France
Email: rahma.hellali@uvsq.fr

Zaineb Chelly Dagdia
Université Paris-Saclay,
UVSQ, DAVID, France
Université de Tunis,
Institut Supérieur de Gestion
de Tunis, LARODEC, Tunisia
Email: zaineb.chelly-dagdia@uvsq.fr

Karine Zeitouni
Université Paris-Saclay,
UVSQ, DAVID, France
Email: karine.zeitouni@uvsq.fr

***Abstract*—Performing data mining tasks in the medical domain poses a significant challenge, mainly due to the uncertainty present in patients' data, such as incompleteness or missingness. In this paper, we focus on the data mining task of clustering corticosteroid (CS) responsiveness in sepsis patients. We address the issue and challenge of missing data by applying Game-Theoretic Rough Sets (GTRS) as a three-way decision approach. Our study considers the APROCCHS cohort, comprising 1240 sepsis patients, provided by the Assistance Publique–Hôpitaux de Paris (AP-HP), France. Our experimental results on the APROCCHS cohort indicate that GTRS maintains the trade-off between accuracy and generality, demonstrating its effectiveness even when increasing the number of missing values.**

I. Introduction

DUE TO its high mortality, incidence, and morbidity, sepsis is regarded as one of the most serious diseases that impact people's lives. The Third International Consensus Definition for Sepsis and Septic Shock (Sepsis-3), in 2016, defined sepsis as a "life-threatening organ dysfunction resulting from dysregulated host responses to infection" [1]. Immunologically, the human body releases some immune chemicals into the blood to fight the encountered infection. These released substances cause extensive inflammation, resulting in blood clots and leaking blood vessels. As a consequence, blood flow is disrupted, depriving organs of nutrition and oxygen, and hence, resulting in organ damage. The Sequential Organ Failure Assessment (SOFA) score [2] is used to codify the degree of organ dysfunction. It is difficult to estimate the global burden of sepsis. The study conducted in [3], estimated that in 2017 there were 48.9 million cases and 11 million sepsis-related deaths all over the globe, which accounted for almost 20% of all global deaths. There is no current diagnostic test for sepsis.

Knowing that there are still no specific interventions to control immune responses to invading pathogens [4], for sepsis, researchers have looked at the biological underpinnings of sepsis to see if there are any treatments that could help. Because of their impact on the immune system, corticosteroids have received a lot of attention [5]. The hormonal route from the hypothalamic-pituitary gland to the adrenal glands promotes corticosteroid synthesis in sepsis [6], [7]. These hormones affect inflammation through the formation of white blood cells, cytokines, and nitric oxide. The timing of corticosteroid administration may be a key component in therapy response. Short-term mortality was found to be higher in observational studies when hydrocortisone was started later. It is expected that corticosteroid treatment is advantageous for sepsis patients for these reasons and that variations in dose, timing, or duration of corticosteroid treatment may alter the patient response to treatment differently [8].

This paper delves into the data mining task [9] of clustering corticosteroid (CS) responsiveness in sepsis patients using the APROCCHS cohort provided by Assistance Publique–Hôpitaux de Paris (AP-HP), France. The cohort includes 1240 sepsis patients. A key challenge in this task is the presence of missing data.

Grouping data with missing values is one of the primary difficulties in clustering. There are commonly two strategies to deal with missing values [10]. The first strategy is based on preprocessing techniques [11]. Generally, it adopts deleting the whole row containing missing values or replacing the missing values based on experts' rules [12]. Some common missing values imputation techniques include replacing missing values with the mean, median, or mode of the available data for that feature [13]. The hot Deck Imputation method is replacing missing values by randomly selecting a value from another similar data point in the same dataset [14]. Using the values of K-nearest neighbors in the feature space to estimate the missing value [15]. Linear Regression Imputation aims to predict the missing values using linear regression based on other variables in the dataset [16].

The process of filling in missing values can potentially introduce a significant amount of imputation bias and uncertainty. It is important to recognize that missing values can be informative and carry meaningful implications. In certain instances, the absence of data itself can convey valuable information or signify a particular category or state. Imputing these missing values may result in distorting the original meaning or introducing artificial patterns into the dataset. In such situations, it is advisable to treat the missing values as a distinct category or conduct a separate analysis specifically on the subset of data that contains missing values. Also, imputation during preprocessing has reportedly been found to compromise the accuracy and consistency of classification outcomes. Thus, these methods are not recommended specifically when we deal with medical data because they can bias the medical results.

The second strategy relies on incorporating mechanisms in the clustering model [17] which means that we will not impute or preprocess the missing values, but the internal functioning of the algorithm will handle automatically the missing values. Examples of works belonging to this strategy are [18] which allocated missing value objects to a cluster with a large number of missing values and [19] which assigned objects containing missing data to clusters based on their neighbors.

Moreover, the theory of rough sets provides a valuable framework for analyzing incomplete information through the use of approximations [20]. This approach allows us to delve into the realms of uncertain and imprecise data, aiding in our understanding of complex systems. According to the research conducted by [21], the application of rough set theory has been observed across various fields and domains. In [22], authors integrated the Variable Precision Rough Set (VPRS) approach with Bayesian principles. In [23], the idea is to combine VPRS with fuzzy rough set methods to create flexible decision rules. In essence, both papers share a common objective of tackling information imprecision by employing probabilities (within the framework of VPRS) and fuzziness (which allows for handling partial matching of rules' antecedents). Their ultimate aim is to derive interpretable decision models from the available data. Authors in [24] introduced the Learn++.MF, an innovative ensemble-of-classifiers algorithm designed to address the challenge of missing features in supervised classification. It creates an ensemble of classifiers, each trained on a random subset of available features. When classifying instances with missing values, the algorithm employs majority voting from classifiers that were trained without the missing features. The study demonstrates that Learn++.MF effectively handles significant amounts of missing data, with only a gradual decline in performance as the missing data increases. In biomedicine and healthcare, rough set theory has been applied for disease diagnosis [25], medical image analysis [26], and patient profiling [27].

Focusing on the second strategy, mentioned above, and tackling the challenge of missing data in sepsis patients' records, we apply Game-Theoretic Rough Sets (GTRS) as a three-way decision approach. The aim is to assign patients with incomplete records to the appropriate clusters automatically. In order to study the efficiency of the algorithm application on our clinical data, we aim to answer the following research questions:

- **RQ1:** How can the percentage of missing values affect the performance of the algorithm?
- **RQ2:** Does the k nearest neighbors has an impact on the results?
- **RQ3:** Can the percentage of increasing and decreasing initial values of α and β influence the results?

The rest of this paper is structured as follows. Section II presents the fundamentals of three-way decisions using GTRS. Section III details the application of GTRS, as a three-way decision approach for handling missing data, for clustering CS-responsiveness in sepsis patients. The experimental setup is introduced in Section IV. The results of the performance analysis are discussed in Section V, and conclusions are presented in Section VI.

II. Three-way decisions using rough sets

A. Three-way clustering

The theoretical foundation of three-way clustering is based on the theory of three-way decisions introduced by Yao [28]. Assuming the existence of a set $U = \{o_1, o_2, o_3, ...\}$ which is referred to as the universe of objects, a clustering method will produce a collection of sets $\{c_1, c_2, c_3, ...\}$, where each set c_k contains a group of objects belonging to that specific cluster. Every object o_i in the set has A attributes, represented as $o_i = (o_i^1, ..., o_i^A)$, with o_i^a indicating the value of the a^{th} attribute associated with the i^{th} object.

In traditional clustering, a cluster is usually represented by a single set, indicating that objects within the set definitely belong to a cluster and those outside the set definitely do not belong to it. In situations characterized by uncertainty and a lack of information, two-way decisions are not always feasible from a decision-making perspective, such as in the case of clustering.

A practical and reasonable alternative is to adopt a three-way decision approach, which introduces three options for decision-making, rather than the traditional binary choice. Specifically, we can decide whether an object belongs to a cluster, whether it does not belong to a cluster, or whether it is uncertain whether the object belongs to a cluster or not. This concept of three-way decision-making leads to what is known as three-way clustering.

To define three distinct regions - inside, partial, and outside - an approach involving an evaluation function and a set of thresholds can be employed. The evaluation function quantifies the association or correlation between an object and a cluster, while the thresholds set limits on this relationship for inclusion in each of the regions. Let $e(c_k, o_i)$ be an evaluation function that represents the association between a specific cluster c_k and an object o_i, and let (α, β) be a pair of thresholds. The three regions are defined as follows.

$$Inside(c_k) = \{o_i \in U | e(c_k, o_i) \geq \alpha\}, \tag{1}$$

$$Outside(c_k) = \{o_i \in U | e(c_k, o_i) \leq \beta\}, \tag{2}$$

$$Partial(c_k) = \{o_i \in U | \beta < e(c_k, o_i) < \alpha\}, \tag{3}$$

This means that when the evaluation of an object is equal or above the threshold α, it is considered to be part of the Inside(c_k) group. Conversely, if the evaluation is at or below the threshold β, the object is regarded as being in the Outside(c_k) group. If the object's evaluation falls between the two thresholds, it is included in the Partial(c_k) group. Thus, inclusion in distinct regions is governed by the thresholds (α, β), and varying their settings results in different regions. The automatic determination of these thresholds is a crucial research topic in this context.

In this regard, and based on the work proposed in [29], we utilize the three-way framework to handle data with missing values which involves three steps. The overall functioning is presented in Figure 1. Initially, the set of objects U is partitioned into two sets: C and M. Set C comprises objects that have no missing data, while set M contains those that have missing values. Objects in set C are clustered using conventional algorithms, such as K-means [30], under the assumption that since these objects have no missing values, the level of uncertainty is low, and conventional approaches are more suitable for clustering such objects (Figure 1 (1)).

The second step (Figure 1 (2)) involves creating an incomplete data set from C while maintaining a similar rate of missing values to that of dataset U. For instance, if 30% of objects in the original dataset has missing values, approximately 30% of objects will be randomly chosen from C to induce missing values. This results in partitioning C into two additional sets: the constructed dataset comprising objects with missing values denoted as U_m, and the remaining objects in C with no missing values designated as U_c. This step assists in selecting appropriate values for (α, β) thresholds that will enable the clustering of objects with missing values.

The third step (Figure 1 (3)) involves determining the inclusion of objects with missing values, denoted as M, in the three-way framework. To employ three-way clustering on data with missing values, it is necessary to calculate the evaluation function $e(c_k, o_i)$, as specified in Equations 1, 2, and 3. This function measures the association between an object o_i and cluster c_k and can be defined in various ways. In our case, and as proposed in [29], we utilize an evaluation function that is based on the proportion of nearest neighbors for object o_i that belongs to cluster c_k:

$$e(c_k, o_i) = \frac{\text{Number of } o_i \text{ neighbors belonging to } c_k}{\text{Total neighbors of } o_i} \tag{4}$$

In order to determine the neighbors, a specific distance metric is required. For this example, we utilize the euclidean distance as follows:

$$d(i, j) = \sqrt{\sum_{a=1}^{A} \left(O_i^a - O_j^a\right)^2} \tag{5}$$

Here, o_i^a represents the value of the a^{th} attribute of the i^{th} object and any attributes with missing values are disregarded during distance computation. By utilizing the aforementioned distance metric, it is possible to calculate the distances of each o_i with missing values from all objects in U_c. After sorting these distances, the nearest neighbors for each o_i can be determined. Upon sorting these distances, the nearest neighbors can be identified. After determining the evaluation functions, Equations 1, 2, and 3 can be employed to determine the inclusion of objects into one of the three regions.

The goal of this approach is to enhance the clustering quality of data containing missing values. In this regard, two metrics need to be calculated based on the thresholds (α, β) as follows:

$$Accuracy(\alpha, \beta) = \frac{\text{Correctly clustered objects}}{\text{Total clustered objects}}, \tag{6}$$

$$Generality(\alpha, \beta) = \frac{\text{Total clustered objects}}{\text{Total objects in } U} \tag{7}$$

where *Accuracy* refers to how well we cluster objects with missing values, whereas *generality* refers to the fraction of objects that were clustered in the first place. Thus, as defined in [29], this goal can be approached from the perspective of a trade-off between accuracy and generality of the clustering.

B. Game theoretic rough sets

GTRS is based on a game-theoretic concept and formulation to estimate thresholds of the three-way decisions [31], [32]. The thresholds are interpreted based on a trade-off solution between numerous criteria used to analyze rough sets in a game scenario [33], [32]. Specifically, to increase the overall quality of three-way decisions, GTRS formulates strategies for players in the form of adjustments in thresholds. Each player contributes to the game by configuring the thresholds in order to optimize the game's benefits/rewards and utilities. The overall goal of a game in GTRS is to choose appropriate thresholds for three-way decisions with respect to the available criteria and presented information.

In GTRS (Figure 1 (4)), a typical game consists of three main elements: (i) game players, (ii) strategies, and (iii) payoff or utility functions. These components are usually defined as a tuple $\{P, S, u\}$, where [34]:

- **Game players:** The game players are denoted by a set P. The players in the game are selected to reflect the overall purpose of the game.
- **Strategies:** In the game, each player contributes by playing different strategies. The set of strategies available to player i is denoted by S_i. All possible strategy sets are denoted by the following Cartesian product: $S = S_1 \times S_2 \times \ldots \times S_n$, where S contains ordered pairs of the form $(s_1, s_2, \ldots, s_n)$ such that $s_1 \in S_1, s_2 \in S_2$ and $s_n \in S_n$. Each ordered pair in S is called a *strategy profile* and represents a certain situation encountered in a game.
- **Payoff functions or utility:** The payoff function, also called utility, for the players are defined via a set $u = (u_1, \ldots, u_n)$; where each u_i represents a real-valued utility function for player i and it maps the strategy profiles to real values ($u_i : S \mapsto \Re$). The payoffs reflect the utilities of performing or selecting a specific strategy.

Every player in a game seeks to execute a strategy that maximizes its payoff. The players' strategies, on the other hand, have an impact on their opponents' payoffs. The game solution is used to select a balanced and trade-off point based on all players' utilities. The *Nash equilibrium* is generally used to determine game solution or game outcome in GTRS.

Let us consider a strategy profile $s_{-i} = (s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$. s_{-i} is a strategy profile of all the players in the game except player i, and which can be further denoted as $s = (s_i, s_{-i})$. This means that all the players except i are committed to play s_{-i} and player i choosing s_i. The strategy profile $(s_1, s_2, \ldots, s_n) = (s_i, s_{-i})$ is a Nash equilibrium, when [35],

$$\forall i, \forall s_i^{'} \in S_i, u_i(s_i, s_{-i}) \geq u_i(s_i^{'}, s_{-i}), where(s_i^{'} \neq s_i) \quad (8)$$

This means that for all players i, their respective strategies, i.e., s_i is the best response to s_{-i}. In other words, a strategy profile constitutes a *Nash equilibrium* when no player is benefited from changing his/her strategy alone. The description presented above formulates a game in GTRS. It is to be noted that we may not be able to reach effective thresholds that meet the demands of the underlying applications with a single and non-repeated game. We, therefore, need to play the game several times; where in each play the goal is to keep modifying and refining the thresholds until we attain certain performance goals; e.g., a balance between accuracy and generality. The GTRS seeks an appropriate design of the threshold levels that are used in the three-way decisions framework, presented in Section II-A, by forming a game and applying concepts such as game solution and repeating games.

III. Application

A. Data Source

RECORDS[1] is a European research project that aims to quickly detect whether a patient is sensitive or resistant to the treatment of sepsis with corticosteroids. The project's clinical trial is an adaptive clinical trial that evaluates the efficacy of biomarkers and machine learning algorithms in defining patients' corticosteroid resistance, with the goal of optimizing their management. The project has adopted a distinctive approach to effectively analyze the severity of sepsis cases by collecting data on patients' demographics, health outcomes, and samples. This data collection has resulted in the creation of a first sepsis cohort, known as APPROCHS, which serves as an exceptional resource for medical research.

The paper considers the APROCCHS cohort that has been provided by the Assistance Publique–Hôpitaux de Paris (AP-HP) which is the university hospital trust operating in Paris, France, and its surroundings. It is the largest hospital system in Europe and one of the largest in the world. The goal of the cohort is to allow the investigation of qualitative interactions between clinical phenotypes and survival benefits or harms from corticosteroids (CS), i.e., to permit defining sensitivity and resistance to CS.

B. Data Description

The APROCCHS cohort gathers 1240 adult septic shock patients who are treated with or without CS. Each patient is characterized by 5645 features, also, called *risk factors* reflecting characteristics until Day 90.

Data were collected with a specification indicating whether the patients were treated with corticosteroid or with a placebo. A placebo is a substance or treatment that is given in the same manner as an active drug or treatment being tested but does not have any active ingredients or therapeutic effects [36].

C. Data Pre-processing

In this section, we explain the different data pre-processing tasks that we have performed on the APROCCHS cohort, namely: feature selection, data enrichment, data labeling and data cleaning.

1) Feature selection: Because sepsis is a time-sensitive disease, the likelihood of survival is significantly increased by early detection and treatment. This study focuses on using variables accessible at the earliest stage, especially at Day 0 of hospitalization, for predicting patients' responsiveness to corticotherapy in order to optimize accurate intervention. Specifically, from the initial pool of 5645 features –which reflects features from Day 0 until Day 90–, and by focusing only on features at Day 0, we were able to carefully choose a selection of 24 critical attributes following significant consultation with the respected medical specialists at APHP. This selection procedure entailed careful study and examination of each feature's relevance and significance in regard to our research with respect to the guidance and experience of the APHP healthcare experts.

The collected data is divided into two categories: static and dynamic. The first category includes information on the patient's current condition as well as personal information such as identification number, sex, weight, age, origin, date of hospitalization, and whether or not an antibiotic was administered before Day 0. These traits are noted at the time of admission and remain constant during hospitalization. The second category consists of dynamic elements that can be captured once or more times daily during hospitalization and are related to patient vital signs and laboratory testing. Admission type, infection date, infection place, and examination type are a few examples of dynamic characteristics that have only been once recorded. These data are often gathered before administering treatment. The sequential organ failure assessment (SOFA) score [37], ventilation, vasopressor use, and prescribed therapy dose are a few examples of characteristics that were recorded during the whole hospital stay and are associated with patients' responsiveness to treatment.

2) Data enrichment: Data enrichment relies on the process of adding new variables based on pre-existing ones in order to further explain the data and increase the precision of prediction algorithms. It improves detecting previously hidden relationships and patterns in the data. Following the guidelines of the APHP medical specialists, we generated the variable AR_INF_Type, which represents the source of infection and which was obtained from the diagnosis date and the hospital admission date variables. Furthermore, the values of the cortisol variable have been adjusted using a dataset that was given

[1] https://www.fhu-sepsis.uvsq.fr/rhu-records-4

by medical professionals and provides appropriate values for this characteristic. For a proper diagnosis and treatment plan, it is essential to know whether a patient received an antibiotic before being brought to the hospital. The medical staff's choice of the proper doses for the patient during their hospital stay will depend on this information, in addition to the machine learning model. A new feature entitled "ANTIBIOTIC" was created in order to acquire this information. A value of 1 of this characteristic implies that the patient took an antibiotic while a value of 0 means that he/she did not. The new variable "ANTIBIOTIC" comprises information on 690 patients who did not receive antibiotics prior to admission to the hospital and 550 patients who did.

3) Data labelling: For a patient who is enrolled in the study on Day 0, either corticosteroid medication or a placebo is administered every 4 to 6 hours while a number of features that indicate the patient's improvement are tracked. Daily feature values are recorded while each patient is observed for 90 days. The APHP healthcare experts have created precise standards for figuring out whether a patient would benefit from corticotherapy or not. Patients are specifically categorized as cortico-sensitive (i.e., responders) if all four of the following conditions are satisfied after 14 days of therapy:

- The patient survived.
- For at least 24 hours, there has been no vasopressor treatment.
- For at least 24 hours, the patient has been off of mechanical ventilation.
- The SOFA score is under 6.

The patient is generally considered cortico-resistant or a non-responder if the conditions are not satisfied, which is regarded a negative therapy response. As a result, the label is set to 1 or 0, indicating whether or not the patient reacted to the therapy on Day 14. Finally, patients who did not adhere to the aforementioned rule were eliminated from the cohort, leaving 1234 patients. This was done to preserve the integrity of our data and in accordance with the guidelines provided by medical specialists. The distribution of patients in the APPROCCHS cohort is shown in Table I.

4) Data cleaning: Particularly in important domains such as health, data cleaning and feature engineering are crucial steps in the data analysis process. These aspects have a significant impact on the decision-making process and the performance and accuracy of machine learning models. Dealing with the raw sepsis data that was gathered presented multiple challenges for this investigation. The APROCCHS cohort has a low rate of duplicate data, but in order to have accurate results with the three-way approach clustering, we have dropped duplicated patients. As a result, 1233 sepsis cases were still included in the cohort.

D. Three-way clustering with Game-theoretic rough sets

In this section, the application of the three-way clustering with GTRS, recently proposed in [29], is demonstrated using the pre-processed APROCCHS cohort. The objective is to cluster sepsis patients into two groups to reflect their responsiveness or not to CS while the model internally handles missing values.

1) Missing data description and handling: The only data pre-processing step that was not applied so far to the APROCCHS cohort is the task of handling missing values. As previously mentioned in Section I, the fact of imputing (replacing) or deleting the tuples containing missing values may significantly influence the conclusions drawn from the applied data mining task; specifically when it comes to a sensitive and critical domain as such is the medical domain. Pre-processing missing values may jeopardize the quality and reliability of the machine learning results; which is in our case the clustering task. As mentioned in Section I, a more appropriate and suitable strategy, to handle missing values, is to equip the clustering model with a mechanism able to handle data with missing values. In our study, this will be achieved by applying GTRS for three-way clustering.

However, it is still important to mention that, with respect to the medical experts' guidelines, some missing values had to be filled based on the following received recommendations:

- *Risk factors which are tied to the vasopressor treatment, life status, mechanical ventilation, and SOFA score:* Replace the missing value found at Day_i using the same non-missing value which is registered at Day_{i-1}. This is explained by the fact that if the value has not been registered at Day_i then this means that there has been no change in the patient's risk factor at Day_{i-1}.
- *The label:* To ensure the data's integrity and in accordance with the guidelines of medical experts, some patients have been updated from cortico-sensitive to cortico-resistant.

By applying these guidelines, the APROCCHS cohort still witnesses some missing values. These are distributed over 7 risk factors which are tied to the *KNAUS* score indicating the impact of a disease (i.e., sepsis) on the patient's activities, the *MACCABE* score indicating the presence of an additional fatal disease and its severity, the *SOFA* score in the last 3 hours after admission to intensive care, the *body temperature* at the entrance to the unit of intensive care, the *severity index*, the *glycemic index*, and the *blood lactate level*.

These will be taken care of at the GTRS clustering model level instead of modifying the data itself, i.e., will neither be imputed nor deleted; as will be explained in the next sections.

2) Game formalization: As described in Section II-B, the players, the strategies, and the payoff or utility functions, are the three components which are needed to be defined to analyze problems with GTRS. The game formalization is as follows:

a) The objective of the game: The aim of this game is to improve the clustering performance of datasets with missing values. As stated in [29], this objective can be achieved by balancing the accuracy and generality of the clustering, as described by Equations 6 and 7.

b) The players: The game's ultimate objective and goal should be reflected in the players. In this regard, the players in this game present the clustering's accuracy and generality

TABLE I
DISTRIBUTION OF PATIENTS IN APPROCCHS

Cohort		APPROCCHS		
Group	Features	Sensitive/Improved	Resistant/Not improved	Total
Corticosteroid	5645	233	379	612
Placebo		213	409	622
Total	5645	446	788	1234
Characteristic		APPROCHS randomized controlled trial		

features. Let A denote player *Accuracy* and let G denote the player *Generality*. P = {A, G} represents the player's set.

c) The strategies: The strategies denote the different actions that a player can take in a game. To maximize her/his rewards/benefits, each player adopts a strategy. As demonstrated in [29], when different thresholds are used in the game, the properties of accuracy and generality are influenced differently. Consequently, changes and variations in thresholds can be considered as feasible strategies. Three strategies are considered in our context:

- Decreasing the threshold α — defined as $(\alpha \downarrow)$
- Increasing the threshold β — defined as $(\beta \uparrow)$
- Decreasing α and increasing β simultaneously — defined as $(\alpha \downarrow \beta \uparrow)$

d) The utility functions: The outcomes of choosing a specific strategy are measured using a payoff function. The utility function is defined to reflect a player's potential performance gains or benefits from pursuing a specific strategy. As previously mentioned, different threshold values effect the two players A and G. Considering a certain strategy profile, say (s_m, s_n) leading to thresholds (α, β), the associated payoffs of the players are described as follows:

$$u_A(s_m, s_n) = Accuracy(\alpha, \beta), \tag{9}$$

where u_A is the payoff function of player A, and $Accuracy(\alpha, \beta)$ is defined in Equation 6, and

$$u_G(s_m, s_n) = Generality(\alpha, \beta), \tag{10}$$

where u_G is the payoff function of player G, and $Generality(\alpha, \beta)$ is defined in Equation 7.

For player A and player G, a value of 1 refers to a maximum utility while a value of 0 reflects a minimum payoff.

3) The trade-off between accuracy and generality:

a) Determining the Nash equilibrium: The game is viewed as a competition between the accuracy and generality measures of clustering. This is highlighted in Table II, where the table's rows refer to the strategies of player A and the columns refer to the strategies of player G. Each cell in Table II corresponds to a strategy profile, (s_m, s_n), where s_m represents player A's strategy and s_n represents player G's strategy. The goal of each player is to choose a strategy that configures the (α, β) thresholds in order to maximize her/his utility. $u_A(s_m, s_n)$ and $u_G(s_m, s_n)$ are the payoffs for players A and G, respectively, according to the strategy profile (s_m, s_n).

The logic in a game is that a player chooses a strategy with a larger payoff over other strategies with a lower payoff. For the two-player game under consideration, a strategy profile will be Nash equilibrium, with respect to the definition given in Equation 8, if,

$$Accuracy : \forall s_m \in S_A, u_A(s_m, s_n) \geq u_A(s_m^{'}, s_m), \tag{11}$$

where $(s_m^{'} \neq s_m)$, and

$$Generality : \forall s_n \in S_G, u_G(s_m, s_n) \geq u_G(s_m, s_n^{'}), \tag{12}$$

where $(s_n^{'} \neq s_n)$. This signifies that no player will gain from changing her/his strategy other than the strategy specified by the profile (s_m, s_n).

b) Determining the changes in the thresholds: Essentially, there are four ways for changing the thresholds (α, β) [29]:

1) A single player proposes to decrease the value of α — denoted as $(\alpha-)$;
2) Both of the two-game players propose to decrease the value of α — denoted as $(\alpha--)$;
3) A single player proposes to increase the value of β — denoted as $(\beta+)$;
4) Both of the two-game players propose to increase the value of β — denoted as $(\beta++)$;

These four ways can be used to associate threshold pairs with a certain strategy profile. For example, a strategy profile with (s_2, s_2) which is equal to $(\beta \uparrow, \beta \uparrow)$ is represented as $(\alpha, \beta++)$, since player A and player G propose to increase the value of β.

4) The learning mechanism defining the values of the thresholds: A single game run has minimal utility in terms of finding appropriate values for the (α, β) thresholds. A learning process will emerge as a result of iteratively changing the thresholds with the goal of improving the payoffs for the players. In this regard, the learning rule or criterion is based on the relationship between threshold modification and the influence on the players' utility. This relationship is used to define the four variables $(\alpha-, \alpha--, \beta+, \beta++)$. This is accomplished through the use of an iterative game.

Let (α, β) be the initial thresholds for a particular iteration of an iterative game. As previously mentioned, the Nash equilibrium will be utilized to compute and decide the game solution as well as the associated thresholds; which will be denoted as (α', β'). The four variables $(\alpha-, \alpha--, \beta+, \beta++)$ are calculated based on a fixed percentage of either increasing or decreasing the strategies' values in every iteration. For example, if the initial values of (α, β) = (1, 0), the percentage of increasing and decreasing the strategies is equal to 5%, and

TABLE II
PAYOFF TABLE FOR THE GAME.

		Generality (G)		
		$s_1 = \alpha\downarrow$	$s_2 = \beta\uparrow$	$s_3 = \alpha\downarrow\beta\uparrow$
Accuracy (A)	$s_1 = \alpha\downarrow$	$u_A(s_1, s_1)$, $u_G(s_1, s_1)$	$u_A(s_1, s_2)$, $u_G(s_1, s_2)$	$u_A(s_1, s_3)$, $u_G(s_1, s_3)$
	$s_2 = \beta\uparrow$	$u_A(s_2, s_1)$, $u_G(s_2, s_1)$	$u_A(s_2, s_2)$, $u_G(s_2, s_2)$	$u_A(s_2, s_3)$, $u_G(s_2, s_3)$
	$s_3 = \alpha\downarrow\beta\uparrow$	$u_A(s_3, s_1)$, $u_G(s_3, s_1)$	$u_A(s_3, s_2)$, $u_G(s_3, s_2)$	$u_A(s_3, s_3)$, $u_G(s_3, s_3)$

a strategy profile with (s_1, s_2) which equals to ($\alpha\downarrow, \beta\uparrow$) is represented as ($\alpha-$, $\beta+$). The new values of (α, β) = $(0.95, 0.05)$. The process can be halted once a satisfactory level of performance has been attained.

IV. Experiment setup

In this section, we will present a comprehensive description of the experimental setup for the three-way clustering with the GTRS approach to cluster Corticosteroid sensitivity with missing values.

A. Considered cohort

The used APROCCHS cohort includes patients who received corticotherapy and placebo treatment. A total number of 1233 patients is maintained after selecting the most important features, applying data enrichment, labeling the data, and deleting the duplicates (1 duplicate raw was found in the data and was deleted). In our preliminary study, and based on a ranking strategy, we worked with only 10 risk factors, presented in Table III, among the 24 features. The initial APROCCHS dataset contains 26 instances having missing values (i.e., 2%) which will form the set M.

B. Experimental Plan, Tests, and Tools

Our experimental protocol is divided into three stages. The first stage focuses on simulating data with missing values that aims to answer the question of the performance of the algorithm when adding more missing values. The second stage is devoted to exploring the impact of a parameter of the algorithm. Specifically, we study the impact of changing the value K of the nearest neighbors component which is part of the evaluation function $e(c_k, o_i)$ (see Section II-A). Finally, in the third stage, various percentages of the strategies' initial values are considered to study the influence of these values on the obtained results. Below is an outline of the three stages:

- Experiment 1: We evaluated the performance of the three-way clustering approach by using in each experiment several percentages of the missing values. As a first investigation, the algorithm was tested on four different missing data versions. The rate of missing values randomly chosen in this regard is based on 5%, 10%, 15%, and 20%. This experiment will enable us to respond to the following research question (**RQ1**): How can the percentage of missing values affect the performance of the algorithm?
- Experiment 2: The aim of this experiment is to explore the k nearest neighbors used in calculating the evaluation function to investigate its impact on the results. For this purpose, we choose to work with k = 5 and k = 7. Conducting this experiment will lead us to answer the following question (**RQ2**): Does the k nearest neighbors has an impact on the clustering results?
- Experiment 3: We assessed the choice of the strategies' initial values percentages and their effect on the obtained results. In this experiment, the algorithm takes as input a different set of $\alpha-$, $\alpha--$, $\beta+$ and $\beta++$. In our case, we tried to decrease α and increase β by 7% having initial values of $\alpha-$ equals to 0.93, $\alpha--$ equals to 0.86, $\beta+$ equals to 0.07, and $\beta++$ equals to 0.14, and by 10% having initial values of $\alpha-$ equals to 0.90, $\alpha--$ equals to 0.80, $\beta+$ equals to 0.10, and $\beta++$ equals to 0.20. By carrying out this experiment, we will be able to respond to the following question (**RQ3**): can the percentage of increasing and decreasing initial values of α and β influence the results?

Although the number of iterations is not defined, a maximum number is given to prevent the algorithm from continuing in an endless loop if it does not converge. While setting a maximum iteration of 20, the algorithm often converged between 3 and 4 iterations, based on the APROCCHS cohort. As for the clustering part, the k-means algorithm was used with k=2.

V. Results and discussion

A. Experimental results of GTRS approach

The results obtained from different GTRS-based approach runs with the various percentages of missing values inputs are shown in Tables IV – VII. The tables present the following observations:

- From Table IV (and similarly to all other Tables V – VII), it can be observed that in most runs the algorithm converge in the third iteration. We can also see how the thresholds are altered across the game's several iterations and how this affects generality and accuracy. For the experiment with 5% missing values, the initial thresholds of (α, β) = (1, 0) are set before the game starts, resulting in an accuracy of 0.98 and a generality of 0.88. However, in the second iteration, the accuracy and generality are still the same, while the threshold α is decreased and β increased by 0.14. For the experiment with 10% missing values, the accuracy is stable while the generality increased from 84% to 93%. For 15% and 20% missing values, we can notice a slight decrease in the accuracy (for 15% missing values: from 1 to 98%, for 20% missing values: from 99% to 96%) with an

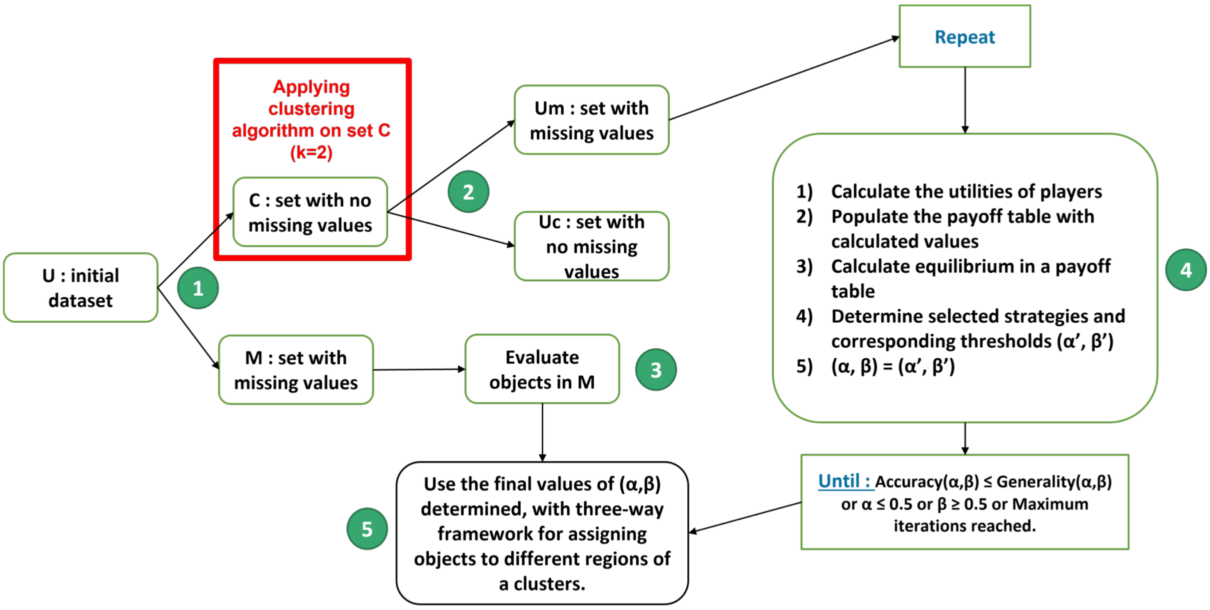

Repeat
Applying clustering algorithm on set C (k=2)
Um : set with missing values
C : set with no missing values
2
Uc : set with no missing values
U : initial dataset
1
1) Calculate the utilities of players
2) Populate the payoff table with calculated values
3) Calculate equilibrium in a payoff table
4) Determine selected strategies and corresponding thresholds (α', β')
5) (α, β) = (α', β')
4
M : set with missing values
Evaluate objects in M
3
Use the final values of (α,β) determined, with three-way framework for assigning objects to different regions of a clusters.
5
Until : Accuracy(α,β) ≤ Generality(α,β) or α ≤ 0.5 or β ≥ 0.5 or Maximum iterations reached.

Fig. 1. Main functioning of the three-way clustering with Game-theoretic rough sets

TABLE III
CONSIDERED SET OF RELEVANT FEATURES AT DAY 0

Reference	Description	Format
DATINF	Diagnosis date	Precision = JJ/MM/YYYY, Min = DATHOSP (Hospital admission date), Max = Current date
SITINF	Infection location	0 = Lung, pleura, 1 = Peritoneal, 2 = Urogenital, 3 = Central Nervous System (CNS), 4 = Endocarditis, mediastinum, 5 = Sepsis, 6 = Soft tissue, 7 = Bones and joints, 8 = Other
SEX	Indicates patient sex	1 = Male, 2 = Female
PATWGHT	Indicates the weight of the patient	Min = 36, Max = 154
ORIGIN	Indicates the patient ORIGIN	1 = City, 2 = Hospital, 3 = Institution
AGE	Indicates patient age	Min = 18, Max = 97
KNAUS_J0	Activity and medical follow-up in the six months prior to admission	1 = Stage D Major activity restriction due to illness, including bedridden or hospitalized patients, 2 = Stage C Chronic illness causing significant but not total activity restriction, 3 = Stage B Moderate or moderate activity limitation due to illness (limited work activities), 4 = Stage A Good health, no activity limitation
MACCABE_J0	Description of the patient's condition before the episode leading to ICU	1 = Absence of underlying disease or underlying disease not life-threatening, 2 = Underlying disease life-threatening within 5 years, 3 = Underlying disease estimated to be fatal within one year
SOFA_ADM	Indicates the worst case value up to 3 hours after admission	Min = 2, Max = 16
IGSII_ADM_TYP	Indicates the admission type of the patient	0 = Scheduled surgery, 6 = Medical, 8 = Unscheduled surgery

increase in generality (for 15% missing values: from 87% to 94%, for 20% missing values: from 83% to 91%) – a trade-off maintaining the required balance. To explore the research question **RQ1** of whether the percentage of missing values affects the performance of the algorithm, we performed the first experiment. By examining the outcomes of the GTRS algorithm via results presented in Tables IV – VII, when increasing the number of randomly chosen missing values, the trade-off accuracy/generality will not be lost.

- In the results presented in Table V, we have increased the value of k from 5 to 7. In comparison with the initial thresholds $(\alpha,\ \beta)$ = (1, 0) and when testing with only 5% of missing values, we can observe that there is a slight decrease in accuracy (i.e., 1%) while the generality improved with 9% reaching 95%. When testing with 10% and 20% missing values, the GTRS algorithm shows a minor reduction in accuracy varying from 1% to 3% with an increase in generality (between 8% and 13%). Thus, the GTRS model delivers an acceptable trade-off between accuracy and generality. For the experiment with 15%, and while comparing the results to the 5%

missing values, we can note that there is an increase in the accuracy showing 98% (97% with 5% missing values) with a 1% decrease in generality. In order to investigate the research question of whether the value of k nearest neighbors has an impact on the results **RQ2**, Experiment 2 was carried out. Through the interpretation of the results obtained from the GTRS algorithm, when increasing the percentage of missing values, and by increasing k to 7, we can notice a slighter loss in the trade-off accuracy/generality in comparison to k = 5.

- Table VI and Table VII show the results obtained when varying the strategies' initial values with decreasing α and increasing β by 10% having initial values of $\alpha-$ equals to 0.90, $\alpha--$ equals to 0.80, $\beta+$ equals to 0.10, and $\beta++$ equals to 0.20. Consistent with the findings in Table IV, from Table VI, we can notice that from 5% to 15% of missing values the accuracy demonstrates stability in its values with 98% while the generality presents a significant increase varying from 88% to 94%. For 20% missing values, the obtained results show a slight decrease in accuracy with 3% (reaching 96%) and a slight increase in generality with 2% (reaching 91%).

  As was the case in Table V, by analyzing the obtained results in Table VII, we can notice that when compared to the initial thresholds (α, β) = (1, 0) and tested with only 5% of missing values, we observed a slight decrease in accuracy (by 1%), but a significant improvement in generality (by 9%). Moreover, for experiments with 10%, 15%, and 20% one can observe that accuracy values were decreased by approximately 2% while generality increased by up to 14%. Also, for instance, with initial values equal to 7%, k = 5, and 20% of missing values (Table IV), the final values are 96% and 91% for accuracy and generality, respectively. With initial values equal to 10%, k = 5, and 20% of missing values (Table VI), the final values are the same registering 96% and 91% for accuracy and generality, respectively. To answer the third research question **RQ3** to what extent can variations in the percentage of initial values of α and β, whether increased or decreased, impact the results, Experiment 3 was implemented. By looking at the obtained results and interpreting them (Table IV, V, VI, and VII), it is noticeable that the final output of the GTRS algorithm is relatively stable regardless of the initial values of the strategies.

As expected, from the different tables, when using k = 5, the execution time is observed to be lower than when using k = 7, indicating that a smaller value of k can lead to faster computations. However, for more exploration, the execution time can be minimized by using several techniques such as Multithreading [38], Single Instruction Multiple Data (SIMD) [39], and Open Multi-Processing (OpenMP) [40]. By employing these parallelism techniques, the GTRS algorithm execution time can be reduced, leading to marked improvements in both its performance and efficiency.

B. Three-way clustering approach evaluation

The previously obtained results show the effectiveness of the three-way clustering approach with GTRS in handling missing values. Therefore, in almost all the experiments, for clustering CS responsiveness, the trade-off accuracy/generality is maintained. The best trade-off found is with an accuracy value of 97%, and the generality presents 95%; with k =7 and 5% of missing values.

The final step (Figure 1 (3,5)) in the GTRS algorithm is to evaluate objects with missing values in M using Equation 4 and then select the best values of (α, β) and test them on the set M with missing values. As mentioned in Section IV-A, set M contains 26 patients having missing data (i,e., only 2%). Table VIII summarizes the obtained results after applying the three-way clustering approach on the set M and using k = 7 as value of k nearest neighbors. It can be observed that the accuracy/generality trade-off was preserved, presenting 96% accuracy and 92% generality. The results revealed that the best thresholds values for (α, β) = (0.58, 0.42). These final (α, β) values are used for assigning objects to different regions of a clusters as follows:

$$Inside(c_k) = \{o_i \in U | e(c_k, o_i) \geq 0.58\}, \tag{13}$$

$$Outside(c_k) = \{o_i \in U | e(ck_, o_i) \leq 0.42\}, \tag{14}$$

$$Partial(c_k) = \{o_i \in U | 0.42 < e(c_k, o_i) < 0.58\}, \tag{15}$$

.

After applying Equations 13, 14, and 15 to assign objects in set M to clusters, we observed that the algorithm's non-deterministic nature resulted in some sepsis patients being found in the partial region. This means that these sepsis patients could not be clustered to a specific region as CS(placebo) sensitive(improved) or resistant(not improved); despite that we had their correct label in the cohort. In addition to this, when we examined the patients clustered by GTRS, we found some false negatives. This suggests that the results were not entirely deterministic, and further statistical analysis is required to validate them. One possible explanation to these preliminary results is that we have only considered 10 risk factors out of the 24 variables. Despite this, we still can consider that the initial results in terms of trade-off accuracy and generality are promising and indicate that GTRS has potential in addressing the issue of missing data in sepsis patients.

VI. Conclusion

The aim of this paper is to investigate the issue of clustering with missing values in clinical data using a three-way approach with GTRS. The study utilized data from the APPROCHS cohort, which included 1240 sepsis patients enrolled in a randomized controlled trial, and collected by clinicians from APHP. An important challenge in implementing this approach was setting appropriate thresholds to determine the three types of decisions. GTRS was found to be a promising alternative for clustering objects with missing values.

TABLE IV
Obtained results of GTRS algorithm applied on APROCCHS cohort using multiple missing values percentages and fixed values of k = 5 and initial values = 7%

Missing values	k	Initial values	Iteration	Alpha	Beta	Accuracy	Generality	Execution Time
5%	5	7%	1	1	0	0.98	0.88	22 min
			2	**0.86**	**0,14**	**0.98**	**0.88**	
10%	5	7%	1	1	0	0.98	0.84	47 min
			2	0.86	0.14	0.98	0.84	
			3	**0.72**	**0.28**	**0.98**	**0.93**	
15%	5	7%	1	1	0	1	0.87	79 min
			2	0.86	0.14	1	0.87	
			3	**0.72**	**0.28**	**0.98**	**0.94**	
20 %	5	7%	1	1	0	0.99	0.83	154 min
			2	0.86	0.14	0.99	0.83	
			3	**0.72**	**0.28**	**0.96**	**0.91**	

TABLE V
Obtained results of GTRS algorithm applied on APROCCHS cohort using multiple missing values percentages and fixed values of k = 7 and initial values = 7%

Missing values	k	Initial values	Iteration	Alpha	Beta	Accuracy	Generality	Execution Time
5%	7	7%	1	1	0	0.98	0.86	28 min
			2	0.93	0.07	0.98	0.86	
			3	0.72	0.28	0.98	0.89	
			4	**0.58**	**0.42**	**0.97**	**0.95**	
10%	7	7%	1	1	0	0.99	0.8	54 min
			2	0.86	0.14	0.99	0.8	
			3	0.72	0.28	0.96	0.88	
			4	**0.58**	**0.42**	**0.95**	**0.93**	
15%	7	7%	1	1	0	1	0.87	78 min
			2	0.86	0.14	1	0.87	
			3	**0.72**	**0.28**	**0.98**	**0.94**	
20%	7	7%	1	1	0	0.99	0.8	180 min
			2	0.86	0.14	0.99	0.8	
			3	0.72	0.28	0.99	0.85	
			4	**0.58**	**0.42**	**0.96**	**0.93**	

TABLE VI
Obtained results of GTRS algorithm applied on APROCCHS cohort using multiple missing values percentages and fixed values of k = 5 and initial values = 10%

Missing values	k	Initial values	Iteration	Alpha	Beta	Accuracy	Generality	Execution Time
5%	5	10%	**1**	**1**	**0**	**0.98**	**0.88**	26 min
10%	5	10%	1	1	0	1	0.82	50 min
			2	0.8	0.2	0.98	0.9	
			3	**0.66**	**0.34**	**0.98**	**0.9**	
15%	5	10%	1	1	0	1	0.87	79 min
			2	0.8	0.2	0.98	0.94	
			3	**0.66**	**0.34**	**0.98**	**0.94**	
20 %	5	10%	1	1	0	0.99	0.82	104 min
			2	0.80	0.20	0.96	0.91	
			3	**0.66**	**0.34**	**0.96**	**0.91**	

To evaluate the effectiveness of the GTRS model, three experiments were conducted. In the first experiment, the algorithm was tested with varying percentages of missing data, and the results showed that accuracy and generality can be preserved despite an increase in the number of missing values. The second experiment examined how the selection of the k nearest neighbors in the evaluation function affected the results. The third experiment evaluated the impact of the percentages of initial values of the strategies on the results, and the stability of the final output of the GTRS algorithm was apparent as it did not significantly vary with the initial values of the strategies.

As future work, we aim to use four clusters, instead of two, to further represent sepsis patients (Cortico-sensitive, Cortico-resistant, improved status with placebo, and unimproved status with placebo). This may improve the performance of the algorithm. Also, we aim to explore alternative approaches such as Reinforcement learning [41]. This approach would consider accuracy and generality as agents, and increasing and decreasing α and β strategies as actions to be taken in the environment. Players would learn a policy through trial and error that maximizes their rewards. Additionally, one can expand the evaluation of the results achieved by taking into account the quality of the model to address concerns related

TABLE VII
Obtained results of GTRS algorithm applied on APROCCHS cohort using multiple missing values percentages and fixed values of k = 7 and initial values = 10%

Missing values	k	Initial values	Iteration	Alpha	Beta	Accuracy	Generality	Execution Time
5%	7	10%	1	1	0	0.98	0.86	28 min
			2	0.9	0.1	0.98	0.86	
			3	**0.7**	**0.3**	**0.97**	**0.95**	
10%	7	10%	1	1	0	1	0.78	39 min
			2	0.8	0.2	1	0.86	
			3	**0.66**	**0.34**	**0.98**	**0.92**	
15%	7	10%	1	1	0	1	0.83	78 min
			2	0.8	0.2	1	0.89	
			3	**0.66**	**0.34**	**0.99**	**0.94**	
20%	7	10%	1	1	0	0.99	0.80	106 min
			2	0.8	0.2	0.99	0.85	
			3	**0.66**	**0.34**	**0.96**	**0.93**	

TABLE VIII
Best (α, β) values evaluation on the set M with missing values

Missing values	k	Iteration	Alpha	Beta	Accuracy	Generality
2%	7	1	1	0	0.95	0.85
		2	0.86	0.14	0.95	0.85
		3	0.72	0.28	0.95	0.85
		4	**0.58**	**0.42**	**0.96**	**0.92**

to overlearning [42], overfitting [43], and the assessment parameters used to measure the model's performance.

Statements of ethical approval

For the APROCCHS study, the protocol and qualification of all investigators were approved by the Ethics Committee (Comité de Protection des Personnes, CPP) of Saint-Germain-en-Laye, France, on November 22, 2007. The trial was registered at ClinicalTrials.gov under NCT00625209.

Acknowledgment

The national program "*Programme d'Investissements d'Avenir (PIA)*" (as part of the France 2030 programme) under the reference ANR-18-RHUS-0004. This work is part of the Federation Hospitalo-Universitaire (FHU) Saclay and Paris Seine Nord Endeavour to PerSonalize Interventions for Sepsis (SEPSIS). This work was also supported by ANR PIA funding: ANR-20-IDEES-0002 and by the iRECORDS project, funded by ERA PerMed (JTC_2021) to KZ and DA (ANR-21-PERM-0005)".

References

[1] M. Singer, C. S. Deutschman, C. W. Seymour, M. Shankar-Hari, D. Annane, M. Bauer, R. Bellomo, G. R. Bernard, J.-D. Chiche, C. M. Coopersmith *et al.*, "The third international consensus definitions for sepsis and septic shock (sepsis-3)," *Jama*, vol. 315, no. 8, pp. 801–810, 2016.

[2] J. Matsuda, S. Kato, H. Yano, G. Nitta, T. Kono, T. Ikenouchi, K. Murata, M. Kanoh, Y. Inamura, T. Takamiya *et al.*, "The sequential organ failure assessment (sofa) score predicts mortality and neurological outcome in patients with post-cardiac arrest syndrome," *Journal of cardiology*, vol. 76, no. 3, pp. 295–302, 2020.

[3] K. E. Rudd, S. C. Johnson, K. M. Agesa, K. A. Shackelford, D. Tsoi, D. R. Kievlan, D. V. Colombara, K. S. Ikuta, N. Kissoon, S. Finfer *et al.*, "Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the global burden of disease study," *The Lancet*, vol. 395, no. 10219, pp. 200–211, 2020.

[4] A. Rhodes, L. E. Evans, W. Alhazzani, M. M. Levy, M. Antonelli, R. Ferrer, A. Kumar, J. E. Sevransky, C. L. Sprung, M. E. Nunnally *et al.*, "Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016," *Intensive care medicine*, vol. 43, no. 3, pp. 304–377, 2017.

[5] D. W. Cain and J. A. Cidlowski, "Immune regulation by glucocorticoids," *Nature Reviews Immunology*, vol. 17, no. 4, pp. 233–247, 2017.

[6] D. Annane, S. M. Pastores, W. Arlt, R. A. Balk, A. Beishuizen, J. Briegel, J. Carcillo, M. Christ-Crain, M. S. Cooper, P. E. Marik *et al.*, "Critical illness-related corticosteroid insufficiency (circi): a narrative review from a multispecialty task force of the society of critical care medicine (sccm) and the european society of intensive care medicine (esicm)," *Intensive care medicine*, vol. 43, no. 12, pp. 1781–1792, 2017.

[7] N. Heming, S. Sivanandamoorthy, P. Meng, R. Bounab, and D. Annane, "Immune effects of corticosteroids in sepsis," *Frontiers in Immunology*, p. 1736, 2018.

[8] D. Annane, "Corticosteroids for severe sepsis: an evidence-based guide for physicians," *Annals of intensive care*, vol. 1, no. 1, pp. 1–7, 2011.

[9] J. Cleve and U. Lämmel, *Data mining*. Walter de Gruyter GmbH & Co KG, 2020.

[10] S. Gavankar and S. Sawarkar, "Decision tree: Review of techniques for missing values at training, testing and compatibility," in *2015 3rd international conference on artificial intelligence, modelling and simulation (AIMS)*. IEEE, 2015, pp. 122–126.

[11] S. A. Alasadi and W. S. Bhaya, "Review of data preprocessing techniques in data mining," *Journal of Engineering and Applied Sciences*, vol. 12, no. 16, pp. 4102–4107, 2017.

[12] L. Yu, R. Zhou, R. Chen, and K. K. Lai, "Missing data preprocessing in credit classification: One-hot encoding or imputation?" *Emerging Markets Finance and Trade*, vol. 58, no. 2, pp. 472–482, 2022.

[13] M. S. Osman, A. M. Abu-Mahfouz, and P. R. Page, "A survey on data imputation techniques: Water distribution system as a use case," *IEEE Access*, vol. 6, pp. 63 279–63 291, 2018.

[14] R. R. Andridge and R. J. Little, "A review of hot deck imputation for survey non-response," *International statistical review*, vol. 78, no. 1, pp. 40–64, 2010.

[15] U. Pujianto, A. P. Wibawa, M. I. Akbar *et al.*, "K-nearest neighbor (k-nn) based missing data imputation," in *2019 5th International Conference on Science in Information Technology (ICSITech)*. IEEE, 2019, pp. 83–88.

[16] N. Karmitsa, S. Taheri, A. Bagirov, and P. Mäkinen, "Missing value imputation via clusterwise linear regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1889–1901, 2020.

[17] R. A. Hughes, J. Heron, J. A. Sterne, and K. Tilling, “Accounting for missing data in statistical analyses: multiple imputation is not always the answer,” *International journal of epidemiology*, vol. 48, no. 4, pp. 1294–1304, 2019.

[18] S. Goel and M. Tushir, “Different approaches for missing data handling in fuzzy clustering: a review,” *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, vol. 13, no. 6, pp. 833–846, 2020.

[19] D.-T. Dinh, V.-N. Huynh, and S. Sriboonchitta, “Clustering mixed numerical and categorical data with missing values,” *Information Sciences*, vol. 571, pp. 418–442, 2021.

[20] Z. Pawlak, “Rough sets,” *International journal of computer & information sciences*, vol. 11, pp. 341–356, 1982.

[21] A. Skowron and D. Ślęzak, “Rough sets turn 40: From information systems to intelligent systems,” in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 23–34.

[22] T.-F. Fan, C.-J. Liau, and D.-R. Liu, “Variable precision fuzzy rough set based on relative cardinality,” in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2012, pp. 43–47.

[23] L. S. Riza, A. Janusz, C. Bergmeir, C. Cornelis, F. Herrera, D. Śle, J. M. Benítez *et al.*, “Implementing algorithms of rough set theory and fuzzy rough set theory in the r package “roughsets”,” *Information sciences*, vol. 287, pp. 68–89, 2014.

[24] R. Polikar, J. DePasquale, H. S. Mohammed, G. Brown, and L. I. Kuncheva, “Learn++. mf: A random subspace approach for the missing feature problem,” *Pattern Recognition*, vol. 43, no. 11, pp. 3817–3832, 2010.

[25] B. Panda, S. Gantayat, and A. Misra, “Rough set rule-based technique for the retrieval of missing data in malaria diseases diagnosis,” *Computational Intelligence in Medical Informatics*, pp. 59–71, 2015.

[26] P. Maji, “Advances in rough set based hybrid approaches for medical image analysis,” in *Rough Sets: International Joint Conference, IJCRS 2017, Olsztyn, Poland, July 3–7, 2017, Proceedings, Part I*. Springer, 2017, pp. 25–33.

[27] K. B. Nahato, K. N. Harichandran, K. Arputharaj *et al.*, “Knowledge mining from clinical datasets using rough sets and backpropagation neural network,” *Computational and mathematical methods in medicine*, vol. 2015, 2015.

[28] Y. Yao *et al.*, “An outline of a theory of three-way decisions.” in *RSCTC*, vol. 7413, 2012, pp. 1–17.

[29] M. K. Afridi, N. Azam, J. Yao, and E. Alanazi, “A three-way clustering approach for handling missing data using gtrs,” *International Journal of Approximate Reasoning*, vol. 98, pp. 11–24, 2018.

[30] C. M. Poteraş and M. L. Mocanu, “Evaluation of an optimized k-means algorithm based on real data,” in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2016, pp. 831–835.

[31] J. P. Herbert and J. Yao, “Game-theoretic rough sets,” *Fundamenta Informaticae*, vol. 108, no. 3-4, pp. 267–286, 2011.

[32] J. Yao and J. P. Herbert, “A game-theoretic perspective on rough set analysis,” *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 20, no. 3, pp. 291–298, 2008.

[33] N. Azam and J. Yao, “Analyzing uncertainties of probabilistic rough set regions with game-theoretic rough sets,” *International journal of approximate reasoning*, vol. 55, no. 1, pp. 142–155, 2014.

[34] Y. Shoham, “Computer science and game theory,” *Communications of the ACM*, vol. 51, no. 8, pp. 74–79, 2008.

[35] K. Leyton-Brown and Y. Shoham, *Essentials of game theory: A concise multidisciplinary introduction*. Springer Nature, 2022.

[36] R. S. Hotchkiss, E. Colston, S. Yende, D. C. Angus, L. L. Moldawer, E. D. Crouser, G. S. Martin, C. M. Coopersmith, S. Brakenridge, F. B. Mayr *et al.*, “Immune checkpoint inhibition in sepsis: a phase 1b randomized, placebo-controlled, single ascending dose study of anti-pd-l1 (bms-936559),” *Critical care medicine*, vol. 47, no. 5, p. 632, 2019.

[37] T. Z. J. Teng, J. K. T. Tan, S. Baey, S. K. Gunasekaran, S. P. Junnarkar, J. K. Low, C. W. T. Huey, and V. G. Shelat, “Sequential organ failure assessment score is superior to other prognostic indices in acute pancreatitis,” *World Journal of Critical Care Medicine*, vol. 10, no. 6, p. 355, 2021.

[38] I. Oz and S. Arslan, “A survey on multithreading alternatives for soft error fault tolerance,” *ACM Computing Surveys (CSUR)*, vol. 52, no. 2, pp. 1–38, 2019.

[39] M. Steinegger, M. Meier, M. Mirdita, H. Vöhringer, S. J. Haunsberger, and J. Söding, “Hh-suite3 for fast remote homology detection and deep protein annotation,” *BMC bioinformatics*, vol. 20, no. 1, pp. 1–15, 2019.

[40] S. Bernabé, C. García, R. Fernández-Beltran, M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, “Open multi-processing acceleration for unsupervised land cover categorization using probabilistic latent semantic analysis,” in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 9835–9838.

[41] Y. Li, E. Fadda, D. Manerba, R. Tadei, and O. Terzo, “Reinforcement learning algorithms for online single-machine scheduling,” in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020, pp. 277–283.

[42] C. Song and V. Shmatikov, “Overlearning reveals sensitive attributes,” *arXiv preprint arXiv:1905.11742*, 2019.

[43] X. Ying, “An overview of overfitting and its solutions,” in *Journal of physics: Conference series*, vol. 1168. IOP Publishing, 2019, p. 022022.

DOI: 10.15439/2023F5836
ISSN 2300-5963 ACSIS, Vol. 35

Tenure and Background of CIOs in Germany - Influencing Factors and International Comparison

Patrick Hillebrand
OTH Regensburg
Regensburg, Germany
patrick.hillebrand@st.oth-regensburg.de

Markus Westner
OTH Regensburg
Regensburg, Germany
markus.westner@oth-regensburg.de

***Abstract*—The average tenure of Chief Information Officers (CIOs) has increased over the past few years. Nevertheless, the average tenure of CIOs is shorter than that of Chief Executive Officers (CEOs). While most studies on tenure and background are based on data from US IT executives, studies on German CIOs are missing. This study analyzes the tenure of German CIOs as a proxy for management effectiveness and how certain factors influence it. An original and unique dataset of 384 IT executives from German companies is examined. The data include the size and industry sector of the companies, educational and professional backgrounds of the CIOs, and the CIOs' reporting lines. Data were analyzed using the chi-square test and Fisher's exact test. The German CIOs had a median tenure of 4.0 years. However, if we examine executives who are currently in office and executives with a completed term of office separately, the median tenure differs. The results also show that German CIOs do not have shorter tenures than German CEOs. When compared with US CIOs, the results depend on the values selected for comparison. In addition, the analysis shows that neither the size and industry sector of the companies nor the educational and professional backgrounds of the CIOs and managers of the CIO reports have a statistically significant influence on the tenure of IT executives. The factors examined in this study can be considered as preconditions for the CIO position. In the future, factors that play a role during tenure should be examined.**

***Index Terms*—CIO, Chief Information Officer, CIO tenure, CIO background**

I. Introduction

A LONG tenure for Chief Executive Officers (CEOs) is linked to superior firm performance [1]. This indicates successful work by the executives. Consequently, the CEO is often replaced when the company performs poorly because there is doubt about his or her ability to effectively implement strategies to increase the company's value [1].

In addition to the CEO, the Chief Information Officer (CIO), as the top information technology (IT) executive, has been an established position in companies for several years. The average tenure of CIOs has increased over the past 10 years but has stagnated in recent years [2]. Despite this increase, CEOs on average still have a longer tenure; according to a study by the consulting firm Korn Ferry, CIOs have significantly shorter tenures at 4.6 years compared to CEOs at 6.9 years [3].

Although other studies have examined the tenures of CIOs during a similar period, they have arrived at different conclusions. CIO Magazine's 2020 State of the CIO Report shows that CIOs have an average tenure of 6.5 years in 2020 [4]. This result is consistent with that determined by Kappelman et al. [2]. Nevertheless, all studies conclude that most IT executives hold their positions for only three or fewer years [2, 4].

Many factors can influence the tenure of CIOs. Dawson et al. [5] examine the tenure of senior executives, particularly CIOs. The aim of their work was to determine differences in tenure based on factors such as gender, size, and type of organization. Recently, Jones et al. [6] examined the background of CIO. Their findings suggest that the background of a CIO has little to do with what a CIO does once in office. For example, they found that whether an IT executive was hired externally or promoted internally did not influence their reporting line. However, this study did not investigate whether these factors had an impact on the tenure of CIOs.

All the previously mentioned studies were based on data from American CIOs. German IT executives were not included in these studies. This is consistent with the results of a recent literature review, which found that quantitative research on IT executives is primarily based on data from the US [7].

This study aims to address these gaps. It examines the tenure of CIOs in Germany and attempts to answer the following research question: 'Which factors significantly influence the tenure of CIOs?'

To address this question, the remainder of this article is structured as follows. First, the existing literature is reviewed, and hypotheses are derived. Next, the methodology used is described, including the collection of data, description of the data sample, and statistical methods. The results of the analysis are presented and discussed in the following section. Finally, we present the limitations of this study and avenues for future research.

II. Literature Review and Hypotheses Building

A long CEO tenure is linked to superior firm performance and can indicate a successful executive [1]. Therefore, in this study, long tenure was used as a proxy for successful work. Even though the CIO has different tasks and responsibilities than the CEO, it can be assumed that if a CIO performs poorly, he or she will also be dismissed. In the following section, we hypothesize the factors that may influence tenure and, thus, the success of CIOs.

Studies have yielded different results regarding the tenure of CIOs. According to a study by the consulting firm Korn Ferry, CIOs have significantly shorter tenures at 4.6 years compared to CEOs at 6.9 years [3]. Other studies arrive at an average tenure of 6.5 years and 6.6 years respectively [2, 4]. There may be cultural differences between Germany and the US.

However, organizational culture, rather than the societal culture of each country, largely influences how the CIO operates in its role [8]. Therefore, the tenure of the CIOs should not differ by country. Based on these arguments, we propose the following hypothesis H1.

H1: The tenure of German CIOs is comparable to that of US CIOs.

In 2018, German-speaking CEOs who left office had an average tenure of 6.6 years [9]. In a study of how CEO age and CEO tenure moderates the relation between certain financial factors, Neifar and Ajili [10] determined a tenure of 6.5 years for German CEOs. These figures are consistent with the average tenure of US-American CEOs, which is 6.9 years [3]. Since the CIO has a significantly shorter tenure than the CEO, we propose the following hypothesis H2.

H2: The tenure of German CIOs is shorter than that of German CEOs.

Dawson et al. [5] analyzed the tenure of executives in relation to company size. They also found that managers in smaller organizations had the longest tenures, followed by slightly shorter tenures in larger organizations. According to this study, executives have the shortest tenures in medium-sized companies. They argue that medium-sized companies could be used as steppingstones, or that large companies could attract better pay. To measure size, we used turnovers from companies. It is therefore interesting to see whether the results differ when the number of employees is used to measure size. In summary, we propose hypotheses H3a and H3b.

H3a: The tenure of CIOs differs according to the size of the company, based on the number of employees.

H3b: The tenure of CIOs differs according to the size of the company, based on turnover.

The role of IT is unique to different organizations and industries. Therefore, the role of the CIO differs depending on whether companies have a high or low level of operational and strategic dependence on IT [11]. Industry-specific experience and knowledge of a CIO could therefore be particularly helpful for new business-oriented opportunities [6]. However, in a survey conducted by Mazzola et al. [12], IT executives changed industries particularly frequently on their way to become top IT executives. In addition, the industry in which they started their careers was often not the one in which they became CIO [12]. Mazzola et al. [12] concluded that IT skills are either industry-independent, or that industry knowledge is not a significant factor in the success of a CIO. Therefore, hypothesis H4 is proposed.

H4: The tenures of CIOs do not differ by industry.

CIOs are perceived as innovative and technologically knowledgeable, yet highly detail-oriented [13]. Many CIOs have a technological or engineering background. They often seem to lack political skills or the ability to communicate technical problems in such a way that other executives understand their messages [11]. These skills are relevant to an increasing business focus, and IT executives are expected to actively participate in business discussions [14]. Moreover, a CIO should be a business leader, rather than an IT manager [11]. Managers with an economic background already possess these skills.

Regardless of the subject area, a high level of education is an important prerequisite for becoming a CIO [12]. Therefore, we propose the following two hypotheses H5a and H5b.

H5a: CIOs with a background in economics have the longest tenures.

H5b: A higher level of education is associated with a longer tenure.

Whether a CIO has been promoted to this position internally or externally depends on the circumstances. For example, a successful CIO is often followed by an executive from within the organization to sustain success. The new manager already knows the organization and does not have to familiarize himself with it first [15]. In addition, companies with the primary goal of increasing efficiency beyond their IT function are more likely to hire a CIO from their ranks [6]. Companies that want to make strategic changes tend to hire a CIO outside the organization [15]. An externally hired CIO can use the beginning of his or her tenure to influence perception through the choice of activities [13]. However, a disadvantage is that they do not know the organization and may find conditions other than expected [15]. Whether a CIO is promoted internally or hired externally has advantages and disadvantages and depends on the circumstances of the organization. Therefore, hypothesis H6 is proposed.

H6: There is no difference in tenure whether CIOs were hired internally or externally.

Most CIOs in the US come to office from an IT position. However, this number has steadily decreased in recent years [2]. Furthermore, internal promotions mostly occur in the IT position [12]. The steady decline in previous IT positions and the need for a stronger business focus may indicate that a previous IT position is no longer the most promising [14]. However, according to Jones et al., the background of a CIO has little influence on the tasks a CIO performs once in office [6]. Thus, we propose the following hypotheses: H7a, H7b, and H7c.

H7a: The tenure of the CIOs is independent of the previous position.

H7b: Similar to findings from the US, the majority of externally hired CIOs in Germany also come from a position outside IT.

H7c: Similar to findings from the US, the majority of internally hired CIOs in Germany also rise from an IT position.

Most US CIOs report to the CEO, followed by the Chief Financial Officer (CFO). The number of CIOs reporting to the CEO has increased in recent years, while the number of CIOs reporting to the CFO has decreased [2, 16]. Whether a CIO is promoted within the organization or hired from outside the organization can affect CIO reports. According to Jones et al., internally promoted IT executives are more likely to report to the CFO and IT executives hired from outside the organization are likely to report to the CEO [6]. Based on this finding, we propose the following hypotheses H8a, H8b, and H8c.

H8a: Similar to findings from the US, the majority of German CIOs also report to the CEO.

H8b: Similar to findings from the US, the majority of German CIOs who have been promoted internally also report to the CFO.

H8c: Similar to findings from the US, the majority of German CIOs who are hired externally also report to the CEO.

III. Methodology

A. Data Collection

To test the hypotheses, data from two sources were collected and combined into a single dataset. The basis for the selection of CIOs is the "Top-500" company database of the German edition of CIO Magazine [17]. The database lists companies based in Germany with a turnover of more than one billion euros and was chosen for several reasons: (a) it contains information on companies, such as turnover, number of employees, and type of industry; (b) key figures on the company's IT organization are available; and (c) current IT executives are already listed. Companies with more than 5,000 employees were included in this study. Smaller companies often do not have their own separate IT organizations and therefore do not have a CIO. After applying this criterion, 330 companies remained for further investigation. The database was queried in mid-October 2020. No updates or changes after that date were included in the study.

The LinkedIn social network is primarily used to gather information about CIOs. On LinkedIn, users can create profiles that contain information about their professional careers. If a CIO did not have a LinkedIn profile, the German social network alternative Xing or press releases were used. The following information was collected from these sources: (a) the beginning and end of tenure, (b) previous position, (c) degree and specialization of education, and (d) current title of the CIO. The previous position was used to determine whether the CIO was promoted internally or externally. In addition, the department in the previous position was determined. Current IT executives of a company and their predecessors were included in the analysis. The LinkedIn and Xing data were collected until the end of October 2020. Thus, the length of the tenures spans until and included October 2020. All transitions of the CIOs to new positions from this date were not included in this data sample.

B. Data Sample

Data were collected from 384 CIOs from the previously mentioned 330 companies. However, CIOs could only be identified for 268 companies. Thus, no data on IT executives were available for these 62 companies. The tenures of all CIOs cover the period from 1990 to 2020, but only nine CIOs became CIO before 2000. Sixty CIOs took office between 2000 and 2010 and 315 from 2010 onwards.

Table I shows the distribution of the previously mentioned 268 companies based on (a) revenue, (b) IT expenditure, (c) number of employees, and (d) number of IT employees. The companies in which CIOs are employed have a median turnover of €4.5 billion, with most having a turnover of €2 to €5 billion. The median number of employees at these companies was 16,885. IT organizations have a median IT budget of € 77.50 million available for IT expenditure. However, at 32.6%, most companies had a budget of only €2 million. The CIOs had a median number of 365 IT employees. However, many companies have only 100–250 IT employees.

TABLE I. Company Profiles

Turnover in billion €	N	Percentage
<= 2	38	14.2
> 2 to 5	114	42.5
> 5 to 10	48	17.9
> 10 to 20	31	11.6
> 20 to 50	24	9.0
> 50	13	4.9
Total	268	100.0
IT budget in million €	**N**	**Percentage**
<= 50	93	34.7
> 50 to 100	73	27.2
> 100 to 250	51	19.0
> 250 to 500	22	8.2
> 500 to 1,000	17	6.3
> 1.000	12	4.5
Total	268	100.0
Number of employees€	**N**	**Percentage**
5000 to 10,000	72	26.9
> 10,000 to 20,000	89	33.2
> 20,000 to 50,000	60	22.4
> 50,0000 to 100,000	25	9.3
> 100,000	22	8.2
Total	268	100.0
Number of IT employees€	**N**	**Percentage**
<= 100	20	7.5
> 100 to 250	87	32.5
> 250 to 500	69	25.7
> 500 to 1,000	45	16.8
> 1,000 to 5,000	36	13.4
> 5,000	11	4.1
Total	268	100.0

Table II presents the distribution of the CIOs by industry type. Most CIOs in this dataset are from the (a) manufacturing, (b) retail, and (c) automotive sectors.

TABLE II. Industry Types of CIOs

Industry type	N	Percentage
Automotive	43	11.2
Chemical	34	8.9
Construction	11	2.9
Energy and raw materials	23	6.0
Finance	8	2.1
Food	16	4.2
Health	9	2.3
Manufacturing	128	33.3
Media	15	3.9
Retail	59	15.4
Transportation	38	9.9
Total	268	100.0

Table III shows that most of the CIOs (94%) were male. Only 23 of 384 CIOs in this dataset were female. This is well below the 17% reported by Fortune 500 companies in 2016 [16]. However, in 2018, only 2.1% of new German-speaking CEOs were female, which suggests that a low percentage of women were not limited to the position of the CIO [9].

TABLE III. Gender of CIOs

Industry type	N	Percentage
Female	23	6.0
Male	361	94.0
Total	384	100.0

As indicated in Table IV, responsible IT executives in Germany have different titles. By far the most common title is "CIO / Group CIO", which is held by 65% of IT executives. Furthermore, 28 executives hold the title "Director / Managing Director IT" and 37 hold the title "Head of IT /Group IT". However, in some companies, the Chief Digital Officer (CDO) and Chief Technology Officer (CTO) are responsible for IT. This could indicate that there is no CIO position in the organization.

TABLE IV. TITLES OF CIOS

Industry type	N	Percentage
CDO	5	1.3
CIO & CDO	12	3.1
CIO & additional title	13	3.4
CIO (Regional /BU)	7	1.8
CIO / Group CIO	252	65.6
CTO	7	1.8
Director / Managing Director IT	28	7.3
EVP/SVP/VP Group IT	11	2.9
Head of IT /Group IT	37	9.6
Other	12	3.1
Total	384	100.0

As Table V shows, most CIOs have a degree in economics followed by science and engineering. In this study, business informatics and industrial engineering were attributed to economics. These figures clearly show that German CIOs not only have technical or engineering backgrounds [11]. Only three CIOs came from a discipline that could not be assigned. Data were missing for 49 CIO because the field of education could not be specified.

TABLE V. CIOS' FIELDS OF EDUCATION

Field of education		N	Percentage	Valid Percentage
Valid	Engineering	58	15.1	17.3
	Science	97	25.3	29.0
	Economics	177	46.1	52.8
	Other	3	0.8	0.9
	Total	335	87.2	100.0
Missing	Not specified	49	12.8	
Total		384	100.0	

As shown in Table VI, almost all the CIOs have an academic degree. Most CIOs have a master's degree or are equivalent to their highest educational level. 74 CIOs also had doctorate holders. The category "Other" includes non-academic educational backgrounds. If a higher degree means a higher qualification, it is notable that ten people in this dataset have become IT executives without academic education. Data were missing for 36 CIO because the degree of education could not be specified.

TABLE VI. CIOS' DEGREE OF EDUCATION

Degree of education		N	Percentage	Valid Percentage
Valid	Bachelor	12	3.1	3.4
	Master	252	65.6	72.4
	Doctorate	74	19.3	21.3
	Other	10	2.6	2.9
	Total	348	90.6	100.0
Missing	Not specified	36	9.4	
Total		384	100.0	

In addition to education, the previous positions a CIO has held also shape his skills and experience. To test the hypotheses regarding the professional background of the CIO, previous positions were divided into three categories. The entry "CIO" indicates that the executive was already the CIO of another company in the previous position. "IT" stands for a previous position within an IT organization. "Business unit" includes all positions outside the IT organization. As shown in Table VII, the previous positions were distributed almost equally. The percentage of German CIOs who come from a business unit is similar to 31.1% of CIOs in the US [2]. The previous position could not be specified for eight CIOs.

TABLE VII. CIOS' PREVIOUS POSITIONS

Previous position		N	Percentage	Valid Percentage
Valid	CIO	114	29.7	30.3
	IT	138	35.9	36.7
	Business unit	124	32.3	33.0
	Total	376	97.9	100.0
Missing	Not specified	8	2.1	
Total		384	100.0	

As illustrated in Table VIII, most CIOs were hired externally. The percentage of externally hired individuals was 79.3%, which was significantly higher in the US. [2]. It would be interesting to learn the reasons for this discrepancy through future research. Data are missing for five CIO because it could not be specified if the CIO was promoted internally or hired externally.

TABLE VIII. CIOS HIRED INTERNALLY OR EXTERNALLY

Previous position		N	Percentage	Valid Percentage
Valid	Internal	138	35.9	36.4
	External	241	62.8	63.6
	Total	379	98.7	100.0
Missing	Not specified	5	1.3	
Total		384	100.0	

Table IX shows that the CIO reports to many executives. The majority of German CIOs report to the CFO, followed by the CEO. Therefore, hypothesis H8a, which states that CIOs in Germany also report to the CEO by a majority, is not supported. 13 CIOs in this data sample report to the Chief Digital Officer (CDO). This is surprising because the position of the CDO is quite new compared to the CIO. Data are missing for 163 CIOs because the reporting structure could not be specified.

TABLE IX. CIOS' REPORTING STRUCTURES

Previous position		N	Percentage	Valid Percentage
Valid	CDO	13	3.4	5.9
	CEO	55	14.3	24.9
	CFO	95	24.7	43.0
	COO (Operating)	9	2.3	4.1
	CTO	15	3.9	6.8
	Group CIO	4	1.0	1.8
	Other C-Level	17	4.4	7.7
	Is a board member	13	3.4	5.9
	Total	221	57.6	100.0
Missing	Not specified	163	42.4	
Total		384	100.0	

As Table X shows, most CIOs under two-thirds have a tenure of less than five years. However, 59 CIOs had been in office for more than 10 years and had an above-median tenure.

TABLE X. TENURES OF CIOS

Tenure	N	Percentage	Cumulated Percentage
<= 1	41	10.7	10.7
> 1 to 3	114	29.7	40.4
> 3 to 5	87	22.7	63.0
> 5 to 10	83	21.6	84.6
> 10 to 20	50	13.0	97.7
> 20	9	2.3	100.0
	384	100.0	

As shown in Figure 1, CIO tenures were not normally distributed. In addition, owing to the chosen data collection approach, the dataset contained many CIOs that are currently in office. As a result, the average tenure may be distorted. Therefore, current and completed tenures must be considered separately. Figure 1 shows that 36 CIOs had only recently moved to their positions and still had a tenure of less than one year. In addition, most IT executives currently in office have tenures between one and three years. If, on the other hand, one looks at CIOs with a completed tenure, it is noticeable that the tenures are more evenly distributed.

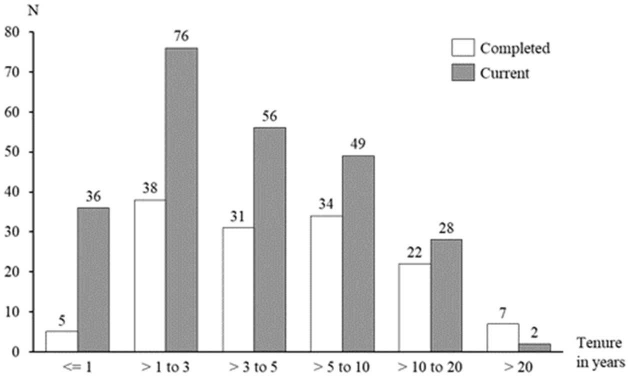


Fig. 1. Distribution of completed and current tenures.

Table XI displays the descriptive statistics for tenure. The median tenure of all 384 is 4.0 years while it is 4.7 years for completed tenures and 3.7 years for current tenures. The arithmetic mean is given in Table XI because it was used to test Hypotheses 1 and 2.

TABLE XI. CIOS HIRED INTERNALLY OR EXTERNALLY

Tenure	N	M	SD	Mdn
Current	247	4.76	4.15	3.70
Completed	137	6.57	5.51	4.70
Total	384	5.50	4.80	4.00

The tenure of the CIOs was analyzed using statistical tests. As shown in Table XI, most CIOs (64%) are currently in office. It was not possible to determine the length of tenure of these CIOs. To avoid excluding 247 CIOs, two groups were formed based on the median of the completed tenures. One group includes all CIOs who have a tenure above or equal to 4.7 years, regardless of whether they are currently in office or not. The other group includes all CIOs with a completed tenure of less than the median of 4.7 years. Figure 2 illustrates the formation of the two groups and highlights the CIOs included in the tests in grey. 152 executives are currently in office and have below-median tenure at the time of October 2020. However, these CIOs may also have an above-median tenure of a few years and might be deemed “successful” in the future. However, since the length of tenure cannot be predicted, and thus it cannot be determined whether the CIOs will be successful in the future, they are excluded from the tests.

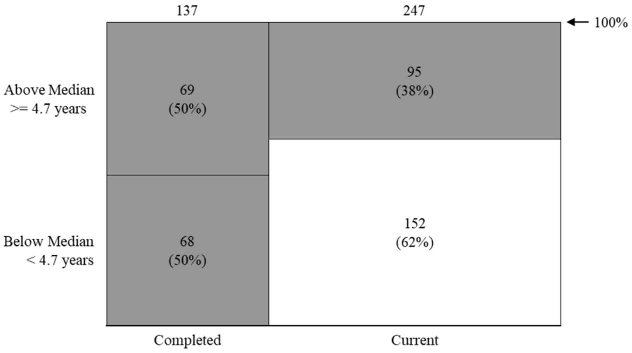


Fig. 2. Distribution of **completed** and current tenures.

As shown in Table XII, 232 CIOs consisting of two groups were used for the statistical tests. Of these, 164 had tenures greater than or equal to 4.7 years. 68 CIOs had tenures of less than 4.7 years.

TABLE XII. GROUPS OF CIOS FORMED

Groups of CIOs	N
Above Median >= 4.7 years (Completed & Current)	164
Below Median < 4.7 years (Completed)	68
Total	232

C. Statistical Methods

IBM SPPS Statistics 27 was used to analyze the dataset and test the hypotheses. As the CIOs were split into two groups, tenure was now a categorical variable. The data collected on the CIO background were categorical variables. Therefore, the chi-squared test was used for the analysis. The chi-squared test can be used to determine whether two or more independent samples differ in their distribution over a variable. A significant test statistic indicated that the groups differed in the distribution of the variables of interest. However, the test did not indicate which group was different [18]. These differences were identified in a post-hoc analysis using Bonferroni correction. If the cells of a cross table had an expected frequency of less than five, Fisher’s exact test was used.

IV. Results

A. Tenure

Since no statistical test was necessary for the first two hypotheses, 384 CIOs were examined here. The average completed tenure shown in Table XI corresponds to the values of 6.5 years and 6.6 years [2, 4]. The average current tenure is also close to the value of 4.6 years [3]. However, it is unclear whether the abovementioned studies included only completed or current tenures. Whether German CIOs have a longer tenure than their US counterparts depends on comparative values. Therefore, H1 was only partially supported.

With 6.6 years, the average tenure of CEOs leaving office corresponds to the mean value of the completed tenures of CIOs, as shown in Table V [9]. This comparison shows that German CIOs do not have shorter tenures than German CEOs do. Therefore, H2 was not supported.

B. Size of the Organizations

Next, we examined whether being a CIO in a large organization can be advantageous for an above-average tenure. A chi-squared test showed that there was no statistically significant difference in corporate turnover between CIOs with below-median and above-median tenures ($\chi 2(5, N = 232) = 6.128, p = .294$). Figure 3 illustrates these results. Even if the difference is not statistically significant, it is noticeable that more CIOs in companies with turnover equal to or less than €2 billion have a tenure that is below average.

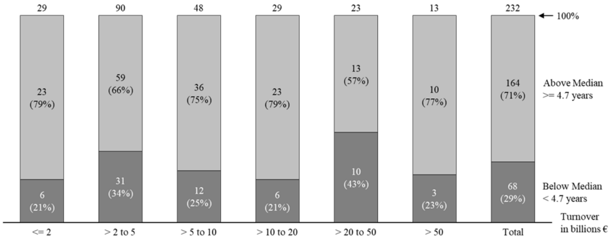


Fig. 3. Comparison of above-/below-median tenures by turnover

In the same way, there is no statistically significant difference in the number of employees of organizations between CIOs with an above-median tenure and CIOs with a below-median tenure, $\chi 2(4, N = 232) = 3.316, p = .506$. Figure 4 shows that the percentage of CIOs with above-median tenure differs between companies with 50,000 and 100,000 employees and companies with more than 100,000 employees.

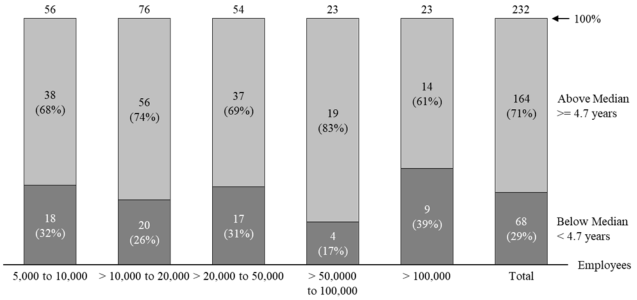


Fig. 4. Comparison of above-/ below-median tenures by employees

Therefore, CIOs in large companies do not have a longer tenure than those in smaller companies, which is why Hypotheses H3a and H3b are not supported. It must be considered that this study only considered companies with more than 5,000 employees. Therefore, further investigations should be conducted without limiting the number of employees.

C. Industry Type of the Organizations

Next, the industry type of the organization in which the CIOs operate is analyzed. An Exact Fisher test was conducted to determine whether there was a difference in the industry type between CIOs with below-median and above-median tenures. The results of this test also showed no statistically significant difference (p = .756). Although Figure 5 shows that all the CIOs in the construction industry have above-median tenures, this can be neglected because of the small number of CIOs. Hypothesis H4, which states that CIOs' tenure does not differ by industry type, is supported.

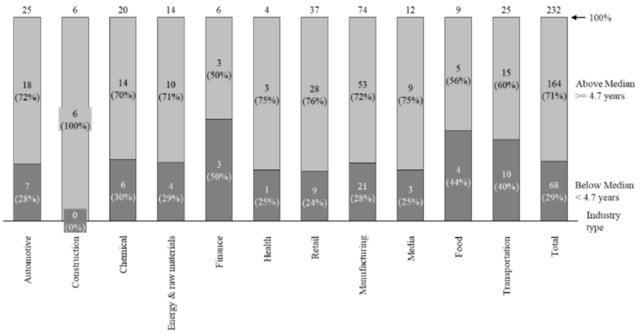


Fig. 5. Comparison of above-/ below-median tenures by industry type

D. Educational Background

The next step was to examine educational background in more detail. This includes the fields of education and the degree of education. An Exact Fisher test was conducted to determine whether there was a difference in the field of education between CIOs with below-median and above-median tenures. The results of this test also showed no statistically significant differences (p = .200). It should be noted that the field of education could not be specified for the 29 CIOs. Even if the difference is not statistically significant, Figure 6 shows that CIOs with a degree in economics are the only group of CIOs with below-median rather than above-median tenure. The test rejects hypothesis H5a, that CIOs with a degree in economics have the longest tenure. Thus, a degree in economics is not a factor that automatically leads to long-term office.

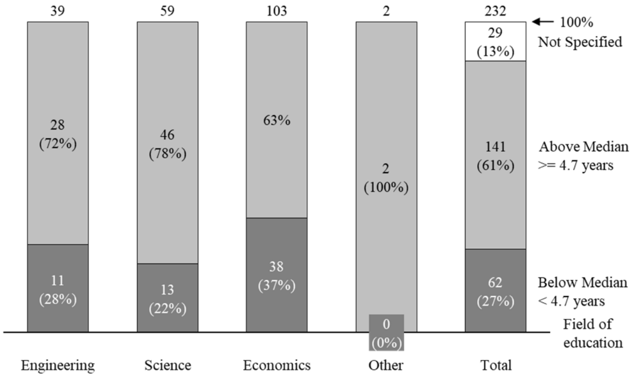


Fig. 6. Comparison of above-/ below-median tenures by field of education

In addition, we examined whether there was a difference in the degree of education between CIOs with below-median and above-median tenure. The results of Fisher's exact test showed no statistically significant difference (p = .751). Therefore, CIOs with above-median tenures do not have a certain degree of education more often than expected. The test shows that a higher degree is not associated with longer tenure. Thus, Hypothesis H5b was not supported. Figure 7 also shows that more CIOs with a doctorate degree have below-median tenures than those with above-median tenures. An additional doctorate did not extend the average tenure of the CIOs. However, the frequency of CIOs with a master's or doctorate degree shows that a higher education degree is a prerequisite for the position of the CIO. It should be mentioned again that the degree of education could not be specified for 20 of the 232 CIOs.

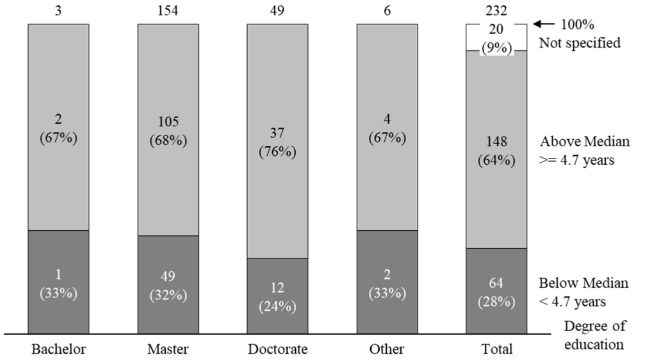


Fig. 7. Comparison of above-/ below-median tenures by the degree of education

E. Professional Background

Next, their professional background was examined. The focus here is on the previous position and whether the CIOs have been promoted internally or moved externally. First, we examined whether there was a difference in the previous position between below-median and above-median tenures. A chi-square test showed no statistically significant difference ($\chi 2(2, N = 225) = 1.059, p = .589$. This supports hypothesis H7a, which states that CIO tenure is independent of the previous position. The results show that there is no position that is a particularly good prerequisite for long tenure. Figure 8 illustrates the results because only the previous CIO position is more likely to have a below-median tenure than an above-median tenure. The previous position could not be determined for 7 of the 232 CIOs.

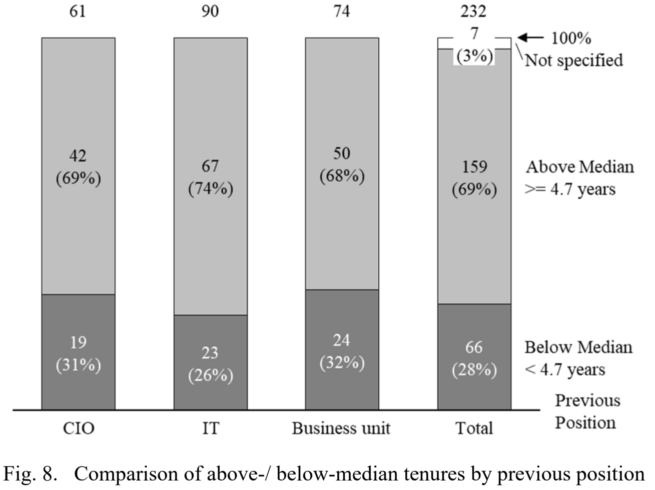
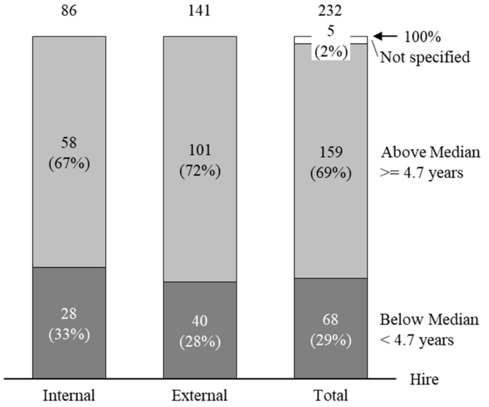


Fig. 8. Comparison of above-/ below-median tenures by previous position

Second, we analyzed whether there was an impact on whether CIOs would be promoted internally or hired from outside. A comparison of above- and below-median tenures showed no statistically significant difference between internally and externally hired CIOs, $\chi 2(2, N = 227) = 0.447$, p = .504. The results of the chi-square test indicate that CIOs with above-median tenures are not promoted internally or hired externally more often than expected. Figure 9 illustrates the results. At approximately 70%, the proportion of CIOs with below-median tenure is similar for both internal and external hires. The previous position could not be determined for 7 of the 232 CIOs. In summary, Hypothesis H6 is supported.

Fig. 9. Comparison of above-/ below-median tenures by internal/ external hire



The next step was to determine whether there was a connection between the previous position and an internal or external hire. As tenure was not a factor, 384 CIOs were included. A chi-square test was carried out to check whether the majority of externally hired CIOs came from a business unit and whether the majority of internally hired CIOs were promoted from an IT position. The test showed a statistically significant difference in the previous position between internally and externally hired CIOs, $\chi 2(2, N=376) = 104.679$, $p < .001$. The effect size for these results was relatively strong (Cramer's V = 0.528) (Cohen, 1988). Post-hoc comparisons using the Bonferroni correction showed that only the difference in the previous IT position was statistically significant. Thus, CIOs with previous IT positions are promoted internally more often than expected. Figure 10 shows that the difference was also significant for the previous CIO positions. However, this difference can be neglected because of the categorization used. The previous position could not be determined for 8 of the 384 CIOs. Therefore, Hypothesis H7c, which states that internally hired CIOs in Germany also rise from an IT position by a majority, is supported. In contrast, IT executives from a position outside IT are not increasingly hired externally. Therefore, Hypothesis H7b is not supported.

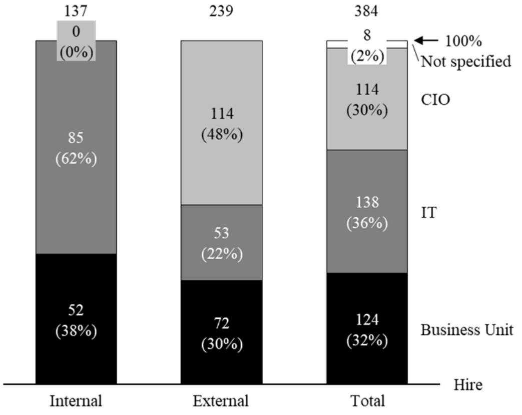


Fig. 10. Percentage of the previous position by internal/ external hire

F. Reporting Structure

Finally, the reporting structure was examined. As Figure 11 shows, the majority of German CIOs report to the CFO, followed by CEO. A more detailed report structure is provided in Table IX. Therefore, Hypothesis H8a, which states that the majority of CIOs in Germany also report to the CEO, is not supported. To test the other hypotheses, the CIOs who report to the CEO or the CFO are primarily relevant. All others were grouped under "Other." To test whether externally hired CIOs reported by a majority to the CEO and internally promoted CIOs reported by a majority to the CFO, a chi-square test was performed. The results show no statistically significant difference in the reporting structure between internally and externally hired CIOs, χ2(2, N = 206) = 2.121, p = .346. Thus, Hypotheses H8b and H8c were not supported. Whether a CIO is promoted internally or hired externally, therefore, has no effect on the executive he reports to. Figure 11 shows the distribution of CIO reports and whether they were hired internally or externally. There were also no differences in the category "Other." When analyzing the reporting structure, it should be noted that for many CIOs, the reporting structure could not be specified.

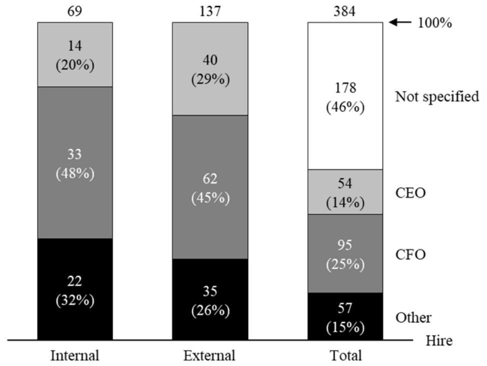


Fig. 11. Percentage of the previous position by internal/ external hire

G. Summary

Table XII summarizes all of the hypotheses tested, the tests used, and the results.

TABLE XIII. SUMMARY OF THE TESTED HYPOTHESES AND RESULTS

Hypothesis	Support	N	χ2	p-value	Effect size
H1: The tenure of German CIOs is comparable to that of US-American CIOs.	Partially *	384	-	-	-
H2: The tenure of German CIOs is shorter than that of German CEOs.	No	384	-	-	-
H3a: The tenure of CIOs differs according to the size of the company, based on the number of employees.	No	232	3.316	.506	-
H3b: The tenure of CIOs differs according to the size of the company, based on turnover.	No	232	6.128	.294	-
H4: The tenures of CIOs do not differ per industry.	Yes	232	**	.756	-
H5a: CIOs with a background in economics have the longest tenures.	No	203	**	.200	-
H5b: A higher level of education is associated with a longer tenure.	No	212	**	.751	-
H6: There is no difference in tenure whether CIOs were hired internally or externally.	Yes	227	0.447	.504	-
H7a: The tenure of the CIOs is independent of the previous position.	Yes	225	1.059	.589	-
H7b: The majority of externally hired CIOs come from a position outside IT.	No	376	104.679	.120	-
H7c: The majority of internally hired CIOs rise from an IT position.	Yes	376	104.679	<.001	Relatively strong
H8a: The majority of German CIOs report to the CEO.	No	384	-	-	-
H8c: The majority of externally hired CIOs report to the CEO.	No	206	2.121	.346	-
H8b: The majority of internally promoted CIOs report to the CFO.	No	206	2.121	.346	-
** Depends on values selected for comparison. ** Fisher's Exact test was used*					

V. Discussion

The results show that neither (a) the size of the company, (b) the industry type, (c) the educational background, (d) the previous positions, nor (e) the reporting structure have a significant impact on the tenure of CIOs. However, several findings require further investigation.

When examining tenures, perspectives and comparative values play a decisive role. The mean value of completed tenures of 6.6 years paints a picture of relatively long tenures. The median of 4.7 years of service paints a more differentiated picture. The perception that CIOs have a short tenure is possibly influenced by the large number of position changes in recent years. For example, the average tenure of IT executives currently in office is only 4.8 years. The median value were 3.7 years. However, the fact that CIOs have only short tenures was not proven by the analysis in this study.

The size of the companies was examined based on their turnover and the number of employees. Due to the exclusion criteria, the companies had a minimum size. Therefore, it cannot be excluded that this is the reason why no differences were observed. An investigation without limitations could provide further insights into whether company size influences CIO tenure.

The analysis clearly shows that German CIOs do not only have a technical or engineering background. Half of CIOs have a degree in economics. However, IT executives with a degree in economics do not have the longest tenures. This study cannot confirm that German CIOs are only technically oriented managers.

IT executives in Germany are well-educated. The vast majority of CIOs have at least one master's degree or diploma at the highest educational level. Many also have doctorate degrees. In terms of current tenure, the proportion of CIOs with only a bachelor's degree increased. An additional master's degree or even a doctorate degree takes more time. It would therefore be interesting to examine whether there is a trend towards an earlier career start and an increase in other forms of further education.

Even if the differences in completed tenures are not statistically significant, a previous position as a CIO does not automatically result in a long tenure in the next CIO position. For example, executives who have moved from an IT position or business unit have a longer tenure. This is surprising, because the executive has already gained experience in the same position in another company. It is also possible, however, that higher expectations are associated with the new manager. For example, a CIO may have been hired for strategic change and may not have been fully implemented.

CIOs that move upward internally are particularly likely to move from an IT position. One reason for this could be that IT executives are familiar with the IT organization and landscape and do not require a long transition period.

Most German CIOs moved to office from an external position. In addition, external hires are most often obtained from previous CIO positions. However, this is not surprising. If an external CIO is hired, he should have already demonstrated leadership qualities in the previous CIO position.

In this study, we only examined whether CIOs moved to the CIO position from a certain previous position, particularly frequently either internally or externally. Therefore, it would be interesting to investigate whether a combination of these criteria allows for long tenure.

Unlike IT executives from the US, German CIOs do not report most frequently to the CEO but to the CFO. As the results show, the German CIO reports to many different managers. Surprisingly, 13 IT executives in this data sample reported to the CDO. This could indicate that IT on its own cannot contribute sufficiently to the digitization of the company. It would therefore be interesting to learn in interviews why the CIOs in these companies are located under the CDO.

VI. Conclusion

This study examines the tenure and background of German CIOs and thereby adds to the current body of knowledge of CIO research [19].

The results show that two-thirds of IT executives have tenures of less than five years. However, almost 15% of CIOs have a tenure of more than 10 years. Furthermore, the results of the statistical tests show that none of the factors considered has a decisive influence on the length of tenure of CIOs alone. Future research could combine these factors to obtain more detailed results.

Besides this limitation, this study also has several other limitations that offer possibilities for further research. The CIO Magazine's Top 500 database and profiles in social networks were chosen as sources for data collection. Therefore, the results of this study depend on the reliability of the data. Not all relevant information can be collected from every executive through these sources. Therefore, future research should focus on other sources of data.

The dataset contains the current CIO of a company, as well as its predecessor. In the case of a predecessor, the length of tenure can be specified, whereas in the case of the current CIO, the length of tenure is still uncertain. Therefore, the two groups were considered separately. However, whether the frequencies of these factors have changed over time has not been examined. For example, the proportion of CIOs from the US that came into office from an IT position has fallen in recent years [2]. Further studies should investigate whether this development can also be observed among German IT executives.

Although the results show that none of the factors significantly influence tenure, we cannot conclude that the background of the CIO is unimportant. For example, a non-IT background does not necessarily mean that the CIO has no IT knowledge or experience [6]. When appointing new CIO positions, the skills of the CIO should align with the company's requirements. The type of CIO organization needs vary with time and industry [6]. Based on this background, an organization can select a suitable IT executive. However, the CIO must understand why he was hired [15]. This allows him to align his studies with the expectations of the organization.

In this study, we examined factors that can be regarded as preconditions for the position of the CIO. These factors do not significantly influence IT executives' tenure. Therefore, we endorse Jones et al. 's proposal to investigate what a CIO does once he is in office, regardless of his background [6]. This will help to draw a more complete picture of the factors that significantly influence the tenure of CIOs.

References

[1] S. S. Dikolli, W. J. Mayew, and D. Nanda, "CEO Tenure and the Performance-Turnover Relation," *Rev Account Stud*, vol. 19, no. 1, pp. 281–327, 2014, https://doi.org/10.1007/s11142-013-9247-6.

[2] L. Kappelman et al., "The 2019 SIM IT Issues and Trends Study," *MISQE*, vol. 19, no. 1, pp. 69–104, 2020.

[3] Korn Ferry, *Age and Tenure in the C-Suite*. [Online]. Available: https://ir.kornferry.com/news-releases/news-release-details/age-and-tenure-c-suite-korn-ferry-study-reveals-trends-title-and

[4] CIO Magazine, *2020 State of the CIO*. [Online]. Available: https://www.idg.com/tools-for-marketers/2020-state-of-the-cio/

[5] G. S. Dawson, M.-W. Ho, and R. J. Kauffman, "How Are C-Suite Executives Different? A Comparative Empirical Study of the Survival of American Chief Information Officers," *Decision Support Systems*, vol. 74, pp. 88–101, 2015, https://doi.org/10.1016/j.dss.2015.03.005.

[6] M. C. Jones, L. Kappelman, R. Pavur, Q. N. Nguyen, and V. L. Johnson, "Pathways to Being CIO: The Role of Background Revisited," *Information & Management*, vol. 57, no. 5, pp. 1–14, 2020, https://doi.org/10.1016/j.im.2019.103234.

[7] K. Drechsler, "Information Systems Executives: A Review and Research Agenda," *ECIS 2020 Research Papers*, pp. 1–16, 2020.

[8] G. M. Hunter, "The Chief Information Officer: A Review of the Role," *Journal of Information*, vol. 5, no. 1, pp. 125–143, 2010, http://dx.doi.org/10.28945/1328.

[9] Strategy&, 2018 *CEO Success Study*. [Online]. Available: https://www.strategyand.pwc.com/de/de/studien/ceo-success/ceo-success-gsa-deep-dive-2018.pdf

[10] S. Neifar and H. Ajili, "CEO Characteristics, Accounting Opacity and Stock Price Synchronicity: Empirical Evidence From German Listed Firms," *J. Corp. Acct. Fin*, vol. 30, no. 2, pp. 29–43, 2019, http://dx.doi.org/10.1002/jcaf.22386.

[11] V. Krotov, "Bridging the CIO-CEO Gap: It Takes Two to Tango," *Business Horizons*, vol. 58, no. 3, pp. 275–283, 2015, https://doi.org/10.1016/j.bushor.2015.01.001.

[12] D. J. Mazzola, R. D. St. Louis, and M. R. Tanniru, "The Path to the Top: Insights From Career Histories of Top CIOs," *Communications of the ACM*, vol. 60, no. 3, pp. 60–68, 2017, http://dx.doi.org/10.1145/2959086.

[13] P. A. Gonzalez, L. Ashworth, and J. McKeen, "The CIO Stereotype: Content, Bias, and Impact," *The Journal of Strategic Information Systems*, vol. 28, no. 1, pp. 83–99, 2019, https://doi.org/10.1016/j.jsis.2018.09.002.

[14] R. Babin and K. A. Grant, "How Do CIOs Become CEOs?," *Journal of Global Information Management*, vol. 27, no. 4, pp. 1–15, 2019, https://dx.doi.org/10.4018/JGIM.2019100101.

[15] A. B. Gerth and J. Peppard, “The Dynamics of CIO Derailment: How CIOs Come Undone and How to Avoid it,” *Business Horizons*, vol. 59, no. 1, pp. 61–70, 2016, https://doi.org/10.1016/j.bushor.2015.09.001.

[16] SpencerStuart, *State of the CIO in 2018*. [Online]. Available: https://www.spencerstuart.com/research-and-insight/the-state-of-the-cio-in-2018

[17] CIO Magazin, *Top-500*. [Online]. Available: https://www.cio.de/top500

[18] T. M. Franke, T. Ho, and C. A. Christie, “The Chi-Square Test,” *American Journal of Evaluation*, vol. 33, no. 3, pp. 448–458, 2012, https://doi.org/10.1177/1098214011426594.

[19] S. Kratzer, M. Westner, and S. Strahringer, „Four Decades of Chief Information Officer Research: A Literature Review and Research Agenda Based on Main Path Analysis,” *The Data Base for Advances in Information Systems,* vol. 54, no. 3, 2023.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 567–576
DOI: 10.15439/2023F5469
ISSN 2300-5963 ACSIS, Vol. 35

A Reusability-Oriented Use-Case Model Specification Language

Bogumiła Hnatkowska, Piotr Zabawa
0000-0003-1706-0205
0000-0002-5078-9869
Wroclaw University of Science and Technology
Wyb. Wyspianskiego 27, 50-370 Wroclaw, Poland
Email: {bogumila.hnatkowska, piotr.zabawa}@pwr.edu.pl

***Abstract*—Use-case models play an essential role in software development processes. They are used to specify functional requirements, estimate software development project efforts, and plan iterations. The use-case model is subject to change as requirements are modified, or the model is refactored. Therefore, it is essential that the use-case model is not redundant and its parts are reusable. Existing approaches for use-case model specification support reusability in a limited way. This paper fills the gap. It introduces a new approach to conveniently yet semi-formally specifying the entire use-case model. The paper presents the Use-Case Flow Language metamodel, consisting of its abstract syntax and a description of the semantics of the metamodel elements. A concrete textual syntax of the language is also provided and informally described. An example of a use-case model specified in the proposed notation is presented in the paper.**

I. Introduction

THE SOFTWARE requirements specification (SRS) is one of the most important artifacts documenting the qualities of a software product. It is always produced regardless of the development methodology used. The SRS can take different forms, including use-case models, product backlogs with user stories, or documents written in free natural language. In the case of a use-case model, the SRS consists of a use-case diagram and the associated use-case specification documents, typically documented using structured texts, tables, or graphical notations (e.g., activity diagrams).

Textual specifications are the most widely used because they are easy to understand and quick to define, even for non-technical people. Still, on the other hand, they can be misinterpreted or incomplete [1]. Therefore, many researchers (e.g., [2], [3], [4]) try to define templates, a set of patterns or rules that help to keep use-case specifications complete, coherent, and consistent.

The important aspect of use-case specification, not fully covered by the existing research, is the specification reusability. The same steps, step sequences, flows, or subflows can be applied in many places when the change in one place will influence all their instances. There are already defined some reusability mechanisms in the UML, like «include» and «extend» dependencies and generalization. However, they are defined at the use-case level. To have an advantage of these mechanisms for use-case fragments, one should introduce new use-cases just for the reusability, which would increase the use-case number significantly, making the use-case diagram and the whole use-case model more complicated.

The paper aims to define a general-purpose language for writing textual use-case model specifications, emphasizing reusability. The proposed notation is consistent with existing good practices, and the result of their application, i.e., the textual use-case specification should have the necessary features to allow its further processing, e.g.:

- Checking use-case models' completeness and correctness.
- Bi transformation into diagrammatic notations, e.g., activity diagrams or flow visualization.

The motivation to cover the use-case model by a standardization process was and still is very strong. The reasons for standardization efforts are as follows:

- The use-case model is used for the specification of functional requirements.
- Use-cases are the source information not only for the implementation of a software product but also for the verification of the product by functional tests; the functional tests can be implemented directly from the use-cases in parallel with the implementation.
- The use-case model is used to estimate the development efforts (use-case points method [5]).
- Use-cases play a crucial role in iterative software development projects as the iteration plans are organized for a set of use-cases or similar constructs.

The contribution of this paper is a notation specification of a use-case model flow language (Use-Case Flows Language, UCFL) used for scenario definition as a supplementary part of a use-case diagram. The specification includes the language metamodel (abstract syntax) – see section III, and concrete textual syntax (T-UCFL) – see section IV. The metamodel takes the form of a UML class diagram, while the concrete syntax is given in the form of context-free grammar. The notation is presented with several examples. It is characterized by a minimal set of keywords used in use-case flow steps. Examples of the T-UCFL usage are contained in section V. And, finally, the content of the research presented in the paper is summarized in section VI.

II. Related Work

A metamodel is a typical form of abstract syntax representation, also used for use-case models ([6], [7]). Such a metamodel can have many different representations, both graphical and textual. The authors decided to propose their own metamodel for the use-case specification formalism in order to overcome the limitations of existing, e.g., the lack of iterations or interruptions.

The concrete syntax of UCFL called T-UCFL takes the form of a free context grammar – such a solution was used in [8] for a similar purpose. The grammar has been developed with best practices in mind. As this is a textual specification, the authors draw inspiration from many existing books [2]-[4] and papers [9]-[10]. These references suggest, among others, different templates of use case scenarios, keyword sets, and ways of identifying steps. To the best of the authors' knowledge, none of them pay attention to the reusability of the step, step ranges, or global flows. The existing reusability mechanisms are defined as reusable templates [11] or patterns [6], [7].

Use-case specifications with globally visible flows are collected in a use-case model. A similar idea is given in [12], where the authors suggest using "several mechanisms to factor out common usage like error handling from sets of use-cases", but the idea is not formalized.

Common elements proposed in this paper include:

- use case name,
- documentation – can be a substitute for a goal, brief description, primary and secondary actors,
- pre- and post-conditions – similar to [13], post-conditions can be divided into subgroups depending on the scenario and return a specific state [9]; such a construct can be used to model minimal guarantees and success guarantees,
- subflows – as potential elements of reuse,
- the main flow of events – sometimes called scenario ([4]) or basic/normal course of events ([8], [12]),
- flows – called alternative flows ([4]), alternative courses (e.g. [2]), or extensions (e.g. [3]).

Use-case flow is typically expressed in terms of actions. Sometimes these actions have no implicit structure, e.g. [12], [4], when the scenario is simply a sequence of sentences. In the proposed approach, the actions are classified and uniquely identified by step identifiers, which enables their reusability. The step identification resembles the one proposed in [2], [11] and implemented in some tools, e.g., CaseCompleted [14] or Enterprise Architect [15].

The use-case semantics, especially the control flow, must be clearly defined. This can be achieved by using specific keywords. The keywords used in the literature to represent the control flow are as follows:

- GOTO step ([11]) or USE-CASE CONTINUES AT step ([14]).
- IF-THEN-ELSE-ELSEIF-ENDIF, MEANWHILE, VALIDATES THAT, DO-UNTIL, ABORT, RESUME step, INCLUDE use-case, EXTENDED BY use-case ([13]).
- IF, VALIDATES, RESUME FLOW, goto, and resume statements are also defined indirectly in separate columns with appropriate names (alternative FlowId, resume FlowId) as identifiers to steps ([16]).
- USE-CASE CONTINUES AT, RETIRED n TIMES, ENDS IN state ([9]).
- COND, INVOKE, REJOIN, FINAL: state ([7]).

Most of them were adapted in the proposed language, e.g., goto, validates that, includes, extends, and final.

III. UCFL Metamodel

This section presents the abstract syntax and semantics of the Use-Case Flow Language specification. The UCFL abstract syntax, in the form of the UML class diagram, is shown in Fig. 1. As the notation focuses on the specification of use-case behavior, the UCFL abstract syntax does not contain either actors or the relationships between them.

A. UCFL Containers

Container is a named element containing flows or their refinements – subflows. We have two types of containers: use-case model and use-case.

1) Use-Case Model: Use-case model is a container of use-cases. It can define publicly visible flows and subflows. Optionally, the use-case model can specify so-called use-case model interruptible regions or flow-interruptible regions (see section III-G) and be documented by a string.

2) Use-Case: Use-case is a basic modeling element that represents interactions between the system and its actors via flows and subflows. It may have optional documentation describing the use-case goal. The use-case may also specify use-case interruptible regions (see section III-G).

Pre and postconditions: A use-case may require some preconditions to be met in order to enable the use-case behavior. These preconditions (if any) are sentences in natural language. The use-case behavior can change the state of the system. The state changes are represented by postconditions. Each postcondition is a sentence in natural language with a state name, e.g., success, partial success, failure, or other, defined by a modeler.

Generalization: Use-cases can be related to each other with generalization relationships. A use-case can be the parent of many use-cases (children). Only leaves of the inheritance tree can have flows defined. A justification for this decision is given later in this section.

B. UCFL Container Elements

1) Flow: Flow is a key element used to structure the use-case behavior. It is a sequence of steps referring to actions performed either by an actor or by the system. A step has a sequence number and a step identifier constructed from the flow identifier.

From the perspective of a graphical language representation, a flow is a path (possibly looped) in the graph without any branches. Flows can be assembled into a graph using specific actions, e.g., conditional. The first action in the flow can

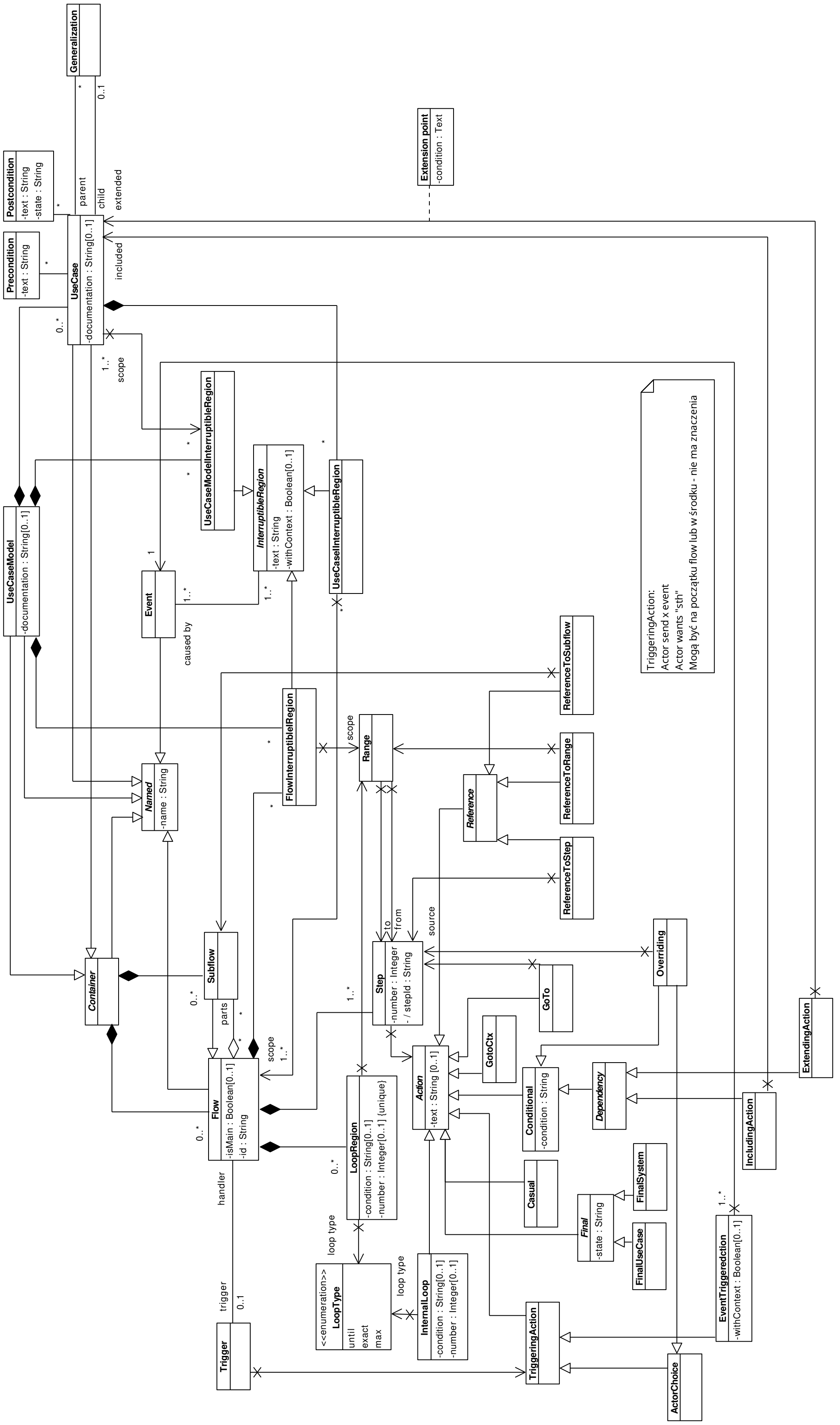


Fig. 1. UCFL abstract syntax

connect it to another flow (as a branch of another flow) - see Fig. 2.

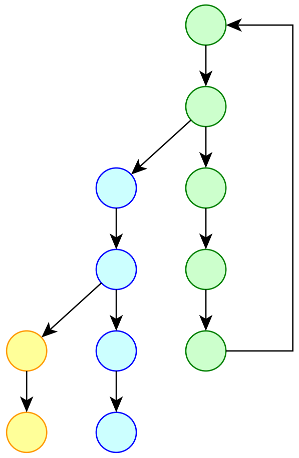


Fig. 2. Flow visualization – different flows are represented by different colors

The flow declaration introduces a flow identifier and a name (both of which must be unique within the context of the flow owner) and, optionally, a trigger. A trigger specifies an action (called a triggering action) that enables the flow. If the flow has a trigger, the flow is called a handler. If the owner of the flow is a use-case then the flow can be marked as a main flow (a use-case must define exactly one main flow; other flows are alternatives).

A flow can additionally define flow interruptible regions (see section III-G) and loop regions (see section III-E).

A flow can be constructed from subflows.

2) Subflow: Subflow is a specialized flow with the restriction that its steps must refer to actions that form a sequence that is casual, finals, and internal loops actions (see section III-F). The subflow is a primary reusable element. It can be shared by several flows; however, a subflow cannot contain interruptible regions or loop regions.

C. Range

Range is a sequence of steps (from–to) included in one flow. Ranges define the scope of flow interruptible regions (see section III-G) or loop regions (see section III-E).

D. Loop Type

Loop type is an enumeration of literals defining different types of loops: `until` (do something until condition), `exact` (do something the exact number of times), `max` (do something the maximum number of times). The type is specified when defining a loop region or an internal loop action.

E. Loop Region

Loop region is the specification of a range that can be repeated in the manner defined by a Loop type. If the Loop type is set to `until`, the condition for the loop region must be defined. Otherwise, the number attribute must be set.

F. Actions

Each step of the flow must refer to one action describing the actor-system interaction in an informal way (*text* attribute in *Action* metaclass).

1) Triggering Actions: A trigger specifies an action (so-called triggering action) that enables the flow. It is the only action that does not need to be referenced by a flow step because it is assumed that it will be performed by an actor to start the flow. There are two types of triggering actions: actor choice action and event-triggered action.

Actor choice action: A flow can be started at the request of the actor, represented by the Actor's choice triggering action.

Event-triggered action: A flow can be started by an actor sending an event to the system, which is modeled by an event-triggered action.

The event-triggered action must refer to an event and optionally can contain a request to store the context before the event is handled (attribute *withCtx*). The event is understood as something that happens at a specific time that requires the system reaction. The event has a name that serves as an event identifier. The context defines the name of the running flow or subflow within the region scope (if any) and its running step, which allows the behavior to be resumed later.

2) Casual action: Casual action is the most general. It is used to model anything the actor or system must do, that cannot be expressed by other actions.

3) Finals: A modeler can define a final action to express that the system has completed its operation (Final system action) or that a use-case has completed its operation in a particular state (Final use-case action). Such an action should be the last one in the flow (or subflow in the case of the final system action).

4) Conditional: Conditional action represents a decision made by the system under specific conditions. Such an action may check whether another use-case has ended in a particular state. It is usually the first action of the alternative flows.

5) Internal Loop: Internal loop represents a case where a particular action is to be repeated in the manner defined by the loop type (see section III-D for details).

6) GoTos:

GoTo: Goto action is used to define unconditional loops. You can jump to a particular step in the same flow or any flow in the same use-case provided that the referenced step exists.

GoTo Ctx: The special version of goto action - `Goto ctx` - allows you to return to the previously saved context (the interrupted action is executed again).

7) Overriding: Overriding is a specific action used as a branching mechanism in the flow definition. This action points to the step in the base flow that is being overridden. The action in the source step must be of the same type as the parent of the overriding action (Actor choice or Conditional).

8) References: References represent the reusable elements of the T-UCFL. Depending on the scope of reusability, three types of reference are distinguished: reference to step, reference to subflow, and reference to range.

Reference to step: Reference to step is the simplest reference action, where the scope of reusability is limited to a single action defined in the step to which the reference action refers. You can imagine that the reused action is copied in place of the reference to the step action.

Reference to subflow: Reference to subflow is the reference action in which the scope of reusability is a particular subflow. When the subflow activity is finished, the control flow is passed back to the original flow.

Reference to range: Reference to the range is the reference action in which the scope of reusability is limited to a specific range.

9) Dependencies: Including: A use-case can include the behavior of another use-case. The semantics of this action is similar to the «includes» relationship in the UML [17] where the including use-case is the owner of the flow with the including action, and the included use-case is the one indicated by the including action.

Extending: The flow of a use-case can contain an extending action. The semantics of this action is like the «extends» relationship in the UML [17] where the extended use-case is the flow owner with the extending action, and the extending use-case is that indicated by the extending action. The extension point describes a condition that must be satisfied for the extension to take place.

G. Interruptible Regions

The UCFL allows the definition of interruption (exception) handling mechanisms using so-called interruptible regions. Such a region points to its scope. The scope of the region can be either a set of use-cases (use-case model interruptible region), a set of flows defined within a use-case model, or a use-case (use-case interruptible region), a range (flow interruptible region). The scope can be interrupted by any event, that caused the interruption.

1) Use-case model Interruptible Region: Use-case model Interruptible Region enables specification of the interruption mechanisms at the use-case model. The interruption scope can refer to any flow or a use-case defined in this container.

2) Use-case Interruptible Region: Use-case Interruptible Region enables specification of the interruption mechanisms at the use-case level. The interruption scope can refer to any flow defined in this container.

3) Flow Interruptible Region: Flow Interruptible Region enables specification of the interruption mechanisms at the flow level. The interruption scope can refer to a range (step from, step to) defined in the flow context.

H. Use-case generalization relationship

Use cases are classifiers and can inherit one from another. An example of such inheritance is shown in Fig. 3. Assuming that the use-case specification is given in natural language, the question arises of how the use-case generalization influences their specification, which can "include possible variations of its basic behavior, including exceptional behavior and error handling" [17].

Generally, a behavior is a specification of events that may occur during the use-case lifetime. The specification must contain at least one event – the event of its invocation [17]. The behavior is invoked when an instance of the owning classifier (i.e., use-case) is created.

In the case of use-case inheritance, a child's specification of events (including the triggering one) is inherited from the parent use-case, which makes the whole specification ambiguous. Therefore, to avoid possible problems and misinterpretations, we assume that any parent use-case must serve only as a root of a use-case hierarchy. Use-case triggers for the hierarchy leaves should determine which child to run.

IV. T-UCFL Informal Description

This section demonstrates the use of the T-UCFL concrete syntax ([18]) with several examples. The language grammar has been designed to keep the language flexible and concise. However, as the specification is intended to be processed by computers, the grammar may impose some constraints on the use of the language, such as the need to enclose elements in quotes or the use of certain keywords.

A. T-UCFL Containers

The container – as an abstract class – has no textual representation.

1) Use-case Model: A use-case model is a container and a namespace for all other elements. Its declaration defines the model's name (e.g., Buying) and optional documentation. Its definition contains shareable elements with global visibility (flows and subflows), an optional declaration of interruptible regions, and a list of use-cases. The concrete syntax assumes that the documentation is textual; however, for readability purposes, the authors decided to use a graphical version in the example presented below (see Fig. 3).

```
Use-Case Model: Buying
Documentation:
```

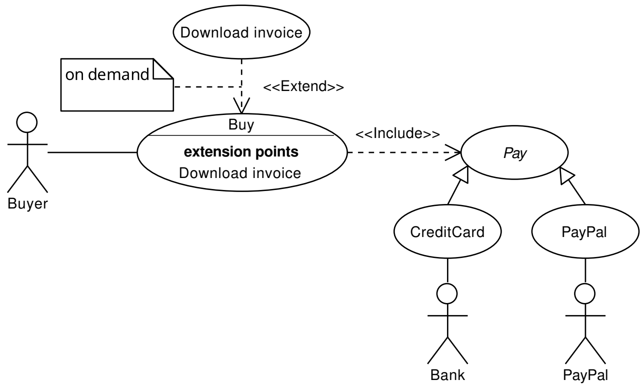


Fig. 3. Buying use-case model documentation in the form of a use-case diagram

2) Use-case: Use-case specification consists of a use-case declaration followed by the use-case definition. The use-case declaration defines a unique use-case name within the use-case model (e.g., `CreditCard`) and, optionally, use-case documentation.

```
Use-Case: CreditCard
Documentation: "Use-case enables payment
  with a credit card."
```

3) Pre and postconditions: A use-case declaration can also contain pre- or post-conditions placed after documentation (if any). The precondition section has one or more statements expressing conditions, for example

```
Preconditions:
- "Actor is logged in the system."
```

Quotation marks are required by formal grammar and can be skipped if the use-case specification is not going to be automatically translated.

Each post-condition section, if any, should define a name of a final system state name (e.g., success, partial-success, error) followed by one or more conditional statements, e.g.,

```
Postcondition(success):
- "An order is stored by the system."
```

4) Generalization: If a use-case has a parent, its name follows the child use-case name and "`-->`" symbol, e.g., `Pay` is the parent use-case for the `CreditCard` use-case:

```
Use-Case: CreditCard --> Pay
```

B. T-UCFL Container Elements

1) Flow: Flow is a named element with an additional string identifier. A flow can be defined in the context of a use-case model, typically as a handler for some event or in the context of a particular use-case. A use-case should have exactly one flow with the reserved name: `Main flow`, and any number of alternative flows with unique identifiers.

Each flow defines a sequence of numbered steps. The step number is constructed with a sequence number preceded by the flow identifier (skipped for the main flow), e.g.:

```
Use-Case: Buy
Trigger: ...
Main flow:
  1. ...
  2. ...
Flow B: The_order_data_invalid
  B1. ...
Flow C: Unsuccessful_payment
  C1. ...
```

The example shown above presents the `Buy` use-case with the main flow and two alternative flows `B` and `C` (`B` is the identifier, `The_order_data_invalid` – is the flow name). The main flow of the use-case has a triggering action defined.

2) Subflow: A subflow is an element of reuse. It can be visible globally (subflows defined at the use-case model level) or locally (subflows defined at the use-case level). They serve to split long flow definitions into manageable fragments. Only casual, final, and internal loop actions are allowed in the subflow definition.

A subflow example is given below:

```
Subflow P: Car_info
  P1. ...
  P2. ...
```

C. Range

Range defines a subsequence in a flow, identified by two steps identifiers, e.g., `2.-3.` consists of 2 steps (2 and 3 in the `Main flow`), `A.5.-A.7.` consists of 3 steps in the flow `A`.

D. Loop Type

Loop type is a keyword (one of `until`, `exact`, `max`) used together with a loop region or internal loop action to specify the loop type – see section V for examples.

E. Loop Region

The loop region works in a similar way to an interruptible flow region. It specifies a range of steps to be repeated as specified by the associated loop type. The loop region is placed after all the steps of the flow, e.g.,

```
Main flow:
  1. ...
  2. ...
Steps 1-2 can be repeated exactly 3 times
Steps 1-2 can be repeated
  until "condition."
```

F. Actions

1) Triggering Actions: A triggering action is typically used to specify how an actor starts a particular flow. In this case, it is specified before the flow, after the `Trigger` keyword. Examples of triggering actions include Actor choice action or Event-triggered action. However, the triggering actions can also be referenced by flow steps.

Actor choice action: A flow can be started by an actor (their decision). Such action must start with `Actor wants` and be followed with "decision" written in quotation marks, e.g.,

```
Use-Case: CreditCard --> Pay
Trigger: Actor wants "to pay with a credit
card"
Main flow: ...
```

2) Event-triggered action: A flow can also be started by an event sent asynchronously by an actor. Such an action must start with `Actor sends` and be followed with the event name and one of `event` or `event with ctx`, e.g.,

```
Actor sends cancelling_service event
Actor sends cancelling_service event
```

```
  with ctx
```

The latter action contains the request to store the context before the event is handled.

3) Casual action: Casual action is the most general. It is a free text without keywords present in other types of actions like `verifies`, `includes`, `ends with` or `goto` (e.g., `"System asks about the order data"`), representing something that the actor or the system must do. The grammar requires this action to be enclosed in quotation marks.

4) Finals: The modeler can define a final action expressing that the system finishes operation (`The system ends`) or a use-case finishes in a specific state (e.g., failure) with the phrase: `The use-case ends with failure`. Such an action should be the last one in the flow. The first one means that the system stops running.

5) Conditional: The conditional action represents a decision made by the system. It must contain the phrase `System verifies` or `System verifies that`, followed by a phrase containing a condition, e.g., `System verifies that "the order data are valid"`. Such an action may check whether another use-case ended in a specific state, e.g. (`System verifies that Pay use-case ended with failure`).

6) Internal loop: One can define that a given action should be repeated a specific number of times specifying its loop type, e.g.,

- `"Actor selects products" max 3 times.`
- `"Actor selects products" until "he is satisfied".`

7) GoTos: GoTo: GoTo action is used to define unconditional loops. We can jump to a particular step in any flow defined in the specific context (a use-case model or a use-case) provided that the referenced step exists, e.g., `Goto 2.` (a jump to the 2nd step in the `Main flow`), `Goto A3.` (a jump to the 3rd step of flow `A`).

GoTo Ctx: GoTo ctx is a special version of the goto action that passes the control flow to the previously saved context (if any). If no context is stored, the semantic is undefined. The interrupted action defined by the context is executed again.

8) Overriding: Overriding actions are used to link flows in a graph. They point to the action in another flow and should be of the same type as the overridden action. Typically, they start alternative flows in a use-case. An example of an overriding action when a decision is made by the system might look like this:

```
B1.3. System verifies that "the order
      data are invalid"
```

The 3rd step in the main flow will be overridden in the `B` flow with the action given above.

9) References: A reference is a basic reusability mechanism. One can reuse another step, step range, or subflow behavior. Examples of such actions are given below:

- `A1.2.` (a step reference; in the 1st step of flow `A`, the 2nd step of the main flow is reused)
- `A2.B3.` (a step reference; in the 2nd step of flow `A`, the 3rd step of flow `B` is reused)
- `B3.A1.-A2.` (a range reference; in the 3rd step of the flow `B`, the range of two steps 1-2 from the flow `A` is reused)

Technically, the reference to a singular step or step range can be thought of as a shortcut for a preprocessing mechanism that copies the referenced elements to the places where they are used and renumbers the steps respectively. Let us assume that flow A contains the steps:

```
A3. Action 1
A4. Action 2
```

and that flow B contains the steps:

```
B1. Action 3
B2. A3.-A4.
B3. Action 4
```

The result of such preprocessing can look like this:

```
B1. Action 3
B2.a. Action 1
B2.b. Action 2
B3. Action 4
```

Subflows must be directly referenced (keyword `subflow` followed by the subflow name) in the appropriate actions, e.g.

```
A2.subflow Car_Info
```

10) Dependencies: Including: One use-case can include or extend another use-case behavior. This is modeled with dependency actions: including or extending. An example of the including action is given below:

```
System includes Pay use-case.
```

When the included use-case reaches the final action, the control returns to the including use-case.

Extending: Two use-cases can also be linked with an extension relationship. The flow of the extended-use case should contain the extension point definition, e.g.,

```
Extension point: "Actor requires the
  invoice downloading."
```

The flow is extended with `Download_invoice` use-case

The extension point specifies a condition under which the flow is extended with another behavior (here: "Actor requires the invoice downloading"). The control returns to the extended use-case when the extending use-case reaches the use-case final action.

G. Interruptible Regions

An interruptible region defines a scope for which the normal operation of the system can be interrupted by a specific event (its name is given) coming from an actor.

1) Use-case model Interruptible Region: Use-case model interruptible region is the one with the widest scope. If it is present, it is placed at the beginning of the use-case model definition, e.g., where any use case can be interrupted by the `close_system` event.

```
Use-Case Model: Document_Editor
Any use-case can be interrupted
  by close_system event
```

2) Use-case Interruptible Region: The scope of a use-case interruptible region is limited to a specific use-case. If it is present, it is placed at the beginning of the use-case definition, e.g.,

```
Use-Case Model: Buy
Any flow can be interrupted by
  close_system event
```

3) Flow Interruptible Region: The scope of a flow interruptible region is limited to a range within a specific flow. If it is present, it is placed after all flow actions, e.g.,

```
1. ...
10. The use-case ends with success
Steps 1.-3. can be interrupted by
  cancelling_service event with ctx
```

The flow interruptible region narrows the scope of the event handling mechanism, e.g., the interruption will be only handled within between steps 1-3 (inclusively).

V. Example Specification

This section contains a small example of a use-case model from Fig. 3, which presents most of the constructs introduced informally in section IV.

The first part of the T-UCFL model specification is related just to the model:

```
Use-Case Model: Buying
Trigger: Actor sends
  cancelling_service event
Flow A: Cancelling_service_event_handler
  A1. "System asks
      for cancellation confirmation"
  A2. Actor wants
      "to cancel the operation"
  A3. The use-case ends with failure

Flow B: Cancellation_denied
  B1.A2. Actor wants "to deny cancellation"
  B2. Goto ctx
```

This part of the model specification is composed of the use-case model called `Buying`; global flow `A` named `Cancelling_service_event_handler`, which is shared among all use-cases and can be triggered by the `cancelinig_service` event generated by an actor; the global flow `B` named `Cancellation_denied` which is a branch of the `A` flow (see a reference step `B1.A2.`). The `Goto ctx` action (if performed) will pass the control flow to the context.

The remaining parts of the T-UCFL specification contain subsequent use-case specifications.

```
Use-Case: Buy
Postcondition (success):
- "An order is stored by the system"
- "An invoice is generated, assigned
  to the order, and stored by the system"

Trigger: Actor wants "to buy an item"
Main flow:
  1. ...
  2. "System asks about order data
     (including payment method)"
  3. "Actor delivers the order data"
  4. System verifies that "the order
     data are valid"
  5. System includes Pay use-case
  6. System verifies that "the Pay
     use-case ended with success"
  7. "System stores an order,
     generates an invoice,
     and sends it by e-mail"
  8. "System informs about
     order completion and enables
     an option to download the invoice"
  9. Extension point: "Actor requires
     invoice downloading"
The flow is extended with
  Download_invoice use-case
  10. The use-case ends with success

Steps 1.-3. can be interrupted
  by cancelling_service event with ctx
```

The main flow contains several conditional actions (e.g., 4, 6). The 5th step contains an including action (`Pay` use-case is included). The 9th step contains an extending action with the condition defined. Finally, there is a flow interruptible region consisting of step range `1.-3.`.

Then two alternative flows (`B` and `C`) are defined:

```
Flow B: The_order_data_invalid
  B1.4. System verifies that
        "the order data are invalid"
  B2. "System informs about invalid data"
  B3. Goto 2.

Flow C: Unsuccessful_payment
  C1.6. System verifies that
        "the payment ended with failure"
  C2. "System informs about the lack
     of payment"
  C3. Goto 2.
```

In both cases, the first step refers to the step with conditional action in the main flow and contains the condition, which complements the condition from the main flow.

The specification continues with the `Download_invoice` use-case specification:

```
Use-Case: Downolad_invoice
Postcondition (success):
- "An invoice is downloaded to
   the Buyer's computer"

Main flow:
  1. "System presents the invoice details
     and asks for confirmation
     of the invoice download"
  2. Actor wants "to download the invoice"
  3. "System sends the last buyer invoice
     to the buyer's computer"
  4. The use-case ends with success
Steps 1.-3. can be interrupted
  by cancelling_service event with ctx

Flow B : Downloading_not_confirmed
  B1.2. Actor wants "to skip downloading"
  B2. The use-case ends
      with partial success
```

The use-case has only one alternative flow `B`, which – in contrast to the `Buy` use-case, is started by the actor's choice action.

There is also a group of three interrelated use-cases in the `Buying` use-case model. The first is the `Pay` use-case, which is abstract and has no flow. It has the following form:

```
Use-Case: Pay
Documentation: "Abstract use-case.
  A root hierarchy for different
  payment methods"
Postcondition (succes):
- "payment succesfull"
Postcondition (failure):
- "payment unsuccesfull"
```

The specification also contains two concrete use-cases (`CreditCard`, `PayPal`) that inherit from the `Pay` use-case. Because of limited space only the first is presented:

```
Use-Case: CreditCard --> Payment
Documentation: "Use-case enables payment
  with a credit card"
Trigger: Actor wants "to pay with a
  credit card"
Main flow:
  1. "System asks for credit card details"
  2. "Actor delivers credit card details"
  3. "System sends a request to a bank
     for payment and waits
     for bank response"
  4. System verifies that "the payment
     was successful"
  5. The use-case ends with success
Steps 1.-3. can be interrupted
  by cancelling_service event

Flow B: Payment_unsuccessfull
  B1.4. System verifies that
        "the payment was unsuccessful"
  B2. The use-case ends with failure
```

Other examples, together with the language abstract and concrete syntax, are available at [18].

VI. Summary

The concept of a new use-case model specification language (UCFL) consisting of the metamodel, and a textual concrete syntax (T-UCFL) was introduced in the paper. The main purpose of the language is the specification of use-case behaviors. It has commercial origins, as the need for the reusability-oriented approach to use case modeling was recognized during the authors' commercial activities.

The T-UCFL syntax was stabilized on plenty of advanced experiments focused on modeling non-trivial behaviors of some invented software systems. The UCFL metamodel was inferred from these experiments.

Concepts introduced in the paper are designed to extensively support the reusability and avoid redundancy in use-case flows for the whole use-case model. The reusability is achieved at different granulation levels, from a singular step, steps' range to a subflow. Flow initial fragments are reused by definition as they are shared with alternative flows. Inclusion, extension, and generalization between use-cases are also supported.

The language helps to introduce changes into the use-case model. A change made in one place "is visible" in many places referring to the changed element.

The UCFL introduced in the paper seems to be very promising and could be further developed. It is internally consistent, concise, and semi-formal - the specification mimics those written in natural language.

It is worth noting that the paper only introduces a textual concrete syntax. However, other syntaxes may be introduced, especially graphical ones.

In the future, the authors intend to extend the proposed notation with tool support. They also work on graphical concrete syntax and bidirectional transformations between concrete syntaxes. Of course, the usability of the language needs to be validated by external users, first in academia and then in industry.

References

[1] S. Liu, J. Sun, Y. Liu, Y. Zhang, B. Wadhwa, J. Dong, and X. Wang, "Automatic early defects detection use case documents," in *Proc. 29th ACM/IEEE international conference on Automated software engineering*, 2014, pp. 785–790.

[2] S. Adolph, P. Bramble, and A. Pols, *Patterns for Effective UseCases*. Addison-Wesley Professional, 2003.

[3] A. Cockburn, *Writing Effective Use-Cases*. Addison-Wesley, 2000.

[4] G. Overgaard and G. Palmkvist, *Use-cases: Patterns and Blueprints*. Addison-Wesley, 2005.

[5] S. Diev, “Use cases modelling and software estimation: applying use case points,” *ACM SIGSOFT Software Engineering Notes*, vol. 31, no. 6, pp. 1–4, 2006.

[6] M. Śmiałek, J. Bojarski, W. Nowakowski, A. Ambroziewicz, and T. Straszak, “Complementary use case scenario representations based on domain vocabularies,” in *Proc. MODELS’07*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 544–558.

[7] M. Śmiałek, A. Ambroziewicz, and P. R, “Pattern library for use-case-based application logic reuse,” in *Proc. Databases and Information Systems. Communications in Computer and Information Science*, vol. 838. Cham: Springer, 2018, pp. 90–105.

[8] S. Iqbal, I. Al-Azzoni, A. G, and K. HU, “Extending uml use case diagrams to represent non-interactive functional requirements,” *e-Informatica Software Engineering Journal*, vol. 14, no. 1, pp. 97–115, 2020.

[9] S. Mustafiz, J. Kienzle, and H. Vangheluwe, “Model transformation of dependability-focused requirements models,” in *Proc. ICSE Workshop on Modeling in Software Engineering*, 2009, pp. 50–55.

[10] I. Santos, R. Andrade, and P. Santos Neto, “Templates for textual use cases of software product lines: results from a systematic mapping study and a controlled experiment,” *Journal of Software Engineering Research and Development*, vol. 3:5, 2015.

[11] M. Ochodek, K. Koronowski, A. Matysiak, P. Miklosik, and S. Kopczynska, “Sketching use-case scenarios based on use-case goals and patterns,” *Software Engineering: Challenges and Solutions. Advances in Intelligent Systems and Computing*, vol. 504, pp. 17–30, 2017.

[12] D. Rosenberg and S. Kendall, *Applying Use Case Driven Object Modeling with UML: an Annotated e-Commerce Example*, 1st ed. Boston: Addison-Wesley, 2001.

[13] T. Yue, L. Briand, and Y. Labiche, “A systematic review of transformation approaches between user requirements and analysis models,” *Requirements Eng*, vol. 16, pp. 75–99, 2011.

[14] “CaseCompete,” Tech. Rep. [Online]. Available: https://casecomplete.com

[15] “Enterprise architect,” Tech. Rep. [Online]. Available: https://www.sparxsystems.com

[16] J. Thakur and A. Gupta, “Automatic generation of sequence diagram from use case specification,” in *Proc. 7th India Software Engineering Conference. Association for Computing Machinery*, New York, NY, USA, 2014, pp. 1–6.

[17] S. Cook, C. Bock, P. Rivett, T. Rutt, E. Seidewitz, B. Selic, and D. Tolbert, “Unified modeling language (UML) version 2.5.1,” Object Management Group (OMG), Standard, Dec. 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1

[18] B. Hnatkowska and P. Zabawa, “Use-case flow (UCF) case-studies,” Repository, 2023. [Online]. Available: https://github.com/bhnatkowska/UCF

DOI: 10.15439/2023F2125
ISSN 2300-5963 ACSIS, Vol. 35

AI-based Maize and Weeds Detection on the Edge with CornWeed Dataset

Naeem Iqbal*
DFKI
Plan-based Robot Control
Osnabrueck, Germany.
naeem.iqbal@dfki.de

Christoph Manss*
DFKI
Marine Perception
Oldenburg, Germany.
christoph.manss@dfki.de

Christian Scholz†, Daniel König‡, Matthias Igelbrink§, Arno Ruckelshausen¶
Faculty of Engineering and Computer Science
University of Applied Sciences Osnabrueck
Osnabrueck, Germany.
†c.scholz@hs-osnabrueck.de, ‡philipp-daniel.koenig@hs-osnabrueck.de,
§matthias.igelbrink@hs-osnabrueck.de, ¶a.ruckelshausen@hs-osnabrueck.de

***Abstract*—Artificial intelligence (AI) is used more heavily in agricultural applications. Yet, the lack of wireless-fidelity (Wi-Fi) connections on agricultural fields makes AI cloud services unavailable. Consequently, AI models have to be processed directly on the edge. In this paper, we evaluate state-of-the-art detection algorithms for their use in agriculture, in particular plant detection. Thus, this paper presents the *CornWeed* data set, which has been recorded on farm machines, showing labelled maize crops and weeds for plant detection. The paper provides accuracies for the state-of-the-art detection algorithms on the CornWeed data set, as well as frames per second (FPS) metrics for the considered networks on multiple edge devices. Moreover, for the FPS analysis, the detection algorithms are converted to open neural network exchange (ONNX) and TensoRT engine files as they could be used as future standards for model exchange.**

***Index Terms*—plant detection, deep learning, agriculture, maize data, data acquisition, vision transformer**

I. Introduction

When it comes to smart agriculture on farm devices, the evaluation speed of obtained images plays a crucial role [1]. If the processing of the images is too slow, the farm device has to adjust its speed, which results in a lower time efficiency. Object detection algorithms are already capable to provide object recognition at real-time speed. Especially neural networks are utilized for fast object detection, but the performance of a neural network - inference speed and accuracy - is influenced by its structure and size which determines if the network can run on an edge device.

Often the desired detection is marked by a bounding box which surrounds the identified object. Object detectors that use bounding boxes can be categorized into one-stage and two-stage detectors. Two-stage detectors first identify regions of interest using a heuristic and, then, detect the object in this region. One-stage detectors do both tasks in a single network. One-stage detectors are therefore easier to train and are considered to be computationally faster than two-stage detectors [2], [3]. Two-stage detectors generally have a higher accuracy on the location information of the object and they identify smaller objects much better. For one-stage detectors this lower accuracy often originates from poor anchor boxes and the class imbalance problem. Recently, one-stage detectors with an anchor-less approach yielded better accuracy for smaller objects [4]. This is useful for agricultural applications as plants need to be detected in early growth stages and as farm machines might have limited computational power. Also, nowadays an new form of object detectors emerged - transformer networks for object detection [5]. Such networks tend to be large, but they yield high accuracies.

Yet, does it make sense to deploy algorithms directly on farm machines? In [6], the authors discuss the importance of deploying algorithms directly on the farm machines for better responsiveness and reducing the load on cloud computing. On larger farmlands the network connection might be unreliable such that no cloud services are reachable. It might also be possible to use alternative sensors that are already available such as satellite images and drone imagery. These could be preprocessed before the field work. However, satellite imagery can only give guidance for larger patches of land and can not provide insightful information on individual plants due to limited geometric resolution [7]. Even the alternative of using drones prior to field cultivation or the application of herbicides, is not scaling well as presented in [8]. For example, drone imagery is expensive, as it requires additional personnel, and it is most often limited to good weather [9]. Moreover, the collected information can be outdated by a few days or even a week. For weeding applications, these delays can be critical because weeds can be growing fast. Thus, sensor data should be directly processed on the farm machine especially because the capabilities of edge devices are increasing [10].

For example, in [11], the authors present an object detection algorithm for sugar beets that is able to detect the sugar beets and count their leaves based on red, green, and blue (RGB) and

*Both authors contributed equally.

near infra red (NIR) data. The data set is described in [12]. In [13], a robotic platform is presented that utilizes the detector from [11]. This system is able to distinguish weeds from crops such that it can destroy the weeds with a mechanical stamp. As this robot relies on the aforementioned object detector, the system requires RGB and NIR data. However, often only RGB data is available.

In this paper, we empirically evaluate typical object detection networks for their applicability on the edge for the detection of maize and weeds with RGB data. Because such networks require large amounts of data to be trained, we also present a data set that provides box labelled maize and weeds. The networks are then trained from scratch with the presented data set. Our contribution is therefore as follows:

- We present an agricultural dataset, named CornWeed dataset where maize and weeds plants have been labelled for box object detection[*].
- We evaluate object detection algorithms with various neural network architectures based on their detection accuracy (mean average precision (mAP)).
- Each of the detection algorithms is evaluated on farm edge devices based on a Nvidia Jetson Xavier NX and Jetson AGX Orin regarding their real-time capabilities (frames per second).

II. Data Set

A. Hardware Setup and Data Acquisition

For data acquisition, we utilized a previously designed sensor system [14]. This system comprises a computer, power supplies, and sensors. System and sensors communicate via the robot operating system (ROS)[†] such that data can be stored into *ROS Bags* (see Fig.1) which is a ROS specific data format for time-dependent data. The benefit of this system is that it is sensor agnostic, i.e. any sensor can be integrated and connected. Here we used an Intel Realsense D435i (3D stereo camera) and a real time kinematic (RTK) enabled global navigation satellite systems (GNSS) receiver, as presented in Fig. 1. For a robust and consistent data base, data collection was conducted using two different agricultural machines, an implement on a tractor and on a remotely steered research platform BoniRob [15], see Fig. 2. In a first step, we integrated the sensor system into the BoniRob platform (Sensorbox 2) to evaluate optimal camera angles, heights, resolution, light conditions, etc. on a small scale.In a second step, the sensor system (Sensorbox 1) was mounted on a conventional hoe with a shifting frame and pulled through the field trials with a tractor. For this setup, based on the first data acquisition with Sensorbox 2 (640 x 480 pixel), the resolution of the RGB camera on Sensorbox 1 was increased to 1280x720 pixel for a higher quality of the image data. Yet, both resolutions are kept in the data set for variability. In both sensor setups, the Intel Realsense D435i camera with a vertical field of view (FOV) of 69 ° was mounted at a height of 0.5 m, looking downwards. Therefore, the geometric size of the each obtained image spans a distance of 0.68 m along the driving direction.

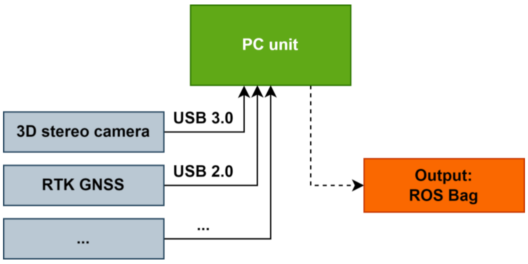


Fig. 1. System perspective of the utilized sensor system.

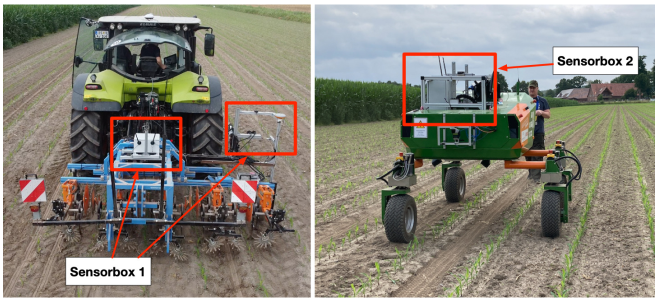


Fig. 2. Platforms for data acquisition. On the left an implementation on the tractor on a conventional hoe with shifting frame (Sensorbox 1). On the right the BoniRob with Sensorbox 2.

B. Data Variability

To represent different stages of growth and weed pressures, we conducted the field trials on multiple days. Therefore, the data samples were recorded over a period of three weeks to ensure different growth stages. Here, the primary focus of the application was to root out the weeds early enough to ensure maximum crop growth. Thus, only the early growth stages of maize crops were considered for the detection application because only at that time crops compete with weeds for resources (water, sunlight, etc.) and otherwise the crops outgrow the weeds. Hence, later growth stages of maize are less relevant for weeding applications. The data set only contains samples in the daylight with cloudy and sunny weather conditions, however, evening and early morning samples in future could be added to extend the domain knowledge for deep neural networks. The field trials always took place on the same field such that the same soil conditions and the same types of weeds persist throughout the data set.

C. Data Labelling

The number of detected weeds instances plays a crucial role for selective weeding. To keep track of the number of detected objects, bounding boxes were chosen as the medium of annotation. The data set contains 3574 outdoor field images of *maize and weeds*, which are also the annotated classes in the data set. An example image of the data set

[*] Dataset Zenodo DOI 10.5281/zenodo.7961764

[†] Open Source Robotics Foundation https://www.ros.org, accessed on 25th of June, 2023

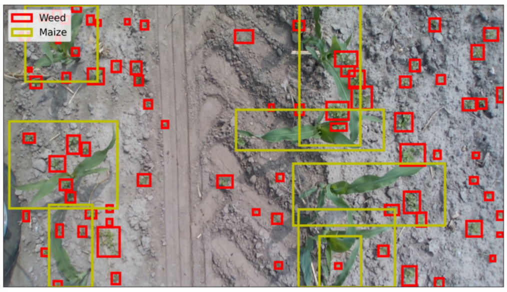


Fig. 3. An example image of the data set taken with the setup on the conventional hoe (Sensorbox 1). The images have resolutions of 640 × 480 pixel with Sensorbox 2 and 1280 × 720 pixel with Sensorbox 1. Here, the labelled instances of Maize are shown in yellow and the weed instances are shown in red.

for Sensorbox 1 with labels is displayed in Fig. 3. The annotations were generated by human annotators and reviewed by a different human reviewer. We used the open-source computer vision annotation tool (CVAT) labelling tool [16] provided by CVAT.ai corporation. The model trained on the data set can be subsequently incorporated into this tool to further reduce the average labelling time. Thus, to speed up the process of annotation, intermediate object detectors have been trained during the annotation process with the interim data to provide proposal annotations. The annotator then fine tuned the proposed annotations by adding not detected weeds and maize, adjusting the class labels of false positives, and changing the sizes of the boxes. Such an interim detector can also be provided by models trained on a synthetic training data as done by Naeem et. al. [17] for a similar use-case.

III. Detection algorithms

For a real-time detection scenario the accuracy is as important as the achievable detection rate. In the considered use-case of selective weeding, the movement speed of the farm device constraints the minimum frames per second (FPS). To cover the whole ground with an average velocity of 8 km/h (2.2 m/s), we require at least 3-4 FPS. Higher frame-rates are of course desirable and would make the system more reliable. Given the low frame-rate requirement, two-stage detectors such as Faster region-based convolutional neural network (R-CNN) [18] can be used also as they have higher accuracy than single stage detectors as shown by Garcia and Mateo et. al. [19]. The authors show that while one-stage detectors are generally faster in inference speeds at lower image resolution, two-stage detectors outperform in terms of accuracy and detecting small objects in the image. This is especially relevant for the considered use-case here, since most of the weeds should be rooted out in the early growth stages, when they are small, before they start competing with the actual crop for resources.

This leads to an accuracy aspect: while weeds are small, detectors might have poor object detection performance. For example, anchor-based approaches [18], [20], [21] have difficulties to find very small objects in the image if the anchor boxes are not small enough. There are, however, object detectors that use an anchor-free approach [4], [22] and these are supposed to have a substantially better performance on small objects. More recently, object detectors based on transformer networks yield high accuracies in multiple applications [5]. Accordingly, for the considered use-case, we chose networks of the aforementioned categories. The networks are introduced in the following subsections.

A. Faster R-CNN

R-CNN is a two-stage detector where the first stage produces region proposals that are then fed into the second stage where the object detection takes place. First versions of R-CNN have been published in 2014 [23] and the following versions have improved to be more computationally effective and more accurate. In this paper, the considered version is the Faster R-CNN [18]. This version uses a convolutional neural network (CNN) as backbone to identify feature maps, which are then sent to a region proposal network and a detection network.

B. RetinaNet

The RetinaNet [21] is a one-stage object detector that is based on the single-shot detector (SSD) [24]. The main idea of SSD is that the detection requires information at different scales. Therefore this network pools directly from multiple convolutional layers, which are referred to as *convolutional predictors for detection*. This network utilizes default boxes and aspect ratios, which have to be determined beforehand. Each default box is then used for prediction on a grid on the image. As the number of predicted boxes can become large, hard negative mining is applied. Due to this only few candidate boxes are considered during the training of an SSD network, which is also known as the foreground-background class imbalance problem [25]. To address this problem, RetinaNet introduces the focal loss to put more emphasis on the hard training examples instead of easy ones. The authors showed that the focal loss substantially improves the performance of one-stage detectors.

C. FCOS

Another one-stage detector that does not use anchor boxes is the fully convolutional one-stage (FCOS) detector [4]. This detector does a pixel-wise detection and computes then a *center-ness* of each pixel according to the ground-truth boxes. The benefits of this are that the intersection over union (IoU), which is computationally expensive, does not need to be computed and that no anchor boxes are required. A downside of this approach is that in the detection ambiguities can occur as one pixel might be the center of multiple boxes. In such cases the larger box is ignored such that the detector has a better accuracy for smaller objects. For the use-case at hand, this is actually good as there are many small weeds.

D. YOLO

The you only look once (YOLO) detector, initially published in [26], has become very popular and has been extended in various aspects. It is a single-stage detector that outputs class probabilities and bounding box coordinates in a single step that are filtered with non-maximum suppression. The YOLOv5 is an efficient implementation [27] in PyTorch, which uses basically the same network as introduced in [28]. In this detector, the authors make excessive use of the so called *bag of freebies* – methods that only change the training strategy or the training cost – and *bag of specials* – methods of plugins that have a good performance to inference cost ratio. The bag of freebies are, for example, data augmentation methods that increase the robustness of the detector. The bag of specials on the other hand are spatial pyramid pooling, a spatial attention module, or other activation functions.

YOLOv5 comes with many variants with different model layers and backbones. For the scope of this paper, we only use YOLOv5 medium and large variants of the YOLO architecture.

E. DINO Transformer

Most of the above object detection models require a prior knowledge of the task in the form of anchors (one-stage) or region proposals (two-stage). The prior knowledge makes the model specialized to a specific task but loses performance when moved to a different detection task making transfer learning difficult. Carion et. al. [29] propose detection transformer (DETR) that is an end-to-end object detection transformer. With this architecture, there is no need to post process the bounding boxes or risk counting the same object twice due to its bipartite matching loss function. Following the DETR architecture, Zhang et. al. [30] came up with an improved variant of DETR called DETR with improved denoising anchor boxes (DINO) transformer. Zhang et. al. proposed the following improvements to the DETR architecture: 1) Adding a noisy version of the ground truth labels during training to speed up the training process. 2) Mixed query selection 3) Box update based on the current layer and the next layer during back propogation. Despite the recent trends in the transformer-based detector architectures, Carion et. al. pointed out the reduced performance of DETR in detecting small objects in the image, while Zhang et. al. argued that the anchor-based approaches still show superior accuracy compared to the transformers. For our experiments, we used the DINO transformer to represent the transformer-based family of object detectors.

IV. Experimental Setting

This section goes through the lifecycle of the neural networks:

1) Training a neural network and selecting the best variant from the *training pipeline*.
2) *Deployment pipeline* which explains how the network is optimized for a particular edge device to maximize performance throughput.
3) Edge devices used to evaluate inference speed of neural networks on an agricultural use-case.

A. Training Pipeline

For all the models mentioned above, official Github repositories already exist [31] [32] [27]. Thus, for the Faster R-CNN, the RetinaNet, and the FCOS, we used the Detectron2 repository from Meta [31]. For each of these models, we set the batch size to 32, the learning rate to 0.01, and the optimizer was stochastic gradient decent. For YOLOv5, we utilized the implementation of Ultralytics [27]. There, we set the batch size to 32 and 16 for YOLOv5 medium and YOLOv5 large variants, respectively. Also, we specified the image size to be 800 pixel and to be rectangular to have comparable results with the other networks. During training, we also used the *multi-scale* option, where the image size is varied during training. For the DINO transformer, we used the Detrex research platform [32], which is based on the Detectron2 repository. Due to the size of the DINO transformer, we had to set the batch size to 4. The learning rate was set to 0.0001. All other parameters of the algorithms were left to the default values. Also, we did not tune the augmentations if any are used by the model.

We trained each model on a NVidia Tesla V100 DGXS 32GB GPU with the CUDA Version 11.1, and used a train-validation split of 80% (2862 images) and 20% (712 images), respectively.

B. Deployment Pipeline

After training, the best model file is selected based on the validation data split and then converted to an open neural network exchange (ONNX) model. ONNX is a framework that optimizes and acts as an intermediate representation of neural networks to support conversion to any standard frameworks such as PyTorch, TensorFlow, OpenVINO, TensorRT, etc. However, a user can also use this intermediate representation directly. In this paper, both the ONNX models and TensorRT models are evaluated to highlight the impact of framework choices. The code repository for testing all the trained models is provided at this link [‡]. The ONNX models of the DINO transformer and YOLO network were then converted to a TensorRT engine file with precision of 16-bit and 32-bit floating points and 8-bit integer. The different precision can make the model more memory efficient and also advanced build-in function can be used for faster computation [33]. The other models consist of the layers that can not be optimized by the TensorRT engine (up until version 8.5[§]), hence failing to do the conversion to a TensorRT engine. At the time of this publication, TensorRT version 8.5 was available.

[‡] Inference speed testing: https://github.com/niqbal996/deployment_testing

[§] We are hopeful that with upcoming TensorRT 8.6, the performance metrics can be updated for remaining networks in Table V-A

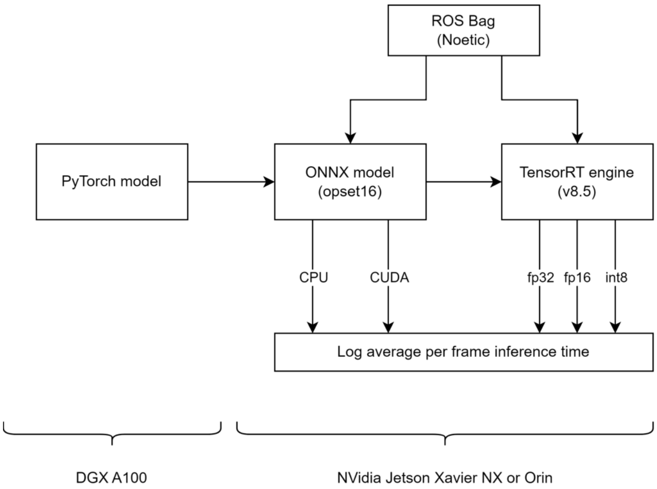


Fig. 4. Deployment workflow showing how the models are trained in PyTorch, converted to open neural network exchange (ONNX) models, and then to TensorRT engine files.

C. Edge Device

For all the experiments in this paper, two NVIDIA Jetson devices were used:

1) the Jetson Xavier NX and
2) the Jetson AGX Orin.

Both edge devices have L4T 35.3.1 with Ubuntu 20.04, *ROS Noetic*, CUDA 11.4 running on them. Jetson devices come with predefined power modes utilizing a varying number of on-board CPUs and online CPU cores. For the experiments shown in Table V-A (see later in Sec. V-B), the Xavier NX was set to mode ID 6 with all cores online and 1400 MHz CPU frequency. In this mode, the Xavier NX board consumes about 20 W of power. Similarly, we activated the MAXN power profile on the AGX Orin to utilize all GPU and CPU cores and to remove clock restrictions. In this power profile, the AGX Orin consumes about 60 W. The input to the model is fed in the form of image sequences via a *ROS Bag*. Each consecutive image sample is read from the *ROS Bag*, downscaled, and normalized according to the input size expected by the model. The performance metrics are logged from the moment an image is received until the detections are ready to be published into the ROS ecosystem as detection messages.

V. Results

This section presents the accuracy results of the presented networks for the data set and the speed results on the considered edge devices.

A. Training Results

TABLE I
Mean average precision (mAP) of all object detectors at various confidence values and class IDs for Maize and weeds

Model	mAP50	mAP50:95	AP50 Maize	AP50 Weeds
Faster R-CNN FPN	71.6	41.8	88.1	55.1
RetinaNet R50 FPN	68.0	40.2	**89.7**	46.3
FCOS R50 FPN	69.3	39.8	87.0	51.7
YOLOv5 medium	**85.4**	**53.3**	**93.0**	**77.8**
YOLOv5 large	**85.4**	**53.9**	**93.7**	**77.0**
DINO Transformer	75.8	45.0	92.3	59.3

Table I shows the mAP with the mean over both classes (namely *Maize* and *Weeds*) at 50 % confidence value i.e. mAP50 in the first column. The second column shows cumulative average of mAP varied from 50 % to 95 % with step size of 5 % i.e. mAP50:95. In the last two columns, we show the average precision (AP) per class at 50 % confidence to show the accuracy trend on larger sized *maize* objects and smaller sized *weed* objects.

Across all networks, the YOLOv5 networks have the highest performance of 85.4 and 53.9 for mAP50 and mAP50:95, respectively. The larger variant YOLOv5 large does not proportionally increase the performance while being much larger in size than YOLOv5 medium. This implies that the accuracy has already saturated and increasing network size does not always necessarily lead to increased performance. In terms of comparison, the YOLOv5 repository already offers many variations and augmentations to the input data, see bag freebies in Sec. III-D. This bag of freebies is also optimized during training by the code in the repository. The networks based on Detectron2 do not have such an optimization. Thus, if only the networks based on the Detectron2 repository are compared with each other, the DINO transformer stands out in detecting maize but still gives a lower accuracy of 59.3 for AP50 for weeds compared with YOLOv5. This is due to its reduced capability to detect tiny objects such as weeds which can be in some instances only be a few pixels in size. This behaviour was first brought to light by Carion et. al. [29] for the detection transformers.

If the one-stage networks of the Detectron2 repository are compared with the two-stage detector Faster R-CNN, the RetinaNet reached a better performance of 89.7 on detecting maize but surprisingly a lower AP of 46.3 on weeds. This higher accuracy of 89.3 than 88.1 from Faster R-CNN is due to the usage of Focal loss proposed by [21]. While focal loss helps in outperforming a two-stage detector such as Faster R-CNN, the AP for weeds is still lower. We believe the lower accuracy of 46.3 on weeds to be an outlier due to anchor boxes not modified to the task of weed detection prior to training. Comparing RetinaNet with YOLOv5, the Detectron2 framework does not provide any flexibility to change the anchor boxes to the task of maize or weed detection prior to training. This leads to reduced performance. Ahmed et. al. [34] observed similar behaviour from RetinaNet in detecting tiny objects from aerial images and used anchor optimization to mitigate that.

To achieve the best possible accuracy from an anchor-based detector architecture, the anchor boxes have to be modified to the specific use-case. Task specific modification of anchor

TABLE II

AVERAGE INFERENCE RATE (IN FRAMES PER SECOND (FPS)) FOR ALL THE DETECTORS FROM TABLE I ON EDGE DEVICES WITH ONNX *CUDA Execution Provider* (CEP) AND TENSORRT. THE INPUT IMAGE RESOLUTION WAS 800 X 1067 PIXEL FOR THE THREE NETWORKS (FCOS, RETINANET, FASTER R-CNN), 800 X 1088 PIXEL FOR BOTH YOLO NETWORKS, AND 512 X 683 PIXEL FOR THE DINO TRANSFORMER. THE VALUES MARKED WITH A * WERE DUE TO A TENSORRT BUG WHERE THE DINO MODEL NOT UTILIZED CUDA.

Framework	**ONNX with *CEP***		**TensorRT**					
Edge device	**Jetson Xavier NX**	**Jetson AGX Orin**	**Jetson Xavier NX**			**Jetson AGX Orin**		
			int8	**fp16**	**fp32**	**int8**	**fp16**	**fp32**
Faster R-CNN FPN	0.62	1.8	X	X	X	X	X	X
RetinaNet R50 FPN	1.37	4.3	X	X	X	X	X	X
FCOS R50 FPN	1.30	4.0	X	X	X	X	X	X
YOLOv5 medium	**2.5**	**6.0**	**19.3**	**12**	**3.7**	**51**	**33.5**	**16.8**
YOLOv5 large	1.4	4.0	12.7	6.5	1.65	37	22	10.9
DINO Transformer	0.24	0.55	1.35*	1.47*	0.86*	3.2*	3.1*	3*

boxes can be crucial in choosing optimum network architecture for any agricultural use-case where each crop has different size in different growth stages. DINO Transformer, FCOS and other anchor-less implementations reduce the complexity of the task by not having to provide any prior knowledge or anchor optimization before training while giving slightly lower accuracy in general.

B. Speed Results

As described in section IV-B, the trained models are optimized and then deployed for both NVIDIA Jetson Xavier NX and Jetson AGX Orin. The average FPS were logged based on the input image which was fed from a ROS bag recorded with Intel RealSense D435i camera for testing purposes. For anchor-based implementations (e.g. RetinaNet, YOLO), the non-maximum suppression is also part of the inference time. This creates a fair comparison between anchor-based and anchor-free models. The corresponding FPS metrics are shown in Table V-A. For the first three models (Faster R-CNN, RetinaNet, and FCOS), the model contains layers that cannot be converted into a TensorRT engine with version 8.5. With the next release of TensorRT v8.6 accompanied with the new Jetpack release, these models should also be convertible to a TensorRT engine. Comparatively looking at the numbers, in general, the ONNX inference framework gives lower inference rates than TensorRT, even when using *CUDA execution provider*. However, the ONNX model serves as a good intermediate model representation that can be later converted to any other inference framework such as TensorFlow, TensorRT, or PyTorch.

The TensorRT engine files with different floating point and integer precisions yield higher FPS in general, see Table V-A. For example, YOLOv5 medium yields 33.5 FPS on fp16 precision via TensorRT compared to a mere 6 FPS via ONNX on Jetson Orin. Especially, the 8-bit integer precision yields the highest inference rate for each network, about 6-9 times faster compared with the ONNX models. When comparing different networks, YOLOv5 gives the highest FPS.

Though the model conversion from PyTorch or Tensorflow to ONNX or TensorRT comes with its own challenges. Depending upon the layers that constitute a particular model, not all layer components are easily convertible to an TensorRT engine. While some architectures are easily transferrable to a TensorRT engine, other model architectures while giving more accuracy may be more difficult in transferring to a TensorRT engine. This makes the model deployment to an edge device more difficult and impacts the choice of model architecture.

When comparing the accuracy (see Table I) versus the inference rate (see Table V-A), the YOLOv5 medium version has the same accuracy as the YOLOv5 large variant but has more FPS e.g., 33.5 versus 22 FPS on Jetson Orin (fp16). This implies that increasing the model size does not necessarily lead to improved performance while the medium variant gives more inference speed on the edge device. On the other hand, the DINO transformer provides high mAP50 on the maize class, but surprisingly lower values on weed class[¶]. While vision transformers usually outperform most networks, they may not always give the best performance depending on the size of the objects. The GPU memory size and the input image resolution also plays a critical role in deciding for the neural network architecture.

VI. CONCLUSION

This paper shows how multiple one-stage object detectors (RetinaNet, FCOS, YOLO), a two-stage object detector (Faster R-CNN), and a transformer object detector (DINO) perform on an agricultural use-case on different edge systems. With an accompanying data set, we found out that the YOLOv5 object detector performed well on detecting maize and reasonably well on detecting tiny objects such as small weeds. For the other neural networks, characteristics are highlighted such as a low accuracy on detecting tiny objects which is a common challenge in agricultural perception tasks. These shortcomings are not immediately visible when training on a different domain data set such as autonomous driving. The model deployment pipeline is also included for readers who embark on a different use-case and want to optimize their model's inference speed. Generally speaking, it is always

[¶] The DINO transformer was set to lower resolution of 512 x 683 because otherwise it does not fit onto Jetson Xavier NX with 8GB VRAM. For a fair comparison between Xavier NX and AGX Orin, it was set to that resolution.

better to convert a well trained model to an edge device specific engine such as the TensorRT engine. This way the FPS can increase by factors up to 8, if additionally the integer precision is reduced. Thus, using TensorRT yields faster networks, not only in the agricultural domain but also other domains, e.g. [35]. Inference framework-wise, ONNX has a relatively simplified model conversion process that works with most of the models and transferable across edge devices with different GPU architecture. By comparison, TensorRT has a more complicated GPU-specific model conversion process e.g. Jetson Orin needs a separate engine than Jetson Xavier and does not work successfully on all models.

VII. Acknowledgements

The DFKI Lower Saxony (DFKI NI) is funded by the Lower Saxony Ministry of Science and Culture and the Volkswagen Foundation. The project Agri-GAIA on which this report is based was funded by the German Federal Ministry for Economics and Climate Action under the funding code 01MK21004A: Responsibility for the content of this publication lies with the author. Many thanks to Hof Langsenkamp, Belm, Germany for providing the fields for data acquisition and the used hardware (tractor and implement). We would like to thank Oliver Zielinski for mainly writing on the parts of the proposal that lead to this work. We would also like to thank our labellers: Qalab Abbas, Jule Fröhlich, Novruz Mammadli, Charles Lennart Müller, Dibyashree Nahak, Turgut Nasrullayev, and Simon Zielinski.

VIII. Acronyms

CNN	convolutional neural network
R-CNN	region-based convolutional neural network
Wi-Fi	wireless-fidelity
SSD	single-shot detector
YOLO	you only look once
FCOS	fully convolutional one-stage
FPS	frames per second
FOV	field of view
IoU	intersection over union
mAP	mean average precision
ONNX	open neural network exchange
AI	artificial intelligence
DETR	detection transformer
DINO	DETR with improved denoising anchor boxes
CVAT	computer vision annotation tool
RTK	real time kinematic
GNSS	global navigation satellite systems
NIR	near infra red
RGB	red, green, and blue
ROS	robot operating system
AP	average precision

References

[1] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, "Machine Learning in Agriculture: A Comprehensive Updated Review," *Sensors*, vol. 21, no. 11, p. 3758, Jan. 2021, DOI:10.3390/s21113758.

[2] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A Survey of Deep Learning-based Object Detection," *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[3] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep Learning for Generic Object Detection: A Survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[4] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9626–9635.

[5] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[6] X. Zhang, Z. Cao, and W. Dong, "Overview of Edge Computing in the Agricultural Internet of Things: Key Technologies, Applications, Challenges," *IEEE Access*, vol. 8, pp. 141 748–141 761, 2020, DOI:10.1109/ACCESS.2020.3013005.

[7] M. Weiss, F. Jacob, and G. Duveiller, "Remote sensing for agricultural applications: A meta-review," *Remote Sensing of Environment*, vol. 236, p. 111402, 2020, DOI:10.1016/j.rse.2019.111402. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425719304213

[8] E. Cai, S. Baireddy, C. Yang, M. Crawford, and E. J. Delp, "Deep transfer learning for plant center localization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 277–284, DOI:10.1109/CVPRW50498.2020.00039.

[9] F. López-Granados, "Weed detection for site-specific weed management: mapping and real-time approaches," *Weed Research*, vol. 51, no. 1, pp. 1–11, 2011, DOI:10.1111/j.1365-3180.2010.00829.x.

[10] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, Aug. 2019, DOI:10.1109/JPROC.2019.2918951.

[11] J. Weyler, A. Milioto, T. Falck, J. Behley, and C. Stachniss, "Joint Plant Instance Detection and Leaf Count Estimation for In-Field Plant Phenotyping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3599–3606, Apr. 2021, DOI:10.1109/LRA.2021.3060712.

[12] N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss, "Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1045–1052, Sep. 2017.

[13] X. Wu, S. Aravecchia, P. Lottes, C. Stachniss, and C. Pradalier, "Robotic weed control using automated weed and crop classification," *Journal of Field Robotics*, vol. 37, no. 2, pp. 322–340, 2020, DOI:10.1002/rob.21938.

[14] D. König, M. Igelbrink, C. Scholz, A. Linz, and A. Ruckelshausen, "Entwicklung einer flexiblen Sensorapplikation zur Erzeugung von validen Daten für KI-Algorithmen in landwirtschaftlichen Feldversuchen," in *42. GIL-Jahrestagung, Künstliche Intelligenz in der Agrar- und Ernährungswirtschaft*. Bonn: Gesellschaft für Informatik in der Land-, Forst- und Ernährungswirtschaft e.V., 2022, pp. 165–170.

[15] W. Bangert, A. Kielhorn, F. Rahe, A. Albert, P. Biber, S. Grzonka, S. Haug, A. Michaels, D. Mentrup, M. Hänsel *et al.*, "Field-robot-based agriculture:"remotefarming. 1" and "bonirob-apps"," in *71th conference LAND. TECHNIK-AgEng 2013*, 2013, pp. 439–446.

[16] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin, B. Hoff, TOsmanov, D. Kruchinin, A. Zankevich, DmitriySidnev, M. Markelov, Johannes222, M. Chenuet, a andre, telenachos, A. Melnikov, J. Kim, L. Ilouz, N. Glazov, Priya4607, R. Tehrani, S. Jeong, V. Skubriev, S. Yonekura, vugia truong, zliang7, lizhming, and T. Truong, "opencv/cvat: v1.1.0," Aug. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4009388

[17] N. Iqbal, J. Bracke, A. Elmiger, H. Hameed, and K. von Szadkowski, "Evaluating synthetic vs. real data generation for ai-based selective weeding," in *43. GIL-Jahrestagung, Resiliente Agri-Food-Systeme*, C. Hoffmann, A. Stein, A. Ruckelshausen, H. Müller, T. Steckel, and H. Floto, Eds. Bonn: Gesellschaft für Informatik e.V., 2023, pp. 125–135.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *arXiv:1506.01497 [cs]*, Jan. 2016, DOI:10.48550/arXiv.1506.01497.

[19] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data," *Remote Sensing*, vol. 13, no. 1, p. 89, Dec. 2020, DOI:10.3390/rs13010089.

[20] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767 [cs]*, Apr. 2018, DOI:10.48550/arXiv.1804.02767.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020, DOI:10.1109/TPAMI.2018.2858826.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," Aug. 2021, DOI: 10.48550/arXiv.2107.08430.

[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587, DOI: 10.1109/CVPR.2014.81.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37, DOI:10.1007/978-3-319-46448-0_2.

[25] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance Problems in Object Detection: A Review," *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1–1, 2020, DOI:10.1109/TPAMI.2020.2981890.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788, DOI:10.1109/CVPR.2016.91.

[27] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934 [cs, eess]*, Apr. 2020.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229, DOI:10.48550/arXiv.2005.12872.

[30] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to-end object detection," *arXiv preprint arXiv:2203.03605*, 2022, DOI:10.48550/arXiv.2203.03605.

[31] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[32] detrex contributors, "detrex: An research platform for transformer-based object detection algorithms," https://github.com/IDEA-Research/detrex, 2022.

[33] S. Markidis, S. Chien, E. Laure, I. Peng, and J. S. Vetter, "Nvidia tensor core programmability, performance & precision," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2018, pp. 522–531. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/IPDPSW.2018.00091

[34] M. Ahmad, M. Abdullah, and D. Han, "Small object detection in aerial imagery using retinanet with anchor optimization," in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, 2020, pp. 1–3, DOI:10.1109/ICEIC49074.2020.9051269.

[35] M. Wolf, K. van den Berg, S. P. Garaba, N. Gnann, K. Sattler, F. Stahl, and O. Zielinski, "Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)," *Environmental Research Letters*, vol. 15, no. 11, p. 114042, Nov. 2020, DOI:10.1088/1748-9326/abbd01.

DOI: 10.15439/2023F1609
ISSN 2300-5963 ACSIS, Vol. 35

BIGOS - Benchmark Intended Grouping of Open Speech Corpora for Polish Automatic Speech Recognition

Michał Junczyk
Adam Mickiewicz University
email: michal.junczyk@amu.edu.pl

Abstract—**This paper presents a Benchmark Intended Grouping of Open Speech (BIGOS), a new corpus designed for Polish Automatic Speech Recognition (ASR) systems. This initial version of the benchmark leverages 1,900 audio recordings from 71 distinct speakers, sourced from 10 publicly available speech corpora. Three proprietary ASR systems and five open-source ASR systems were evaluated on a diverse set of recordings and the corresponding original transcriptions. Interestingly, it was found that the performance of the latest open-source models is on par with that of more established commercial services. Furthermore, a significant influence of the model size on system accuracy was observed, as well as a decrease in scenarios involving highly specialized or spontaneous speech. The challenges of using public datasets for ASR evaluation purposes and the limitations based on this inaugural benchmark are critically discussed, along with recommendations for future research. BIGOS corpus and associated tools that facilitate replication and customization of the benchmark are made publicly available.**

I. Introduction

AUTOMATIC Speech Recognition (ASR) is used in various applications and usage scenarios. Given that multiple aspects impact the difficulty of ASR tasks (vocabulary, acoustic conditions, speech type, etc.), the quality of target systems heavily depends on the effectiveness of the evaluation process. Benchmarking and evaluation ultimately aim to validate the system's ability to adapt to novel and unseen data.[1] To achieve this, multiple evaluation methods, datasets, and metrics are needed. The most commonly used metric for ASR evaluation is the Word Error Rate (WER), which quantifies word-level insertions, deletions, and substitutions between a system and reference transcriptions. WER has known limitations [2, 3]. When used on a narrow set of evaluation data, the assessment of the capabilities of the models, particularly in terms of generalization to unseen data, may be unclear.

Unlike English [4, 5, 6], German [7] and recently Hungarian [8], the Polish language lacks a common public-domain reference dataset for ASR benchmarking. Consequently, the results of Polish speech recognition studies are generally not directly comparable. Although transcribed recordings are available, it is often not practical to find or use all available public-domain datasets.

This study introduces BIGOS, a resource intended to enable systematic benchmarking and tracking of Polish ASR systems over time across a diverse range of publicly available corpora. The primary purpose of BIGOS is to alleviate the painstaking efforts required to discover and compile speech corpora from multiple sources. To ensure that original licenses are respected by BIGOS users, the corpus is distributed on the Hugging Face platform[1], which allows gated access. Alternatively, scripts for self-curation and customization of the dataset are also provided. [2] The first iteration of the benchmark presented in this work is performed using 1,900 utterances sourced from 10 corpora and 3 commercial ASR systems and 5 freely available.

The remainder of this paper is structured as follows: Section 2 reviews the relevant literature, and Section 3 outlines the construction of the BIGOS benchmark and dataset, detailing the source speech corpora, corpus statistics, and ASR systems evaluated. Section 4 presents an exemplary application of BIGOS for the evaluation of ASR systems, Section 5 describes the limitations, and Section 6 concludes the paper by outlining the directions for future research.

II. Related work

A. *ASR evaluation datasets*

Prominent English-only datasets for ASR research and evaluation include the Wall Street Journal, VoxForge, Fisher, CHiME, LibriSpeech, TED-Lium, Common Voice, and Earnings. Wall Street Journal corpus covers news broadcast recordings, while SwitchBoard and Fisher include spontaneous telephone conversations. LibriSpeech [9] and MLS [10] feature narrated audiobooks, while VoxForge includes narrated Wikipedia articles. The TED-LIUM corpus [11] contains oratory educational talks, while the CHiME [12] dataset represents recordings of noisy environments in the real world. Earnings-21 and Earnings-22 contain conversational speech from earnings call recordings [4, 5]. The most voluminous dataset in terms of both the duration of speech content and language coverage is the MLS (Multilingual Librispeech), which contains 41,000 hours of material [10] for 8 languages. The Mozilla Common Voice dataset covers speech for more than 55 languages and boasts the largest number of contributing speakers, with over 10,000 as of March 2023 [13]. Both Common Voice and MLS include Polish language data. All of the aforementioned datasets offer a diverse range of speech sources, speaker demographics, and speech types,

[1] https://huggingface.co/datasets/michaljunczyk/pl-asr-bigos
[2] https://github.com/goodmike31/pl-asr-bigos-tools

providing researchers with valuable resources to investigate various aspects of ASR and to train new systems.

B. ASR benchmarks

The idea of using available speech datasets to benchmark the quality of ASR systems was first implemented nearly a decade ago. Gaida et al. [14] were the first to conduct a comprehensive evaluation of several open-source speech recognition tools. Dernoncourt developed a framework to evaluate seven ASR systems in two different collections and provided scripts to format Common Voice and LibriSpeech.[3] Moore et al. [15] introduced a meta-dataset containing reference text, hypotheses from two separate ASR systems, the Word Error Rate (WER), and annotations about speech intelligibility. Ulasik created a multilingual CEASR dataset for English and German[7], based on reference transcriptions from popular public-domain datasets and transcripts from four undisclosed ASR systems. Siegert et al. [16] performed a longitudinal study and found no significant changes in WER for 4 commercial systems over 8 months. Aksenova et al. [1] conducted a comprehensive survey on existing ASR benchmarking methodologies and proposed a systematic benchmarking framework for the most common use cases. Xu et al.[17] compared 4 commercial ASR services with respect to robustness to acoustic background noise. Varod et al. highlighted that ASR performance is language and system specific and that low-resource languages such as Hebrew can have a performance comparable to high-resource languages such as German.[18] The ASR4REAL benchmark [19] revealed significant accuracy variations depending on the accent of the speaker and socioeconomic status. Papadopoulou evaluated four commercial ASR systems in the context of translation post-editing effort [20]. The challenges associated with the recognition of spontaneous and accented speech were further analyzed in the benchmarks organized by the Rev and Google companies. [4, 5, 21]. Pasandi et al. highlighted that conversational speech is the most challenging and environmentally relevant type of data for speech recognition. Pires et al. constructed the Portuguese Evaluation Benchmark[22] using the Mozilla Common Voice and Voxforge datasets and five commercial ASR engines. Mihajlik et al. conducted an evaluation of open-source Hungarian ASR systems using a comprehensive linguistic dataset [8]. Extending the studies by Ulasik et al. for English and German, Wirth et al. [3] questioned the prevailing statistical ASR evaluation paradigm by performing a manual recognition error assessment. Of paramount importance, the study identified that 18% of the ASR errors originated from flawed ground-truth transcriptions and another 18% from flawed or ambiguous audio within publicly accessible datasets.

C. Polish ASR benchmarks

The first evaluation of commercial ASR systems for the Polish language was carried out in 2018 [23]. The first open benchmark for ASR systems was organized by Korzinek [24]. In 2019, Unai et al. [25] evaluated a self-developed Polish ASR system using 223 hours of speech collected from six datasets, including the Clarin-PL Studio Corpus (EMU)[26], the PELCRA family of corpora [27, 28], the Polish Senate recordings corpus [29], the Simple4All Tundra Corpus, and the test results for the PolEval 2019 competition [24]. The most extensive benchmark to date is Diabiz *Diabiz* performed using a set of 400 dialogs in eight domains and three commercial ASR systems. [30, 31].

III. BIGOS CORPUS DESIGN AND CURATION

As indicated by the Polish ASR Speech Data catalog [4] as of March 2023, approximately 5300 hours of speech in 51 datasets are available for Polish ASR development. Roughly 1000 hours of transcribed speech spread across 13 datasets is freely accessible under permissive licenses, facilitating the curation of a new evaluation dataset detailed in the following section.

A. BIGOS corpus overview

Table III-A summarizes the properties of the BIGOS dataset.

Table I
BIGOS DATASET PROPERTIES

Attribute	Value
Datasets sourced	10
Speech material (hours)	4.5
Test cases total	1900
Speakers	71

B. Sourcing and pre-analysis

Polish ASR Speech Data Catalog was used to identify suitable datasets to be included in the benchmark. The following mandatory criteria were considered:

- Dataset must be downloadable.
- The license must allow for free, noncommercial use.
- Transcriptions must be available and align with the recordings.
- The sampling rate of audio recordings must be at least 8 kHz.
- Audio encoding using a minimum of 16 bits per sample.

The following is an overview of 10 datasets that meet the criteria and were chosen as sources for the BIGOS dataset.

- The Common Voice dataset *(mozilla-common-voice-19)*, developed by Mozilla, is an open source multilingual resource [13]. This project aims to democratize voice technology by providing a wide-ranging, freely available dataset that covers many languages and accents. Contributors from around the globe donate their voices, reading out pre-defined sentences or validating the accuracy of other contributions. Common Voice is recognized as the most comprehensive and diverse voice dataset available, spanning more than 60 languages and representing many underrepresented groups. Datasets are released every

[3] https://github.com/Franck-Dernoncourt

[4] https://github.com/goodmike31/pl-asr-speech-data-survey

three months under a permissive Creative Commons 0 license.

- The Multilingual LibriSpeech (MLS) dataset *(fair-mls-20)* is a large, multilingual corpus created for speech research by Facebook AI Research (FAIR)[10]. This dataset is derived from audiobooks from LibriVox and covers eight languages, including about 44,000 hours of English and a total of around 6,000 hours for other languages. The Polish speech data include 137 hours of read speech from 25 books, recorded by 16 speakers. Humans have evaluated the transcriptions in the test sets.
- The Clarin Studio Corpus *(clarin-pjatk-studio-15)* is provided by CLARIN-PL, a subsection of CLARIN devoted to the Polish language. This corpus includes 13,802 short utterances, which add up to about 56 hours, spread over 554 audio sessions by 317 speakers. Each session contains between 20 and 31 audio files. All utterances were recorded in a studio, guaranteeing clear audio files free from background noise and other environmental factors.
- The Clarin Mobile Corpus *(clarin-pjatk-mobile-15)* is a Polish speech corpus of read speech recorded over the phone. It includes many speakers, each reading several dozen different sentences, and a list of words containing rare phonemes. It is designed for the analysis of modern Polish pronunciation in a telephony environment.
- The Jerzy Sas PWR datasets (Politechnika Wrocławska) *(pwr-viu-unk, pwr-shortwords-unk, pwr-maleset-unk).* According to the documentation available online[5] speech was collected using a variety of microphones and in relatively noise-free acoustic conditions. Three datasets are available: short words, very important utterance (VIU), and male AM set.
- The M-AI Labs Speech corpus *(mailabs-19)*, similar to the MLS corpus, was created from LibriVox audiobooks. This corpus covers nine languages and was created by the European company M AI Labs with the mission of *"enabling (European) companies to take advantage of AI & ML without having to give up control or know-how."* [6] The M-AILABS Speech Dataset is provided free of charge and is intended to be used as training data for speech recognition and speech synthesis. The training data consists of nearly a thousand hours of audio for all languages, including 53.5 hours for Polish.
- The AZON Read and Spontaneous Speech Corpora [7] *(pwr-azon-spont-20, pwr-azon-read-20)* is a collection of recordings of academic staff, mainly in the physical chemistry domain. The corpus is divided into two parts: supervised, where the speaker reads the provided text, and unsupervised spontaneous recordings, such as live-recorded interviews and conference presentations by scientific staff. The dataset contains recordings of 27 and 23 speakers, totaling 5 and 2 hours of transcribed speech, respectively. The AZON database is available under a CC-BY-SA license.

Two additional corpora, the Spelling and Numbers Voice database (SNUV) from the University of Łódź's PELCRA group and the CLARIN Cyfry corpus, initially met the necessary requirements for this study. However, their unique transcription conventions led to high error rates during initial tests. For example, the word *"pstrąg"* in *SNUV* corpus is transcribed as *"py sy ty ry ą gy"*. The conventional normalization employed by most ASR systems is *"p s t r ą g"*. In the case of *Cyfry* corpus, only numeric expressions are transcribed, hence high error rates are produced for correctly recognized nonnumeric expressions. As such, these corpora will be included in the next iteration of the benchmark, following a thorough manual retranscription process to mitigate these issues.

C. Curation and selection

Necessary preprocessing parameters were consolidated into specific configuration files for each dataset, including download links, metadata fields to be extracted, etc. Subsequently, the text data and audio were extracted and encoded in a unified format. Dataset-specific transcription norms are preserved, including punctuation and casing. To strike a balance in the evaluation dataset and to facilitate the comparison of Word Error Rate (WER) scores across multiple datasets, 200 samples are randomly selected from each corpus. The only exception is 'pwr-azon-spont-20', which contains significantly longer recordings and utterances, therefore only 100 samples are selected. Finally, the first version of the BIGOS corpus contains 1900 recordings of the 115,915 available in the 10 datasets (1. 64% of the total available transcribed speech). The table II provides detailed information on the composition of the BIGOS 1.0 corpus.

Table II
NUMBER OF RECORDINGS AND AVERAGE DURATIONS

Dataset	Size[h]	Files	Average length[s]
fair-mls-20	0.81	200	14.52 ±2.82
clarin-pjatk-mobile-15	0.72	200	13.05 ±3.51
pwr-azon-spont-20	0.72	100	25.75 ±7.12
clarin-pjatk-studio-15	0.56	200	10.10 ±4.32
pwr-azon-read-20	0.48	200	8.72 ±1.95
mailabs-19	0.42	200	7.60 ±3.10
mozilla-common-voice-19	0.27	200	4.89 ±1.50
pwr-shortwords-unk	0.24	200	4.41 ±1.42
pwr-maleset-unk	0.19	200	3.44 ±0.44
pwr-viu-unk	0.08	200	1.49 ±0.22
Total	4.49	1900	-
Average	-	-	9.39 ±2.64

D. Preprocessing and format standardization

The following curation methods were applied to the baseline version of the BIGOS dataset:

- validation of audio file availability and validity,
- unification of audio format to WAV 16 bits/16 kHz,
- normalization of audio amplitude to -3 dBFS,
- unification of text encoding to UTF8,

[5] https://www.ii.pwr.edu.pl/ sas/ASR/

[6] https://www.caito.de/2019/01/03/the-m-ailabs-speech-dataset/

[7] https://zasobynauki.pl/zasoby/korpus-nagran-probek-mowy-do-celow-budowy-modeli-akustycznych-dla-automatycznego-rozpoznawania-mowy,53293/

Table III
ATTRIBUTES IN THE BIGOS UTTERANCE DATA OBJECT

Attribute	Description
id_file_pproc	Standardized file identifier
id_file_source	Original file identifier
id_dataset_source	Source dataset identifier
subset_source	Subset in source dataset (train, test, valid)
path_audio_source	Path to original audio file
path_trans_source	Path to original transcription file
path_audio_pproc	Path to audio file after standardization
meta_spkid_source	Original speaker identifier
meta_spkid_pproc	Standardized speaker identifier
meta_spk_age_source	Speaker age info from source
ref_original	Original transcription (reference)
hyp_whisper_cloud	Hypothesis of Whisper cloud service
hyp_google_default	Hypothesis of Google cloud service default model
hyp_azure_default	Hypothesis of Azure cloud service default model
hyp_whisper_tiny	Hypothesis of Whisper local tiny model
hyp_whisper_base	Hypothesis of Whisper local base model
hyp_whisper_small	Hypothesis of Whisper local small model
hyp_whisper_medium	Hypothesis of Whisper local medium model
hyp_whisper_large	Hypothesis of Whisper local large model

- extraction of original transcription,
- removal of redundant characters
- extraction and unification of metadata.

E. Validation and ASR transcripts generation

Upon completing the preprocessing of the entire dataset, the number of obtained recordings, transcriptions, and metadata records in the compiled dataset were checked for consistency. If the validation was successful, the ASR hypotheses for the locally hosted Whisper models were generated. ASR transcriptions for cloud services like Google, Azure, and Whisper were obtained via respective APIs. Table III presents the object of the resulting BIGOS utterance data.

IV. ASR SYSTEMS EVALUATION

A. Evaluated ASR systems

Below is an overview of the ASR systems evaluated in the first iteration of the BIGOS benchmark.

- Google Cloud Speech-to-Text [8] supports more than 125 languages and variants. The "default" model from May 2023 was used for this benchmark.
- Microsoft's Azure Speech Service [9] as of May 2023 supports more than 100 languages and variants. The "default" model from May 2023 was used for this benchmark.
- Whisper is an ASR system developed by the OpenAI company. It is trained on a large amount of weakly supervised multilingual and multitask data collected from the Internet [32]. The web-hosted model available via API and the locally hosted models from May 2023 were used for this benchmark.[10]

B. Metrics

ASR systems predictions were evaluated against the target transcriptions using 3 industry-standard metrics:

- Sentence Error Rate (SER) calculates the proportion of sentences that are not perfectly recognized, i.e., sentences that contain at least one error.
- Word Error Rate (WER) is defined as the minimum number of operations (substitutions, insertions, and deletions) required to transform the system output into the reference transcript, divided by the total number of words in the reference.
- Character Error Rate (CER) metric calculates the minimum number of character-level operations (substitutions, insertions, and deletions) needed to change the system's output into the reference transcript, divided by the total number of characters in the reference.

V. BENCHMARK RESULTS

This section provides an overview and analysis of the results obtained.

A. Quality per system and model type

The performance of various systems was evaluated using average SER, WER, and CER values obtained from ten test datasets available in BIGOS. The "large" model of the Whisper system achieved the highest accuracy, outperforming all other systems in every metric. The "medium" model of the Whisper system came second, and the "cloud" model of the same system came third. Google and Azure's services followed these, with the remaining Whisper models trailing behind.

Interestingly, the two most accurate systems are both freely available. Despite using the same "large-v2" model, the cloud-based variant was outperformed by the locally hosted "large" variant and, even more surprisingly, by the "medium" variant, which theoretically should be less advanced. On average, free systems outperformed well-established paid services.

To understand why this is the case, a more detailed and manual examination of the evaluation results is required. However, it is crucial to note that lower scores in this evaluation do not necessarily indicate inferior performance in real-world scenarios.

One hypothesis is that commercial systems, despite their ability to handle advanced normalization conventions, might actually perform worse when evaluated on publicly available datasets that use written forms of numerals (e.g., "one", "six o'clock") instead of numeric forms (e.g., "1", "6:00"). This paradox suggests that the use of automated evaluation metrics and publicly available datasets used "as-is" (without transcription unification) may not fully represent real-world performance and capabilities.

Tables IV present the average SER, WER, and CER scores for the Azure, Google, and Whisper systems.

B. Quality per dataset

The best overall performance was observed for the PWR corpora, which contain recordings from a single speaker in

[8] https://cloud.google.com/speech-to-text

[9] https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text

[10] https://github.com/openai/whisper/blob/main/model-card.md

Table IV
AVERAGE SER, WER AND CER OF EVALUATED POLISH ASR SYSTEMS

Service	System	Variant	SER	WER	CER
paid	Azure	default	64.3 ±39.2	18.3 ±12.6	10.2 ±9.3
paid	Google	default	59.9 ±38.5	16.1 ±10.0	7.4 ±5.7
paid	Whisper	large	58.7 ±34.6	11.0 ±8.0	4.1 ±3.9
free	Whisper	tiny	90.3 ±14.6	46.8 ±9.4	14.7 ±6.1
free	Whisper	base	83.7 ±19.9	32.9 ±9.9	10.1 ±4.7
free	Whisper	small	67.6 ±28.2	16.5 ±6.5	5.5 ±3.0
free	Whisper	medium	55.4 ±34.1	9.3 ±4.6	3.7 ±2.6
free	**Whisper**	**large**	**50.1 ±34.4**	**7.6 ±3.8**	**3.1 ±2.4**

a quiet acoustic environment. This limited variability led to perfect performance for the Whisper Cloud and Azure systems in *PWR VIUa* and the best average WER for the *Male* set. Interestingly, for single-word utterances, the limited context led the Google and Whisper local systems to recognize foreign language words instead of Polish words. For example, the word 'zapisz' was recognized as a Russian word 'Запись', the word 'zakończ' as the English word 'the coins', and words 'małe litery' and 'duże litery' as the Italian words 'ma vedi tere' and 'due lettere', respectively. The PWR male set dataset had the second-best performance. A median WER of 6% suggests that modern Polish ASR systems handle short utterances from contemporary literature quite effectively.

Slightly worse performance (average WER over 10% for all systems) was observed for the *MLS, M-AI Labs*, and *Common Voice* datasets. Given the widespread use and accessibility of the *MLS* and *Common Voice* datasets within the global ASR community, it is likely that these datasets were used during training, allowing all systems to efficiently handle in-domain recordings and transcriptions. This hypothesis is supported by the performance of Whisper systems family on the *MLS* corpus; however, Google's performance on the *Common Voice* dataset was nearly twice as bad as other systems. Given that Whisper is trained mostly on publicly available data, while commercial systems leverage proprietary datasets, the impact of training and evaluation data leakage is more significant in the case of Whisper.

Performance for the CLARIN mobile dataset was slightly inferior, possibly due to longer utterances and the use of commercial *default* models, which are not optimized to handle speech recorded with an 8 kHz sampling frequency.

As expected, performance declined for the AZON read and spontaneous corpora, which contain scientific vocabulary from the chemistry field. However, the Google and Whisper local systems handled both types of AZON corpora proficiently, despite containing fillers and hesitations.

Table V-B presents the median WER for specific datasets sourced in BIGOS for Azure, Google, Whisper Cloud and Large systems.

VI. LIMITATIONS

The initial version of the benchmark comes with several limitations. First, the quantity and specificity of the datasets, along with the metadata about speakers and acoustic conditions, are limited. To examine ASR performance for particular sociodemographic groups, such as non-native Polish speakers or specific types of speech, such as whispery speech, dedicated datasets[33] should be used. Second, the unification of normalization relies solely on automatic methods and does not involve manual re-transcription. Lastly, the initial evaluation uses a limited number of test recordings, systems, and models, which constrains the precision and breadth of the benchmark.

Table V
AVERAGE WER PER DATASET FOR SELECTED SYSTEMS

	Paid			**Free**	
Dataset	**A**	**G**	**W**	**W**	**WER avg**
pwr-maleset	6.6	**3.2**	6.1	**3.2**	4.8±1.8
pwr-shortwords	7.1	**4.4**	7.8	4.8	6.0±1.7
pwr-viu	0.3	24.4	**0.0**	7.9	8.1±11.4
common-voice-19	**10.2**	19.9	11.2	10.3	12.9±4.7
mailabs-19	19.5	19.6	**8.4**	8.5	14.0±6.4
mls-20	30.0	22.9	5.9	**4.6**	15.8±12.6
azon-read-20	35.9	4.1	23.4	**3.8**	16.8±15.7
pjatk-mobile-15	26.9	30.7	11.8	**10.7**	20.0±10.3
azon-spont-20	28.2	15.7	24.2	**14.3**	20.6±6.7
WER average	18.3±12.6	16.1±10.0	11.0±8.0	**7.6±3.8**	13.2 ±4.9

VII. CONCLUSION AND FUTURE WORK

This work addresses the lack of a publicly available ASR evaluation suite for Polish by providing BIGOS, Benchmark Intended Grouping of Open Speech corpora. BIGOS, as its name suggests, was compiled from 10 existing publicly accessible Polish speech corpora. A test sample comprising 1900 recordings from 71 distinct speakers was used to gauge the performance of 3 commercial ASR systems against 5 freely available ones. Through automatic evaluation metrics, it was discovered that Whisper Cloud consistently outperforms more established services from Google and Azure on the test set representing publicly available speech datasets for Polish. Interestingly, the largest and second largest of the Whisper models exhibit superior performance compared to its paid version. The BIGOS corpus[11] and tools[12] for corpus curation and evaluation of ASR systems are available to the community, allowing reproduction and extension of this benchmark.

As indicated in the Limitations and Related Work sections, there are many interesting research directions to explore. The primary objective of the next BIGOS iteration is to include a subset of manually verified reference transcriptions. Comparison of error rates, calculated using original and manually verified transcriptions, will reveal the evaluation bias resulting from differences in normalization standards in various public-domain corpora. Furthermore, the reliability and informativeness of the evaluation could be significantly improved if the evaluation results were manually annotated, similar to the German study [3], which revealed that the evaluation errors may be caused by the poor quality of the evaluation data and that not all errors are of equal importance. Lastly, it will be interesting to measure the robustness of the systems using larger samples, new data sources, and automatically perturbed recordings.

[11] https://huggingface.co/datasets/michaljunczyk/pl-asr-bigos

[12] https://github.com/goodmike31/pl-asr-bigos-tools

References

[1] Alëna Aksënova et al. “How Might We Create Better Benchmarks for Speech Recognition?” In: Association for Computational Linguistics, 2021, pp. 22–34. DOI: 10.18653/v1/2021.bppf-1.4.

[2] Piotr Szymański et al. “WER we are and WER we think we are”. In: Association for Computational Linguistics, 2020, pp. 3290–3295. DOI: 10.18653/v1/2020.findings-emnlp.295.

[3] Johannes Wirth and Rene Peinl. “ASR in German: A Detailed Error Analysis”. In: (2022). DOI: 10.48550/arXiv.2204.05617.

[4] Miguel Del Rio et al. “Earnings-21: A Practical Benchmark for ASR in the Wild”. In: (2021).

[5] Miguel Del Rio et al. “Earnings-22: A Practical Benchmark for Accents in the Wild”. In: (Mar. 2022). DOI: 10.48550/arXiv.2203.15591.

[6] Sanchit Gandhi, Patrick von Platen, and Alexander M. Rush. “ESC: A Benchmark For Multi-Domain End-to-End Speech Recognition”. In: (Oct. 2022). DOI: 10.48550/arXiv.2210.13352.

[7] Malgorzata Anna Ulasik et al. “CEASR: A corpus for evaluating automatic speech recognition”. In: 2020, pp. 6477–6485.

[8] Péter Mihajlik et al. “BEA-Base: A Benchmark for ASR of Spontaneous Hungarian”. In: *2022 Language Resources and Evaluation Conference, LREC 2022* (Feb. 2022), pp. 1970–1977. DOI: 10.48550/arXiv.2202.00601.

[9] Vassil Panayotov et al. *LIBRISPEECH: AN ASR CORPUS BASED ON PUBLIC DOMAIN AUDIO BOOKS.*

[10] Vineel Pratap et al. “MLS: A Large-Scale Multilingual Dataset for Speech Research”. In: *Proc. Interspeech 2020*. 2020, pp. 2757–2761. DOI: 10.21437/Interspeech.2020-2826.

[11] François Hernandez et al. “TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation”. In: (2018). DOI: 10.1007/978-3-319-99579-3_21.

[12] Heidi Christensen et al. “The CHiME corpus: a resource and a challenge for computational hearing in multi-source environments”. In: ISCA, 2010, pp. 1918–1921. DOI: 10.21437/Interspeech.2010-552.

[13] Rosana Ardila et al. “Common Voice: A Massively-Multilingual Speech Corpus”. In: (2020). DOI: 10.48550/arXiv.1912.06670.

[14] Christian Gaida et al. “Comparing Open-Source Speech Recognition Toolkits”. In: 2014.

[15] Meredith Moore et al. “Say What? A Dataset for Exploring the Error Patterns That Two ASR Engines Make”. In: 2019, pp. 2528–2532. DOI: 10.21437/Interspeech.2019-3096.

[16] Ingo Siegert et al. *Recognition Performance of Selected Speech Recognition APIs – A Longitudinal Study*. 2020. DOI: 10.1007/978-3-030-60276-5_50.

[17] Binbin Xu et al. “A Benchmarking on Cloud based Speech-To-Text Services for French Speech and Background Noise Effect”. In: (2021).

[18] Vered Silber Varod et al. “A cross-language study of speech recognition systems for English, German, and Hebrew”. In: *Online Journal of Applied Knowledge Management* (2021), pp. 1–15. DOI: 10.36965/OJAKM.2021.9(1)1-15.

[19] Morgane Riviere, Jade Copet, and Gabriel Synnaeve. “ASR4REAL: An extended benchmark for speech models”. In: (2021).

[20] Martha Maria Papadopoulou, Anna Zaretskaya, and Ruslan Mitkov. “Benchmarking ASR Systems Based on Post-Editing Effort and Error Analysis”. In: INCOMA Ltd., 2021, pp. 199–207.

[21] Alëna Aksënova et al. “Accented Speech Recognition: Benchmarking, Pre-training, and Diverse Data”. In: (2022). DOI: 10.48550/arXiv.2205.08014.

[22] Regis Pires Magalhães et al. “Evaluation of Automatic Speech Recognition Approaches”. In: *Journal of Information and Data Management* 13 (3 Sept. 2022). DOI: 10.5753/jidm.2022.2514.

[23] Marcin Pacholczyk. *Przegląd I porównanie rozwiązań rozpoznawania mowy pod kątem rozpoznawania zbioru komend głosowych*. 2018.

[24] Danijel Koržinek. “Task 5: Automatic speech recognition PolEval 2019 competition”. In: (2019). URL: http://2019.poleval.pl/files/2019/11.pdf.

[25] Nahuel Unai et al. “Development and evaluation of a Polish ASR system using the TLK toolkit”. 2019.

[26] Danijel Koržinek, Krzysztof Marasek, and Łukasz Brocki. *Polish Read Speech Corpus for Speech Tools and Services*. 2016.

[27] Piotr Pęzik. “Spokes – a search and exploration service for conversational corpus data”. In: 2015.

[28] Piotr Pęzik. “Increasing the Accessibility of Time-Aligned Speech Corpora with Spokes Mix”. In: European Language Resources Association (ELRA), 2018.

[29] Krzysztof Marasek, Danijel Korzinek, and Łukasz Brocki ˇ. “System for Automatic Transcription of Sessions of the Polish Senate”. In: (2014).

[30] Piotr Pęzik et al. *DiaBiz - an Annotated Corpus of Polish Call Center Dialogs*, pp. 20–25.

[31] Piotr Pęzik and Michał Adamczyk. *Automatic Speech Recognition for Polish in 2022*. University of Łódź, 2022. URL: https://clarin-pl.eu/dspace/bitstream/handle/11321/894/ASR_PL_report_2022.pdf.

[32] Alec Radford et al. “Robust Speech Recognition via Large-Scale Weak Supervision”. In: (2022). DOI: 10.48550/arXiv.2212.04356.

[33] Piotr Kozierski et al. “Acoustic Model Training, using Kaldi, for Automatic Whispery Speech Recognition”. In: 2018. DOI: 10.15439/2018F255.

DOI: 10.15439/2023F5438
ISSN 2300-5963 ACSIS, Vol. 35

MLP-COMET-based decision model re-identification for continuous decision-making in the complex network environment

Bartłomiej Kizielewicz
0000-0001-5736-4014
Dept. of Artificial Intelligence Methods and Applied Mathematics,
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: bartlomiej-kizielewicz@zut.edu.pl

Jakub Więckowski
0000-0002-9324-3241
National Institute of Telecommunications
ul. Szachowa 1, 04-894 Warsaw, Poland
Email: j.wieckowski@il-pib.pl

Jarosław Jankowski
0000-0002-3658-3039
Dept. of Information Systems Engineering,
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: jjankowski@zut.edu.pl

Abstract—**In recent years, complex networks have gained significant attention for their practical potential in data analysis and decision-making. However, assessing node relevance in complex networks poses challenges, including subjectivity and difficulty reproducing criteria relationships. To address these issues, we propose MLP-COMET. This novel approach combines the Multi-Layer Perceptron (MLP) with the Characteristic Objects Method (COMET) in Multi-Criteria Decision Analysis (MCDA). MLP-COMET aims to re-identify decision models using MLP to evaluate characteristic objects. We evaluate the approach to assessing the complex network and demonstrate its effectiveness in evaluating without heavy reliance on domain experts. The MLP-COMET performance is evaluated through ranking comparisons, showing a strong correlation with reference expert rankings. We also analyze the impact of training sample size and number of characteristic objects on ranking similarity, observing high stability and similarity using the r_w metric. MLP-COMET offers an effective and reliable tool for evaluating complex networks and facilitating decision-making processes.**

I. Introduction

COMPLEX networks have been significantly developed in recent years due to their high practical potential [1]. They have been used effectively in the areas of quantum systems [2], information processing [3], decision tree analysis [4], or node relevance assessment problems [5]. The increasing computational capabilities of computer technology allow more complex and efficient solutions to be developed [6]. It has also translated into strengthening the position of techniques included in complex networks in their use for data analysis [7]. By using such approaches in a wide range of practical problems, more efficient and effective solutions can be achieved, and benefits can be derived from the conclusions drawn from the analyzed data.

One highly popular area considering the developed models based on complex networks is connected to blockchain and cryptocurrencies [8]. Since the field related to virtual payments is expanding, there was a need to propose solutions that could be used to make more rational and conscious decisions. Complex networks are applied for analyzing the blockchain structure [9], the performed transactions [10], or for the automation processes [11], among others. Those techniques can be used to extract knowledge that can be used for further analysis and constitute making more effective steps in the area of blockchain and cryptocurrencies.

Complex networks are based on the usage of nodes in the analysis process [12]. It often leads to incompatible nodes assessment taking into account the centrality measures. Since the occurrence of this phenomenon should be limited, various techniques are used to reduce it. For this purpose, Multi-Criteria Decision Analysis (MCDA) methods can be used [13]. The MCDA techniques allow for assessing decision variants regarding multiple criteria that are considered in the evaluation process [14], [15]. Moreover, those methods enable modeling different preferences depending on the outcome and objectives that are expected as the final results [16], [17]. It can be achieved by modifying the criteria weights, representing the relevance of subsequent decision factors [18]. With this approach, the determined models are highly configurable and reusable in different initial conditions.

To model the criteria importance, different approaches can be used [19]. Many decision models rely on professional knowledge and experience that is extracted from the domain expert through the criteria judgment process [20]. The subjective weighting methods can be used for this purpose [21]. They allow for a structured and systematic judgment process, indicating assessment steps that aim to simplify the criteria importance evaluation for the expert. Multiple approaches can be used since various techniques are being developed in this area. One of the most popular methods is the Analytical Hierarchy Process (AHP) [22], which is based on the criteria pairs comparison aiming to establish the relationship between assessed criteria. The other approach that can be applied to extract expert knowledge is the new method of Ranking Comparison (RANCOM) [23], which proved to handle expert judgment inaccuracies significantly better than the mentioned AHP method. The other techniques that can be used for this purpose are Best-Worst Method (BWM) [24], Full Consistency Method (FUCOM) [25], Fixed Point Scoring [18], or Simple Multi-Attribute Rating Technique (SMART) [26], among others.

Despite multiple advantages that can be benefited from engaging the domain expert in the evaluation process, there are also some drawbacks of this approach [27]. The main disadvantages of determining the decision models based on expert knowledge are their unavailability, a certain level of hesitance and inaccuracies of the judgments, or the impossibility of reproducing the previously defined relationships between criteria relevance [28]. It can lead to multiple difficulties in making the determined model reusable in various applications. However, it is worth developing ready-to-use models that guarantee highly effective and reliable results.

In this paper, we propose a multi-criteria decision analysis model for evaluating complex networks, which is an artificial expert in the form of a MultiLayer Perceptron (MLP) combined with a Characteristic Object METhod (COMET). The MLP-COMET approach aims to re-identify the decision model based on the evaluated decision variants. The MLP in the regressor variant is used to represent the domain expert that assesses the Characteristic Objects (COs) in the COMET method. The practical problem of assessing the Bitcoin network is used to verify the model's performance. The main contributions of the study are

- presenting an approach that enables re-identification of decision model
- indicating the methodology that can be used to replace the domain expert in the decision process
- analyzing the Bitcoin network with the determined MLP-COMET technique

The rest of the paper is organized as follows. Section 2 presents the literature review on centrality metrics in complex networks and MCDA and its usage in the practical problems connected to the blockchain field. Section 3 presents the preliminaries of the complex network and MCDA. Section 4 presents the proposed approach for re-identifying the multi-criteria decision model. Section 5 shows the study case of using MLP-COMET to re-identify the decision model based on the evaluated alternatives in the practical problem of analyzing the Bitcoin network. Finally, Section 6 presents the conclusions drawn from the research and the further directions for developing the presented approach.

II. Literature review

A. Selection problems based on centrality metrics

In the area of complex networks, it is possible to distinguish metrics that define the attractiveness of a node concerning the others available in the analyzed network. Their analysis allows more efficient choices to be made regarding selecting nodes that are key to information propagation. Measures of centrality such as degree, closeness, betweenness, and eigenvector are a group of factors that allow efficient analysis of network nodes and their selection [29]. Alexandrescu et al. used the four mentioned centrality measures to identify the sustainability communicators in urban regeneration [30]. The presented measures were applied as the decision criteria in one of the three dimensions that were determined in the assessment, namely, the informal network influence. Karczmarczyk et al. applied the MCDA techniques to select seeds for targeted influence maximization within social networks, where the centrality, betweenness, closeness, and eigenvector centrality measures were also considered in the evaluation of the node [31]. Muruganantham et al. focused on the problem of discovering and ranking the influential users in social media networks by applying the selected MCDA methods, namely Preference Ranking Organization Method for Enrichment of Evaluations (PROMETHEE) II, ELimination Et Choix Traduisant la REalité (ELECTRE), AHP, Statistical Design Institute Matrix method (SDI), Pugh (also known as Decision Matrix Method), and Technique for the Order of Prioritisation by Similarity to Ideal Solution (TOPSIS) [32]. The authors also used the four above-mentioned measures to assess the social influence in the network. Moreover, the centrality metrics can be grouped based on their scope of operation. The closeness, betweenness, eigenvector, coreness, average clustering coefficient, average shortest path length, and PageRank measures belong to global measures, while degree or semi-local centrality are classified as measures with local scope [33]. The global measures are identified based on the necessity of having access to the whole network to determine the global information for the specific factor. On the other hand, local measures can be calculated using the local information of the node.

B. Blockchain and cryptocurrencies in MCDA

MCDA methods are used in many application areas due to the ability to flexibly select decision criteria based on which decision variants are assessed. This configurability and versatility allow decision-making models to be used in the area related to blockchain and cryptocurrencies. Lai and Liao proposed an approach for MCDM based on Double Normalization-based Multiple Aggregation (DNMA) and Criteria Importance Through Inter-criteria Correlation (CRITIC) for blockchain platform evaluation [34]. The authors considered 8 decision criteria, namely performance efficiency, interactivity, scalability, reliability, security, portability, maintainability, and cost. Erol et al. examined blockchain applicability in sustainable supply chains by the MCDM framework determined with Fuzzy Step-wise Weight Assessment Ratio

Analysis (SWARA), Complex Proportional Assessment (COPRAS), Evaluation based on Distance from Average Solution (EDAS) assessment, and COPELAND method [35]. The evaluation considered 6 decision variants and 8 criteria. Öztürk and Yildizbaşi focused on indicating the barriers that keep the implementation of blockchain into supply chain management [36]. Based on the Fuzzy AHP and Fuzzy TOPSIS, the assessment was conducted considering the uncertainties in the problem. The results from the research showed that high investment costs, data security, and utility play the most important role in the evaluation. Çolak et al., on the other hand, directed their research toward an assessment of blockchain technology in supply chain management [37]. Using the Hesitant Fuzzy Sets (HFS) combined with the AHP (HF-AHP) and TOPSIS (HF-TOPSIS), it was possible to examine the decision alternatives and take into account the potential uncertainties. The authors identified 5 main criteria and 17 sub-criteria, which were used to evaluate 5 decision variants. The sensitivity analysis approach was also used to examine if differences in criteria weights could significantly influence the proposed rankings. Based on the obtained results, the authors indicated that the medicine/drug industry seems to be the most suitable sector for introducing blockchain technology. Table I presents the selected approaches used in multi-criteria problems directed to blockchain and cryptocurrency fields.

C. Blockchain and cryptocurrencies in complex networks

The problems connected to blockchain analysis are also addressed by researchers using complex network techniques. Since it is important to identify the most significant nodes in the networks that play a crucial role in the information spread, many approaches have been used for this purpose. Moreover, the centrality measures are eagerly used to investigate the network structures allowing for an in-depth analysis. Tao et al. performed a complex network analysis of the Bitcoin blockchain network, using degree distribution, clustering coefficient, shortest path length, assortativity, and rich-club coefficient [10]. Bielinskyi and Soloviev attempted to identify the complex network precursors of crashes and critical events in the cryptocurrency market [38]. The authors used time series of data considering the days in correction, Bitcoin's high price in $, Bitcoin's low price in $, the decline in %, and the decline in $. As the centrality measures, the authors selected eigenvector values and average path length. Lin et al. focused on understanding Ethereum transaction records with a complex network approach [39]. The authors modeled the transaction records using time and amount features and designed several flexible temporal walk strategies. The degree distribution of the Ethereum transaction network was analyzed with an actual feasible path for money flow. Serena et al. represented cryptocurrency activities ad a complex network to analyze the transaction graphs [40]. Four prominent Distributed Ledger Technologies (DLTs), namely Bitcoin, DogeCoin, Ethereum, and Ripple, were considered. The authors considered three selected centrality measures: degree distribution, average clustering coefficient, and average shortest path length of the main component.

D. Expert knowledge in multi-criteria problems

Multi-Criteria Decision Analysis models can be personalized with the different preferences of criteria importance. This approach can be used to propose an individual and specific set of results compliant with the expert preferences and expectations. To extract experts' knowledge and use it as the input data in MCDA models, subjective criteria weighting methods are used. Since multiple techniques are being developed to assist the expert in identifying the criteria importance, it is important to select methods that are intuitive and reflects the experts' opinion reliably. Various Decision Support Systems (DSSs) were determined to evaluate alternatives using the domain expert knowledge in the specific field. Dweiri et al. proposed a DSS based on the AHP method for supplier selection in the automotive industry, where the AHP method was used to identify the expert preferences regarding the criteria importance [41]. Mahendra used the FUCOM-SAW method to determine the DSS for e-commerce selection in Indonesia [42]. The FUCOM method served as a measure for extracting the expert knowledge based on which the assessment was performed. Sarabi and Darestani applied the Fuzzy Multiple Objective Optimizations on the basis of Ratio Analysis plus full Multiplicative Form (MULTIMOORA) and BWM approach for determining the DSS for logistics service provider selection in mining equipment manufacturing [43]. The BWM method allowed for defining the criteria relevance based on the expert experience in the given field. The RANCOM method was used to identify the decision-maker preferences regarding the laptop selection, and the identified weights were then used in the selected six MCDA methods [23]. Fahlepi proposed a DSS for employee discipline identification, where the SMART method was used for establishing the criteria relevance based on the expert judgment [44]. It can be seen that various approaches are used to define the decision models based on expert knowledge. However, it should be borne in mind that the experts' availability limits these solutions. Moreover, expert knowledge can change over time, translating into assigning different criteria relevance within the same decision problem. The subjectivity of the assessment should also be considered in developing such systems. It should be limited to providing results with high objectivity of the evaluation, increasing the results' reliability. Since it could be challenging to re-identify the experts' preferences over time, it is worth proposing approaches to fill this gap. To this end, the MLP-COMET technique is proposed, which is based on the complex network analysis and aims to identify the decision model which can be applied to assess new decision variants within the same problem.

III. Preliminaries

A. Centrality measures

Complex network centrality metrics are network analysis tools used to identify nodes of high importance or influence

TABLE I
SELECTED APPROACHES FOR SOLVING BLOCKCHAIN AND CRYPTOCURRENCIES PROBLEMS WITH MCDA METHODS

Method	No. of alts.	No. of crit.	Problem	Year	Reference
PROMETHEE II	80	6	Cryptocurrency exchanges evaluation	2021	[45]
AHP, PROMETHEE II	9	7	Cryptocurrency portfolio selection	2021	[46]
Q-Rung Orthopair Fuzzy Hypersoft Sets	4	8	Cryptocurrency market analysis	2022	[47]
MARCOS, Fuzzy MARCOS	6	5	Blockchain software selection	2021	[48]
AHP, TOPSIS	3	16	Cryptocurrency mining strategies	2021	[49]
Fuzzy BWM	3	15	Cryptocurrency trading system	2023	[50]
Fuzzy TOPSIS	3	27	Object selection in blockchain-enabled IoT platforms	2022	[51]
Fuzzy AHP, Fuzzy VIKOR	9	8	Feasibility evaluation of blockchain in logistics operations	2020	[52]

* where: 'No. of alts.' - Number of decision variants, 'No. of crit.' - Number of criteria.

in a network. Over the past few years, the trend of introducing new centrality metrics has continued to grow. The mainly used centrality metrics of complex networks are closeness centrality, degree centrality, eigenvector centrality, or betweenness centrality. In addition, there are also centrality metrics such as Katz centrality, harmonic centrality, or percolation centrality. In assessing the relevance of social network nodes, several centrality metrics are mainly used due to the need for knowledge related to the systematic distinction of these measures of [53]. Therefore, in this article, we will focus on the following measures of social network centrality [54], [55], [56]:

1) **Degree centrality:**

$$D_c(i) = \sum_{j}^{n} x_{ij} \tag{1}$$

where i is the considered node, j is the other nodes present in the network, n is the number of all nodes, and x_{ij} is the connection between node i and node j.

2) **Betweenness centrality:**

$$B_c(i) = \left(\sum_{s \neq i \neq t} \frac{g_{st}(i)}{g_{st}} \right) \frac{n(n-1)}{2} \tag{2}$$

where g_{st} is the count of binary shortest paths from node s to node t, and $g_{st}(i)$ is the count of those paths that pass through node i.

3) **Eigenvector centrality:**

$$E_c(i) = \lambda^{-1} \sum_{j=1}^{n} A_{ij} e_j \tag{3}$$

where e_j is the node score j, A is the adjacency matrix of the network, n is the number of nodes present in the network, and λ is a constant.

4) **Closeness centrality:**

$$C_c(i) = \frac{n-1}{\sum_{j=1}^{n} d_{ij}} \tag{4}$$

where d_{ij} is the distance from node i to node j.

5) **Harmonic centrality:**

$$H_c(i) = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{\operatorname{dist}(x_i, x_j)} \tag{5}$$

where i is the considered node, j is the other nodes present in the network, d_{ij} is the distance from node i to node j.

B. The Multi-Layer Perceptron Regressor

Artificial neural networks are computational models inspired by the structure and operation of the human brain. One type of artificial neural network is the Multi-Layer Perceptron, which is widely used in classification and regression problems. It consists of multiple perceptrons, the structure of which is based on the original approach proposed by Frank Rosenblatt in 1957. A Multi-Layer Perceptron consists of three main layers: an input, hidden, and output layer. The input layer accepts input data, passed on to subsequent layers. Hidden layers are intermediate between input and output and consist of multiple perceptrons. The output layer generates the final results of the network. The connections between perceptrons in the different layers are weighted, meaning each connection is assigned a weight. These weights determine how much the output of one perceptron affects the input of other perceptrons.

A Multi-Layer Perceptron uses supervised learning, which requires a set of learning data consisting of pairs of input and expected output data. The goal is to train the network to learn a function that transforms the input data into the expected output data. The backward error propagation algorithm is most commonly used, which propagates the error from the network's output to the hidden layers and the input layer to adjust the connection weights. The effectiveness of a multilayer perceptron depends on several hyperparameters, such as the number of hidden layers, the number of perceptrons in each layer, the learning rate, and the activation function. Proper selection of hyperparameters is crucial to the effectiveness and efficiency of the network. An example of neural network visualization is shown with Fig. 1.

C. The Characteristic Objects Method

The Characteristic Objects METhod (COMET) is an approach proposed by Sałabun in 2015 to eliminate the paradox

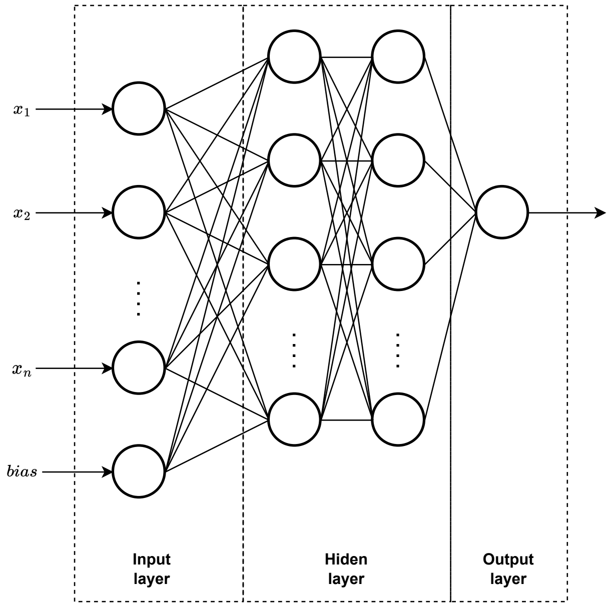


Fig. 1. Example structure of a multilayer perceptron.

of reversed rankings [57]. Here, the evaluation of decision alternatives is done by measuring the distance between them and the characteristic objects that play a key role in the model. In addition, this method has seen many extensions for uncertain environments such as Normalized Interval-Valued Triangular Fuzzy Numbers (NIVTFN) [58], Intuitionistic Fuzzy Sets (IFS) [59] and Hesitant Fuzzy Sets (HFS) [60], For the COMET method, the following sequence of steps is used:

Step 1. Identify the problem's dimensionality. Expert selects r number of criteria and their fuzzy values, which is represented by a Eq. (6).

$$\begin{array}{c} C_1 = \left\{\tilde{C}_{11}, \tilde{C}_{12}, ..., \tilde{C}_{1c_1}\right\} \\ C_2 = \left\{\tilde{C}_{21}, \tilde{C}_{22}, ..., \tilde{C}_{2c_2}\right\} \\ ... \\ C_r = \left\{\tilde{C}_{r1}, \tilde{C}_{r2}, ..., \tilde{C}_{rc_r}\right\} \end{array} \tag{6}$$

where $C_1, C_2, \dots, C_r$ are the criteria represented by the fuzzy numbers.

Step 2. Creating Characteristic Objects (COs) with the Cartesian product from fuzzy number cores. An example of the construction of characteristic objects can be illustrated by the Eq. (7).

$$CO = \langle C(C_1) \times C(C_2) \times ...C(C_r)\rangle \tag{7}$$

The result is a set of characteristic objects. This set can be expressed as follows:

$$\begin{array}{c} CO_1 = \langle C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r1})\rangle \\ CO_2 = \langle C(\tilde{C}_{11}), C(\tilde{C}_{21}), ..., C(\tilde{C}_{r2})\rangle \\ ... \\ CO_t = \langle C(\tilde{C}_{1c_1}), C(\tilde{C}_{2c_2}), ..., C(\tilde{C}_{rc_r})\rangle \end{array} \tag{8}$$

Step 3. Formation of Matrix of Expert Judgments (MEJ) using comparisons of characteristic objects among themselves. The Expert Judgment Matrix (MEJ) is represented by the Eq. (9).

$$MEJ = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1t} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2t} \\ \dots & \dots & \dots & \dots \\ \alpha_{t1} & \alpha_{t2} & \dots & \alpha_{tt} \end{pmatrix} \tag{9}$$

where α_{ij} is the degree of preference of comparing one characteristic object to another. If object CO_i is more reflective than object CO_j assign the value 1. If they are equal, assign the value 0.5. If CO_i is less reflective than CO_j assign the value 0. It can be shown by the Eq. as follows:

$$\alpha_{ij} = \begin{cases} 0.0, & f_{expert}(CO_i) < f_{expert}(CO_j) \\ 0.5, & f_{expert}(CO_i) = f_{expert}(CO_j) \\ 1.0, & f_{expert}(CO_i) > f_{expert}(CO_j) \end{cases} \tag{10}$$

Once the expert matrix MEJ is determined, the Summed Judgements (SJ) vector must be determined using Eq. (11).

$$SJ_i = \sum_{j=1}^{t} \alpha_{ij} \tag{11}$$

where t is the number of characteristic objects.

After computing the Summed Judgements (SJ) vector, the vector of preferences (P) for the COs should be computed. This is shown as follows [57].

Step 4. Formation of a rule base from characteristic objects and a preference vector. This can be expressed using an Eq. (12).

$$IF\ C\left(\tilde{C}_{1i}\right)\ AND\ C\left(\tilde{C}_{2i}\right)\ AND\ \dots\ THEN\ P_i \tag{12}$$

Step 5. Make an inference to compute the scores of the given alternatives. The alternative A_i comprises the values of every criterion, i.e., $A_i = \{\alpha_{1i}, \alpha_{2i}, \dots, \alpha_{ri}\}$. By employing Mamdani fuzzy inference, a preference P is computed for every alternative according to [61].

IV. Proposed approach

This paper proposes an approach to evaluate nodes in a complex network using the MultiLayer Perceptron Regressor (MLP Regressor) and the Characteristic Objects METhod (COMET). This approach aims to construct a multi-criteria model to evaluate network nodes. In traditional expert-based multi-criteria models, problems often arise due to dynamically changing knowledge and the limited availability of experts. Our approach uses MLP Regressor as an artificial expert trained from existing node evaluations. This allows us to avoid relying on experts and obtain node evaluations based on the artificial expert model. After constructing the artificial expert model, we use it to evaluate Characteristic Objects in the COMET approach. Characteristic Objects are reference points with information about the decision maker's preferences. This allows us to construct a multi-criteria model that considers the decision-makers preferences and allows us to evaluate the nodes with these preferences in mind. The approach described in the paper aims to combine machine learning techniques, such as MLP Regressor, with multi-criteria analysis to evaluate nodes in a complex network efficiently. This hybrid approach can be helpful in various fields where there is a need to make decisions based on network analysis considering the decision maker's preferences.

Fig. 2 represent the proposed MLP-COMET approach. The first step in this approach is to determine the set of evaluated decision alternatives, the hyperparameters for the MLP Regressor model, and the decision criteria with their characteristic values needed for the COMET method. This is followed by training the MLP-Regressor model, an artificial expert for this approach. Once a stable model maps the decision maker's preferences to the designated set of decision alternatives created, the decision model is initialized. Then the structure of the COMET method is modeled based on the characteristic values and the reidentification of the decision model using the artificial expert (model: MLP). After determining the preference of Characteristic Objects, the newly created decision options can be evaluated. In the case of this article, the implementation of the entire algorithm was created using the `sklearn` library (class: MLPRegressor) and the `pymcdm` library (class: COMET) [62], [63].

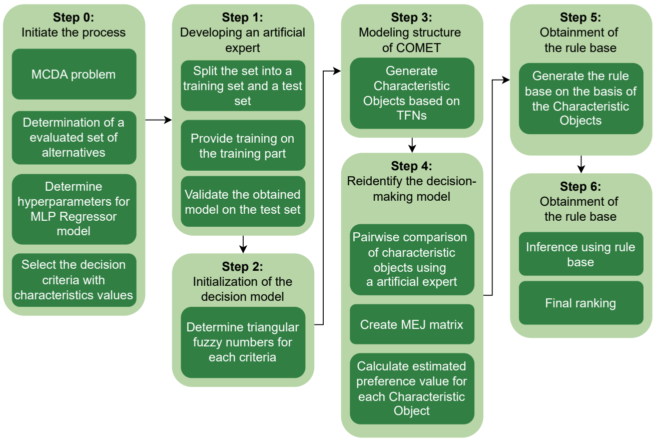

Step 0:
Initiate the process
MCDA problem
Determination of a evaluated set of alternatives
Determine hyperparameters for MLP Regressor model
Select the decision criteria with characteristics values
Step 1:
Developing an artificial expert
Split the set into a training set and a test set
Provide training on the training part
Validate the obtained model on the test set
Step 2:
Initialization of the decision model
Determine triangular fuzzy numbers for each criteria
Step 3:
Modeling structure of COMET
Generate Characteristic Objects based on TFNs
Step 4:
Reidentify the decision-making model
Pairwise comparison of characteristic objects using a artificial expert
Create MEJ matrix
Calculate estimated preference value for each Characteristic Object
Step 5:
Obtainment of the rule base
Generate the rule base on the basis of the Characteristic Objects
Step 6:
Obtainment of the rule base
Inference using rule base
Final ranking

Fig. 2. MLP-COMET approach procedure.

V. Study case

In this section, a study will be conducted related to the proposal of the MLP-COMET approach for evaluating the composite network. First, the dataset associated with the Bitcoin composite network will be described. Then, research on the accuracy of the MLP Regressor model, which will serve as an artificial decision expert, will be conducted. After the research on its accuracy, an example of re-identifying the decision model and examining the similarity of the rankings derived from MLP-COMET at different characteristic values and the size of the learning set demonstrated is.

A. Description of the data

For this article, a complex network related to cryptocurrencies and, more specifically, Bitcoin was chosen [64], [65]. The selected network is people who trust those using this cryptocurrency. The network presented has 5881 nodes and 35592 edges. In addition, the network directed is in this way, and a weight is assigned to each of its edges. For this article, the weights of each edge were taken into account in determining centrality measures such as betweenness and eigenvector. In the case of the present network, nodes will play the role of decision variants. A visualization of the Bitcoin user network is shown in Fig. 3.

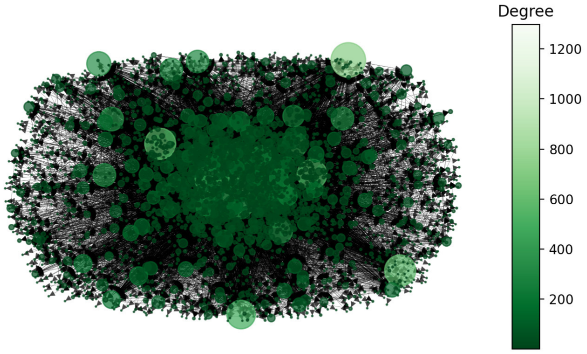


Fig. 3. Complex network of Bitcoin users [64], [65].

To evaluate the nodes of the present complex network, centrality metrics were used as criteria. Five centrality metrics were selected, i.e., betweenness centrality, degree centrality, eigenvector centrality, closeness centrality, and harmonic centrality. These metrics are presented in the Section III-A. Due to the low values found among some centrality metrics, the number of nodes is shown on histograms on a logarithmic scale.

Fig. 4 shows the distribution of values of the betweenness centrality metric, chosen as the first criterion for evaluating nodes (C_1). The minimum value of the centrality measure of indirectness is 2.89182e-08, which means that there are nodes with a shallow indirect role. The mean value of the measure is 0.01798, suggesting that most nodes have a low mediating role. The highest recorded value of the centrality of agency measure is 0.84816, suggesting the existence of a few nodes with a highly high mediating role. The standard deviation value is 0.06324, indicating some variation in the distribution of the centrality measure. The skewness value is 4.39799, indicating that the distribution of the centrality measure of intermediation is skewed to the right. This means there are a few nodes with very high centrality, which may indicate the existence of crucial nodes in the Bitcoin network.

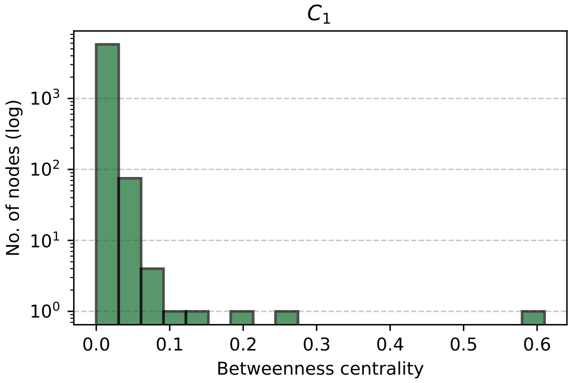


Fig. 4. Distribution of betweenness centrality values for Bitcoin users complex network.

The distribution of the degree centrality metric's value, which is selected as the second criterion for evaluating nodes (C_2), is shown using Fig. 5. The distribution of degree centrality measure values for users of the Bitcoin comprehensive network is as follows: the minimum value of the degree measure is 0.00017, indicating the existence of nodes with a low degree of connectivity. The average value of the degree measure is 0.00205, implying that most nodes have a lesser degree of connection. However, the highest recorded value of the degree measure is 0.22074, indicating a few nodes with an extremely high degree of connection. The standard deviation is 0.00651, indicating a rather diverse distribution of the degree measure. In addition, the skewness value is 13.84915, implying that the distribution is significantly skewed to the right. This means there are a few nodes with a very high degree of connectivity.

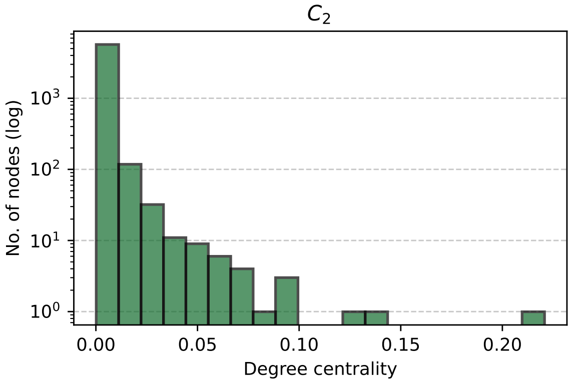


Fig. 5. Distribution of degree centrality values for Bitcoin users complex network.

Eigenvector centrality is another metric used as the third criterion (C_3) for evaluating nodes, and Fig. 6 represents it. The minimum value of the measure is -0.17028, which indicates the presence of negatively influenced nodes in the network. The mean value of the measure is 0.00038, revealing that most nodes have little influence in the network. The highest recorded value of the measure is 0.32039, indicating the existence of a few nodes with high importance in the Bitcoin network. The standard deviation value is 0.01303, which hints at some variation in the distribution of the vector centrality measure. The skewness value is 3.97669, pointing to a skewed distribution to the right.

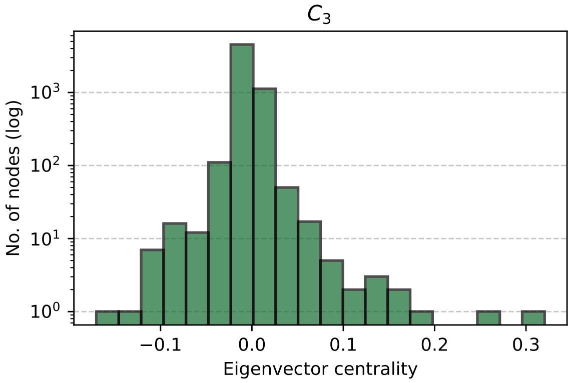


Fig. 6. Distribution of eigenvector centrality values for Bitcoin users complex network.

The closeness centrality metric for evaluating nodes was chosen as the fourth criterion (C_4), and its distribution is shown in Fig. 7. The minimum value of the measure is 0.0, which means there are nodes that not directly connected are to any other node in the network. The average value of the measure is 0.21886, which suggests that most nodes have a moderate degree of proximity to other nodes in the Bitcoin network. The highest recorded value of the measure is 0.33939, indicating that a few nodes exceptionally well connected are to other nodes. The standard deviation value is 0.03196, indicating some variation in the distribution of the proximity centrality measure. The skewness value is -1.65533, indicating that the distribution is slightly skewed to the left.

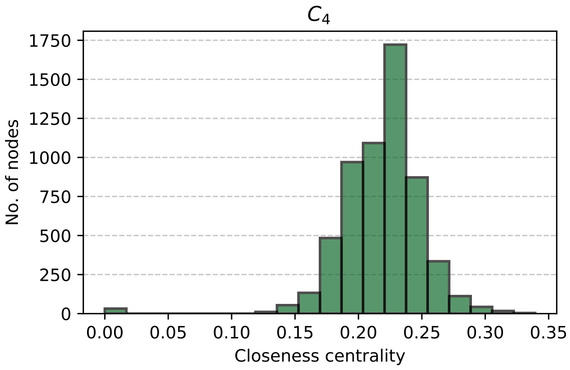


Fig. 7. Distribution of closeness centrality values for Bitcoin users complex network.

Fig. 8 shows the distribution of the value of the harmonic centrality metric, which is chosen as a criterion of the five to evaluate nodes (C_5). Based on statistical information, the minimum value of harmonic centrality in the studied network was 0.0, which means that some nodes did not have an essential role in transmitting the information. The mean value of harmonic centrality was 1346.77502, reflecting that most nodes in the network have moderate importance. The highest value recorded was 2233.24999, indicating that there are a few nodes with a vital role in the complex network. The analysis of the standard deviation of 210.28290 indicates the dispersion of harmonic centrality values in the studied network. This means there are significant variations in the level of centrality between different nodes. The value of the skewness coefficient, amounting to -1.26434, indicates an asymmetric distribution of harmonic centrality values, with a predominance of nodes with lower centrality.

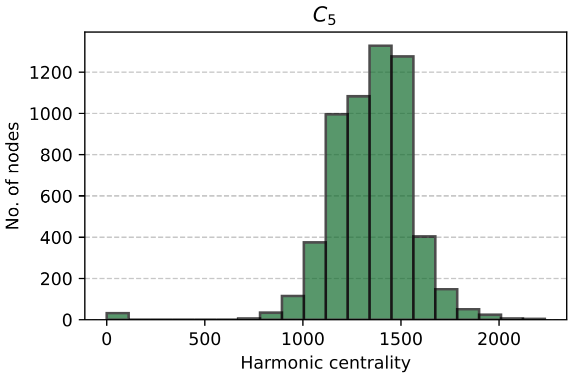


Fig. 8. Distribution of harmonic centrality values for Bitcoin users complex network.

For the studies conducted, min-max normalization was applied to the network centrality metrics. This normalization is intended to scale the values of the metrics within a fixed range to allow comparison and interpretation of the results. The model training process can be sensitive to differences in the scale of metrics values. If no normalization is performed, metrics with a more extensive range of values may significantly impact the training process, and metrics with a smaller range may be ignored. Min-max normalization allows the values of metrics to be adjusted to a range of 0 to 1, eliminating scale differences and ensuring that each metric has an equal impact on the learning process.

B. Artificial expert study: MLP regressor

In this section, a study related to the accuracy of the MLP Regressor model will be conducted. Since the MLP Regressor model will be responsible for evaluating character objects acting as reference preference points of the decision maker, it is necessary to investigate the possibilities related to the model's accuracy concerning the training sample. Therefore, a 10-fold cross-validation was carried out for a given size of the learning set. For the MLP Regressor model, its hyperparameters were adjusted using the `GridSearchCV` class, where the following results were obtained: `max_iter=1000`, `batch_size=64`, `solver= 'lbfgs'`, `hidden_layer_sizes=[1000]`, `activation='relu'`, `alpha=0.0001`.

Using Fig. 9 shows the 10-fold cross-validation on the training set. The dashed line denotes the limiting values obtained for the coefficient of determination obtained for the learning set of the obtained model. In comparison, the solid line with points denotes the average coefficient of determination values. The present results show that as the size of the training set increases, the R^2 values tend to increase. These results indicate that a larger training set size tends to translate into better results for the coefficient of determination. The R^2 values are high for all training set sizes, suggesting that the model can describe the data well.

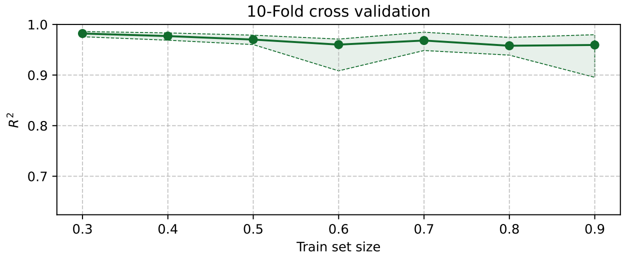


Fig. 9. Relationship between the coefficient of determination and the size of the training set for 10-fold crosvalidation(validated set: train, metric: R^2).

A 10-fold cross-validation on the test set is shown using Fig. 10. In the graph, the dashed line marks the limit values of the coefficient of determination obtained for the test set, and the solid line with points marks the average values of the coefficient of determination. Analysis of the results indicates that the model can significante explain variation in the data. The average values of the coefficient of determination are high, indicating a good fit of the model to the data. The maximum values of the coefficient of determination are also high, which means that in some cases, the model achieves a significant fit for the data.

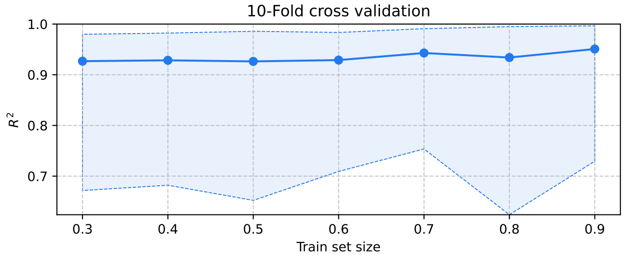


Fig. 10. Relationship between the coefficient of determination and the size of the training set for 10-fold crosvalidation(validated set: test, metric: R^2).

Using Table II, the results of 10-fold cross-validation on the learning set for different sizes of the training set are presented. Analyzing the results for a training set of size 0.3, the lowest R^2 value is 0.9859, indicating a high fit of the model to the data. In contrast, for a training set of 0.9, the lowest R^2 value is 0.8955, indicating a lower model fit to the data than other training set sizes. The average R^2 value ranges from 0.9579 to 0.9820, depending on the size of the training set. The standard deviation is most significant for a training set size of 0.9, suggesting greater variability in results for this set size.

TABLE II
10-FOLD CROSVALIDATION ON THE TRAINING SET FOR ITS PARTICULAR SIZE (VALIDATED SET: TRAIN, METRIC: R^2).

Stats	**Train set size**						
	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Min	0.9757	0.9689	0.9602	0.9083	0.9485	0.9393	0.8955
Mean	0.9820	0.9770	0.9701	0.9601	0.9682	0.9579	0.9594
Max	0.9859	0.9832	0.9788	0.9711	0.9847	0.9743	0.9797
Std	0.0034	0.0047	0.0063	0.0176	0.0088	0.0109	0.0229

The Table III shows the results of 10-fold cross-validation on the learning set for different training set sizes, using the R^2 metric on the test set. The minimum values of R^2 range from 0.6522 to 0.7536, depending on the size of the training set. The average values of R^2 are high, ranging from 0.9263 to 0.9508 for different training set sizes. The maximum values of R^2 range from 0.9798 to 0.9966, indicating a high fit of the model to the data for some cases. The standard deviation of R^2 measures the variability of the results and ranges from 0.0679 to 0.1050.

TABLE III
10-FOLD CROSVALIDATION ON THE TRAINING SET FOR ITS PARTICULAR SIZE (VALIDATED SET: TEST, METRIC: R^2).

Stats	**Train set size**						
	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Min	0.6716	0.6820	0.6522	0.7092	0.7536	0.6239	0.7294
Mean	0.9268	0.9284	0.9263	0.9289	0.9429	0.9339	0.9508
Max	0.9798	0.9821	0.9857	0.9834	0.9909	0.9949	0.9966
Std	0.0883	0.0854	0.0944	0.0768	0.0679	0.1050	0.0754

Based on the results presented above, the MLP Regressor model is stable to perform its function in the present problem as an artificial expert.

C. Study of the stability: MLP-COMET

The study will focus on the similarity of the rankings obtained from the MLP-COMET approach. In order to test the applicability of the MLP model for evaluating the characteristic objects of the COMET method, a study related to the evaluation of 15 selected nodes of a complex network derived from a test set was conducted. In this study, a division of the set into a train set (size: 80%) and a test set (size: 20%) was used to test the MLP-COMET model. The same set was used for the hyperparameters for the MLP model, as shown in the previous study. On the other hand, for the COMET method, 2 characteristic values were selected for each criterion based on the limit values of the normalized criteria.

Table IV shows the selected 15 nodes of the network composed of the test set and their rankings obtained from the MLP-COMET model and the reference expert model. For this study, a high similarity between the two rankings can be observed, as the difference in positions occurs only for four decision variants, i.e., A_1, A_{12}, A_{13} and A_{14}. In addition, the differences in ranking positions are slight and occur mainly at the end of the ranking, which may have a negligible effect on the order.

TABLE IV
SAMPLE NODES OF THE COMPLEX NETWORK SELECTED FROM THE TEST SET AND THEIR CRITERION VALUES (C_1-C_5) AND RANKINGS.

$\mathbf{A_i}$	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$	$\mathbf{C_5}$	**Obt.**	**Ref.**
A_1	0.0003	0.00154	0.34915	0.66337	0.61881	7	8
A_2	0.0001	0.00231	0.32911	0.69508	0.65103	3	3
A_3	0.0009	0.00616	0.34907	0.61402	0.56985	12	12
A_4	0.0001	0.00000	0.34738	0.59391	0.54956	14	14
A_5	0.0001	0.00000	0.34791	0.69165	0.65592	5	5
A_6	0.0001	0.00000	0.34703	0.54707	0.50306	15	15
A_7	0.0003	0.00693	0.37048	0.73539	0.69548	2	2
A_8	0.0003	0.00693	0.35113	0.75381	0.71646	1	1
A_9	0.0006	0.00462	0.34813	0.69907	0.66312	4	4
A_{10}	0.00056	0.00385	0.38118	0.69595	0.65366	6	6
A_{11}	0.00036	0.00077	0.34609	0.60992	0.56510	13	13
A_{12}	0.05522	0.00539	0.34817	0.63432	0.59286	9	10
A_{13}	0.00091	0.00771	0.34825	0.65936	0.61782	8	7
A_{14}	0.00065	0.00616	0.34797	0.64370	0.60106	10	9
A_{15}	0.00036	0.00077	0.34792	0.63731	0.59520	11	11

* where: 'Obt.' - MLP-COMET rank, 'Ref.' - reference model rank.

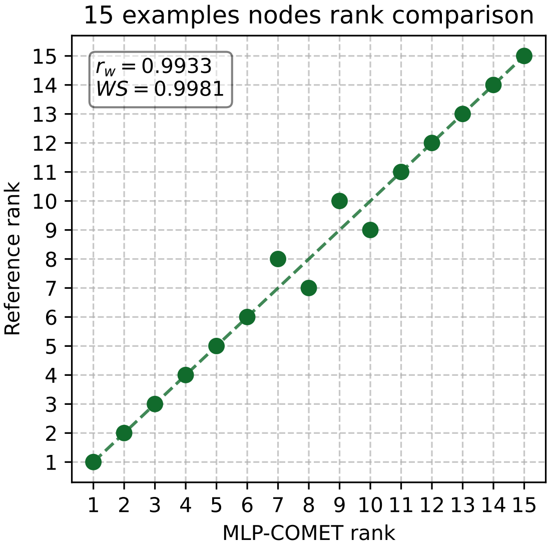


Fig. 11. Relationship between the ranking obtained from the MLP-COMET model and the reference ranking of the expert model for 15 nodes from the test set.

With the help of Fig. 11, the relationship between the ranking obtained from the MLP-COMET model and the reference ranking of the expert model is shown for 15 selected nodes of the composite network. The nodes usually occupy the same positions in both rankings, indicating high similarity. In addition, high similarity can also be observed by analyzing the similarity metrics of the rankings, such as r_w (weighted Spearman correlation coefficient) and WS (ranking similarity coefficient). The values of these metrics were 0.9933 for r_w and 0.9981 for WS, respectively.

After a sample study related to the applicability of the MLP model as an artificial expert for evaluating characteristic objects in the COMET method was performed, it was necessary to investigate the effect of the number of characteristic values on the similarity of rankings depending on the size of the training set. For this study, the r_w measure was used as a metric of ranking similarity for each case studied.

Fig. 12 shows the similarity matrix of node rankings derived from the training set for a given size of the training set and several characteristic values for each criterion. As can be seen from the heatmap, a very stable model was obtained based on samples from the training set. The values of the coefficient r_w were in the range [0.98,1.00], which shows the high similarity of the rankings. The most stable models were obtained for the learning set of 50% and 80% of the initial set and the number of characteristic values 6, 7, 8. In contrast, the slightest similarity was obtained for the learning set of 30% of the initial set.

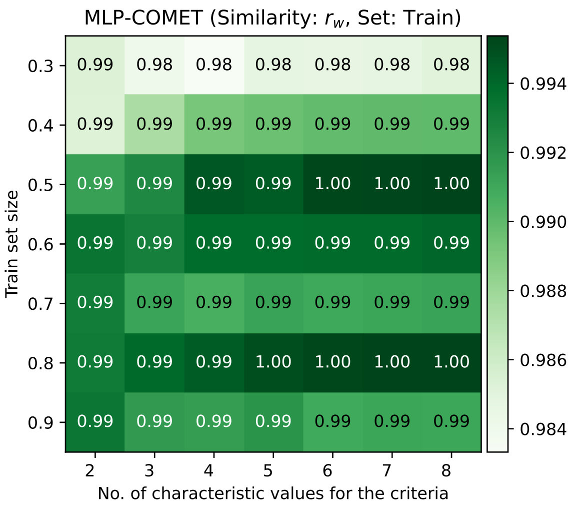


Fig. 12. Ranking similarity matrix for given learning set size and number of characteristic values (MLP-COMET, metric: r_w, studied set: train).

Using Fig. 13, the similarity matrix of node rankings derived from the test set is shown for a given size of the train set and several characteristic values for each criterion. As in the case of the learning set, the high similarity of rankings derived from comparisons of MLP-COMET rankings and reference rankings was shown on the test set. The range of obtained values of the coefficient r_w is [0.98, 1.00], which indicates a high mapping of the rankings of the complex network nodes by the MLP-COMET model. The highest similarity of rankings was obtained for a learning set of size 50% of the base set and 6,7,8 numbers of characteristic values for all considered criteria.

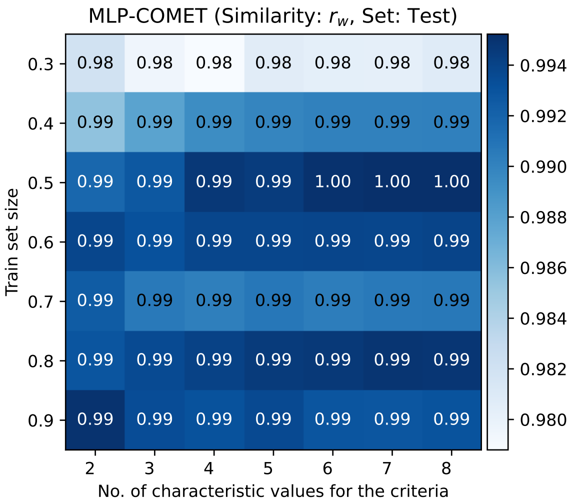


Fig. 13. Ranking similarity matrix for given learning set size and number of characteristic values (MLP-COMET, metric: r_w, studied set: test).

VI. Conclusions and Future works

This paper presents a study on applying the MLP-COMET approach to evaluating the nodes of a complex network of Bitcoin users. The accuracy of the representation of the decision maker's preferences in the decision-making process investigated was using the MLP Regressor model, which achieved high accuracy on both the test set and the training set. Therefore, another study conducted was con the MLP-COMET model, where the results indicate that the MLP-COMET model reproduces well the ranking obtained from the reference expert model, suggesting that it can be an effective tool in evaluating the nodes of a complex network.

The effect of the training sample size and the number of characteristic objects on the similarity of rankings between MLP-COMET and the reference rankings was also investigated. Similar results were obtained, where the similarity measured using the r_w metric for both the train and test sets was in the range [0.98, 1.00]. This demonstrates the tested model's high stability and applicability to the tested complex network to evaluate its nodes.

Future research directions of the proposed approach include other complex networks or multi-criteria decision-making problems. In addition, it would also be appropriate to consider a study related to the consistency of the obtained MEJ matrices of the MLP-COMET approach. Also, more research on its accuracy and consideration of the uncertain environment would need to be conducted. In addition, future research should focus on other cryptocurrencies such as Ethereum.

References

[1] T. C. Silva and L. Zhao, *Machine learning in complex networks.* Springer, 2016.

[2] J. Biamonte, M. Faccin, and M. De Domenico, "Complex networks from classical to quantum," *Communications Physics*, vol. 2, no. 1, p. 53, 2019.

[3] C. W. Lynn, L. Papadopoulos, A. E. Kahn, and D. S. Bassett, "Human information processing in complex networks," *Nature Physics*, vol. 16, no. 9, pp. 965–973, 2020.

[4] B. Yang and J. Li, "Complex network analysis of three-way decision researches," *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 973–987, 2020.

[5] L. Qiu, J. Zhang, and X. Tian, "Ranking influential nodes in complex networks based on local and global structures," *Applied intelligence*, vol. 51, pp. 4394–4407, 2021.

[6] A. S. d. Mata, "Complex networks: a mini-review," *Brazilian Journal of Physics*, vol. 50, pp. 658–672, 2020.

[7] M. Zanin, D. Papo, P. A. Sousa, E. Menasalvas, A. Nicchi, E. Kubik, and S. Boccaletti, "Combining complex networks and data mining: why and how," *Physics Reports*, vol. 635, pp. 1–44, 2016.

[8] H. Xiong, M. Chen, C. Wu, Y. Zhao, and W. Yi, "Research on progress of blockchain consensus algorithm: a review on recent progress of blockchain consensus algorithms," *Future Internet*, vol. 14, no. 2, p. 47, 2022.

[9] S. Ferretti and G. D'Angelo, "On the ethereum blockchain structure: A complex networks theory perspective," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 12, p. e5493, 2020.

[10] B. Tao, I. W.-H. Ho, and H.-N. Dai, "Complex network analysis of the bitcoin blockchain network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2021.

[11] V. Hou Su, S. Sen Gupta, and A. Khan, "Automating ETL and Mining of Ethereum Blockchain Network," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1581–1584, 2022.

[12] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics reports*, vol. 650, pp. 1–63, 2016.

[13] H. Zhao, Z. Li, and R. Zhou, "Risk assessment method combining complex networks with MCDA for multi-facility risk chain and coupling in UUS," *Tunnelling and Underground Space Technology*, vol. 119, p. 104242, 2022.

[14] D. Pamucar, M. Yazdani, M. J. Montero-Simo, R. A. Araque-Padilla, and A. Mohammed, "Multi-criteria decision analysis towards robust service quality measurement," *Expert Systems with Applications*, vol. 170, p. 114508, 2021.

[15] R. Krishankumar and D. Pamucar, "Solving barrier ranking in clean energy adoption: An MCDM approach with q-rung orthopair fuzzy preferences," *International Journal of Knowledge-based and Intelligent Engineering Systems*, no. Preprint, pp. 1–18, 2023.

[16] M. Marttunen, J. Lienert, and V. Belton, "Structuring problems for Multi-Criteria Decision Analysis in practice: A literature review of method combinations," *European journal of operational research*, vol. 263, no. 1, pp. 1–17, 2017.

[17] M. Toslak, A. Ulutaş, S. Ürea, and Ž. Stević, "Selection of peanut butter machine by the integrated PSI-SV-MARCOS method," *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 27, no. 1, pp. 73–86, 2023.

[18] G. Odu, "Weighting methods for multi-criteria decision making technique," *Journal of Applied Sciences and Environmental Management*, vol. 23, no. 8, pp. 1449–1457, 2019.

[19] A. R. Paramanik, S. Sarkar, and B. Sarkar, "OSWMI: An objective-subjective weighted method for minimizing inconsistency in multi-criteria decision making," *Computers & Industrial Engineering*, vol. 169, p. 108138, 2022.

[20] Y. Liu, C. M. Eckert, and C. Earl, "A review of fuzzy AHP methods for decision-making with subjective judgements," *Expert Systems with Applications*, vol. 161, p. 113738, 2020.

[21] K. Rathi and S. Balamohan, "A mathematical model for subjective evaluation of alternatives in fuzzy multi-criteria group decision making using COPRAS method," *International Journal of Fuzzy Systems*, vol. 19, pp. 1290–1299, 2017.

[22] W. Ho and X. Ma, "The state-of-the-art integrations and applications of the analytic hierarchy process," *European Journal of Operational Research*, vol. 267, no. 2, pp. 399–414, 2018.

[23] J. Więckowski, B. Kizielewicz, A. Shekhovtsov, and W. Sałabun, "RANCOM: A novel approach to identifying criteria relevance based on inaccuracy expert judgments," *Engineering Applications of Artificial Intelligence*, vol. 122, p. 106114, 2023.

[24] S. Kheybari, M. Kazemi, and J. Rezaei, "Bioethanol facility location selection using best-worst method," *Applied energy*, vol. 242, pp. 612–623, 2019.

[25] D. Pamučar, Ž. Stević, and S. Sremac, "A New Model for Determining Weight Coefficients of Criteria in MCDM Models: Full Consistency Method (FUCOM)," *Symmetry*, vol. 10, no. 9, p. 393, 2018.

[26] M. R. Patel, M. P. Vashi, and B. V. Bhatt, "SMART-Multi-criteria decision-making technique for use in planning activities," *New Horizons in Civil Engineering (NHCE 2017)*, pp. 1–6, 2017.

[27] K. Yang, N. Zhu, C. Chang, D. Wang, S. Yang, and S. Ma, "A methodological concept for phase change material selection based on multi-criteria decision making (MCDM): A case study," *Energy*, vol. 165, pp. 1085–1096, 2018.

[28] M. Shao, Z. Han, J. Sun, C. Xiao, S. Zhang, and Y. Zhao, "A review of multi-criteria decision making applications for renewable energy site selection," *Renewable Energy*, vol. 157, pp. 377–403, 2020.

[29] A. Karczmarczyk, J. Jankowski, and J. Wątróbski, "Multi-criteria decision support for planning and evaluation of performance of viral marketing campaigns in social networks," *PloS one*, vol. 13, no. 12, p. e0209372, 2018.

[30] F. M. Alexandrescu, L. Pizzol, A. Zabeo, E. Rizzo, E. Giubilato, and A. Critto, "Identifying sustainability communicators in urban regeneration: Integrating individual and relational attributes," *Journal of Cleaner Production*, vol. 173, pp. 278–291, 2018.

[31] A. Karczmarczyk, J. Jankowski, and J. Wątrobski, "Multi-criteria seed selection for targeted influence maximization within social networks," in *Computational Science–ICCS 2021: 21st International Conference, Krakow, Poland, June 16–18, 2021, Proceedings, Part III*, pp. 454–461, Springer, 2021.

[32] A. Muruganantham and M. Gandhi, "Discovering and ranking influential users in social media networks using Multi-Criteria Decision Making (MCDM) Methods," *Indian J Sci Technol*, vol. 9, no. 32, pp. 1–11, 2016.

[33] A. Saxena and S. Iyengar, "Centrality measures in complex networks: A survey," *arXiv preprint arXiv:2011.07190*, 2020.

[34] H. Lai and H. Liao, "A multi-criteria decision making method based on DNMA and CRITIC with linguistic D numbers for blockchain platform evaluation," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104200, 2021.

[35] I. Erol, I. M. Ar, and I. Peker, "Scrutinizing blockchain applicability in sustainable supply chains through an integrated fuzzy multi-criteria decision making framework," *Applied Soft Computing*, vol. 116, p. 108331, 2022.

[36] C. Öztürk and A. Yildizbaşi, "Barriers to implementation of blockchain into supply chain management using an integrated multi-criteria decision-making method: a numerical example," *Soft Computing*, vol. 24, pp. 14771–14789, 2020.

[37] M. Çolak, İ. Kaya, B. Özkan, A. Budak, and A. Karaşan, "A multi-criteria evaluation model based on hesitant fuzzy sets for blockchain technology in supply chain management," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 1, pp. 935–946, 2020.

[38] A. O. Bielinskyi and V. N. Soloviev, "Complex network precursors of crashes and critical events in the cryptocurrency market," in *Ceur workshop proceedings*, vol. 2292, pp. 37–45, 2018.

[39] D. Lin, J. Wu, Q. Yuan, and Z. Zheng, "Modeling and understanding ethereum transaction records via a complex network approach," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 11, pp. 2737–2741, 2020.

[40] L. Serena, S. Ferretti, and G. D'Angelo, "Cryptocurrencies activity as a complex network: Analysis of transactions graphs," *Peer-to-Peer Networking and Applications*, vol. 15, no. 2, pp. 839–853, 2022.

[41] F. Dweiri, S. Kumar, S. A. Khan, and V. Jain, "Designing an integrated AHP based decision support system for supplier selection in automotive industry," *Expert Systems with Applications*, vol. 62, pp. 273–283, 2016.

[42] G. S. Mahendra, "Implementation of the FUCOM-SAW Method on E-Commerce Selection DSS in Indonesia," *Tech-E*, vol. 5, no. 1, pp. 75–85, 2021.

[43] E. P. Sarabi and S. A. Darestani, "Developing a decision support system for logistics service provider selection employing fuzzy MULTIMOORA & BWM in mining equipment manufacturing," *Applied Soft Computing*, vol. 98, p. 106849, 2021.

[44] R. Fahlepi, "Decision support systems employee discipline identification using the simple multi attribute rating technique (SMART) method," *Journal of Applied Engineering and Technological Science (JAETS)*, vol. 1, no. 2, pp. 103–112, 2020.

[45] K. Kądziołka, "The promethee ii method in multi-criteria evaluation of cryptocurrency exchanges," *Economic and Regional Studies/Studia Ekonomiczne i Regionalne*, vol. 14, no. 2, pp. 131–145, 2021.

[46] Z. Aljinović, B. Marasović, and T. Šestanović, "Cryptocurrency portfolio selection—A multicriteria approach," *Mathematics*, vol. 9, no. 14, p. 1677, 2021.

[47] S. Khan, M. Gulistan, N. Kausar, S. Kousar, D. Pamucar, and G. M. Addis, "Analysis of cryptocurrency market by using Q-rung orthopair fuzzy hypersoft set algorithm based on aggregation operators," *Complexity*, vol. 2022, 2022.

[48] G. Ilieva, T. Yankova, I. Radeva, and I. Popchev, "Blockchain software selection as a fuzzy multi-criteria problem," *Computers*, vol. 10, no. 10, p. 120, 2021.

[49] U. Hacioglu, D. Chlyeh, M. K. Yilmaz, E. Tatoglu, and D. Delen, "Crafting performance-based cryptocurrency mining strategies using a hybrid analytics approach," *Decision Support Systems*, vol. 142, p. 113473, 2021.

[50] Y.-C. Yang, W.-S. Shieh, and C.-Y. Lin, "Applying the Fuzzy BWM to Determine the Cryptocurrency Trading System under Uncertain Decision Process," *Axioms*, vol. 12, no. 2, p. 209, 2023.

[51] B. B. Gardas, A. Heidari, N. J. Navimipour, and M. Unal, "A fuzzy-based method for objects selection in blockchain-enabled edge-IoT platforms using a hybrid multi-criteria decision-making model," *Applied Sciences*, vol. 12, no. 17, p. 8906, 2022.

[52] I. M. Ar, I. Erol, I. Peker, A. I. Ozdemir, T. D. Medeni, and I. T. Medeni, "Evaluating the feasibility of blockchain in logistics operations: A decision framework," *Expert Systems with Applications*, vol. 158, p. 113543, 2020.

[53] F. Bloch, M. O. Jackson, and P. Tebaldi, "Centrality measures in networks," *Social Choice and Welfare*, pp. 1–41, 2023.

[54] Y. Du, C. Gao, Y. Hu, S. Mahadevan, and Y. Deng, "A new method of identifying influential nodes in complex networks based on TOPSIS," *Physica A: Statistical Mechanics and its Applications*, vol. 399, pp. 57–69, 2014.

[55] W. Zhang, Q. Zhang, and H. Karimi, "Seeking the important nodes of complex networks in product R&D team based on fuzzy AHP and TOPSIS," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[56] P. Boldi and S. Vigna, "Axioms for centrality," *Internet Mathematics*, vol. 10, no. 3-4, pp. 222–262, 2014.

[57] W. Sałabun, "The Characteristic Objects Method: A New Distance-based Approach to Multicriteria Decision-making Problems," *Journal of Multi-Criteria Decision Analysis*, vol. 22, no. 1-2, pp. 37–50, 2015.

[58] S. Faizi, W. Sałabun, S. Ullah, T. Rashid, and J. Więckowski, "A New Method to Support Decision-Making in an Uncertain Environment Based on Normalized Interval-Valued Triangular Fuzzy Numbers and COMET Technique," *Symmetry*, vol. 12, no. 4, p. 516, 2020.

[59] S. Faizi, W. Sałabun, T. Rashid, S. Zafar, and J. Wątróbski, "Intuitionistic fuzzy sets in multi-criteria group decision making problems using the characteristic objects method," *Symmetry*, vol. 12, no. 9, p. 1382, 2020.

[60] W. Sałabun, A. Karczmarczyk, and J. Wątróbski, "Decision-making using the hesitant fuzzy sets COMET method: An empirical study of the electric city buses selection," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1485–1492, IEEE, 2018.

[61] B. Paradowski and Z. Drążek, "Identification of the decision-making model for selecting an information system," *Procedia Computer Science*, vol. 176, pp. 3802–3809, 2020.

[62] B. Kizielewicz, A. Shekhovtsov, and W. Sałabun, "pymcdm—the universal library for solving multi-criteria decision-making problems," *SoftwareX*, vol. 22, p. 101368, 2023.

[63] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[64] S. Kumar, F. Spezzano, V. Subrahmanian, and C. Faloutsos, "Edge weight prediction in weighted signed networks," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pp. 221–230, IEEE, 2016.

[65] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "Rev2: Fraudulent user prediction in rating platforms," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 333–341, ACM, 2018.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 603–613
DOI: 10.15439/2023F3718
ISSN 2300-5963 ACSIS, Vol. 35

List Of Pareto Optimal Solutions of a Biobjective Shortest Path Problem

Lasko M. Laskov
0000-0003-1833-818
Informatics Department
New Bulgarian University
21 Montevideo Str., 1618 Sofia, Bulgaria
Email: llaskov@nbu.bg

Marin L. Marinov
0009-0003-9544-819X
Informatics Department
New Bulgarian University
21 Montevideo Str., 1618 Sofia, Bulgaria
Email: mlmarinov@nbu.bg

***Abstract*—Many applications in practice involve the search for a shortest path in a network by optimizing two conflicting objective functions. Such problems often are referred to as biobjective optimization problems. Their goal is to find special optimal paths that are *nondominated* and are also known in the specialized literature as to as *Pareto optimal*. While most of the existing methods aim to find the minimum complete set of Pareto optimal paths, we propose an approach that is able to generate a list of all Pareto optimal solutions in a given network.**

The described method solves the biobjective optimization problem in the case in which the first objective function is a linear (MINSUM), while the second objective function is from the "bottleneck" type (MAXMIN). The presented approach is based on two modifications of the Dijkstra's shortest path algorithm that solve the MINSUM and the MAXMIN problems respectively.

We prove the correctness and the computational complexity of the presented algorithms. Also, we provide detailed numerical examples that illustrate their execution.

I. Introduction

PARETO optimal solutions of biobjective (bicriterion) optimization problems are a subject of extensive research in combinatorial optimization and operation research disciplines, and in particular the biobjective shortest path problems [1]. These special type of shortest path problems can arise in numerous applications in practice including transportation problems, computer networking, robot motion, and many others.

The search of biobjective Pareto optimal solutions in a shortest path problem is an optimization problem that is a subject of two objective functions, lets say f and g. The Pareto optimal paths (also called nondominated) are a set of paths, such that for any path α in it, it is impossible to improve either f or g criterion, without getting worse the other [2].

By determining the objective functions f and g, different types of bicriterion path problems can be defined. The first notable work that examines these types of problems is [3] in which Hansen defines ten types of bicriterion path problems, and also introduces their abbreviations. In particular, the MINSUM-MAXMIN problem is solved with an algorithm with a polynomial complexity $O(m^2 \log n)$, where n is the number of vertices of the network, and m is the number of directed edges. In the MINSUM-MAXMIN problem the first objective function is a liner one, while the second objective function is from the *bottleneck* type.

In the literature actually there are quite few works that focus on the solution of the MINSUM-MAXMIN problem. One of them is [4] which proposes an extension of the Martin's algorithm [5] for a multiobjective shortest path problem with a MAXMIN objective function. Most of the methods that can be found in the literature focus on the combination of two linear functions, for example [2], [6], [7], [8] solve the MINSUM-MINSUM bicriteria path problem. Other works focus on MINMAX-MINSUM problem (see [9] and [10]), however in their case the authors do not consider the Pareto optimality, rather they aim to define a singe objective function by combining the MINMAX and MINSUM criteria.

Another subject that is rarely considered in the literature is the calculation of all Pareto optimal paths from a source vertex v_0 and a destination (target, terminal) vertex v_t. Most of the existing algorithms aim to find the minimal complete set of Pareto optimal paths, which means that from each equivalent set of Pareto optimal paths a single path is discovered (see [3]). In [11] the authors look for a set of alternative Pareto optimal paths in a method that solves a concrete practical problem for routing of Hazardous materials and propose shortest path algorithm on a network with two criteria: one that corresponds to road length, and the other that corresponds to a risk measure.

The work [2] is one of the few in the literature that pays special attention to the calculation of all Pareto optimal paths. The authors propose two algorithms depending whether the paths may contain or may not contain loops, both of them based on k shortest paths algorithms in graphs. However, as mentioned above, in this work the two objective functions are linear, and the algorithms do not cover the case in which one of the functions is a bottleneck function.

The exact methods that are present in the literature, are generally classified into *labeling* and *ranking paths* algorithms [12]. Labeling algorithms can be split into two categories: *label setting* [3], [5], [8]; and *label correcting* [6], [7], [11]. In the category *ranking paths* we can classify methods that are based on the k shortest paths algorithms, for example [2].

The other major branch of methods are based on heuristics approaches. For example, in [13] the authors propose a solution to the biobjective shortest path problem that is based on a genetic algorithm for which the authors report to find the Pareto optimal set in 77% of the instances. Also, the probabilistic technique ant colony optimization (ACO) and its modifications are adopted in the solution of various complex combinatorial problems (see for example [14]). Heuristic methods often are adopted in various practical problems, like electric vehicle shortest path problem [15].

In this paper we propose an exact method for calculation of all Pareto optimal paths for the MINSUM-MAXMIN problem in a network. We define two helper problems, MINSUM list and MAXMIN list, and we provide two algorithms that solve them, which are based on generalization of the Dijkstra's algorithm [16]. We use the solutions of the two helper problems to formulate the method for general problem solution. The correctness of all algorithms is proved, and their computational complexity is shown. Also, we illustrate the algorithms with detailed examples that show their execution.

The paper is organized as follows. In Sec. II we introduce the notations and problems formulation. In Sec. III we describe the two algorithms that solve the two helper problems. In Sec. IV we show how the Pareto optimal solutions list is constructed based on the solution of the two helper problems. Finally, Sec. V contains conclusions and discussions.

II. Problem formulation

A. Notations

Let $G = (V, E)$ is a directed graph (digraph) with $n = |V|$ number of vertices and $m = |E|$ number of directed edges. Without loss of generality we will assume that $V = \{1, 2, \dots, n\}$ and $E \subseteq V^2$.

We define the following two functions on the set of edges of the digraph, $f : E \to \mathbb{R}_+$ and $g : E \to \overline{\mathbb{R}}_+$. The function f assigns to each edge $(i, j) \in E$ the positive number $f(i, j)$, which we call the *length* of the edge $e = (i, j)$. The function g assigns to each edge $(i, j) \in E$:

$$g(i,j) = \begin{cases} +\infty, & \text{if} \quad (i,j) \in E \text{ has no restriction} \\ g_{ij} > 0, & \text{if} \quad (i,j) \in E \text{ has a restriction} \end{cases}$$

For convenience, we will call the value $g(i, j)$ the *capacity* of the edge $e = (i, j)$.

The digraph G together with the functions f and g defines the network $G = (V, E, f, g)$ (see [17]). The network is represented by the adjacency list of the outgoing neighbors [18] that is augmented with the length and capacity of the edges in the following way:

$$Adj = \{Adj(1), \dots, Adj(i), \dots, Adj(n)\}, \tag{1}$$

where $Adj(i) = \{(j, f(i, j), g(i, j)) : (i, j) \in E\}, \forall i \in V$. In this way, if $q = Adj(i, k)$ for some $i \in V$, and a positive integer k, $q(1)$ will denote the k-th outgoing neighbor of the vertex i, $q(2)$ will denote the length of the edge $(i, q(1))$, and $q(3)$ will denote the capacity of the edge. To denote the adjacency list of outgoing neighbors of a given network G, we will use the notation $G.Adj$.

Path in the network G is the finite sequence of the type

$$v_0, e_1, v_1, e_2, \dots, v_{(t-1)}, e_t, v_t, \tag{2}$$

where $v_j \in V, \forall j \in \{0, 1, \dots, t\}$ are distinct vertices, and e_i is an edge with starting vertex $v_{(i-1)}$ and ending vertex v_i, that belongs to E for all $i \in \{1, 2, \dots, t\}$. The path consists of $(t + 1)$ vertices and t edges, the vertex v_0 is the *source* of the path, and the vertex v_t is the *destination* of the path.

The path (2) with a source v_0 and a destination v_t connects v_0 with v_t and is called a (v_1, v_t)-path, which we will denote with an ordered sequence α of vertices:

$$\alpha = (v_0, v_1, \dots, v_t). \tag{3}$$

For each path $\alpha = (v_0, v_1, \dots, v_t)$ we define two functions:

$$x(\alpha) = \sum_{j=1}^{t} f(v_{j-1}, v_j) \tag{4}$$

$$y(\alpha) = \min_{j \in \{1,\dots,t\}} \{g(v_{j-1}, v_j)\} \tag{5}$$

We will call the number $x(\alpha)$ the *length* of the path, and the number $y(\alpha)$ the *capacity* of the path α. Both functions $x(\alpha)$ and $y(\alpha)$ define the objective functions of the problems that we will discuss.

We denote all $(1, j)$-paths with the shorter W_j. Then, we will call the number

$$r_j = \min_{\alpha \in W_j} \{x(\alpha)\} \tag{6}$$

a *distance* between the vertex 1 and the vertex j. Also, we will call the number

$$c_j = \max_{\alpha \in W_j} \{y(\alpha)\} \tag{7}$$

the *capacity* of the vertex j.

For each W_j the term *Pareto optimal path* is defined as follows.

Definition 1. *We call the path $\alpha \in W_j$* ***Pareto optimal*** *when there does not exist another path $\beta \in W_j$, for which any of the following two conditions is fulfilled:*

- *$x(\beta) < x(\alpha)$ and $y(\beta) \geq y(\alpha)$;*
- *$x(\beta) \leq x(\alpha)$ and $y(\beta) > y(\alpha)$.*

We say that α and β are *equivalent* ($\alpha \sim \beta$), when $x(\alpha) = x(\beta)$ and $y(\alpha) = y(\beta)$.

The path β is *dominated* by the path α, when $x(\alpha) < x(\beta)$ and $y(\alpha) \geq y(\beta)$ or $x(\alpha) \leq x(\beta)$ and $y(\alpha) > y(\beta)$.

Besides that, we will denote the distance from vertex 1 to any vertex v with $r(v)$.

B. Problems formulation

Based on the above definitions, we formulate the main problem considered in this paper:

Problem 1 (List of Pareto optimal solutions)**.** *Compute a list of all Pareto optimal solutions for W_n.*

To solve the List of Pareto optimal solutions problem, we will use the solutions of the following two helper problems.

The solution of the first helper problem requires the definition of a function $minsum(G.Adj)$ that computes a *list of all shortest paths* in the network.

Problem 2 (MINSUM list)**.** *Compute a list of all $(1, n)$-paths with minimal length, given by:*

$$S_x = \{\alpha \in W_n : x(\alpha) \leq x(\beta), \forall \beta \in W_n\}. \tag{8}$$

The solution of the second helper problem requires the definition of a function $maxmin(G.Adj)$, and it is a version of the first helper problem that computes a *list of all maximum capacity paths* in the network.

Problem 3 (MAXMIN list)**.** *Compute a list of all $(1, n)$-paths with maximal capacity, given by:*

$$S_y = \{\alpha \in W_n : y(\alpha) \geq y(\beta), \forall \beta \in W_n\}. \tag{9}$$

III. Solution of the two helper problems

To solve the two helper problems we propose two modifications of the Dijkstra's algorithm [16], in which the results hold as well in the case in which the source vertex is selected $i_0 \neq 1$. In both modifications we assume that the network $G = (V, E, f, g)$ is defined using the adjacency list of the outgoing neighbors.

In the computer program implementation of the modified versions of Dijkstra's algorithm we apply the Fibonacci heap data structure [19] for all priority queue operations. Even though the relative complexity of its implementation, this advanced data structure introduces a significant speedup of the algorithm to $O(n \log n + m)$, which is proved based on amortized analysis [18].

A. List of all shortest paths

We will solve the problem of computing of a list of all $(1, n)$- shortest paths by finding the subnetwork $\widehat{G} = (V, \widehat{E}, f, g)$ of the shortest paths.

Definition 2. *We will say that $\widehat{G} = (V, \widehat{E}, f, g)$ is a* ***subnetwork of the shortest paths*** *in the network $G = (V, E, f, g)$, if the following two properties hold:*

1) *Every $(1, n)$- shortest path in G is also a $(1, n)$-path in $\widehat{G}$.*
2) *Every $(1, n)$-path in $\widehat{G}$ is a $(1, n)$- shortest path in G.*

The solution of the MINSUM list helper problem is given by the definition of a function $minsum(G.Adj)$ (Alg. 2), which for a given network G calculates the adjacency list of the outgoing neighbors of the shortest paths subnetwork $\widehat{G}$. The implementation of the $minsum(G.Adj)$ function is based on a modification of the Dijkstra's algorithm [16], as follows.

The algorithm splits the set of network vertices in into subsets. The first subset V_0 denotes the vertices that are not yet traversed by the algorithm. The second subset $U = V \setminus V_0$ stores the traversed vertices. The procedure that traverses the network guarantees that

$$r(v) \geq r(u), \tag{10}$$

for each vertex $v \in V_0$ and each vertex $u \in U$. Initially, $V_0 = V$ and $U = \varnothing$. In each of n consecutive iterations of execution the function $minsum(G.Adj)$ a selected vertex is transferred from V_0 into U.

Algorithm 1 Function $relaxS(u, v, d, pr)$

```
Input: vertices $u$, $v$, and vectors $d$, $pr$
Output: vectors $d$, $pr$
  $q \in G.Adj(u)$, such that $q(1) = v$
  $r \leftarrow d(u) + q(2)$
  if $d(v) > r$ then
      $d(v) \leftarrow r$
      $pr(v) \leftarrow \{u\}$
  else if $d(v) = r$ then
      $pushback(pr(v), u)$
  end if
  return $\{d, pr\}$
```

The algorithm uses two vectors d and pr, both of them with n components. Initially, $d(1) = 0$, and all other components of d are ∞. The initial values of pr are equal to the empty set $\varnothing$. After the completion of the algorithm, d will store the distances from the source vertex to each of the other vertices in the network, in other words $d(j) = r(j), \forall j \in V$, and pr will store the adjacency list of the ingoing neighbors of the digraph $(V, \widehat{E})$. The last step of the $minsum(G.Adj)$ function composes the adjacency list of the outgoing neighbors of the shortest paths subnetwork $\widehat{G}$.

In the proposed variant of the Dijkstra's algorithm that solves Prob. 2, the function that implements the relaxation procedure is modified. We define the function $relaxS(u, v, d, pr)$ that performs relaxation of the edge (u, v) by changing the current state of d and pr, as it is given in Alg. 1.

We use two more helper functions in our modified Dijkstra's implementation of the $minsum(G.Adj)$ procedure: $extract(V_0)$ and $outadj(pr, G.Adj)$.

The input of $extract(V_0)$ is the subset $V_0 \subset V$, and the output is $\{v_1, V_1\}$, where $v_1 \in V_0$, $V_1 = V_0 \backslash \{v_1\}$ and $d(v_1) = \min\limits_{v \in V_0}\{d(v)\}$.

The purpose of the $outadj(pr, G.Adj)$ function is to build the adjacency list of the outgoing neighbors of the network $\widehat{G}$ out of the digraph presented by the adjacency list of the ingoing neighbors pr. The function composes each edge $e = (i, j)$ from the digraph given by pr, and takes the corresponding edge length $f(i, j)$ and capacity $g(i, j)$ from the outgoing adjacency list of the original input network $G.Adj$.

Proposition 1. *The function $minsum(G.Adj)$ is correctly defined.*

Proof: The proof of the correctness of the function $minsum(G.Adj)$ is analogous to the proof of the Dijkstra's

Algorithm 2 Function $minsum(G.Adj)$

```
Input: $G.Adj$
Output: distance $d_0$ to vertex $n$, and $\widehat{G}.Adj$
  $V_0 \leftarrow \{1, 2, \dots, n\}$
  $d \leftarrow (0, \infty, \dots, \infty)$
  while $V_0 \neq \varnothing$ do
      $\{u, V_0\} \leftarrow extract(V_0)$
      for each $q \in G.Adj(u)$ do
          $\{d, pr\} \leftarrow relaxS(u, q(1), d, pr)$
      end for
  end while
  $\widehat{G}.Adj \leftarrow outadj(pr, G.Adj)$
  return $\{d(n), \widehat{G}.Adj\}$
```

algorithm (see [18]). We will only note that after the each iteration of the **while** loop the following properties hold for each vertex $v \in V_0$:

1) $d(v) \geq r(v)$;
2) $pr(v)$ is the set of all vertices $u \in U$ for which there exists a path $\alpha = (i_0, i_1, \dots, i_k, u, v)$ with length $x(\alpha) = d(v)$.

Besides that, when $v_1 \in V_0$ and $d(v_1) = \min\limits_{v \in V_0}\{d(v)\}$, then following the proof of the Dijkstra's algorithm, we find out that $d(v_1) = r(v_1)$. Then in the set V_0 does not exist a vertex that is the one before the last one in a (i_0, v_1)-path with length $d(v_1)$. In fact, if we assume that for the path $\alpha = (i_0, i_1, \dots, i_k, v_1)$ holds $i_k \in V_0$ and $x(\alpha) = d(v_1)$, we reach contradiction because

$$r(v_1) = d(v_1) = \sum_{s=1}^{k} f(i_{s-1}, i_s) + f(i_k, v_1) \geq r(i_k) + f(i_k, v)$$

and using the inequality (10), we get $r(v_1) \geq r(v_1) + f(i_k, v)$.

The **while** loop of the algorithm stops when $V_0 = \varnothing$, and then $d(j) = r(j), \forall j \in V$.

We define the network $\widehat{G} = (V, \widehat{E}, f, g)$, where $\widehat{E} = \{(i, j) \in E : i \in pr(j)\}$. In this way, $\widehat{G}$ is the shortest paths subnetwork of the network G, and pr is the adjacency list of the ingoing neighbors of the digraph $(V, \widehat{E})$, which is verified directly, by using the fact that $pr(v)$ is the set of those vertices $u \in V$ for which there exists a path $\alpha = (i_0, i_1, \dots, u, v)$ such that $x(\alpha) = r(v)$. ■

It is clear that using the adjacency list $\widehat{G}.Adj$, it is easy to compose the list of all $(1, n)$- shortest paths. The following example illustrates this observation.

Example 1. *Let G_1 is the network given on the Figure 1. We will find all $(1, 5)$- shortest paths in G_1.*

Solution: The outgoing adjacency list of G_1 is given by:

$$\begin{aligned} G_1.Adj = \{&\{(2,2,4),(3,5,3)\},\\ &\{(3,3,5),(4,6,4),(5,5,3)\},\\ &\{(4,3,6),(5,1,1)\},\{(5,1,7)\},\{\}\}. \end{aligned} \tag{11}$$

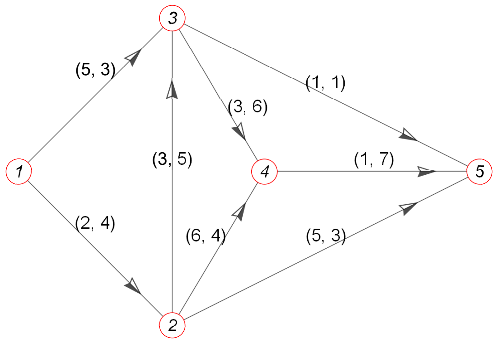


Fig. 1. Example network G_1 composed by five vertices with length and capacity of each edge given next to it

The function $minsum(G_1.Adj)$ produces the following result:

$$\{d_0, \widehat{G}_1.Adj\} \leftarrow minsum(G_1.Adj), \tag{12}$$

where $\widehat{G}_1.Adj = \{\{(2,2,4),(3,5,3)\}, \{(3,3,5),(4,6,4)\}, \{(4,3,6),(5,1,1)\}, \{\}, \{\}\}$ and $d_0 = 6$.

The network with outgoing adjacency list $\widehat{G}_1.Adj$ has exactly two $(1, 5)$-paths

$$\alpha = (1, 3, 5) \text{ and } \beta = (1, 2, 3, 5).$$

Obviously, $x(\alpha) = x(\beta) = 6 = d_0$.

Proposition 2. *The function $minsum(G.Adj)$ has computational complexity $O(n \log n + m)$.*

The proof follows directly from the computational complexity of the Dijkstra's algorithm, in the case in which the Fibonacci heap data structure is used for the implementation of the priority queue operations (refer to [18]).

B. List of all maximum capacity paths

We denote the capacity of each vertex of the network $v \in V$ with $c(v)$.

Definition 3. *We say that $\widetilde{G} = (V, \widetilde{E})$ is a **maximal capacity** digraph of the network $G = (V, E, f, g)$, if the following two properties hold:*

1) *Every $(1, n)$- maximal capacity path in G is also a $(1, n)$-path in $\widetilde{G}$.*
2) *Every $(1, n)$-path in $\widetilde{G}$ is also a $(1, n)$- maximal capacity path in G.*

In this section we will define the function $maxmin(G.Adj)$ which calculates the adjacency list of the outgoing neighbors $\widetilde{G}.Adj$ of the maximal capacity graph $\widetilde{G} = (V, \widetilde{E})$. For its implementation, first we will formulate the function $capacity(G.Adj)$ (Alg. 3) which calculates the capacity of a vertex n in the network G, again following the Dijkstra's algorithm.

We denote with d a vector of n components that initially has all its elements equal to $-\infty$, except the first element $d(1) = \infty$. During the calculations of the function $capacity(G.Adj)$ the components of d are growing, and when the function

Algorithm 3 Function *capacity*$(G.Adj)$

```
Input: $G.Adj$
Output: the capacity $c_n$ of the vertex $n$
  $V_0 \leftarrow \{1, 2, \dots, n\}$
  $d \leftarrow (\infty, -\infty, \dots, -\infty)$
  while $V_0 \neq 0$ do
      $\{u, V_0\} \leftarrow extract(V_0, d)$
      for each $q \in G.Adj(u)$ do
          $relax(u, q(1), G.Adj, d)$
      end for
  end while
  return $d(n)$
```

Algorithm 4 Function $maxmin(G.Adj)$

```
Input: $G.Adj$
Output: the capacity $c$ of the vertex $n$ and $\widetilde{G}.Adj$
   $c \leftarrow capacity(G.Adj)$
2: for $i \leftarrow 1$ to $n$ do
       for each $q \in Adj(i)$ do
4:         if $q(3) \geq c$ then
               $pushback(\widetilde{G}.Adj, q(1))$
6:         end if
       end for
8: end for
   return $\{c, \widetilde{G}.Adj\}$
```

completes, the j-th component of d stores the capacity of the vertex j, or in other words $d(j) = c(j)$. Note that by definition $c(1) = \infty$.

We will use the following helper functions in the implementation of *capacity*$(G.Adj)$.

The function $extract(V_0, d)$ for an arbitrary subset $V_0 = \{j_1, j_2, \dots, j_k\} \subseteq V$ and using the vector d, calculates the pair $\{j_s, V_1\}$, where

$$j_s \in V_0,\ d(j_s) = \max_{j_p \in V_0}\{d(j_p)\} \text{ and } V_1 = V_0 \setminus \{j_s\}. \quad (13)$$

The function $relax(i, j, G.Adj, d)$ for each two vertices i and j, and based on $G.Adj$ and d, performs the following calculations:

1) Defines $r = \min\{d(i), g(i, j)\}$.
2) If $d(j) < r$, it sets $d(j) = r$.

Proposition 3. *The function capacity*$(G.Adj)$ *is correctly defined.*

Proof: Let $1 \leq k < n$. We assume that after k iterations of the **while** loop (see Alg. 3), from the initial set $V_0 = V = \{1, 2, \dots, n\}$ are excluded k number of vertices u_s, and for the resulting set $V_0 = V \setminus \{u_1, \dots, u_k\}$ the following properties hold:

1) For each $u \in U = \{u_1, \dots, u_k\}$ it holds that $d(u) = c(u)$.
2) For each $u \in U$ and each $v \in V_0$ it hods that $d(u) \geq d(v)$.
3) Let $v_k \in V_0$ and $d(v_k) \geq d(v)$ for each $v \in V_0$. Then $d(v_k) = c(v_k)$.

The above three properties are true for $k = 1$.

After the $(k+1)$-st iteration of the **while** loop, the vertex v_k is excluded from v_0 and we get $V_0' = V_0 \setminus \{v_k\}$. We will prove that the three properties hold for the set V_0'.

From the definition of v_k and the fact that the three properties hold for V_0, it follows directly that properties 1) and 2) are fulfilled for V_0'.

Now, let $v_{k+1} \in V_0'$ and $d(v_{k+1}) \geq d(v)$ for each $v \in V_0'$. We denote $U' = U \cup \{v_k\} = V \setminus V_0'$. Also, let α be an arbitrary $(1, v_{k+1})$-path and

$$\alpha = (1, u_1, \dots, u_s, v_{j_1}, \dots, v_{j_r}, v_{k+1}),$$

where $u_i \in U', \forall i \in \{1, \dots, s\}$ and $v_{j_1} \in V_0'$.

From the definition of $d(v_{j_1})$ it follows that for the capacity $y(\beta)$ of the path $\beta = (1, u_1, \dots, u_s, v_{j_1})$ it is fulfilled that $y(\beta) \leq d(j_1)$. Then, for that capacity $y(\alpha)$ of the path α we have that

$$y(\alpha) \leq y(\beta) \leq d(v_{j_1}) \leq d(v_{k+1}),$$

which proves that $d(v_{k+1}) = c(v_{k+1})$. ■

Having the function *capacity*$(G.Adj)$, we can define the function $maxmin(G.Adj)$, given in Alg. 4.

Proposition 4. *The function* $maxmin(G.Adj)$ *is correctly defined.*

Proof: Since the function *capcity*$(G.Adj)$ that is triggered on the first line of the Alg. 4 is correct, it follows that c is the capacity of the vertex n. The outer **for** loop on the line 2 defines that adjacency list $\widetilde{G}.Adj$ of the graph $\widetilde{G} = (V, \widetilde{E})$, where $\widetilde{E} = \{(i, j) \in E : g(i, j) \geq c\}$. Now we will prove that the graph $\widetilde{G} = (V, \widetilde{E})$ is the maximum capacity graph.

If α is a $(1, n)$-path in the network G with capacity $y(\alpha) = c$, then for each edge (i, j) of α it holds that $g(i, j) \geq c$, and hence, $(i, j) \in \widetilde{E}$. This shows that α is a $(1, n)$-path of $\widetilde{G}$.

Now, let β is a $(1, n)$-path in $\widetilde{G}$. This means that β is a $(1, n)$-path in the network G, and for each its edge (i, j) it is fulfilled that $g(i, j) \geq c$. From here it follows that $y(\beta) \geq c$. However, since c is the capacity of the vertex n, then $y(\beta) = c$. ■

Proposition 5. *The computational complexity of the function* $maxmin(G.Adj)$ *is* $O(n \log n + m)$.

The proof follows directly from the complexity of the Dijkstra's algorithm implemented with Fibonacci heap, since the function *capacity*$(G.Adj)$ repeats exactly it steps.

We will introduce the function $list(m, \widetilde{G}.Adj)$ that will help us to clarify the following examples. The argument of the function m is a natural number or ∞. The function maps a list S of $(1, n)$-paths in $\widetilde{G}$ that satisfies the following properties:

1) If $m = \infty$, the list S contains all $(1, n)$-paths of $\widetilde{G}$.
2) If m is a natural number, S contains all $(1, n)$-paths of $\widetilde{G}$ if their number is not greater than m. Otherwise, S contains m number of all $(1, n)$-paths of $\widetilde{G}$.

Example 2. *For the network G_1 (see Fig. 1) we will compose the list S_y (9) of all $(1,5)$-paths with maximal capacity.*

Solution: The solution is composed by the following two steps:

1) $\{c, \widetilde{G}_1.Adj\} \leftarrow maxmin(G_1.Adj)$.
2) $S_y \leftarrow list(\infty, \widetilde{G}_1.Adj)$.

On the first step, using the function $maxmin(G_1.Adj)$, we calculate that:

$$c = 4 \text{ and } \widetilde{G}_1.Adj = \{\{2\},\{3,4\},\{4\},\{5\},\{\}\}.$$

On the second step, the function $list(\infty, \widetilde{G}_1.Adj)$ calculates the list

$$S_y = \{(1,2,4,5),(1,2,3,4,5)\}.$$

Then, it is directly verified that

$$y((1,2,4,5)) = y((1,2,3,4,5)) = 4 = c.$$

IV. Pareto optimal solutions set

We denote with P the set of Pareto optimal paths in the network $G = (V, V, f, g)$. It is clear that

$$P = \bigcup_{i=1}^{k_0} P_i, \tag{14}$$

where P_i are the classes of Pareto equivalent paths.

We will find the set P by composing a list Q of all classes of Pareto equivalent paths P_i. We will compose the list Q using the following procedure, which we will call *Pareto Optimal Paths (POP)*:

1) Set $W = W_n$.
2) Calculate $d = \min\limits_{\beta\in W}\{x(\beta)\}$ and define $X = \{\alpha \in W : x(\alpha) = d\}$.
3) Calculate $c = \max\limits_{\beta\in X}\{y(\beta)\}$ and define $P_0 = \{\alpha \in X : y(\alpha) = c\}$. Store P_0 into the list Q.
4) Define $Y = \{\beta \in W : y(\beta) > c\}$.
5) Define $Z = W \setminus (Y \cup P_0)$.
6) If $Y = \varnothing$, then end. Otherwise, set $W = Y$ and go back to step 2.

Lemma 1. *The POP procedure correctly composes the list Q.*

Proof: We will prove the correctness of the POP procedure by induction.

Base case. The *first iteration* of POP defines:

$$d_1 = \min_{\beta\in W_n}\{x(\beta)\}, X_1 = \{\alpha \in W_n : x(\alpha) = d_1\},$$
$$c_1 = \max_{\beta\in X_1}\{y(\beta)\}, P_1 = \{\alpha \in X_1 : y(\alpha) = c_1\}, Q = \{P_1\},$$
$$Y_1 = \{\beta \in W_n : y(\beta) > c_1\}, Z_1 = W_n \setminus (Y_1 \cup P_1).$$

The following properties hold:

1) From the definitions of the sets P_1, Y_1 and Z_1 it follows:

$$W_n = Z_1 \cup P_1 \cup Y_1, Z_1 \cap P_1 = \varnothing,$$
$$Z_1 \cap Y_1 = \varnothing, Y_1 \cap P_1 = \varnothing.$$

2) If $\alpha \in P_1$, then by the definition of P_1 the equalities $x(\alpha) = d_1$ and $y(\alpha) = c_1$ hold. Hence, the paths that belong to P_1 are equivalent.
3) If $Z_1 \neq \varnothing$ and $\beta \in Z_1$, then from step 5 it follows that one of the following two statements is fulfilled:
   a) $y(\beta) < c_1$ and $x(\beta) \geq d_1$, or
   b) $y(\beta) = c_1$ and $x(\beta) > d_1$.

   Hence, β is dominated by each $\alpha \in P_1$.

If $Y_1 = \varnothing$, then $W_n = P_1 \cup Z_1$ and P_1 is the set of Pareto optimal paths. This means that in (14) the constant $k_0 = 1$, and the calculations of the POP procedure will stop.

If $Y_1 \neq \varnothing$, then the constant $k_0 > 1$. In this case, for each $\beta \in Y_1$ the inequalities are fulfilled:

$$c_1 < y(\beta) \text{ and } d_1 < x(\beta). \tag{15}$$

As a result, also in this case P_1 is a set of equivalent Pareto optimal paths α, and it is correctly included in the list Q. Here, we set $W = Y_1$, and we go back to step 2 of the second iteration of the procedure.

The *second iteration* of the POP procedure defines:

$$d_2 = \min_{\beta\in Y_1}\{x(\beta)\}, X_2 = \{\alpha \in Y_1 : x(\alpha) = d_2\},$$
$$c_2 = \max_{\beta\in X_2}\{y(\beta)\}, P_2 = \{\alpha \in X_2 : y(\alpha) = c_2\},$$
$$Q = \{P_1, P_2\}, Y_2 = \{\beta \in Y_1 : y(\beta) > c_2\},$$
$$Z_2 = Y_1 \setminus (Y_2 \cup P_2).$$

In analogy with the first iteration of POP, the following properties are proved.

1) From the definitions of the sets P_2, Y_2 and Z_2 it follows:

$$\begin{aligned} Y_1 = Z_2 \cup P_2 \cup Y_2, Z_2 \cap P_2 = \varnothing, \\ Z_2 \cap Y_2 = \varnothing, Y_2 \cap P_2 = \varnothing \end{aligned} \tag{16}$$

2) If $\alpha \in P_2$, then by the definition of P_2 the equalities $x(\alpha) = d_2$ and $y(\alpha) = c_2$ hold, and hence the paths that belong to P_2 are equivalent. Besides that, from (15) follows that $d_1 < d_2$ and $c_1 < c_2$.

If $Z_2 \neq \varnothing$, then for each path $\beta \in Z_2$ holds that $x(\beta) \geq d_2$ because $\beta \in Y_1$. Besides that, $y(\beta) \leq c_2$, because $\beta \notin Y_2$. Then, for β one of the following statements hold:

- $y(\beta) < c_2$ and $x(\beta) \geq d_2$, or
- $y(\beta) = c_2$ and $x(\beta) > d_2$, because $\beta \notin P_2$.

Therefore, each path $\beta \in Z_2$ is dominated by any path from P_2.

Let $\alpha \in P_2$, and β is such a $(1,n)$-path, so that $\beta \notin Y_1$. Then $y(\beta) \leq c_1 < y(\alpha)$. The following two cases are possible:

- $x(\beta) \geq d_2 = x(\alpha)$. In this case α dominates β.
- $x(\beta) < d_2 = x(\alpha)$. In this case α and β cannot be compared.

That proves that if $Y_2 = \varnothing$, then the elements of P_2 are Pareto optimal and the equation (14) has the form $P = P_1 \cup P_2$, in other words, $k_0 = 2$. From here it follows that P_2 is correctly included in the list Q, and the termination of the computation of POP procedure is correct.

If $Y_2 \neq \varnothing$, then $k_0 > 2$. In this case, for each $\beta \in Y_2$ the following inequalities are fulfilled.

$$c_2 < y(\beta) \text{ and } d_2 < x(\beta) \tag{17}$$

The first inequality follows from the definition of Y_2, and the second one – from the definition of d_2 and c_2.

Therefore, also in the case in which $Y_2 \neq \varnothing$, the set P_2 is a set of Pareto optimal paths, and it is correctly included in the list Q. In this case, we set $W = Y_2$ and we go back to step 2 in the procedure to start its third iteration.

It is clear that in the third iteration of the procedure $W_n = Z_1 \cup P_1 \cup Z_2 \cup P_2 \cup Y_2$.

Inductive step. We assume that after $k \geq 2$ iterations of the POP procedure, the following components are defined.

- The sets P_i, Z_i, Y_k, $i \in \{1, 2, \dots, k\}$, that have no common elements.
- The numbers d_i and c_i $i \in \{1, 2, \dots, k\}$, for which the following five properties are fulfilled.
  1) $W_n = \bigcup\limits_{i=1}^{k} (Z_i \cup P_i) \bigcup Y_k$.
  2) $P_1, P_2, \dots, P_k$ are sets of equivalent Pareto optimal paths, for which
     a) $d_i = x(\alpha)$ and $c_i = y(\alpha)$, $\forall \alpha \in P_i$ and $\forall i \in \{1, 2, \dots, k\}$;
     b) $d_i < d_{i+1}$ and $c_i < c_{i+1}$, $\forall i \in \{1, 2, \dots, (k-1)\}$.
  3) Every path $\beta \in Z_i$ is dominated by every path $\alpha \in P_i$.
  4) For every $\beta \in Y_k$ the inequalities hold:
  $$c_k < y(\beta) \text{ and } d_k < x(\beta). \tag{18}$$
  5) $Q = \{P_1, P_2, \dots, P_k\}$.

  These five properties follow directly from the proof of the *second iteration* of the procedure.

It is clear that if $Y_k = \varnothing$, then $W_n = \bigcup\limits_{i=1}^{k} (Z_i \cup P_i)$ and the list Q is correctly composed.

If $Y_k \neq \varnothing$, we implement the $(k+1)$-st iteration of the POP procedure. Using steps from 2 to 5, we define:

$$d_{k+1} = \min_{\beta \in Y_k} \{x(\beta)\}, X_{k+1} = \{\alpha \in Y_k : x(\alpha) = d_{k+1}\},$$
$$c_{k+1} = \max_{\beta \in X_{k+1}} \{y(\beta)\},$$
$$P_{k+1} = \{\alpha \in X_{k+1} : y(\alpha) = c_{k+1}\},$$
$$Q = \{P_1, P_2, \dots, P_{k+1}\}, Y_{k+1} = \{\beta \in Y_k : y(\beta) > c_2\},$$
$$Z_{k+1} = Y_k \setminus (Y_{k+1} \cup P_{k+1}).$$

Repeating the proof of the second iteration, we find out that for

- sets P_i, Z_i, Y_{k+1}, $i \in \{1, 2, \dots, k, (k+1)\}$, and
- numbers d_i and c_i $i \in \{1, 2, \dots, k, (k+1)\}$

the five properties are fulfilled.

Since W_n has finite number of elements and $P_i \neq \varnothing, \forall i$, then after a finite number of k_0 iterations the procedure POP stops, and we prove that:

1) $W_n = \bigcup\limits_{i=1}^{k_0} (Z_i \cup P_i)$;
2) $P_1, P_2, \dots, P_{k_0}$ are sets of equivalent Pareto optimal paths for which
   a) $d_i = x(\alpha)$ and $c_i = y(\alpha)$, $\forall \alpha \in P_i$ and $\forall i \in \{1, 2, \dots, k_0\}$;
   b) $d_i < d_{i+1}$ and $c_i < c_{i+1}$, $\forall i \in \{1, 2, \dots, (k_0 - 1)\}$.
3) Every path $\beta \in Z_i$ is dominated by every path $\alpha \in P_i$.
4) $Q = \{P_1, P_2, \dots, P_{k_0}\}$.

■

Corollary 1. *For the classes P_i of Pareto optimal paths the following holds*

$$P_i = \{\alpha \in W_n : \, x(\alpha) = d_i \text{ and } y(\alpha) = c_i\},$$

for each $i \in \{1, 2, \dots, k_0\}$.

Corollary 2. *For P_{k_0} the following equality hods:*

$$P_{k_0} = \{\alpha \in X_1' : \, x(\alpha) = d_1'\},$$

where $X_1' = \{\alpha \in W_n : y(\alpha) = \max\limits_{\beta \in W_n} \{y(\beta)\}$ and $d_1' = \min\limits_{\beta \in X_1'} \{x(\beta)\}$.

Remark 1. *The list Q can be composed by: first apply the Corollary 2 to separate the set P_{k_0}; after that, consecutively separate the sets $P_{k_0 - 1}$, $P_{k_0 - 2}$, and so on, until P_1 is separated.*

Every subset P_i turns out to be a set of all $(1, n)$-paths in a special digraph G_i. The algorithm that finds a *List of Pareto Optimal Paths* which we will call LPOP (given in Alg. 5), composes a list S of the adjacency lists $G_i.Adj$ for each $i \in \{1, 2, \dots, k_0\}$ by implementing the POP procedure.

We will note that once we have the list S, we can easily obtain a list P' of Pareto optimal solutions by taking predefined number of elements for each class P_i, as well we can obtain a list P of all optimal solutions.

Besides the previously defined functions $minsum(G.Adj)$, $maxmin(G.Adj)$ and $capacity(G.Adj)$, in the formulation of the LPOP algorithm (Alg. 5), we will use the function $restrict(G.Adj, c)$, that is defined as follows.

The function $restrict(G.Adj, c)$ takes as an input the outgoing adjacency list $G.Adj$ and the number c. It calculates the adjacency list of those edges (i, j) from $G.Adj$, for which $g(i, j) > c$.

Theorem 1. *The LPOP algorithm (Alg. 5) is correct.*

Proof: The correctness of the Alg. 5 follows from the correctness of the functions $capacity(G.Adj)$, $minsum(G.Adj)$, $maxmin(G.Adj)$, and from Lemma 1. It is easily verified by proving that the **while** loop of the $lpop(G.Adj)$ function implements the POP procedure using these functions.

We will examine the first iteration of the **while** loop. The function $minsum(G.Adj)$ calculates $d_0 = \min\limits_{\beta \in W_n} \{x(\beta)\}$ and defines the adjacency list $\widehat{G}.Adj$ of the shortest paths subnetwork $\widehat{G}$. From the correctness of the function $minsum(Adj)$

Algorithm 5 Function $lpop(G.Adj)$

```
Input: $G.Adj$
Output: the list $S$ with $k_0$ number of elements
    $RAdj \leftarrow G.Adj$
 2: $c_0 \leftarrow capacity(G.Adj)$
    $more \leftarrow$ true
 4: while $more =$ true do
      $\{d_0, G.Adj\} \leftarrow minsum(G.Adj)$
 6:   if $d_0 = \infty$ then
        $more \leftarrow$ false
 8:   else
        $\{c_1, \widetilde{G}.Adj\} \leftarrow maxmin(G.Adj)$
10:     $pushback(S, \widetilde{G}.Adj)$
        if $c_1 = c_0$ then
12:       $more \leftarrow$ false
        else
14:       $RAdj \leftarrow restrict(RAdj, c_1)$
        end if
16:     $G.Adj \leftarrow RAdj$
      end if
18: end while
    return $S$
```

we know that α is a $(1,n)$-path in $\widehat{G}$, exactly when α is a $(1,n)$-path in the network G and $x(\alpha) = d_0$. Hence, the set X from the POP procedure is the set of all $(1,n)$-paths of the subnetwork, defined by the adjacency list $G.Adj$.

By using the definition of the function $maxmin(G.Adj)$, we will prove that its function call implements step 3 of the POP procedure.

Indeed, when applied on the adjacency list of the subnetwork $\widehat{G}$, the function $maxmin(Adj)$ calculates the maximal capacity c_1 of a $(1,n)$-path in $\widehat{G}$, and defines the adjacency list $\widetilde{G}.Adj$ of the maximal capacity digraph $\widetilde{G}$ of the subnetwork $\widehat{G}$. This means that α is a $(1,n)$-path in the digraph $\widetilde{G}$, if and only if it is a $(1,n)$-path in the network $\widehat{G}$ and $y(\beta) = c_1$. In the notations of the POP procedure, this means that $c_1 = \max_{\beta \in X}\{y(\beta)\}$ and α is a $(1,n)$-path in the digraph $\widetilde{G}$ if and only if $\alpha \in P_0$. For that reason the adjacency list $\widetilde{G}.Adj$ is included in the list S (line 10 of Alg. 5).

The step 6 from the POP procedure is implemented in the alternative branch of the **if** statement which verifies whether the set Y, defined by the step 4 of the procedure, is the empty set. If $c_0 = c_1$, then $Y = \varnothing$, and the algorithm is terminated. Otherwise, we define the adjacency list $RAdj$ of the subnetwork for which β is a $(1,n)$-path if and only if it is a $(1,n)$-path in G, and $y(\beta) > c_1$. ■

We will illustrate the above proof with the following example.

Example 3. *For the input network G_1, given on Fig. 1, we will trace how the algorithm LPOP (Alg. 5) implements the POP procedure.*

Solution: Initially, the algorithm stores a copy of the adjacency list in the variable $RAdj$, which will be modified in the body of the **while** loop. On line 2, the $capacity(G.Adj)$ function calculates that the capacity of the vertex 5 is $c_0 = 4$.

First iteration. The $minsum(G.Adj)$ function calculates the distance $d_0 = 6$ to the vertex 5, and the adjacency list of the subnetwork $\widehat{G}$:

$$\widehat{G}.Adj = \{\{(2,2,4),(3,5,3)\},\{(3,3,5),(4,6,4)\},\\ \{(4,3,6),(5,1,1)\},\{\},\{\}\}.$$

It is apparent that the above defined subnetwork $\widehat{G}$ has exactly two $(1,5)$-paths: $\alpha_1 = (1,3,5)$ and $\beta_1 = (1,2,3,5)$. Besides that, $x(\alpha_1) = x(\beta_1) = 6$. Since the set of all $(1,5)$-paths in G_1 is $W = \{(1,2,5),(1,3,5),(1,2,3,5),(1,2,4,5),(1,3,4,5),(1,2,3,4,5)\}$, it is directly verified that α_1 and β_1 are the only $(1,5)$-paths with length $d_0 = 6$ in the network G_1. The latter follows from the correctness of the function $minsum(G.Adj)$. In the procedure POP the set $\{\alpha_1, \beta_1\}$ is denoted by X.

Since $d_0 = 6 \neq \infty$ the algorithm enters the body of the **else** statement on line 8. The function $maxmin(G.Adj)$ calculates that in the subnetwork $\widehat{G}$ the capacity of the vertex 5 is $c_1 = 1$, and:

$$\widetilde{G}.Adj = \{\{2,3\},\{3,4\},\{4,5\},\{\},\{\}\}.$$

The outgoing adjacency list $\widetilde{G}.Adj$ defines the maximal capacity digraph $\widetilde{G}$ of the subnetwork $\widehat{G}$. It is apparent that $\widetilde{G}$ has exactly two $(1,5)$-paths. In this case these are $\alpha_1 = (1,3,5)$ and $\beta_1 = (1,2,3,5)$. Also, $y(\alpha_1) = y(\beta_1) = 1$. In the procedure POP we denote the set $\{\alpha_1, \beta_1\}$ by P_0. From Lemma 1 it follows that $P_0 = \{\alpha_1, \beta_1\}$ is the first class of equivalent Pareto optimal solutions. For that reason the algorithm includes $\widetilde{G}.Adj$ in the list S.

Since $c_1 = 1 \neq c_0 = 4$, the function $restrict(RAdj, c_1)$ modifies the adjacency list $RAdj$ by removing all edges with capacity not greater than c_1. The new adjacency list is:

$$RAdj = \{\{(2,2,4),(3,5,3)\},\{(3,3,5),(4,6,4),(5,5,3)\},\\ \{(4,3,6)\},\{(5,1,7)\},\{\}\}.$$

It is directly verified that the set Y_1 of all $(1,5)$-paths in the network defined by $RAdj$ is the set of all $(1,5)$-paths in the network G_1 with capacity bigger than $c_1 = 1$, which is verified by Theorem 1. In the procedure POP Y_1 is denoted by Y and is defined in the step 4 of the procedure.

Setting $G.Adj = RAdj$ the LPOP algorithm moves to the next iteration. In the procedure POP, it corresponds to the assignment $W = Y$, and the start of the new iteration by transition to the step 2.

Second iteration. The $minsum(G.Adj)$ function calculates $d_0 = 7$ and

$$\widehat{G}Adj = \{\{(2,2,4),(3,5,3)\},\{(3,3,5),(4,6,4),(5,5,3)\},\\ \{(4,3,6)\},\{\},\{\}\}.$$

The resulting subnetwork $\widehat{G}$ has a single $(1,5)$-path $\alpha_2 = \{1,2,5\}$. The length of the path α_2 is $x(\alpha_2) = 7$. In this case $X = \{\alpha_2\}$.

The result of the function $maxmin(G.Adj)$ is:

- the capacity of the vertex 5 in the network $\widehat{G}$ is $c_1 = 3$, and
- the digraph $\widetilde{G}$ of the maximal capacity has adjacency list

$$\widetilde{G}.Adj = \{\{2,3\},\{3,4,5\},\{4\},\{\},\{\}\}.$$

The above verifies the fact that in this case $P_0 = \{\alpha_2\}$ is the second class of equivalent Pareto optimal solutions. For that reason $\widetilde{G}.Adj$ is included as second element in the list S.

Since the condition for the loop stop is not fulfilled,

$$RAdj = \{\{(2,2,4)\},\{(3,3,5),(4,6,4)\},\{(4,3,6)\},\\ \{(5,1,7)\},\{\}\}.$$

It is immediately apparent that the set Y_2 of all $(1,5)$-paths in the network defined by $RAdj$ is the set of $(1,5)$-paths of G_1, that have capacities grater than $c_1 = 3$.

Third iteration. The $minsum(G.Adj)$ function calculates that this time the distance to the vertex 5 is $d_0 = 9$, and the new minimal paths subnetwork has adjacency list:

$$\widehat{G}Adj = \{\{(2,2,4)\},\{(3,3,5),(4,6,4)\},\{(4,3,6)\},\\ \{(5,1,7)\},\{\}\}.$$

In this case the set of all $(1,5)$-paths in $\widehat{G}$ is $X = \{(1,2,4,5),(1,2,3,4,5)\}$, and these are all $(1,5)$-paths of Y_2 with length $d_0 = 9$.

Using the $maxmin(G.Adj)$ function, we find out that in the network $\widehat{G}$ the vertex 5 has capacity $c_1 = 4$, and the maximal capacity digraph $\widetilde{G}$ of the subnetwork $\widehat{G}$ has adjacency list $\widetilde{G}.Adj = \{\{2\},\{3,4\},\{4\},\{5\},\{\}\}$.

Apparently, the digraph $\widetilde{G}$ has exactly two $(1,5)$-paths $\alpha_3 = (1,2,4,5)$ and $\beta_3 = (1,2,3,4,5)$. According to Lemma 1, the set $P_0 = \{\alpha_3,\beta_3\}$ is the third class of equivalent Pareto optimal paths. In this case $x(\alpha_3) = x(\beta_3) = 9$ and $y(\alpha_3) = y(\beta_3) = 4$. $\widetilde{G}.Adj$ is included as third element in the list S.

The condition of the **if** statement on line 11 $c_0 = c_1$ will be evaluated to true. This means that there does not exist a $(1,5)$-path with a capacity greater than the current c_1. As a result, the **while** loop is terminated. In the procedure POP this means that $Y_3 = \{\beta \in W : y(\beta) > 4\} = \varnothing$, and the procedure stops.

After the end of the calculations

$$S = \{\{\{2,3\},\{3,4\},\{4,5\},\{\},\{\}\},\\ \{\{2,3\},\{3,4,5\},\{4\},\{\},\{\}\},\\ \{\{2\},\{3,4\},\{4\},\{5\},\{\}\}\},$$

where each element of S determines one class of Pareto optimal paths:

- $S(1) = \{\{2,3\},\{3,4\},\{4,5\},\{\},\{\}\}$ defines the class $P_1 = \{(1,3,5),(1,2,3,5)\}$;
- $S(2) = \{\{2,3\},\{3,4,5\},\{4\},\{\},\{\}\}$ defines the class $P_2 = \{\{1,2,5\}\}$; and
- $S(3) = \{\{\{2\},\{3,4\},\{4\},\{5\},\{\}\}\}$ defines the class $P_3 = \{(1,2,4,5),(1,2,3,4,5)\}$.

The network G_1 has the set of Pareto optimal paths $P = P_1 \cup P_2 \cup P_3$.

Theorem 2. *The LPOP algorithm (Alg. 5) has computational complexity $k_0 O(n \log n + m)$, where k_0 is the number of classes of Pareto equivalent paths.*

The proof follows from Prop. 2 and Prop. 5. It is enough to note that on line 2 of Alg. 5 the call to the function $capacity(G.Adj)$ has complexity $O(n \log n + m)$, and the **while** loop has k_0 number of iterations, where k_0 is the number of classes of Pareto equivalent classes (14). Each iteration involves a single call to the functions $minsum(G.Adj)$ and $maxmin(G.Adj)$, where both have complexity $O(n \log n + m)$. Besides that, the function $restrict(R.Adj, c_1)$ has computational complexity that is lower than $O(n \log n + m)$.

Example 4. *Let the network G_2 be defined with the adjacency list*

$$\begin{aligned} G_2.Adj = \{&\{(2,1,17),(3,1,20),(4,1,19)\},\\ &\{(5,1,7),(6,15,15),(7,1,12)\},\\ &\{(6,1,9),(7,1,18),(8,1,19),(4,1,15)\},\\ &\{(5,14,15),(6,1,12),(7,1,12),(8,1,9),\\ &(10,1,2)\},\{(9,10,22),(10,1,2),(6,1,2)\},\\ &\{(11,1,4),(9,14,20),(10,1,6),(7,1,3)\},\\ &\{(8,1,11),(10,2,7),(5,8,15)\},\\ &\{(10,7,10),(11,10,11)\},\\ &\{(11,8,19),(10,1,20)\},\{(11,3,21)\},\{\}\}. \end{aligned} \tag{19}$$

We will find the list of all Pareto optimal solutions using the LPOP algorithm.

Solution: Using the LPOP algorithm, we calculate the list

$$\begin{aligned} S = \{&\{\{2,3,4\},\{5,7\},\{6,7,8\},\{6,7,8\},\{9\},\\ &\{11\},\{\},\{\},\{\},\{\},\{\}\},\\ &\{\{2,3,4\},\{5,7\},\{6,7,8\},\{6,7,8\},\{9\},\{10\},\\ &\{\},\{\},\{\},\{11\},\{\}\},\\ &\{\{2,3,4\},\{5,7\},\{6,7,8\},\{6,7,8\},\{9\},\{\},\\ &\{10\},\{\},\{\},\{11\},\{\}\},\\ &\{\{2,3,4\},\{7\},\{7,8\},\{6,7\},\{\},\{9\},\{5\},\\ &\{11\},\{\},\{11\},\{\}\},\\ &\{\{2,3,4\},\{7\},\{7,8\},\{6,7\},\{\},\{9\},\{5\},\\ &\{\},\{10\},\{11\},\{\}\},\\ &\{\{2,3,4\},\{6\},\{7,8\},\{\},\{9\},\{\},\{5\},\{\},\\ &\{10\},\{11\},\{\}\}\}. \end{aligned} \tag{20}$$

Every element of S defines a class of equivalent Pareto optimal paths. Using the function $list(m, \widetilde{G}.Adj)$ we get the Pareto optimal paths as a sequence of vertices. For example, the first element of S defines the class P_1, that contains two equivalent Pareto optimal paths $\alpha_1 = (1,3,6,11)$ and $\beta_1 = (1,4,6,11)$.

To each $(1,11)$-path in α we will map a point

$$A_\alpha(x(\alpha), y(\alpha)) \tag{21}$$

The points defined in this way are plotted on Fig. 2. For example, to α_1 and β_1 we map a single point $A_1(3,4)$, because

TABLE I
THE ELEMENTS OF THE LIST S WITH THE CORRESPONDING PARETO OPTIMAL CLASSES P_j AND THEIR POINT REPRESENTATIONS A_j

S_j	P_k	A_j
S_1	$P_1 = \{(1,3,6,11),(1,4,6,11)\}$	$A_1(3,4)$
S_2	$P_2 = \{(1,3,6,10,11),(1,4,6,10,11)\}$	$A_2(6,6)$
S_3	$P_3 = \{(1,2,7,10,11),(1,3,7,10,11),$ $(1,4,7,10,11)\}$	$A_3(7,7)$
S_4	$P_4 = \{(1,3,8,11)\}$	$A_4(12,11)$
S_5	$P_5 = \{(1,4,6,9,10,11)\}$	$A_5(20,12)$
S_6	$P_6 = \{(1,3,7,5,9,10,11)\}$	$A_6(24,15)$

$x(\alpha_1) = x(\beta_1) = 3$ and $y(\alpha_1) = y(\beta_1) = 4$. Following this scheme, by using consecutively the elements S_j of the list S, we calculate the remaining classes P_j and we map the point A_j defined by (21), for each $j \in \{2,3,4,5,6\}$. The results are given in Tab. I.

Therefore, in example (20) the set of all Pareto optimal paths is

$$P = P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5 \cup P_6.$$

The network G_2 has 10 Pareto optimal paths, distributed in 6 classes of equivalent paths. To illustrate graphically the result in Fig. 2, we define the set $\mathcal{P}$ of all $(1,11)$ paths in digraph G_2. In the example (20) there are 118 such paths. In this case we obtain 71 points. If the point A_α illustrates the path α that is not a Pareto optimal, we plot it in black. The points given in Tab. I are plotted in white, and they represent the Pareto optimal paths. Each point A_j is a vertex of an angle γ_j with rays given in dashed lines. Let α_j be a $(1,11)$-path that is represented by the point A_j. Inside the angle γ_j lie all points that illustrate paths that are dominated by α_j. The vert. opp. angle of γ_j is plotted in gray color, and inside it might lie points that illustrate paths that dominate α_j. As we may expect, such points does not exist.

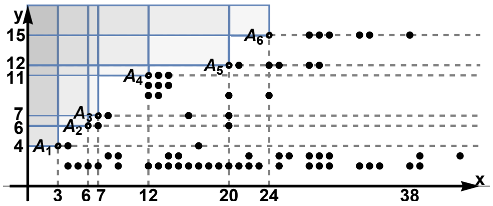


Fig. 2. Plot of the points A_j that illustrate the Pareto optimal classes P_j on the Cartesian plane defined by the values of the functions x and y

V. CONCLUSION

In [3] Hansen solves the MAXSUM-MAXMIN shortest path problem by presenting an algorithm that discovers a special subset of the Pareto optimal solutions, called "minimal complete set of efficient paths (MCS)". The list S that is composed by the LPOP algorithm presented in this paper, gives a more complete information for the Pareto optimal solutions. It is enough to select just one $(1,n)$-path from each element of S to obtain a MCS. For example, using the list S that we get for (11) that defines the network G_1, we get the following MCS: $P^{(1)} = \{(1,3,5),(1,2,5),(1,2,4,5)\}$. Besides that, apparently for the network G_1 three more different MCS can be composed.

Hansen proves that the number of Pareto optimal solutions grows exponentially with the increasing of the number of vertices in the network. However, it has been shown that in various real practical applications this number can be much smaller [20]. In the latter work authors discover key characteristics in the input data that lead to a number of Pareto optimal solutions on each vertex that is restricted by a small constant. In our case, this leads to the restriction of the constant k_0 in Theorem 2.

In Sec. III-A we solve the MINSUM list problem. We prove the correctness of $minsum(G.Adj)$ function (Alg. 2) that helps us to describe all shortest paths by calculation of the shortest paths subnetwork. We prove that its computational complexity is $O(n \log n + m)$.

In Sec. III-B we solve the MAXMIN list problem with Alg. 3 that allows us to calculate the capacity of a vertex with computational complexity $O(n \log n + m)$. Based on it, we define the function $maxmin(G.Adj)$ (Alg. 4) that describes all maximum capacity paths by defining the maximum capacity digraph of the network G. The complexity of the algorithm is again shown to be $O(n \log n + m)$.

Besides the solution of the corresponding MINSUM list and MAXMIN list problems, the two functions $minsum(G.Adj)$ and $maxmin(G.Adj)$ using Alg. 5 allow the solution of Prob. 1. The resulting description of all Pareto optimal solutions separates the classes of Pareto equivalent paths, and allows to visualize a predefined number of elements from each class of Pareto equivalent paths. The correctness of the algorithm is proved (Th. 1), and also its computational complexity is proved to be $k_0 O(n \log n + m)$ (Th. 2).

REFERENCES

[1] R. Beier, H. Röglin, C. Rösner, and B. Vöcking, "The smoothed number of pareto-optimal solutions in bicriteria integer optimization," *Mathematical Programming*, vol. 200, pp. 319–355, September 2022. doi: 10.1007/s10107-022-01885-6

[2] J. C. Namorado Climaco and E. Queirós Vieira Martins, "A bicriterion shortest path algorithm," *European Journal of Operational Research*, vol. 11, no. 4, pp. 399–404, 1982. doi: 10.1016/0377-2217(82)90205-3

[3] P. Hansen, "Bicriterion path problems," *Multiple Criteria Decision Making Theory and Application*, pp. 109–127, 1980. doi: 10.1016/S1097-2765(03)00225-9

[4] X. Gandibleux, F. Beugnies, and S. Randriamasy, "Martins' algorithm revisited for multi-objective shortest path problems with a maxmin cost function," *4OR*, vol. 4, no. 1, pp. 47–59, 2006. doi: 10.1007/s10288-005-0074-x

[5] E. Q. V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984. doi: 10.1016/0377-2217(84)90077-8

[6] J. Brumbaugh-Smith and D. Shier, "An empirical investigation of some bicriterion shortest path algorithms," *European Journal of Operational Research*, vol. 43, no. 2, pp. 216–224, 1989. doi: 10.1016/0377-2217(89)90215-4

[7] A. Skriver and K. Andersen, "A label correcting approach for solving bicriterion shortest-path problems," *Computers & Operations Research*, vol. 27, no. 6, pp. 507–524, 2000. doi: 10.1016/S0305-0548(99)00037-4

[8] A. Sedeño-noda and M. Colebrook, "A biobjective dijkstra algorithm," *European Journal of Operational Research*, vol. 276, no. 1, pp. 106–118, 2019. doi: 10.1016/j.ejor.2019.01.007

[9] M. Minoux, "Solving combinatorial problems with combined minmax-min-sum objective and applications," *Mathematical Programming*, vol. 45, no. 1-3, pp. 361–372, 1989. doi: 10.1007/bf01589111

[10] A. P. Punnen, "On combined minmax-minsum optimization," *Computers & Operations Research*, vol. 21, no. 6, pp. 707–716, 1994. doi: 10.1016/0305-0548(94)90084-1

[11] P. Dell'Olmo, M. Gentili, and A. Scozzari, "On finding dissimilar pareto-optimal paths," *European Journal of Operational Research*, vol. 162, no. 1, pp. 70–82, 2005. doi: 10.1016/j.ejor.2003.10.033

[12] F. Guerriero and R. Musmanno, "Label correcting methods to solve multicriteria shortest path problems," *Journal of Optimization Theory and Applications*, vol. 111, no. 3, pp. 589–613, 2001. doi: 10.1023/A:1012602011914

[13] C. Mohamed, J. Bassem, and L. Taicir, "A genetic algorithms to solve the bicriteria shortest path problem," *Electronic Notes in Discrete Mathematics*, vol. 4, no. 1, pp. 851–858, 2010. doi: 10.1016/j.endm.2010.05.108

[14] S. Fidanova, M. Ganzha, and O. Roeva, "Intercriteria analyzis of hybrid ant colony optimization algorithm for multiple knapsack problem," in *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2021. doi: 10.15439/2021F22 pp. 173–180.

[15] A. Cassia, O. Jabali, F. Malucelli, and M. Pascoal, "The electric vehicle shortest path problem with time windows and prize collection," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022. doi: 10.15439/2022F186 pp. 313–322.

[16] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959. doi: 10.1007/bf01386390

[17] R. Diestel, *Graph Theory*, 5th ed. Berlin: Springer Publishing Company, Incorporated, 2017. ISBN 3662536218. doi: 10.1007/978-3-662-53622-3

[18] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 3rd ed. Cambridge, Massachusetts: The MIT Press, 2009. doi: 10.5555/1614191

[19] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their uses in improved network optimization algorithms," *Journal of the ACM*, vol. 34, no. 3, pp. 596–615, July 1987. doi: 10.1145/28869.28874

[20] M. Müller-Hannemann and K. Weihe, "Pareto shortest paths is often feasible in practice," *Algorithm Engineering*, pp. 185–197, 2001. doi: 10.1007/3-540-44688-5_15

DOI: 10.15439/2023F2715
ISSN 2300-5963 ACSIS, Vol. 35

Comparative Analysis of Exact, Heuristic and Metaheuristic Algorithms for Flexible Assembly Scheduling

Octavian Maghiar, Teodora Selea, Adrian Copie, Flavia Micota, Mircea Marin
West University of Timişoara,
Department of Computer Science,
blvd. Vasile Pârvan, 4,
300223, Timişoara, Romania
Email: {octavian.maghiar98, teodora.selea, adrian.copie, flavia.micota, mircea.marin}@e-uvt.ro

***Abstract*—Real-world manufacturing scenarios usually lead to difficult assembly scheduling problems. Besides strict precedence constraints between jobs or operations, such problems incorporate constraints related to maintenance activities on working stations (machines) and specific setup times when different operations are executed on the same machine. This paper analyzes the performance of several approaches, based on mathematical programming and on (meta)heuristics, to solve flexible assembly scheduling problems characterized by an arbitrary tree-like structure of the operation network. In this context, a specific encoding of candidate solutions and some specific perturbation operators are proposed. The encoding and the operators allow the distribution of sub(batches) of operations on several machines which leads, for some assembly scheduling problems, to a significant decrease of the makespan.**

I. Introduction

SCHEDULING represents a class of optimization problems with significant practical impact. The most studied problem is the Job Shop Scheduling problem (JSSP) [1] aiming to find an assignment of a set of inter-related jobs on a set of working resources (machines) such that some performance measures are optimized.

Assembly production scheduling is a class of scheduling problems (as identified in [2]), where a product is obtained by assembling several components (also referred to as subassemblies, subproducts or make-parts) which are a result of either some production operations or other assembling steps. Hence, the final product is the result of a particular set of operations, which are inter-related according to a hierarchical structure. Numerous factors, such as the number of operations, the number of products, the number of machines, the complexity of product structure, and the number of constraints, can increase the scheduling problem's complexity. Therefore, ongoing research in the field of scheduling is always necessary.

As is mentioned in [3] flexible assembly job-shop scheduling is less studied than job-shop scheduling. Previous work on Assembly Production Scheduling includes effective heuristics, genetic algorithms [4], or machine learning-based solutions. In [5], the authors propose a heuristic, based on the concept of the critical path, for solving the problem of large assembly production having as objective the minimization of the makespan. Another heuristic that targets the problem of assembly scheduling by taking into account the splitting of the operations in several (sub)batches was introduced in [6]. The load is distributed among the available machines using the proposed heuristic, followed by actual scheduling based on the critical path approach. In [7], the authors also include a batch splitting procedure; however, it is followed by a genetic algorithm to perform the scheduling.

The aim of this paper is to analyze several assembly scheduling strategies that are flexible enough to accommodate specific characteristics of the production process, e.g. the maintenance activities. The main contributions of the paper include:

- specific encoding of candidate solutions and specific perturbation operators which ensure the feasibility of generated schedules and allow the distribution of sub(batches) of operations on several machines, leading to a significant decrease of the makespan;
- design and implementation of a highly configurable generator of assembly scheduling test problems;
- a comparative analysis of the performance of an exact solver, a heuristic based on the critical path concept, and two metaheuristic algorithms.

The rest of the paper is organized as follows. Section II proposes a motivating manufacturing scenario, while the particularities and the formal description of the scheduling problem are presented in Section III. A short review of recent works addressing similar problems is presented in Section IV. Sections V and VI provide details on the heuristic based on the critical path and present the proposed encoding, the algorithm for generating feasible initial schedules, the decoding and evaluation procedures as well as the proposed search operators used by the metaheuristic algorithms. The data generator structure, the experimental setup and results for some test

This work was supported by project POC/163/1/3/ "Advanced computational statistics for planning and monitoring production environments" (2022-2023)

problems are presented in Section VII, while Section VIII concludes the paper.

II. A REAL-WORLD SCENARIO

A real-world problem that triggered our study is the production of flexible metal tubes for car exhaust systems. The final product is composed of subproducts that are result of production (SP) or assembly (As) operations. The manufacture process of the product is described using the bill of materials (BOM), that contains all information used to produce an item: the raw materials (RM), the (sub)products, the quantity needed to by for each manufactured product (BOM for mill-tube production - Figures 1).

A mill tube is formed from two component tubes (referred as IT and OT in Figure 1) that are covered with a metal mesh. Interlock rings are used as a fastening system. The production process can be shortly described as follows: (1) the inner tube (subproduct IT) and the outer tube (subproduct OT) are produced from a metal sheet (RM) that has to be rolled, welded and cut on a specific machine (operations O_6 and O_7); (2) the resulting tubes are transformed in bellows (subproduct E) through a stuffing process (O_5); (3) meantime: (i) the metal mesh (product F), that covers the tube built in the hydroforming process (O_4), is produced from metal wire (O_{10}, O_9, and O_8); (ii) the interlocks (subproduct C) are produced also from metal wire (O_3).

The assembling process, that has as result the product B, is executed on assembling workstations (O_2) and it consists of: (i) the bellows are wrapped with the metallic mesh; (ii) the interlocks and garnish are added; (iii) the final product is pressed; (iv) some identification information is written on the product. Then a quality assurance control (O_1) is done and the product, A, is packed.

Beside the information related to BOM other production information, like the machine characteristics on which the operations are executed, must be provided. Table I contains the characteristics (setup-time, unit processing time) of the set of machines ($\{M_1, M_2, \ldots, M_{19}\}$) that are used in mill tube production.

III. PROBLEM DESCRIPTION

The result of an assembly manufacturing process is a product that consists of many components, each of the components being either manufactured or obtained by assembling several other (sub)components (also called make-parts). The operations involved in this process are either of manufacturing type or assembling type.

The main difference between these two types of operations is that a manufacturing operation usually requires only one previously produced component (and potentially several raw materials), while an assembling operation requires several other (sub)components which should be produced by the time the assembling operation starts. In the tree-like structure of an assembly (see Figures 1 and 2) the manufacturing operations correspond to nodes having only one child, while the assembly operations correspond to nodes having at least two children.

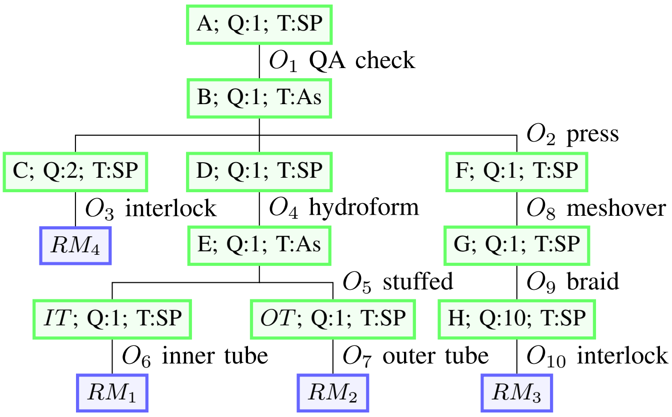


Fig. 1. Bill of materials networks for mill tube scenario. Each node contains information regarding the obtained (sub)product: name (A, B, C, . . .) quantity (Q:number), type (corresponding to a production process - SP or to an assembly process - As)

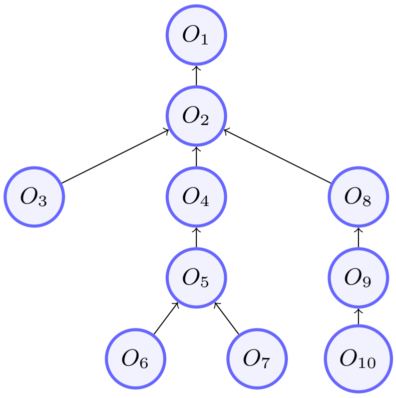


Fig. 2. Operation network for mill tube example

The Assembly Scheduling Problem (ASP) aims to schedule all types of operations on a set of machines in such a way that all make-parts are finalized before they are required for an assembly operation. The main particularity of this problem with respect to the standard Job Shop Scheduling Problem is that the precedence relation between operations corresponding to one assembly product (i.e. one job) is not a total order relation.

This allows the extension of one assembly product scheduling to several products by just considering a dummy assembly operation that would virtually group all products.

Therefore, in the following, we will consider the operations corresponding to all products (jobs) being grouped in one set of operations on which there is a partial order precedence relation.

A. *Characteristics of the assembly scheduling problem*

The class of assembly scheduling problems addressed in this paper is characterized by:

- all required raw materials are available, thus at the beginning of the scheduling time horizon all operations that do not have predecessors can start;

TABLE I
PRODUCTS, OPERATIONS, AND ELIGIBLE MACHINES FOR MILL TUBE EXAMPLE

Product	Operation	Machine (ID, setup time, unit production time)
A	O_1 QA check	(M_1, 600s, 2s)
B	O_2 press	(M_2, 600s, 40s), (M_3, 600s, 40s), (M_4, 500s, 40s), (M_5, 600s, 40s), (M_6, 500s, 40s)
C	O_3 interlock	(M_7, 600s, 4s), (M_{19}, 600s, 4s)
D	O_4 hydroform	(M_8, 800s, 1s), (M_9, 800s, 1s), (M_{10}, 800s, 1s), (M_{11}, 700s, 2s), (M_{12}, 700s, 2s)
E	O_5 stuffed	(M_{13}, 1000s, 5s)
IT	O_6 inner tube	(M_{14}, 900s, 2s),
OT	O_7 outer tube	(M_{15}, 900s, 2s),
F	O_8 meshoverline	(M_{16}, 900s, 12s), (M_{17}, 900s, 12s)
G	O_9 braid	(M_{18}, 900s, 30s)
H	O_{10} interlock braid	(M_7, 600s, 4s), (M_{19}, 600s, 4s)

- at a given moment, a machine can process only one operation and once started, an operation cannot be interrupted (non-preemptive);
- the operations can be grouped in batches and (producing a batch of components or products) started, it cannot be interrupted;
- once established, the size of a batch is not modified;
- for a given operation only one batch of executions can be scheduled on a machine (re-entrance is not allowed);
- there is no cost for transferring components between machines;
- no setup is required if the operations (corresponding to different batches of *same type* of (sub)components or products) are scheduled in sequence on the same machine;
- the setup times are sequence-independent and non-anticipatory (the machine is available and the operation is ready to be started);
- the maintenance activities are considered as machine-assigned operations with fixed starting time and duration.

B. Formal description

1) Notations and input data:

- Set of n operations: $\{O_j | j = \overline{1,n}\}$;
- Set of m machines: $\{M_i | i = \overline{1,m}\}$;
- t_{ji} = time to execute on machine M_i one unit of the (sub)component which is a result of operation O_j;
- s_{ji} = setup time for executing the operation O_j on machine M_i;
- Q_j = total number of executions of operation O_j derived from the quantities specified in the BOM structure;
- F = matrix of eligibility, an $n \times m$ matrix specifying which machines are eligible for each operation, i.e.

$$F_{ji} = \begin{cases} 1 & \text{if } O_j \text{ can be executed on } M_i \\ 0 & \text{if } O_j \text{ cannot be executed on } M_i \end{cases} \tag{1}$$

- R = array of n entries which can be used to identify the subset of maintenance operations, i.e.

$$\mathcal{O}_m = \{O_j | R_j > -1\},$$

in which case R_j denotes the maintenance starting time (SM_i) on the machine M_i which satisfies the condition $F_{ji} = 1$. More specifically, the elements of R are defined as:

$$R_j = \begin{cases} SM_i & \text{if } O_j \text{ is a maintenance operation for } M_i \\ -1 & \text{if } O_j \text{ is not a maintenance operation} \end{cases} \tag{2}$$

The maintenance starting times are fixed and known before the scheduling process is started.

- $\pi(j) = \{k | O_k$ is a child of O_j in the operation network$\}$ describes the direct predecessor relation, i.e. $\pi(j)$ is the set of indices of operations which directly precedes O_j, thus they should be finalized before the starting moment of O_j (one operation can have several direct predecessors);
- $\sigma(j) = k$ where $k \in \pi(j)$ (one operation can have only one direct successor) and $\sigma(j) = -1$ if O_j is a final operation (its result is a final product);
- Deadline for the final operations: $D_j \in [0, T]$ for any $j \in \{1, \ldots, n\}$ such that $\sigma(j) = -1$.

2) Decision variables:

For $j = \overline{1,n}$ and $i = \overline{1,m}$:

- Assignment matrix:

$$A_{ji} = \begin{cases} 1 & \text{if } O_j \text{ is executed on } M_i \\ 0 & \text{if } O_j \text{ is not executed on } M_i. \end{cases} \tag{3}$$

- Batch splitting matrix:

$$B_{ji} = \begin{cases} b_{ji} & \text{if } O_j \text{ is executed on } M_i \\ 0 & \text{if } O_j \text{ is not executed on } M_i \end{cases} \tag{4}$$

where $b_{ji} \in \mathbf{N}$ denotes the number of consecutive executions of operation O_j on machine M_i leading to the production of a batch of b_{ji} (sub)components that are specific to operation O_j;

- $S_{ji} \in [0, T)$: starting time of O_j on M_i (T is the time horizon for the assembly production);
- $C_{ji} \in [0, T]$: completion time of O_j on M_i.

C. Constraints

- (C1) All operations are assigned on eligible machines:

$$\sum_{j=1}^{n} \sum_{i=1}^{m} (1 - F_{ji}) A_{ji} = 0 \tag{5}$$

- (C2) The assignment and batch splitting matrices are consistent (an operation assigned to a machine should be executed at least once and the number of executions of an operation on a machine is nonzero only if the operation is assigned to that machine):

$$B_{ji} > 0, \quad \forall j, i \text{ such that } A_{ji} = 1 \tag{6}$$

$$B_{ji} = 0, S_{ji} = 0, C_{ji} = 0 \quad \forall j, i \text{ such that } A_{ji} = 0$$

- (C3) The completion time of any operation O_j is smaller than the starting time of its succeeding operation $O_{\sigma(j)}$:

$$C_{ji_1} \leq S_{\sigma(j)i_2} \tag{7}$$

$\forall j \in \{1, \ldots, n\}, \forall i_1, i_2 \in \{1, \ldots, m\}, A_{ji_1}{=}A_{\sigma(j)i_2}{=}1$

- (C4) The time intervals corresponding to operations executed on the same machine are disjoint:

$$[S_{j_1 i}, C_{j_1 i}) \cap [S_{j_2 i}, C_{j_2 i}) = \emptyset, \quad \forall j_1 \neq j_2, \tag{8}$$

$\forall i \in \{1, \ldots, m\}$ such that $A_{j_1 i} = A_{j_2 i} = 1$

- (C5) Completion time of a batch of operations:

$$C_{ji} = S_{ji} + s_{ji} + B_{ji} \cdot t_{ji} \tag{9}$$

$\forall j \in \{1, \ldots, n\}, i \in \{1, \ldots, m\},$ such that $A_{ji} = 1$

- (C6) The sum of all batch sizes corresponding to an operation equals the total quantity which should be produced by that operation:

$$\sum_{i=1, A_{ji}=1}^{m} B_{ji} = Q_j, \quad \forall j \in \{1, \ldots, n\} \tag{10}$$

- (C7) All final operations are finalized before the corresponding deadlines:

$$C_{ji} \leq D_j, \quad \forall j \in \{1, \ldots, n\}, i \in \{1, \ldots, m\} \tag{11}$$

such that $\sigma(j) = -1, A_{ji} = 1$.

- (C8) The starting time for maintenance operations is fixed:

$$S_{ji} = R_j, \forall j \in \{1, \ldots, n\}, i \in \{1, \ldots, m\} \tag{12}$$

such that $R_j \neq -1, A_{ji} = 1$.

D. Objective function

The goal of the scheduling is to minimize the makespan, C_{max}, defined as:

$$C_{max} = \max_{j \in E} \max_{i \in \{1, \ldots, m\}} \{C_{ji} \mid A_{ji} = 1\} \tag{13}$$

where $E = \{j | \sigma(j) = -1\}$ is the set of final operations. Thus, the optimization problem is:

$$\min_{A,B,S,C} C_{max} \tag{14}$$

IV. Related work

The authors of [8] addressed the problem of flexible assembly job-shop scheduling with lot streaming. The analyzed production structure is a two-stage one, characterized by the presence of an assembly stage at the end of a flexible job shop. The jobs in the first stage are processed in large batches which might lead to waiting time. The proposed approach is to split the batch into sub-batches, thus the initial scheduling problem is split into two sub-problems: batch splitting and batch scheduling. The proposed solution is based on an Artificial Bee Colony (ABC) algorithm using a population of candidates encoded based on four one-dimensional arrays: (i) an array containing the number of (sub)batches corresponding to each job; (ii) an array containing the size of each (sub)batch; (iii) an array encoding the sequence of all (sub)batches of operations; (iv) an array specifying the machine on which each (sub)batch of operations is assigned. The size of each candidate solution depends on the number of (sub)batches.

The idea of using sub-batches of unequal sizes is exploited also in our study but in the more general context when the production process involves several assembly stages.

An assembly scheduling problem involving several assembly stages which induce additional precedence constraints between jobs is approached in [3] where a genetic algorithm (GA) is used to solve it. Each element of the population evolved by the GA consists of two parts: (i) a one-dimensional array used to encode the machine assignment; (ii) a two-dimensional array for encoding the sequence of operations based on the concept of level in the operations' network. A schedule is constructed through a decoding process in which the operations are allocated to the first time slot where they fit, in the order given by the structure of the two-dimensional array.

With respect to the structure of the operations' network, the approach proposed in [3] is consistent with the particularities of the problem addressed in this paper, but it does not allow batch size control and to distribute sub-batches of operations to different machines.

In [9] a lot streaming technique, that splits jobs into sub-jobs, is applied on an assembly job shop scheduling problem (AJSP). The authors split the problem into two sub-problems (i) determine a sub-lot split; (ii) solve AJSP problem. In order to solve the sub-lot split, a genetic algorithm is used that uses a matrix, that stores on each line the number of lots for each operation and the quantity for each lot. In order to resolve the assembly problem four simple dispatching rules are used. The lots are scheduled by shortest/longest processing time, earliest due date, minimal slack time (difference between the due date and the total processing time of the operation) such that the constraints derived from the BOM are satisfied. In [10] the authors extend the work for [9] and use genetic algorithm and particle swarm optimization metaheuristics to solve also the second problem, by incorporating into the solutions the information related to the order of operation on each machine.

The approaches proposed in [9], [10] are different from the ones discussed in this paper by the fact that the operations can be execute on a sub-set of machines not on all available machines.

V. Heuristic based on the critical path

In [5], the authors introduced the Lead Time Evaluation and Scheduling Algorithm (LETSA) that uses a critical path heuristic in order to construct a scheduling. Based on the operation networks, the LETSA algorithm creates a list of feasible operations, $\mathcal{F}$, i.e. operations for which their successors have been already scheduled. In the beginning, $\mathcal{F}$ contains the operations generating the final product(s).

The scheduling process consists of several steps (see Algorithm 1) starting with the identification of critical paths that originate in all operations of $\mathcal{F}$. For an operation O_j, a critical path is a sequence $[O_j = O_{l_1}, O_{l_2}, \ldots, O_{l_r}]$ such that $O_{l_q} = \sigma(O_{l_{q+1}})$ and $\sum_{q=1}^{r} t_{l_q *}$ is maximal. The notation t_{j*} refers to the execution time of operation O_j on the slowest machine (if the machines would have different execution times).

It should be mentioned that if a machine satisfying all conditions specified in Step S3 of Algorithm 1 is not found, then the tentative execution interval is shifted toward the left until the first availability interval is reached. In this way, a machine is always identified.

The problem addressed in [5] is slightly different from that addressed in this paper because the operations are executed in a work-centers that contain one or more *identical* machines. In our approach, the machines are not necessarily identical, so the following adaptations were done: (i) in order to calculate the critical path in Step S1 of Algorithm 1, the maximal execution times over all eligible machines are used; (ii) the list of potential machines analyzed at Step S3 of Algorithm 1 is limited to eligible machines characterized by the smallest execution times for the corresponding operation. This has been done with the aim of increasing the chance to generate schedules with smaller makespan.

VI. Metaheuristic approaches

For the metaheuristic algorithms used in the experimental analysis, the encoding and decoding procedures, as well as the search operators are described in the following.

A. Encoding and search space

According to [8], a flexible assembly scheduling with batch splitting involves several decisions: (i) in which sequence are executed the operations on each machine; (ii) on which machine is executed each sub-batch corresponding to each operation; (iii) which is the size of each sub-batch for each operation.

In order to incorporate the information required by these decisions we propose the following encoding for a candidate solution:

- a list of distinct operation indices, $L = [o_{(1)}, \ldots, o_{(n)}]$ with $o_{(l)} \in \{1, \ldots, n\}$, such that for any $1 \leq l, r \leq n$, if $o_{(l)} \in \pi(o_{(r)})$ then $l < r$;
- the batch-splitting matrix B_{ji} defined as in Eq. (4).

The order of operations given by list L corresponds to a topological order of the nodes in the operation network. L provides the order in which the operations are dispatched to machines, during the decoding step, but not necessarily the order in which they are executed in the production stage. More specifically, if $O_{o(l)}$ and $O_{o(r)}$ belong to the same branch in the operation tree-like network, then $O_{o(l)}$ will be executed before $O_{o(r)}$, but if they belong to different branches then $O_{o(l)}$ will be dispatched before $O_{o(r)}$, but not necessarily executed before $O_{o(r)}$. It should be noted that the maintenance operations are not explicitly included in the encoding, as their starting times and durations are fixed. They are taken into consideration only during the evaluation step.

An operation list, L, corresponding to the operation network described in Figure 2 is illustrated in Figure 3 where the arrows highlight the direct predecessor relations between operations. As follows from this figure, such an operation list corresponds to a topological order of the operation network. This ordering is not unique. In fact, for an oriented tree $\mathcal{T}$ with n nodes the total number of topological orderings is $n!/\prod_{v \in \mathcal{T}} \mu(\mathcal{T}_v)$ where v denotes a node, $\mathcal{T}_v$ denotes the (sub)tree rooted in the node v and $\mu(\mathcal{T}_v)$ denotes the number of nodes in the (sub)tree rooted in v. For the tree corresponding to the operation network from Figure 2 the number of distinct topological orderings is $10!/(10 \cdot 9 \cdot 4 \cdot 3 \cdot 3 \cdot 2) = 560$.

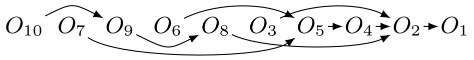


Fig. 3. An operation list corresponding to the tree-like BOM described in Figure 1 which illustrates the topological order of the operations.

Taking into account both the operation list and the batch size matrix, B (see Table II), the proposed encoding allows solving batch splitting and batch sequencing problems simultaneously and corresponds to a structure of size $n + n \cdot m$. This size is larger than that corresponding to traditional encoding based on two one-dimensional arrays of n elements (operation sequence array and machine assignment array). However this traditional encoding does not allow to distribute batches of operations on parallel machines.

B. Generation of initial candidates

The metaheuristics start from one or several initial candidates, which are further modified with the aim of improving its/their quality. The generation of initial candidates should take into account the encoding rules and the structural constraints which should be satisfied by a feasible candidate, i.e. precedence constraints induced by the operation network and constraints related to the number of components to be produced (induced by the corresponding bill of materials). The construction of the list L, corresponding to the topological

Algorithm 1 LETSA Algorithm

```
Input: $n$, $m$, $F_{1..n\times 1..m}$, $O_{1..n}$, $t_{1..n\times 1..m}$, $s_{1..n\times 1..m}$
Output: $A_{1..n\times 1..m}$, $S_{1..n}$, $C_{1..n}$
$\mathcal{F} \leftarrow \{O_j | \sigma(j) = -1\}$
while $\mathcal{F} \neq \emptyset$ do
    (S1): select $O_j \in \mathcal{F}$ s.t. $O_j$ belongs to a critical path
    (S2): set the tentative completion time for $O_j$: $C_{j*} \leftarrow S_{\sigma(j)k}$, where $k$ satisfies $A_{\sigma(j)k} = 1$
    (S3): find $M_i$ such that $F_{ji} = 1$, $M_i$ is available in $[C_{j*} - t_{ji}, C_{j*})$ and maximizes $C_{j*} - t_{ji}$
    (S4): update the assignment matrix and the completion and starting times: $A_{ji} \leftarrow 1$; $C_{ji} \leftarrow C_{j*}$; $S_{ji} \leftarrow C_{ji} - t_{ji}$
    (S5): update the list of feasible operations: $\mathcal{F} \leftarrow \mathcal{F} \backslash \{O_j\} \cup \pi(j)$
end while
return $A$, $S$, $C$
```

TABLE II
BATCH SIZE MATRIX (B) FOR MILL TUBE EXAMPLE

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}
O_1	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O_2	0	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0
O_3	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	10
O_4	0	0	0	0	0	0	0	2	2	2	2	2	0	0	0	0	0	0	0
O_5	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0
O_6	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0
O_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
O_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	5	0	0
O_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0
O_{10}	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	50

order, starts from the root node, O_1, and the frontier, $\mathcal{F}$, consisting of its directly preceding operations, $\pi(O_1)$, and select, at each step, a random element from the current frontier to be added in the front of list L. As soon as an operation is included in L, it is removed from $\mathcal{F}$ which is instead extended with all operations which directly precede the operation included in L. In this way, one of the possible topological orderings of the nodes in the operation network is generated. The matrix B can be initialized using random decisions concerning both the selection of machines from the eligible list and the choice of the batch size. The `random` function used in Algorithm 2 generates an integer value in $\{0, 1, \ldots, q\}$. A minimum batch, min_{Q_j} size of 10% of operation quantity is used in order to restrict small batch sizes.

C. Decoding and evaluation

From a candidate solution, represented by the one-dimensional array L and the two-dimensional array B, a schedule is obtained through decoding. The decoding process, described in Algorithm 3 allows also the computation of the makespan value, and it is activated any time a candidate solution has to be evaluated. The operations are scheduled based on the order specified in the operations list (L). If a part of the quantity (B_{ji}) of operation O_j is planned to be scheduled on machine M_i, the decoding procedure identifies the time interval in which it can be executed on that machine. The operation is scheduled as close as possible to the completion time of all of its preceding operations ($\pi(j)$). This means that for each machine the completion time of the last scheduled operation is dynamically updated and further used to establish the starting time of next operation to be scheduled on that machine. Thus the order defined by L is preserved on each machine.

It should be mentioned that the value of the makespan is influenced by the following decisions:

- split the batch of operations scheduled in a time interval which overlaps with a maintenance interval in such a way that the maximum possible amount operation products is executed before the maintenance activity and the remaining ones immediately after the maintenance ($opt\_maintenance$ flag is activated in Algorithm 3) ;
- no setup is required when two consecutive batches involving the same operation/product are scheduled (opt_setup flag is activated in Algorithm 3).

D. Search operators

Both trajectory and population-based metaheuristics require operators which generate new candidate solutions from existing ones. The metaheuristic algorithms involved in the analysis we conducted involve two main types of operators which generate new candidate solutions:

- *Mutation-like.* This operator generates a new candidate solution in the neighborhood of an existing one by applying a perturbation strategy.
- *Crossover-like.* A new feasible candidate solution is constructed by combining information from two existing candidate solutions.

Algorithm 2 Initialization of a feasible candidate

```
Input: $n$, $m$, $F_{1..n\times 1..m}$, $Q_{1..n}$
Output: $L_{1..n}$, $B_{1..n\times 1..m}$
$L \leftarrow [1]$ // index of the root operation
$\mathcal{F} \leftarrow \pi(1)$ // direct predecessors of the root operation
while $\mathcal{F} \neq \emptyset$ do
    $j \leftarrow \texttt{select}(\mathcal{F})$ // random selection from $\mathcal{F}$
    $L \leftarrow \texttt{prepend}(L, j)$ // add in the front of the list
    $\mathcal{F} \leftarrow \mathcal{F}\backslash\{j\} \cup \pi(j)$ // update the frontier
end while
for $j \leftarrow 1..n$ do
    $q \leftarrow Q_j$; $i \leftarrow 1$
    while $(i \leq m)$ and $(q > 0)$ do
        if $F_{ji} = 1$ then
            $B_{ji} \leftarrow \texttt{max}(min_{Q_j}, \texttt{random}(0, q))$
            $q \leftarrow q - B_{ji}$
            $i_* \leftarrow i$
        end if
        $i \leftarrow i + 1$
    end while
    if $q > 0$ then
        $B_{ji_*} \leftarrow B_{ji_*} + q$
    end if
end for
return $L$, $B$
```

The way of action of these operators is shortly presented in the following, the guiding idea being to preserve the feasibility.

1) Mutation : Let us consider a candidate solution encoded by (L, B). A new candidate is generated in the neighbourhood of (L, B) by following the steps:

- randomly select $o_{(q)} \in L$;
- search for the largest $l \in \{1, \ldots, n\}$ such that $o_{(l)} \in \pi(o_{(q)})$ and for the smallest $r \in \{1, \ldots, n\}$ such that $o_{(q)} \in \pi(o_{(r)})$; if operation $o_{(q)}$ does not have predecessors then $l = 1$ and if $o_{(q)}$ does not have a successor then $r = n$;
- randomly select an insertion position $p \in \{l, l+1, \ldots, r\}$ and insert the element $o_{(q)}$ on position p in L;
- if there are several eligible machines for operation $o_{(q)}$, then randomly select two of them and move the batch (totally or partially) from the source to the destination machine; the decision to perturb B is taken with a given probability.

It is easy to observe that all operations in the sublist of L delimited by l and r ($L_{l..r}$) do not contain operations that are in a precedence relation with $o_{(q)}$, meaning that the perturbed candidate solution is still feasible.

It should be also mentioned that if the list of eligible machines for operation $o_{(q)}$ and the lists of machines corresponding to the operations in the sublist $L_{l..r}$ are disjoint, then any insertion of $o_{(q)}$ in another position of the sublist will have no impact on the makespan of the schedule obtained by decoding the candidate solution. On the other hand, if $o_{(q)}$ is one of the final assembly operations, then $l = r$ and the perturbation will be ineffective.

Algorithm 3 Decoding and evaluation of a solution

```
Input: $L_{1..n}$, $B_{1..n\times 1..m}$, $R_{1..n}$,$D$
Output: $S_{1..n}, C_{1..n}, C_{max}$ // start time, completion time
$S_{1..n} \leftarrow D$; $C_{1..n} \leftarrow 0$
$M^{FT}[1..m] \leftarrow 0$ // current makespan per machine
for $h \leftarrow 1..n$ do
    $j \leftarrow L_h$ // schedule operation $O_j$
    for $i \in \{1, \ldots, m\}$ such that $B_{ji} > 0$ do
        if $opt\_setup = True$ and $O_j$ is identical with the last operation scheduled on $M_i$ then
            $st \leftarrow 0$
        else
            $st \leftarrow s_{ij}$
        end if
        $ST \leftarrow \max(\max\{C_k | k \in \pi(j)\}, M_i^{FT})$
        $ET \leftarrow B_{ji} \cdot t_{ji}$
        if there exists $k$ such that $R_k > 0$ and
            $[ST, ST + st + ET) \cap [R_k, R_k + t_{ki}) \neq \emptyset$ then
            if $opt\_maintenance = True$ then
                $C_j \leftarrow \max\{ST + st + ET + t_{ki}, C_j\}$
            else
                $ST \leftarrow R_k + t_{ki}$
                $C_j \leftarrow \max\{ST + st + ET, C_j\}$
            end if
        else
            $C_j \leftarrow \max\{ST + st + ET, C_j\}$
        end if
        $S_j \leftarrow \min\{ST, S_j\}$
    end for
end for
$C_{max} \leftarrow \max_{j=\overline{1,n}} C_j - \min_{j=\overline{1,n}} S_j$
return $C_{max}$
```

Let us consider the L-part encoding corresponding to the example illustrated in Figure 3: $L = [10, 7, 9, 6, 8, 3, 5, 4, 2, 1]$. If the selected element is $o_{(5)} = 8$ then $l = 4$ and $r = 8$, thus there are four alternative insertion positions for the value 8. However, as the machines M_{16} and M_{17} are not used by the other operations, all of these perturbations will be without impact on the makespan.

2) Crossover : The crossover-like perturbation aims to generate a new feasible candidate solution by using information from two existing ones, usually called parents. The feasibility is preserved if the idea of precedence operation crossover is used: some elements are taken from one parent, while from the other parent is used to order in which the remaining elements are placed. Let us consider $L = [o_{(1)}, \ldots, o_{(n)}]$ and $L' = [o'_{(1)}, \ldots, o'_{(n)}]$ and the corresponding batch size matrices B and B'. The steps followed to construct a new feasible candidate, (L^{new}, B^{new}), are:

- randomly select $l < r$ from $\{1, \ldots, n\}$;
- transfer from L to L^{new} all elements with indices be-

tween l and r;

- scan the elements of L' and for each element $o'_{(l)}$ which is not yet in L^{new} append it to
  - a prefix list L_P, if $o'_{(l)}$ is the predecessor (not necessarily direct) of at least one element of L^{new};
  - a suffix list L_S if there is no element in L^{new} such that $o'_{(l)}$ is its predecessor.
- the new candidate solution is obtained by joining L_P, L^{new} and L_S;
- the rows of B^{new} corresponding to the operations taken in the first step from L will be identical to the corresponding rows from B, while the other rows are taken from B'.

Let us consider, in the case of mill tube example, two candidate solutions: $L = [6, 10, 7, 5, 9, 4, 3, 8, 2, 1]$ and $L' = [10, 7, 9, 6, 8, 3, 5, 4, 2, 1]$ In the case when $l = 4$ and $r = 9$, applying the crossover operator leads to $L^{new} = [10, 7, 6, 5, 9, 4, 3, 8, 2, 1]$ which could be also generated from L by mutation (insertion of the first element on the third position). However, the crossover-like perturbation allows the generation of new candidates which would not be generated through one-step mutation. For instance, for the same L and L' from the above example, if $l = 2$ and $r = 5$ one obtains $L^{new} = [6, 10, 7, 5, 9, 8, 3, 4, 2, 1]$ which could not be obtained by mutation neither from L nor from L'. Ensuring the feasibility of the crossover result requires checking the precedence constraints, thus this operator induces a computational cost larger than that of mutation.

VII. Experimental analysis

A. Data generator

Despite the increasing interest in ASP, there are no benchmarks for the general assembly scheduling problem. Some of the works addressing the flexible assembly scheduling problem [8], [11] use benchmarks that have been originally proposed for flexible job-shop scheduling problems, e.g. Kacem benchmark [12] and Fattahi benchmark [13]. Recently, Talens et al. [14] proposed two extensive sets of instances for the 2-stage assembly scheduling problem, one corresponding to the case of one assembly machine and the other one corresponding to the case of several assembly machines.

Since the problems included in these benchmarks do not capture all characteristics of flexible assembly scheduling, we designed a problem generator that allows the generation of a large variety of assembly scheduling problems characterized by different operation networks and different sets of eligible machines.

The problem generator has a simple architecture, as illustrated in Figure 4. The generator uses:

- a *pool of products* that represent the entities included in the bill of materials corresponding to a client order. Currently, the generator uses a list of 200 of fictitious entities (which can be interpreted as products, components or make-parts), characterized by randomly generated names;
- a *pool of machines* that can be used to execute the operations. Currently, the generator uses a list of 20 available machines.

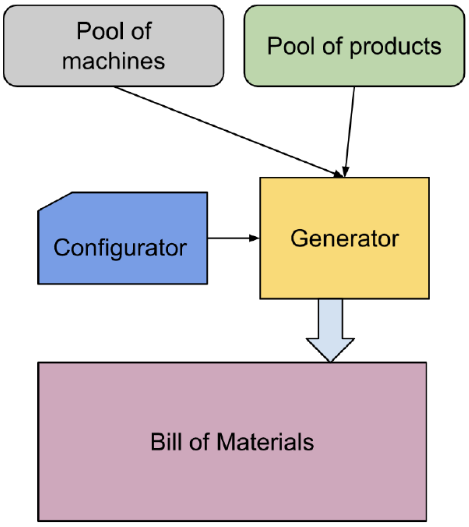


Fig. 4. Architecture of the assembly scheduling problems generator

The characteristics of a problem instance are specified in a configuration file that contains the values of the parameters describing the BOM structure.

In order to provide a high level of flexibility to the generated BOMs, the configurator module can be loaded with a multitude of parameters such as the number of levels in BOM, meaning the number of operations needed to be completed to create the final product, a maximum number of children for each node, meaning that a certain product is the result of assembling one or more simpler parts. One can specify the size of the product pool that is considered for this generator and the time horizon for scheduling. Also, the quantity of the final products can be provided. Another important characteristic that is supported is related to the maintenance applied to the machines involved in the product fabrication. The generator supports the setting of maintenance time intervals, the *Overall Equipment Effectivness* (OEE) for the machines and also one can specify the setup time for the machines if there is a need to change its purpose to support the processing of another product. The values can be generated using either an uniform distribution over the range of feasible values (the default case) or a truncated normal distribution.

The generator produces a JSON file containing the description of a tree-like structure corresponding to a BOM with a specified depth and a variable number of children, each node in the tree representing a specific product and including the generated set of eligible machines. The root node corresponds to the final product, while the children correspond to components that eventually will compose the final product.

B. Test problems

The experimental analysis is based on several problems instances/sets that have been generated such that various characteristics of the problem are emphasized:

- *A real-world case study:* the mill tube problem incorporating three orders of sizes 800, 320, and 160, each order requiring the production of the specified number of tubes according to the BOM described in Figure 1.
- *A deep BOM structure:* it is characterized by branches of up to 50 nodes in the operation network and a branching factor of 2 (for an operation there are at most two operations that directly precede it). The problem instance used in experiments contains 437 nodes.
- *A set of BOM structures with a variable number of operations:* it contains 15 problem instances corresponding to operation networks with a specified depth of 3 and a branching factor ranging between 2 and 16. The number of operations varies between 7 and 273. The quantity corresponding to each component (make-part) in the BOM is set to 10 (in this way each component on to the third level has a quantity equal to 1000). This set is characterized by rather wide structures and it was used to analyze the influence of the number of operations on the performance of exact, heuristic, and metaheuristic methods.

C. Methods and control parameters

The methods involved in the comparative analysis are:

- An *exact solver* (CPLEX) used to solve the problem described in section III-B. The solver is executed using 32 threads and CPLEX control parameters have been used with their default values. It should be mentioned that besides the problem described in section III-B which corresponds to the case when batch splitting is applied, a simplified version that does not require the B matrix as decision variables is also analyzed (referred to as standard in Tables III and IV).
- An implementation of the *LETSA heuristic* as it is described in section V. The standard variant does not use batch splitting while the variant with batch splitting (BS) applies a load-balancing strategy in order to distribute the operations per eligible machines. It should be mentioned that LETSA heuristic starts the construction of the schedule from the final operation, while the other heuristics start from the leaf operations in the operation network. Since the maintenance intervals influence the solution quality, the solution generated by LETSA is shifted such that the leaf operations are scheduled closer to the start time. LETSA does not require control parameters.
- A *Tabu Search (TS)* algorithm based on the mutation-like perturbation described in Section VI-D1. It should be mentioned that when a new candidate (neighborhood element) is constructed, the batch-size matrix (B) is perturbed with a probability equal to 0.15. The neighborhood size is set to 100 and the tabu-list size to 25. If the current candidate solution is not improved in the last 50 iterations then it is replaced with an element selected from the current neighborhood. The implementation uses 32 parallel search processes.
- A *Genetic Algorithm (GA)* which evolves a population of 100 elements by applying the same perturbation as in TS (with the same mutation probability for the batch-size matrix), the crossover operator described in Section VI-D2 and proportional selection. The crossover operator does not use control parameters.

It should be mentioned that the TS and GA implementations are adapted starting from the JSSP implementation available at [1].

All experiments have been run on a machine with 64 vCPUs and 256 GB RAM. The execution timeout was set to 1 hour for CPLEX and 3 minutes for LETSA, TS and GA. For TS and GA the reported makespan is the average value of 30 independent runs.

D. Results and discussion

The results obtained for the mill tube study case are presented in Table III which contains the makespan values (in hours) corresponding to various methods. Since the BOM structure is rather simple and most of the operations use distinct machines the optimal solution is obtained by CPLEX, TS and GA without batch splitting (BS). On the other hand, since the number of products and make-parts corresponding to all orders is rather large, the batch-splitting strategy improves the makespan by around 40% in the case of CPLEX, TS and GA, and around 20% in the case of LETSA.

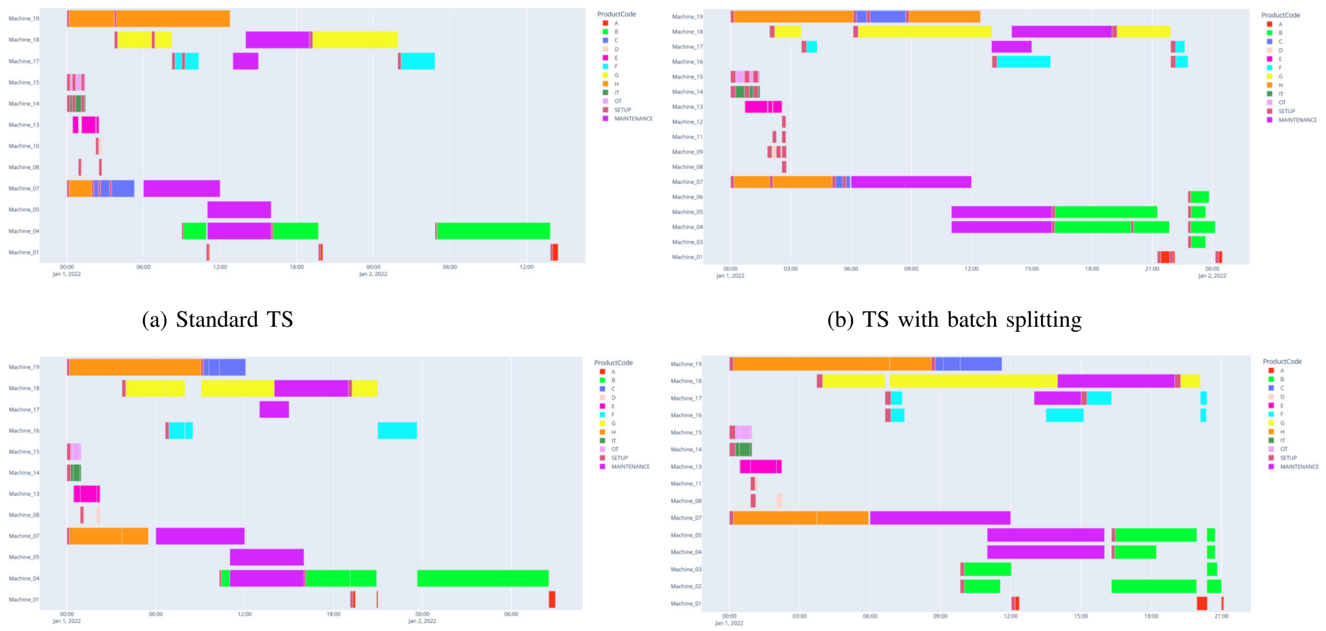
The control of the maintenance intervals and setup times corresponding to successive execution of batches of identical operations can further improve the makespan but not significantly. The Gantt charts (Figure 5) illustrate the fact that the benefit of BS is influenced by the number of machines on which the batch of operations can be distributed.

The positive impact of batch splitting is illustrated also in the case of the operation networks with deep structure (see Table IV), particularly in the case of TS. The poorer behavior of GA can be explained by the fact that the crossover operator is more time expensive than mutation and, because of the imposed limit of time, the exploration of the search space is limited. It should be noted that in this case, CPLEX could not provide a solution in the allocated amount of time (one hour).

TABLE III
MAKESPAN VALUES (IN HOURS) FOR THE TUBE MILL PROBLEM. VARIANTS OF THE ALGORITHMS: STANDARD, BS (WITH BATCH SPLITTING), MSC (WITH CONTROL ON THE MAINTENANCE INTERVALS AND SETUP TIMES)

	CPLEX	LETSA	TS	GA
standard	38.47	40.53	38.47 ± 0	37.42 ± 0.01
BS	21.87	31.23	22.07± 0	24.20 ± 0.75
MSC	38.30	35.22	38.30 ± 0	32.98 ± 0
BS + MSC	21.69	30.37	**21.31 ± 0.15**	21.71 ± 0.52

[1] https://job-shop-schedule-problem.readthedocs.io/en/stable/index.html



(a) Standard TS

(b) TS with batch splitting

(c) TS with MSC

(d) TS with batch splitting and MSC

Fig. 5. Gantt charts for the Tube mill problem generated using Tabu Search

TABLE IV
MAKESPAN VALUES (IN DAYS) FOR THE PROBLEM INSTANCE WITH DEEP STRUCTURE. VARIANTS OF THE ALGORITHMS: STANDARD, BS (WITH BATCH SPLITTING), MSC (WITH CONTROL ON THE MAINTENANCE INTERVALS AND SETUP TIMES)

	LETSA	TS	GA
standard	36.67	36.48 ± 0.87	37.41 ± 0
BS	34.06	27.28 ± 0.46	36.43 ± 0.11
MSC	35.22	35.47 ± 0.23	36.65 ± 0.19
BS + MSC	32.51	**25.79 ± 0.35**	35.17 ± 0.20

To analyze the scalability of the investigated methods we used the set of BOM structures with variable number of operations in the context when a set of 10 machines are available and an operation can be executed on at most 5 machines with different or similar characteristics.

From Figure 6 it can be observed that the exact solver was able to find solutions for problems having up to 183 operations in a time interval of one hour. TS and GA heuristics outperform LETSA heuristic, and TS is slightly better than GA.

VIII. Conclusions and Further Work

The particularities of the addressed assembly scheduling problems required the incorporation of some specific decision variables and constraints. Most mathematical programming models used in flexible job-shop scheduling include binary variables which encode the order between any two operations scheduled on the same machine that leads to $n^2 \cdot m$ binary variables (in the case of n operations and m machines). The proposed mathematical programming model avoids the usage of such a large number of binary variables, but it uses instead the starting and completion time values to enforce the precedence constraints.

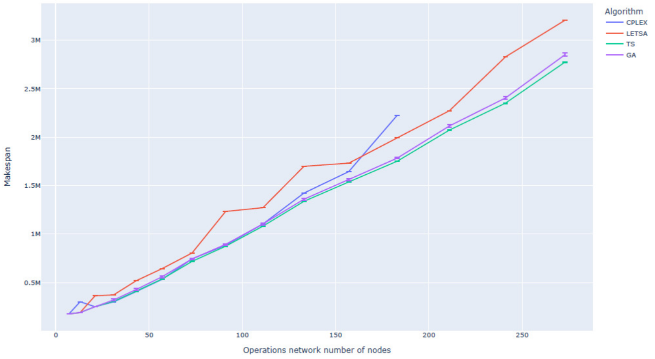


Fig. 6. Scalability results for standard variants of CPLEX, LETSA, TS, and GA for the 15 BOM structures

The proposed problem description and the candidate solution encoding allow specifying the distribution of (sub)batches of identical operations over several machines which led to a significant reduction in the makespan, particularly in the case of large orders.

The strategy that takes into account the maintenance time intervals and removes the unnecessary setup activities proved also to be beneficial but to a lesser extent.

As the operator inspired by the precedence preserving order-based crossover proved to be computationally intensive, we will further investigate other operators which preserve the feasibility of candidate solutions. Since the critical path heuristic incorporated in LETSA generates relatively good solutions in a fraction of the time required by the metaheuristic algorithms, a further step would be to consider the sub-optimal solution produced by LETSA among the initial candidate solutions for the metaheuristic algorithms.

We also plan to conduct a systematic scalability analysis based on sets of test problems including various typologies of operation networks and interactions between operations with respect to the lists of eligible machines, as reflected in the corresponding conjunctive graphs of the scheduling problem.

References

[1] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Computers & Operations Research*, vol. 142, 2022. doi: https://doi.org/10.1016/j.cor.2022.105731

[2] T. Morton and D. W. Pentico, *Heuristic scheduling systems: with applications to production systems and project management*. John Wiley & Sons, 1993, vol. 3.

[3] W. Lin, Q. Deng, W. Han, G. Gong, and K. Li, "An effective algorithm for flexible assembly job-shop scheduling with tight job constraints," *Int. Trans. Oper. Res.*, vol. 29, no. 1, pp. 496–525, 2022. doi: 10.1111/itor.12767. [Online]. Available: https://doi.org/10.1111/itor.12767

[4] Q. Liu, X. Li, H. Liu, and Z. Guo, "Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art," *Applied Soft Computing*, vol. 93, p. 106382, 2020.

[5] A. Agrawal, G. Harhalakis, I. Minis, and R. Nagi, "'just-in-time'production of large assemblies," *IIE transactions*, vol. 28, no. 8, pp. 653–667, 1996.

[6] S.-G. Dastidar and R. Nagi, "Batch splitting in an assembly scheduling environment," *International Journal of Production Economics*, vol. 105, no. 2, pp. 372–384, 2007.

[7] H.-y. Wang, Y.-w. Zhao, X.-l. Xu, and W.-L. Wang, "A batch splitting job shop scheduling problem with bounded batch sizes under multiple-resource constraints using genetic algorithm," in *2008 IEEE Conference on Cybernetics and Intelligent Systems*. IEEE, 2008, pp. 220–225.

[8] X. Li, J. Lu, C. Yang, and J. Wang, "Research of flexible assembly job-shop batch–scheduling problem based on improved artificial bee colony," *Frontiers in Bioengineering and Biotechnology*, vol. 10, p. 909548, 2022.

[9] F. Chan, T. Wong, and L. Chan, "Lot streaming for product assembly in job shop environment," *Robotics and Computer-Integrated Manufacturing*, vol. 24, no. 3, pp. 321–331, 2008. doi: https://doi.org/10.1016/j.rcim.2007.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584507000063

[10] ——, "The application of lot streaming to assembly job shop under resource constraints," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 14 852–14 857, 2008. doi: https://doi.org/10.3182/20080706-5-KR-1001.02514 17th IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016413790

[11] A. Maoudj, B. Bouzouia, A. Hentout, A. Kouider, and R. Toumi, "Distributed multi-agent scheduling and control system for robotic flexible assembly cells," *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1629–1644, 2019. doi: 10.1007/s10845-017-1345-z. [Online]. Available: https://doi.org/10.1007/s10845-017-1345-z

[12] I. Kacem, S. Hammadi, and P. Borne, "Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 1, pp. 1–13, 2002. doi: 10.1109/TSMCC.2002.1009117

[13] P. Fattahi, M. S. Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *J Intell Manuf*, vol. 18, p. 331–342, 2007. doi: https://doi.org/10.1007/s10845-007-0026-8

[14] C. Talens, P. Perez-Gonzalez, V. Fernandez-Viagas, and J. M. Framiñan, "New hard benchmark for the 2-stage multi-machine assembly scheduling problem: Design and computational evaluation," *Comput. Ind. Eng.*, vol. 158, p. 107364, 2021. doi: 10.1016/j.cie.2021.107364. [Online]. Available: https://doi.org/10.1016/j.cie.2021.107364

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 627–636

DOI: 10.15439/2023F9829
ISSN 2300-5963 ACSIS, Vol. 35

Current Trends in Automated Test Case Generation

Tomas Potuzak
0000-0002-8140-5178
Department of Computer Science and Engineering/
NTIS – New Technologies for the Information Society,
European Center of Excellence, Faculty of Applied
Sciences, University of West Bohemia
Univerzitni 8, 306 14 Plzen, Czech Republic
Email: tpotuzak@kiv.zcu.cz

Richard Lipka
0000-0002-9918-1299
NTIS – New Technologies for the Information
Society, European Center of Excellence/Department
of Computer Science and Engineering, Faculty of
Applied Sciences, University of West Bohemia
Univerzitni 8, 306 14 Plzen, Czech Republic
Email: lipka@kiv.zcu.cz

***Abstract*—The testing is an integral part of the software development. At the same time, the manual creation of individu-al test cases is a lengthy and error-prone process. Hence, an intensive research on automated test generation methods is ongoing for more than twenty years. There are many vastly dif-ferent approaches, which can be considered automated test case generation. However, a common feature is the generation of the data for the test cases. Ultimately, the test data decide the prog-ram branching and can be used on any testing level, starting with the unit tests and ending with the tests focused on the behavior of the entire application. The test data are also mostly independent on any specific technology, such as programming language or paradigm. This paper is a survey of existing litera-ture of the last two decades that deals with test data generation or with tests based on it. This survey is not a systematic literature review and it does not try to answer specific scientific questions formulated in advance. Its purpose is to map and categorize the existing methods and to summarize their common features. Such a survey can be helpful for any teams developing their methods for test data generation as it can be a starting point for the exploration of related work.**

***Index terms*—Software testing, test case generation, test data generation, papers survey.**

I. Introduction

TESTING is an essential part of software development. At the same time, the manual creation of individual test cases is a lengthy and error-prone process. In many real-world projects, there is not enough time to ensure sufficient testing of the developed software product, which leads to its lowered quality. The programmers of the test cases can also miss some inputs, which leads to unexpected behavior of the software product. Hence, an intensive research of automated test generation methods is ongoing for more than twenty years, as can be seen, for example, in [1] or [2].

In the existing literature, automated test case generation is used or at least proposed on various testing levels. These levels include unit testing focused on the functionality of isolated features of the developed application (usually a method, procedure, or function), but also the regression and integration testing focused on the correct cooperation of the individual parts of the application. Automated test case gene-ration can also be used during the high-level testing of the functionality of the entire application and its adherence to the specified requirements. Automated test case generation is tempting and seems to be promising, as it should reduce the time the programmers spend on manual test case preparation. Nevertheless, there are several limitations.

First of all, it is difficult to automatically verify that the tested application or its part provides correct results. This would require generating the expected outputs for all the generated inputs, which is an inherently difficult task. Never-theless, this ability is crucial for the usage of the automated test case generations in real software projects. However, it should be noted that, in many cases, it is possible to detect the incorrect behavior of an application even without the known correct outputs. An obvious example is when the application crashes, but there can also be limitations of the outputs, which can be used for incorrectness checking (e.g., the calculated volume of a cube cannot be negative).

Another issue, which is often discussed (e.g., in [3]) is related to the combinatorial explosion. Consider a unit test of a method with several parameters where various combinations of the parameters should be considered. Even when the parameters can be grouped into several discrete classes, the number of all the possible combinations grows very fast with the growing number of parameters and classes. This problem is even more pronounced in higher-level tests, when multiple methods are executed during one higher-level functionality testing. Various settings and running environments of the tested application only worsen this problem. Hence, even in tools, which are used in real projects, such as EvoSuite or Randoop [4], the number of generated test cases can be very high, which leads to long running times. This partially limits the usability of automated testing. Nevertheless, the problem can be mitigated by employing efficient test case selection in

This work was supported by Institutional support for long-term strategic development of research organizations.

order to generate and run the test cases, which provide the highest expected code coverage and/or have the highest expected error detection rate. The increasing power of contemporary computers is also helpful, as running a huge number of tests is more and more feasible.

The last issue, we would like to mention, is the validation of the automated test-generating methods themselves. Several different approaches for the evaluation of the automated test-generating methods can be found in the existing literature. From the practical usability of the methods in real projects point of view, there are two most important questions – how realistic the methods are and how well they perform in finding different types of realistic errors.

There are many different approaches, which can be considered automated test case generation. However, a common feature is the generation of the data for the test cases. Ultimately, the test data decide the program branching and can be used on any testing level, from the unit tests to the tests focused on the behavior of the entire application. The test data are also mostly independent on any specific technology, such as programming language or paradigm.

This paper is a survey of existing literature of the last two decades that deals with automated test data generation or with tests based on it. This survey is not a systematic literature review and it does not try to answer specific scientific questions formulated in advance. Its purpose is to map and categorize the existing methods and to summarize their common features. Such a survey can be useful for any teams developing their methods for test data generation as it can be a starting point for the exploration of related work.

The remainder of this paper is structured as follows. Related surveys are discussed in Section II. The selection of the papers for this survey is described in Section III. The existing methods for test data generation are discussed in Section IV. Their common features and trends are described in Section V. Threats to validity are described in Section VI. The conclusions and the future work are in Section VII.

II. Related Work

There are multiple studies, which survey the existing testing approaches. Our survey is intended to complement them from the automated test data generation point of view.

A. *Existing Methods Studies*

Ref. [5] summarizes the methods for test generation based on control flow analysis, automatic random data generation, and program execution analysis and/or the methods designed to produce tests, which maximizes the code coverage. The majority of the methods described in this survey is designed to deal only with simple program constructions and are often based on the models of the program instead of real programs. This is quite understandable, since the survey is rather old (from 1999). Nevertheless methods based on the same principles repeat again and again in more modern papers, only the methods or at least the examples, on which the methods are demonstrated, are usually more complex. For example, a more recent orchestrated survey [6] is focused on adaptive random testing among other methods.

A thorough review in [7] focuses on the papers dealing with search-based test case generation. The review makes it obvious that there is a constant increase in the number of testing-related publications between 1995 and 2007. The main focus of the review is the quality of the verification of the test generation methods. It is concluded that there is a lack of a standardized rigorous method to perform, asses, and compare the individual methods. Moreover, in many papers, there is even not enough empirical data to perform any comparison. It is also pointed out that, while many methods can achieve relatively high code coverage, it is not clear, whether the tests covering the code are able to find errors in the code. Another survey focused on search-based test case generation can be found in [8].

The search-based testing with an emphasis on mutation-based methods is also the theme of the survey in [9]. The methods described in papers published between the year 1996 and 2014 are based on genetic algorithms, ant colony optimization, simulated annealing, or hill climbing. The survey discusses also the relations and development of the methods in multiple papers. There are several conclusions. One is that the above-mentioned meta-heuristics significantly reduce the number of generated test cases without negative effects on the code coverage. Another is that the automated test generation methods are not designed for the concurrency problems. The last conclusion is that the comparability of the automated methods is difficult, similarly to [7].

The review in [10] focuses on the dynamic symbolic execution. There are twelve tools, which are compared based on various features, such as the number of publications dedicated to each tool, the utilized method for automated test generation, and the environment, in which the tool can be used. The ability of the tools to detect errors in the software is not among the investigated features. This feature is investigated in [11], which is focused on the methods utilizing aspect-oriented programming (namely Wrasp, Aspectra, Raspect, and EAT). One of the conclusions is that the structural evolutionary testing (EAT) shows the most promising results but at the cost of greater effort compared to random testing.

The short survey in [12] focuses on papers dealing with test data generation. It discusses various types of data generation from their architecture and usage points of view. The advantages and disadvantages of the methods as well as the best practices are discussed.

Although the majority of the surveys described above are focused on a technology or a set of technologies, there are also surveys focused on a specific type of software. An example is a systematic literature review [13], which deals with automated functional testing of mobile applications. Another example is a study [14], which discusses application of several different techniques for verification of flight software in Jet Propulsion Laboratory.

B. Practical Usability Studies

There are also studies, which focus on the usage of the automated testing methods in real projects, such as [15]. This study is not focused on published papers, but rather describes how the testing methods are used in real software projects and how the automated testing methods would improve the situation. The study has a bit darker tone than the studies mentioned in Section II.A as it points out that there is a lot of additional effort necessary when a promising method described in a paper should be used in a real industry project.

The study described in [16] is focused on the comparison of existing tools for automated test generation, such as Randoop, AutoTest, AnalitiX, Jtest, and so on. This study describes how the comparison of different methods for automated test generation should look like – precisely the aspect, which was mentioned as missing in [7] and [9]. In [16], a complex benchmark consisting of over 30 cases is described, which enables to empirically determine whether the automated test generation methods are able to uncover specified conditions. The results from this benchmark can be used for comparison of the methods. Although this benchmark is a good basis for the comparison of the automated test generation methods, it still utilizes synthetic cases, not real software [16].

An unorthodox practical study is the Java Unit Testing Tool Contest, which is held annually and its results are reported at various conferences (e.g., in [17] or [18]). The contest is intended for test generation tools designed for Java. Their ability to find errors in programs is tested using a benchmark consisting of real-life classes taken from various open-source GitHub projects. The contesting tools are evaluated based on the code coverage and mutation score [17], [18].

III. Survey Description

This paper is an intermediate result of our exploratory work to create a substance for a systematic literature review, which is the main aim of our current and future work (see Section VII). Although this intermediate result is only a (non-systematic) survey, the collection of the primary studies was performed in a rigorous manner described in following subsections, as the collected papers will also form part of the basis for our future systematic literature review.

A. Papers Searching

As the sources of the papers, we used the IEEE Xplore[1] library, which includes full texts of a large number of technology-related papers from both conferences and journals and the ScienceDirect[2] library, which includes papers from a large number of technology-related journals. Due to the institutional subscription, we have access to the majority of the full texts of the papers contained in both libraries, which is essential for the survey. Both libraries enable basic and advanced searching, but the available filters are quite different. For this reason, we used different settings for each library to obtain manageable numbers of relevant results. We made several attempts with various filters and search strings before we reached the final settings for both libraries.

The final search string for the IEEE Xplore library was "automated test data generating". It was used together with two filters. The year of publication had to be from 2000 to 2022 and the publication topic had to be "Program Testing". Using this setting, 461 results were obtained. The final string for the ScienceDirect was "automated test data generating program testing". It was used with three filters. Similarly to IEEE Xplore, the year of publication had to be from 2000 to 2022. Additionally, the subject area had to be "Computer Science" and the title of the paper had to contain "test data". Using this setting, 58 results were obtained. The searching in both databases was performed in April 2023.

B. Papers Filtering

From the search results, only the papers focused on the issues of automated test data generation for software testing, were selected. In first round, the selection was performed based on the titles. In second round, the selection was performed based on the abstracts, but only from the papers, which passed the first round. After the second round, there were 179 papers left (see Table I). The full texts were downloaded and investigated only for the 179 papers, which passed the second selection round. From these papers, some were eliminated from further processing, because, despite the promising title and abstract, the theme of the paper was outside the scope of this survey. Of the remaining papers, only 67 were included into the study, because they best represent the current trends in test data generation.

It should be noted that many of the obtained papers were already processed during our preliminary work with different search strings and filter settings in 2022. Hence, only the newest papers and papers not obtained previously due to different search settings of the libraries had to be processed. This enabled us to finish the paper in a relatively short time after the final search was performed.

C. Aims of the Survey

As this survey is intended to serve as a starting point for the exploration of related work for research teams dealing with automated test data generation, the aims of the survey can be summarized as follows:

- To categorize existing automated test data generation methods (see Section IV).
- To summarize and discuss common features of the methods (including their verification, implementation availability, testing level, and target platform) and observable trends (see Section V).

TABLE I SUMMARY OF THE NUMBERS OF SELECTED PAPERS

Library	Search results count	Selected papers count
IEEE	461	136
ScienceDirect	58	43

[1] https://ieeexplore.ieee.org

[2] https://www.sciencedirect.com

IV. Existing Methods in Literature

The categorization of the surveyed automated test data generation methods was performed based on the primary technology used for the test data generation. This categorization enables the readers to focus mainly on the papers related to the technology of their interest. It is also consistent with the existing surveys, as they are often focused on a relatively narrow set of technologies (see Section II). The papers of individual categories are discussed in following subsections.

A. Pseudorandom Generation-based Methods

The most basic approach, how to obtain test data, is to generate them using pseudorandom generators. Though the basic method can give relatively good results (e.g., code coverage) for number inputs, its usage for a more complex (and valid) data, such as specific strings or objects is difficult. Nevertheless, pseudorandom number generation is often combined with other approaches. In [19], the stochastic process models of the objects and their random initiation is used together with random method invocation.

In [20], the pseudorandom generating is combined with the constraint solving for the generation of test data for relational database schemas. The testing of object-relational mapping (ORM) based on the pseudorandom generation and formal models is described in [21]. In [22], data description using XML and regular expressions is used together with pseudorandom generating to generate invalid and atypical testing inputs for robustness testing.

B. Control-Flow-based Methods

The control-flow-based methods create control-flow graphs of the tested program using, for example, the static analysis. From these graphs, the tests are generated. A common aim is to achieve a high code coverage, which can be observed for example in [23], [24], or [25].

The method described in [24] is rather basic. It generates input data for the tested program in order to ensure the execution of all branches of the program. The number of generated test cases is limited by the elimination of already explored paths in the control-flow diagram. However, the method is limited to the numerical inputs only. A similar limitation can be also observed in [23].

The control-flow-based methods are often used for web applications. The method described in [25] is designed for the testing of the frontend of web-based applications. It analyzes the content and structure of the investigated website, creates the possible paths of the user, and generates the input testing data for the web forms in order to ensure path coverage. In [26], a method for the generation of test data for testing REST APIs is described. The connected control flow graphs are traversed in order to find patterns of variable usage to produce usable variable values. Another example of the usage for the web application can be found in [27].

The control-flow-based methods are also quite often combined (among other technologies) with the pseudorandom generation of the input data. In [28], stochastic hill climbing is used for the finding the probabilistic distribution. This distribution is then used for the generation of the pseudorandom input testing data. The combination of control-flow diagrams and pseudorandom data generation can be found also in [29].

C. Specification-based Methods

The specification-based methods utilize a form of the specification of the investigated software to generate the test cases. This approach is tempting, as it should compare the actual behavior of the software with the expected behavior given by its specification. The existing methods utilize the UML models (e.g., in [30], [31], [32], or [33]), specification of use cases (e.g., in [31], [34], [35], or [36]), or contracts (e.g., in [37]). Program states description is utilized in [38].

In [30], tests of the entire system are generated from the UML use case and state diagrams. From these diagrams, a usage model is created, which is then used as the basis for the tests. In [32], the activity diagram, the sequence diagram, and the system testing graphs are used to create a combination graph, which is then explored using a modified Depth-First Search (DFS) to generate expected test cases. The contracts in [37] are used similarly to the use case diagrams in [30]. They are transformed into models describing the expected behavior of the investigated program. From this form, the executable test cases are created.

The method described in [35] utilizes textual use case specifications for the generation of acceptance tests. The method is based on natural language processing (NLP) and constraints solving. In [36], the use cases are used to generate a control flow graph and a NLP table, which are, in turn, used for test case generation. The method described in [39] is designed for process-driven applications. The method utilizes analysis of the application and the specification of tests to generate test codes.

D. Program Execution Analysis Methods

The methods based on the program execution analysis utilize the observation of the application behavior in order to generate test cases. There are two main approaches – the approaches based on the instrumentation and on the dynamic symbolic execution (also known as concolic testing).

First approach is based on instrumentation of the tested application in order to enable a simple observation of its behavior. Examples include wrappers around tested functions or methods (e.g., in [40]) or probes near important points of the program, such as control structures (e.g., in [41]), usage of augmented virtual machines (e.g., LLVM [42]), or usage of runtime instrumentation (e.g., in [43]).

Second approach is used for example in [44], [45], [46], [47], [48], or [49]. A dynamic symbolic execution is used in [45] to observe the behavior of the tested application. This observation is used for checking whether new randomly generated input data lead to better path coverage than already stored paths. In [48], the dynamic symbolic execution works with additional attributes enabling to check the efficiency of the paths produced based on the random input data. It is also possible to check whether the expected boundary values described by the contracts are observed. In [50], the dynamic symbolic execution is used for the testing of C++ Qt Framework classes. A source code preprocessing phase is used to find constructors of Qt classes parameters. A similar approach is used in [51], but for C++ templates.

In [52], automated guided symbolic execution combined with constraint solving is used to avoid exploring useless paths in the program. The method is used for system vulnerability detection. In [44], preprocessing of enterprise applications to enable usage of existing symbolic execution tools for their testing is described. In [53], the tested program is transformed into a set of constraints, which are then solved using a symbolic reasoning engine. So, the approach resembles the dynamic symbolic execution. The evaluation of the CREST concolic testing tool's ability to find real-life errors in real embedded applications is described in [54]. In [43], a concolic test generation tool is combined with the automatic generation of test cases from a formal description of the program (e.g., database table definitions, process-flow diagrams, etc.).

E. Data-Description-based Methods

In some papers, the described methods are not focused on a program, but rather on the specification of the input testing data. This approach is quite common in relation to the increasing number of web-based applications and with the necessity to test their text-based APIs. The frequently used description formats include the Web Services Definition Language (WSDL) used for example in [55], [56], and [57] or the JavaScript Object Notation (JSON) used for example in [58]. XML Schema Definition (XSD) is used in [59].

An interesting comparison is described in [57] where a realistic WSDL-based data set is compared to a fully random data set. The conclusion is that the utilization of realistic data leads to a higher code coverage. In [53], a method for generating complex interconnected data from a WSDL specification is described. The method enables to generate both valid and invalid input data. In [60], a method for the preparation of the test data for web forms utilizes an ontology and types of the fields of the web form. In [61], existing data and rules for their converting were used for testing a data warehouse.

A quite different approach is used in [62]. It uses static analysis of existing tests for mining of literals, which can be suitable as input values in generated tests in a specific domain. Yet another different approach is described in [63]. There, the test cases are generated from inputs specification in natural language. Natural language processing (NLP) and key phrases detection are employed for this purpose.

F. Search-based Methods

A common aim of the search-based methods is to provide high code coverage with a relatively low number of generated test cases. These methods typically do not rely on the knowledge of the program structure, but rather employ various search meta-heuristics to find efficient input test data. Regardless of the utilized meta-heuristic, there must be a way to evaluate the solutions found by the heuristic. Hence, these methods are combined for example with models of the tested program behavior, such as the control flow [64] and event flow [65], or with the program instrumentation [66].

The commonly used meta-heuristics include genetic algorithms, which are employed, for example, in [67], [68], [69], [70], [71], or [72], ant colony optimization (e.g., in [73]), or particle swarm optimization (e.g., in [74] or [75]). A genetic algorithm is used for test data generation for unit testing of Java programs in [67]. In [76], a genetic algorithm is combined with grammar-based fuzzing to generate highly structured testing input data. In [77], a genetic algorithm is combined with random search and database instrumentation to generate test data for SQL queries testing. In [78], a genetic algorithm, an evolutionary algorithm, and an alternating variable method combined with an Object Constraint Language (OCL) description of constraints are investigated.

In [73], the ant colony optimization is employed to achieve higher branch coverage with a relatively small set of testing data. The method is based on the simulation of the pheromone path and is reported to provide better branch coverage than a standard genetic algorithm or particle swarm optimization. In [74], the particle swarm optimization is combined with formal specifications (written in SOFL) and mutation testing. Improved particle swarm optimization is also employed together with predicate functions and path similarity calculation in [75] for test case generation. An unspecified meta-heuristic is employed in [79] together with constraint solving of manually added constraints.

G. Machine-Learning-based Methods

The methods based on machine learning usually utilize artificial neural networks (ANNs) for the test data generation. In [80], a neural network is used for black-box testing of the graphical user interface (GUI) of Android applications. The input of the neural network is a set of screenshots of the tested application. In [81], generative adversarial networks are employed for automated test data generation. A neural network for test generation, which uses the execution trace of the program as an input, is employed in [82]. In [83], the dataset for the neural networks training for source code vulnerability detection is prepared using a mutation approach.

In [84], two approaches for test oracle generation are described. One is based on an artificial neural network and the second is based on data mining from decision trees. The advantages and limitations of both approaches are discussed. In [85], no artificial neural network is used. Instead, random forest, which is a generalization of tree-based classification, is employed for predictive mutation testing.

V. Common Features of Existing Methods

Regardless of the technology utilized by the methods described in Section IV, there are common features and issues of these methods discussed in following subsections.

A. Methods Verification

The lack of verification possibilities or of standard ways how to compare various methods is mentioned in several works (e.g., in [7] or [9]). Based on the investigated papers, it can be concluded that an objective comparison and assessment of the methods cannot be done by using the text of the papers only. Simply, there is not enough information and the provided examples and technologies are quite often vastly different. Some papers (e.g., [29]) contain only a very general description of the verification or testing of the proposed method. Some papers (e.g., [33]) contain no testing at all and focus solely on the description of the proposed method.

Nevertheless, some papers provide means for assessing the quality of the described methods, which are "above average". For example, in [38], [49], [61], or [78], very thorough descriptions of the evaluation process of the proposed methods can be found. It is reported that the evaluation process includes tests performed on realistic programs with actual errors found by the methods. This is in contrast with the majority of the paper, in which the methods are often demonstrated on quite simplified examples (e.g., in [67] or [75]).

B. Implementation Availability

It would be beneficial if the implementations of the methods described in individual papers were available for download and further trials. If this is not possible, a complete data set with data supporting the quality of the described method would be also quite informative. However, from the investigated papers, the majority does not enable to perform a replication study without a reimplementation of the methods from the description in the paper. Of the 67 primary studies referred in this survey, there were only 15 studies with direct links to tools with implementation of the described methods.

From the available tools, 11 tools are provided in the form of GitHub repositories (see Table II) and the remaining 4 tools have dedicated websites. The website of the CREST [54] also contains a link to the GitHub repository along with a downloadable `.zip` file. The website of the CATG [43] contains downloadable `.jar` files. The method described in [72] is implemented in the DCRTT, which appears to be a commercial product, as we were unable to find direct download links on the website. Finally, the website of the SDG [79] contains downloadable `.zip` file. As of May 21 2023, all the links are functional. The available tools are summarized in Table II.

TABLE II DIRECTLY AVAILABLE TOOLS

Ref.	Tool name	Link
[35]	UMTG	https://sntsvv.github.io/UMTG
[41]	Ocelot	https://github.com/ocelab/ocelot
[54]	CREST	https://www.burn.im/crest/
[43]	CATG	https://morioh.com/p/bfdc4686b614
[62]	TestMiner	https://github.com/lucadt/testminer
[72]	DCRTT	https://www.gsse.biz/products/DCRTT
[77]	EvoSQL	https://github.com/SERG-Delft/evosql
[79]	SDG	https://people.svv.lu/tools/SDG
[20]	DOMINO	https://github.com/schemaanalyst/ schemaanalyst
[22]	Data- Generators	https://github.com/simonpoulding/ DataGenerators.jl
[21]	CYNTHIA	https://github.com/theosotr/cynthia
[80]	Deep GUI	https://github.com/Feri73/deep-gui
[82]	Agilkia	https://github.com/PHILAE- PROJECT/agilkia
[85]	PMT	https://github.com/sei-pku/ PredictiveMutationTesting
[19]	SDgen	https://github.com/AussieGuy0/Sdgen

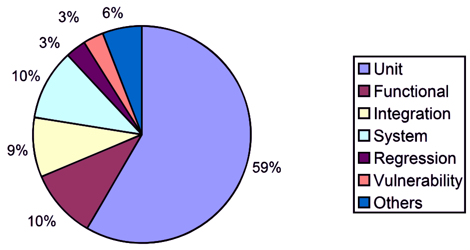


Fig. 1 Percentage of individual testing levels in primary studies

C. Testing Level

As it was stated in Section I, the automated test case generation methods exist for various testing levels. From the primary studies referred in this survey, the vast majority (specifically 39 papers) was focused on unit testing (see Fig. 1), for example [19], [23], [27], [59], or [69]. One of the possible reasons could be that the methods are often demonstrated on quite simple and/or short examples (see Section V.B). Short examples correspond well to unit tests, which usually deal with relatively short part of the source code with limited functionality.

As can be observed in Fig. 1, there were 6 testing levels, which were represented by more than one primary study (including the unit testing). There were papers focused on functional testing (7 papers, e.g., [20], [34], or [80]), integration testing (6 papers, e.g., [26] or [43]), system testing (7 papers, e.g., [25], [29], or [49]), regression testing (2 papers – [39] and [77]), and vulnerability testing (2 papers – [52] and [83]). There were also 4 other testing levels, each represented by a single primary study (4 papers, e.g., [22] or [64]). These papers/methods are grouped as "others" in Fig. 1 and 2.

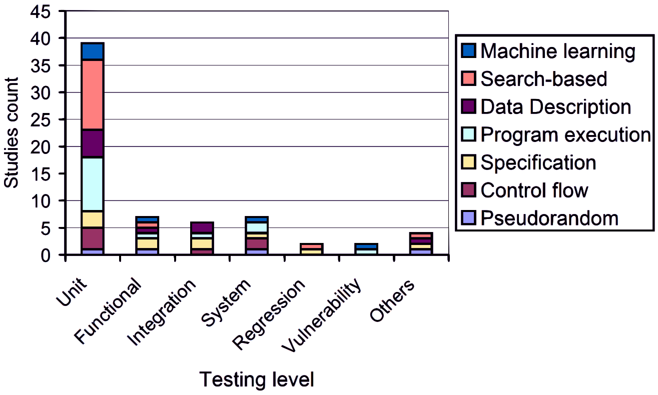


Fig. 2 Main utilized technologies for individual testing levels

In Fig. 2, the portions of the main utilized technologies of the methods for the individual testing levels are depicted. For the unit testing, there are comparatively high numbers of search-based methods (12 papers, e.g., [79], [81], or [84]) and program execution analysis methods (10 papers, e.g., [44], [48], or [51]). Together, they make up more than half of the primary studies focused on unit testing. For other testing levels, the methods are distributed relatively uniformly, but the numbers are too low to draw any further conclusions.

D. Target Platform

The methods described in primary studies are designed for a specific platform, for example for a specific programming language or a specific domain, such as web applications or databases. The methods can be also sufficiently general to be utilizable for multiple platforms. Such general methods usually do not use source code for the generations of the tests, but rather other forms of descriptions of the application, such as UML diagrams (e.g., [35] or [36]). There were 7 target platforms, which were represented by more than one primary study, including the generally utilizable methods (see Fig. 3). The generally utilizable methods also form just the largest group with 19 papers (e.g., [23] or [39]). The specific target platform with the largest number of papers was Java language (18 papers, e.g., [59] or [81]) followed by C/C++ languages (14 papers, e.g., [28] or [42]). Further groups include C#/.NET platform (2 papers – [48] and [56]), web applications (6 papers, e.g., [26] or [55]), databases (DB – 2 papers – [61] and [77]), and programmable logic controllers (PLCs – 2 papers – [46] and [47]). There were also 4 methods designed for other target platforms, each represented by a single primary study (4 papers, e.g., [64] or [74]). These papers/methods are grouped as “others” in Fig. 3 and 4.

In Fig. 4, the portions of the main utilized technologies of the methods for the individual target platforms are depicted. For the Java language, there are mostly search-based (6 papers, e.g., [76] or [79]) and then the machine-learning-based (3 papers – [80], [84], and [85]) and program execution analysis (3 papers – [43], [44], and [53]) methods. The program execution methods are prominent for the C/C++ programming languages (8 papers, e.g., [41] or [50]) and the data-description-based methods for the web applications (4 papers, e.g., [55] or [60]). The generally utilizable methods are mostly specification- (8 papers, e.g., [30] or [36]) and search-based (6 papers, e.g. [65] or [75]).

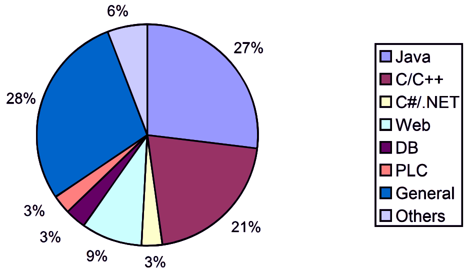


Fig. 3 Percentage of individual target platforms in primary studies

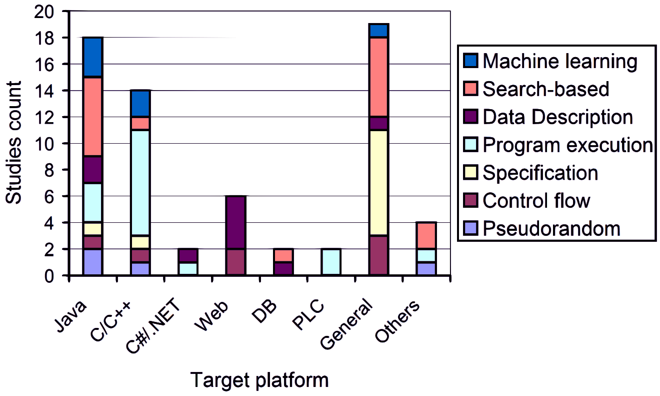


Fig. 4 Main utilized technologies for individual target platforms

E. Observable Trends

Since the time period of the analyzed primary studies is more than two decades (2000 to 2022), there are a few observable trends. Two technologies, which exist for a relatively long time, but are practically used only recently for the test case generation, are natural language processing (e.g., [35] or [36]) and artificial neural networks (e.g., [80] or [81]). Of the primary studies referred in this survey, the oldest study is from 2021 and 2020 for the NLP and the ANNs, respectively. This can be attributed to the relatively recent but significant progress in these fields leading to the practical usability of both technologies.

Another observable trend is the slight increase in the number of studies with direct links to the tools implementing the proposed methods (see Fig. 5). As can be observed in Fig. 5, studies with 11 of 15 available tools were published in 2017 and later. From the primary studies referred in this survey, there was no available tool before 2007.

VI. Threats to Validity

As pointed out in Section I, this survey is not a systematic literature review and does not attempt to answer specific research questions formulated in advance. It also does not attempt to exhaustively list all papers related to the test case or test data generation. Hence, there are papers, which would fit the theme of this survey, but we did not include them. There are several possible reasons:

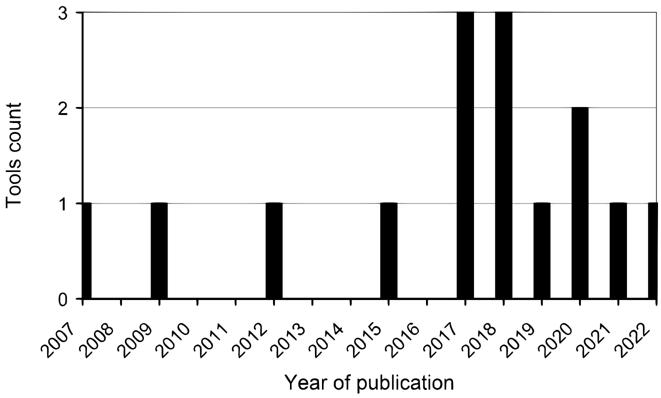


Fig. 5 Number of available tools in individual years

1. The paper was not discovered in the libraries, because it did not pass the utilized filters (see Section III.A).
2. The paper was not present in the two utilized libraries, but may be present in others.
3. The paper was discovered and its full text was read, but because of the similarity to other papers (in the sense of used techniques and/or their combinations), it was not included into the survey.

For the reasons described above, the reader should have in mind that this survey is not exhaustive in any sense, but tries to summarize the approaches and technologies currently in use in the field of automated test data generation.

VII. Conclusion and Future Work

In this paper, the existing literature that deals with test data generation or with tests based on test data generation was summarized. The commonly used approaches were discussed and their common issues and features were described including a few observable trends.

The collected primary studies, which this (non-systematic) survey summarizes, will be used as part of the basis for our future systematic literature review that will cover the theme of this survey, but will add specific research questions and formalization of the entire review process.

Another branch of our current and future work is the creation of a benchmark for the test data generation methods. Such a benchmark would allow us to objectively compare the ability of the methods to find known realistic errors. For this purpose, we are currently developing the Testing Applications Generator (TAG) [86]. This tool is intended to generate applications with selected introduced errors of various types. It enables to introduce errors on the method level meaning that each method can have several different implementations with various introduced errors. The resulting generated application is a general Java application with few limitations and with a structure of the entire project (not only source codes, but also libraries, additional files, and folder structure). The common types of errors should be also obtained during our future research. The tool will be used to create a set of several applications (with several versions each) with multiple introduced errors. This set will serve as the benchmark for automated test generation methods.

References

[1] N. Gupta, A. P. Mathur, and M. L. Soffa, "Generating test data for branch coverage," in Proceedings ASE 2000 - Fifteenth IEEE International Conference on Automated Software Engineering, Grenoble, September 2000, https://doi.org/10.1109/ASE.2000.873666

[2] P. Fröhlich and J. Link, "Automated Test Case Generation from Dynamic Models," in ECOOP '00: Proceedings of the 14th European Conference on Object-Oriented Programming, Cannes, June 2000, pp. 472-491, https://doi.org/10.1007/3-540-45102-1_23

[3] B. S. Ahmed, K. Z. Zamli, W. Afzal, and M. Bures, "Constrained Interaction Testing: A Systematic Literature Study," in IEEE Access, vol. 5, 2017, https://doi.org/10.1109/ACCESS. 2017.2771562

[4] M. M. Almasi, H. Hemmati, G. Fraser, A. Arcuri, and J. Benefelds, "An industrial evaluation of unit test generation: Finding real faults in a financial application," in Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), Buenos Aires, May 2017, pp. 263–272, https://doi.org/10.1109/ICSE-SEIP.2017.27

[5] J. Edvardsson, "A Survey on Automatic Test Data Generation," in Proceedings of the Second Conference on Computer Science and Engineering, Linköping, October 1999, pp. 21–28.

[6] S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, M. Harman, M. J. Harrold, and P. McMinn, "An orchestrated survey of methodologies for automated software test case generation," in The Journal of Systems and Software, vol. 86, no. 8, 2013, pp. 1978-2001, https://doi.org/10.1016/j.jss.2013.02.061

[7] S. Ali, L. C. Briand, H. Hemmati, and R. K. Panesar-Walawege, "A systematic review of the application and empirical investigation of search-based test case generation," in IEEE Trans. Softw. Eng., vol. 36, no. 6, 2009, pp. 742–762, https://doi.org/10.1109/TSE.2009.52

[8] P. McMinn, "Search-based software test data generation: a survey," in Softw. Test. Verif. Reliab., vol. 14, no. 2, 2004, pp. 105–156, https://doi.org/10.1002/stvr.294

[9] R. Jeevarathinam and A. S. Thanamani, "A survey on mutation testing methods, fault classifications and automatic test cases generation," in J. Sci. Ind. Res., vol. 70, no. 2, 2011, pp. 113–117.

[10] T. Chen, X. S. Zhang, S. Z. Guo, H. Y. Li, and Y. Wu, "State of the art: Dynamic symbolic execution for automated test generation," in Futur. Gener. Comput. Syst., vol. 29, no. 7, 2013, pp. 1758–1773, https://doi.org/10.1016/j.future.2012.02.006

[11] R. M. Parizi, A. A. A. Ghani, R. Abdullah, and R. Atan, "Empirical evaluation of the fault detection effectiveness and test effort efficiency of the automated AOP testing approaches," in Inf. Softw. Technol., vol. 53, no. 10, 2011, https://doi.org/10.1016/j.infsof. 2011.05.004

[12] S. Popić, B. Pavković, I. Velikić, and N. Teslić, "Data generators: a short survey of techniques and use cases with focus on testing," in 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, September 2019, https://doi.org/10.1109/ICCE -Berlin47944.2019.8966202

[13] P. Tramontana, D. Amalfitano, N. Amatucci, and A. R. Fasolino, "Automated functional testing of mobile applications: a systematic mapping study," in Software Quality Journal, vol. 27, 2019, pp. 149–201, https://doi.org/10.1007/s11219-018-9418-6

[14] A. Groce, K. Havelund, G. Holzmann, R. Joshi, and R.-G. Xu, "Establishing flight software reliability: testing, model checking, constraint-solving, monitoring and learning," in Annals of Mathematics and Artificial Intelligence, vol. 70, 2014, pp. 315–349, https://doi.org/10.1007/s10472-014-9408-8

[15] M. Bures, "Automated testing in the Czech Republic: the current situation and issues," in Proc. 15th Int. Conf. Comput. Syst. Technol., June 2014, pp. 294–301, https://doi.org/10.1145/2659532.2659605

[16] S. J. Galler and B. K. Aichernig, "Survey on test data generation tools: An evaluation of white- and gray-box testing tools for C#, C++, Eiffel, and Java," in Int. J. Softw. Tools Technol. Transf., vol. 16, no. 6, 2014, pp. 727–751, https://doi.org/10.1007/s10009-013-0272-3

[17] U. R. Molina, F. Kifetew, and A. Panichella, "Java Unit Testing Tool Competition: Sixth round," in SBST '18: Proceedings of the 11th International Workshop on Search-Based Software Testing, May 2018, pp. 22–29, https://doi.org/10.1145/3194718.3194728

[18] X. Devroey, S. Panichella, and A. Gambi, "Java Unit Testing Tool Competition: Eighth Round," in Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops, June 2020, pp. 545–548, https://doi.org/10.1145/ 3387940.3392265

[19] Y. Zheng, Y. Ma, and J. Xue, "Automated large-scale simulation test-data generation for object-oriented software systems," in Proceedings of the 1st International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007), Chengdu, November 2007, pp. 74-79, https://doi.org/10.1109/ISDPE.2007.104

[20] A. Alsharif, G. M. Kapfhammer, and P. McMinn, "DOMINO: Fast and Effective Test Data Generation for Relational Database Schemas," in 2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Västeras, April 2018, pp. 12– 22, https://doi.org/10.1109/ICST.2018.00012

[21] T. Sotiropoulos; S. Chaliasos, V. Atlidakis, D. Mitropoulos, and D. Spinellis, "Data-Oriented Differential Testing of Object-Relational Mapping Systems," in 2021 IEEE/ACM 43rd International Conferen-

ce on Software Engineering (ICSE), Madrid, May 2021, pp. 1535–1547, https://doi.org/10.1109/ICSE43902.2021.00137

[22] S. Poulding and R. Feldt, "Generating Controllably Invalid and Atypical Inputs for Robustness Testing," in Proceedings - 10th IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Tokyo, March 2017, https://doi.org/10.1109/ICSTW.2017.21

[23] N. T. Sy and Y. Deville, "Automatic test data generation for programs with integer and float variables," in Proc. 16th Annu. Int. Conf. Autom. Softw. Eng. (ASE 2001), San Diego, November 2001, pp. 13–21, https://doi.org/10.1109/ASE.2001.989786

[24] N. Gupta, A. P. Mathur, and M. L. Soffa, "Generating test data for branch coverage," in Proc. ASE 2000 15th IEEE Int. Conf. Autom. Softw. Eng., Grenoble, September 2000, pp. 219–227, https://doi.org/10.1109/ASE.2000.873666

[25] H. Huang, W.-T. Tsai, R. Paul, and Y. Chen, "Automated model checking and testing for composite Web services," in Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05), Seattle, May 2005, pp. 300–307, https://doi.org/10.1109/ISORC.2005.16

[26] D. T. Thu, L. D. Quang, D. A. Nguyen, and P. N. Hung, "A Method of Automated Mock Data Generation for RESTful API Testing," in Proceedings - 2022 RIVF International Conference on Computing and Communication Technologies (RIVF 2022), Ho Chi Minh City, December 2022, https://doi.org/10.1109/RIVF55975.2022.10013835

[27] D. T. Thu, D. A. Nguyen, P. N. Hung, "Automated Test Data Generation for Typescript Web Applications," in Proceedings – International Conference on Knowledge and Systems Engineering, Bangkok, November 2021, https://doi.org/10.1109/KSE53942.2021. 9648782

[28] S. Poulding and J. A. Clark, "Efficient software verification: Statistical testing using automated search," in IEEE Trans. Softw. Eng., vol. 36, no. 6, 2010, pp. 763–777, https://doi.org/10.1109/TSE. 2010.24

[29] J. Alava, T. M. King, and P. J. Clarke, "Automatic validation of java page flows using model-based coverage criteria," in Proc. - Int. Comput. Softw. Appl. Conf., Chicaco, September 2006, pp. 439–446, https://doi.org/10.1109/COMPSAC.2006.32

[30] M. Riebisch, I. Philippow, and M. Götze, "UML-Based Statistical Test Case Generation," in LNCS 2591, 2003, pp. 394–411, https://doi.org/ 10.1007/3-540-36557-5_28

[31] L. Bao-Lin, L. Zhi-shu, L. Qing, and C. Y. Hong, "Test Case automate Generation from UML Sequence diagram and OCL expression," in Proc. - 2007 Int. Conf. Comput. Intell. Secur., Harbin, December 2007, pp. 1048–1052, https://doi.org/10.1109/CIS.2007.150

[32] Meiliana, I. Septian, R. S. Alianto, Daniel, and F. L. Gaol, "Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm," in Procedia Computer Science, vol. 116, 2017, pp. 629–637, https://dx.doi.org/10.1016/j.procs.2017.10.029

[33] Y. Zheng, J. Xue, and Y. Zhu, "ISDGen: An automated simulation data generation tool for object-oriented information systems," in 2008 Asia Simul. Conf. - 7th Int. Conf. Syst. Simul. Sci. Comput., Beijing, October 2008, https://doi.org/10.1109/ASC-ICSC.2008. 4675401

[34] M. Zhang, T. Yue, S. Ali, H. Zhang, and J. Wu, "A Systematic Approach to Automatically Derive Test Cases from Use Cases Specified in Restricted Natural Languages," in LNCS, vol. 8769, 2014, pp. 142–157, https://doi.org/10.1007/978-3-319-11743-0_10

[35] C. Wang, F. Pastore, A. Goknil, and L. C. Briand, "Automatic Generation of Acceptance Test Cases from Use Case Specifications: An NLP-Based Approach," in IEEE Trans. on Softw. Eng., vol. 48, no. 2, 2022, https://doi.org/10.1109/TSE. 2020.2998503

[36] M. Lafi, T. Alrawashed, and A. M. Hammad, "Automated Test Cases Generation from Requirements Specification," in 2021 International Conference on Information Technology, Amman, July 2021, https://doi.org/10.1109/ICIT52682.2021.9491761

[37] D. Xu, W. Xu, M. Tu, N. Shen, W. Chu, and C. H. Chang, "Automated Integration Testing Using Logical Contracts," in IEEE Trans. Reliab., vol. 65, no. 3, 2016, pp. 1205–1222, https://doi.org/10.1109/TR.2015.2494685

[38] O. N. Timo and G. Langelier, "Test Data Generation for Cyclic Executives with CBMC and Frama-C: A Case Study," in Electron. Notes Theor. Comput. Sci., vol. 320, 2016, pp. 35–51, https://doi.org/10.1016/j.entcs.2016.01.004

[39] K. Schneid, L. Stapper, S. Thone, and H. Kuchen, "Automated Regression Tests: A No-Code Approach for BPMN-based Process-Driven Applications," in 2021 IEEE 25th International Enterprise Distributed Object Computing Conferenc (EDOC), Gold Coast, October 2021, https://doi.org/10.1109/EDOC52215.2021.00014

[40] C. Fetzer and Z. Xiao, "An automated approach to increasing the robustness of C libraries," in Proc. 2002 Int. Conf. Dependable Syst. Networks, Washington D.C., June 2002, pp. 155–164, https://doi.org/10.1109/DSN.2002.1028896

[41] S. Scalabrino, M. Guerra, G. Grano, A. De Lucia, R. Oliveto, D. D. Nucci, and H. C. Gall, "Ocelot: A search-based test-data generation tool for C," in ASE 2018 - Proceedings of the 33rd ACM/IEEE Int. Conf. on Autom. Softw. Eng., Montpellier, September 2018, pp. 868-871, https://doi.org/10.1145/ 3238147.3240477

[42] H. Riener and G. Fey, "FAuST: A framework for formal verification, automated debugging, and software test generation," in LNCS, vol. 7385, 2012, https://doi.org/10.1007/978-3-642-31759-0_17

[43] H. Tanno, X. Zhang, T. Hoshino, and K. Sen, "TesMa and CATG: Automated Test Generation Tools for Models of Enterprise Applications," in 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, Florence, May 2015, pp. 717–720, https://doi.org/10.1109/ICSE.2015.231

[44] H. Ohbayashi, H. Kanuka, and C. Okamoto, "A Preprocessing Method of Test Input Generation by Symbolic Execution for Enterprise Application," in 2018 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, December 2018, https://doi.org/10.1109/APSEC.2018.00104

[45] T. Su et al., "Automated Coverage-Driven Test Data Generation Using Dynamic Symbolic Execution," in 2014 Eighth Int. Conf. Softw. Secur. Reliab., San Francisco, June 2014, pp. 98–107, https://doi.org/10.1109/SERE.2014.23

[46] L. Hao, J. Shi, T. Su, and Y. Huang, "Automated Test Generation for IEC 61131-3 ST Programs via Dynamic Symbolic Execution," in 2019 International Symposium on Theoretical Aspects of Software Engineering (TASE), Guilin, July 2019, https://doi.org/10.1109/TASE.2019.00004

[47] W. He, J. Shi, T. Su, Z. Lu, L. Hao, and Y. Huang, "Automated test generation for IEC 61131-3 ST programs via dynamic symbolic execution," in Science of Computer Programming, vol. 206, 2021, https://doi.org/10.1016/j.scico.2021.102608

[48] K. Jamrozik, G. Fraser, N. Tillman, and J. De Halleux, "Generating test suites with augmented dynamic symbolic execution," in LNCS, vol. 7942, 2013, pp. 152–167, https://doi.org/10.1007/978-3-642-38916-0_9

[49] B. Chen, Z. Yang, L. Lei, K. Cong, and F. Xie, "Automated Bug Detection and Replay for COTS Linux Kernel Modules with Concolic Execution," in 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London (Canada), February 2020, https://10.1109/SANER48275.2020.9054797

[50] T. A. Bui, L. N. Tung, H. V. Tran, and P. N. Hung, "A Method for Automated Test Data Generation for Units using Classes of Qt Framework in C++ Projects," in 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), Ho Chi Minh City, December 2022, https://10.1109/RIVF55975.2022.10013869

[51] M. H. Do, L. N. Tung, H. V. Tran, and P. N. Hung, "An Automated Test Data Generation Method for Templates of C++ Projects," in 2022 14th International Conference on Knowledge and Systems Engineering (KSE), Nha Trang, October 2022, https://doi.org/10.1109/ KSE56063.2022.9953626

[52] T. Liu, Z. Wang, Y. Zhang, Z. Liu, B. Fang, and Z. Pang, "Automated Vulnerability Discovery System Based on Hybrid Execution," in 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), Guilin, July 2022, pp. 234-241, https://doi.org/10.1109/DSC55868.2022.00038

[53] K. Li, C. Reichenbach, Y. Smaragdakis, Y. Diao, and C. Csallner, "SEDGE: Symbolic example data generation for dataflow programs," in 2013 28th IEEE/ACM International Conference on Automated Software Engineering (ASE), Silicon Valley, November 2013, https://doi.org/10.1109/ASE.2013.6693083

[54] M. Kim, Y. Kim, and Y. Jang, "Industrial application of concolic testing on embedded software: Case studies," in 2012 IEEE Fifth Inte-

rnational Conference on Software Testing, Verification and Validation, Montreal, April 2012, https://doi.org/10.1109/ICST.2012.119

[55] C. Ma, C. Du, T. Zhang, F. Hu, and X. Cai, "WSDL-Based Automated Test Data Generation for Web Service," in 2008 Int. Conf. Comput. Sci. Softw. Eng., Wuhan, December 2008, pp. 731–737, https://doi.org/10.1109/CSSE.2008.790

[56] W. Krenn and B. K. Aichernig, "Test Case Generation by Contract Mutation in Spec#," in Electron. Notes Theor. Comput. Sci., vol. 253, no. 2, 2009, pp. 71–86, https://doi.org/10.1016/j.entcs.2009.09.052

[57] M. Bozkurt and M. Harman, "Automatically generating realistic test input from web services," in Proc. - 6th IEEE Int. Symp. Serv. Syst. Eng., Irvine, December 2011, pp. 13–24, https://doi.org/10.1109/SOSE.2011.6139088

[58] A. Arcuri, "RESTful API Automated Test Case Generation," in 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS), Prague, July 2017, pp. 9–20, https://doi.org/10.1109/QRS.2017.11

[59] N. Havrikov, A. Gambi, A. Zeller, A. Arcuri, and J. P. Galeotti, "Generating unit tests with structured system interactions," in 2017 IEEE/ACM 12th International Workshop on Automation of Software Testing (AST), Buenos Aires, May 2017, pp. 30–33, https://doi.org/10.1109/AST.2017.2

[60] S. Hanna and H. Jaber, "An Approach for Web Applications Test Data Generation Based on Analyzing Client Side User Input Fields," in 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, October 2019, https://doi.org/10.1109/ICTCS.2019.8923098

[61] H. M. Sneed and K. Erdoes, "Testing big data (Assuring the quality of large databases)," in 2015 IEEE Eighth Int. Conf. Softw. Testing, Verif. Valid. Work., Graz, April 2015, pp. 1–6, https://doi.org/10.1109/ICSTW.2015.7107424

[62] L. D. Toffola, C. A. Staicu, and M. Pradel, "Saying 'Hi!' is not enough: Mining inputs for effective test generation," in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), Urbana, October 2017, https://doi.org/10.1109/ASE.2017.8115617

[63] T. Li, X. Lu, and H. Xu, "Automated Test Case Generation from Input Specification in Natural Language," in 2022 IEEE International Symposium on Software Reliability Engineering Workshops, Charlotte, October 2022, https://doi.org/10.1109/ISSREW55968.2022.00076

[64] T. Shu, Z. Ding, M. Chen, and J. Xia, "A heuristic transition executability analysis method for generating EFSM-specified protocol test sequences," in Information Sciences, vol. 370–371, 2016, pp. 63–78, https://doi.org/10.1016/j.ins.2016.07.059

[65] A. Rauf, S. Anwar, M. A. Jaffer, and A. A. Shahid, "Automated GUI test coverage analysis using GA," in 7th Int. Conf. Inf. Technol. New Gener., Las Vegas, April 2010, pp. 1057–1062, https://doi.org/10.1109/ ITNG.2010.95

[66] S. Khor and P. Grogono, "Using a genetic algorithm and formal concept analysis to generate branch coverage test data automatically," in 19th Int. Conf. Autom. Softw. Eng., Linz, September 2004, pp. 346–349, https://doi.org/10.1109/ASE.2004.1342761

[67] Z. J. Rashid and M. Fatih Adak, "Test Data Generation for Dynamic Unit Test in Java Language using Genetic Algorithm," in 6th International Conference on Computer Science and Engineering (UBMK), Ankara, September 2021, https://doi.org/10.1109/UBMK52708.2021.9558953

[68] E. Diaz, J. Tuya, and R. Blanco, "Automated software testing using a metaheuristic technique based on Tabu search," in 18th IEEE Int. Conf. Autom. Softw. Eng., Montreal, October 2003, pp. 310–313, https://doi.org/10.1109/ASE.2003.1240327

[69] J. Khandelwal and P. Tomar, "Approach for automated test data generation for path testing in aspect-oriented programs using genetic algorithm," in Int. Conf. Comput. Com. Autom., Greater Noida, May 2015, pp. 854–858, https://doi.org/10.1109/CCAA.2015.7148494

[70] B. L. Li, Z. S. Li, J. Y. Zhang, and J. R. Sun, "An Automated Test Case Generation Approach by Genetic Simulated Annealing Algorithm," in Third Int. Conf. Nat. Comput., Haikou, August 2007, pp. 106–111, https://doi.org/10.1109/ICNC.2007.187

[71] Z. J. Rashid and M. F. Adak, "Test Data Generation for Dynamic Unit Test in Java Language using Genetic Algorithm," in 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, September 2021, http://dx.doi.org/10.1109/UBMK52708. 2021.9558953

[72] R. Gerlich and C. R. Prause, "Optimizing the Parameters of an Evolutionary Algorithm for Fuzzing and Test Data Generation," in 2020 IEEE 13th International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Porto, October 2020, https://doi.org/10.1109/ICSTW50294.2020.00061

[73] H. Sharifipour, M. Shakeri, and H. Haghighi, "Structural test data generation using a memetic ant colony optimization based on evolution strategies," in Swarm Evol. Comput., vol. 40, 2018, pp. 76-91, https://doi.org/10.1016/j.swevo.2017.12.009

[74] R. J. Cajica; R. E. G. Torres, and P. M. Álvarez, "Automatic Generation of Test Cases from Formal Specifications using Mutation Testing," in 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, November 2021, http://dx.doi.org/10.1109/CCE53527.2021.9633118

[75] H. Cui, L. Chen, B. Zhu, and H. Kuang, "An efficient automated test data generation method," in 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA), Changsha, March 2010, https://doi.org/10.1109/ICMTMA.2010.556

[76] M. Olsthoorn, A. van Deursen, and A. Panichella, "Generating Highly-structured Input Data by Combining Search-based Testing and Grammar-based Fuzzing," in ASE '20: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, December 2020, pp. 1224-1228, http://dx.doi.org/10.1145/3324884.3418930

[77] J. Castelein, M. Aniche, M. Soltani, A. Panichella, and A. Van Deursen, "Search-based test data generation for SQL queries," in Proceedings of the 40th International Conference on Software Engineering, Gothenburg, May 2018, pp. 1220–1230, https://doi.org/10.1145/3180155.3180202

[78] S. Ali, M. Zohaib Iqbal, A. Arcuri, and L. C. Briand, "Generating test data from OCL constraints with search techniques," in IEEE Transactions on Software Engineering, vol. 39, no. 10, 2013, pp. 1376–1402, https://doi.org/10.1109/TSE.2013.17

[79] G. Soltana, M. Sabetzadeh, and L. C. Briand, "Synthetic data generation for statistical testing," in 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), Urbana, October 2017, https://doi.org/10.1109/ASE.2017.8115698

[80] F. Y. B. Daragh and S. Malek, "Deep GUI: Black-box GUI Input Generation with Deep Learning," in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, November 2021, pp. 905–916, https://doi.org/10.1109/ASE51524.2021.9678778

[81] X. Guo, H. Okamura, and T. Dohi , "Automated Software Test Data Generation With Generative Adversarial Networks," in IEEE Access, vol. 10, 2022, https://doi.org/10.1109/ACCESS. 2022.3153347

[82] M. Utting, B. Legeard,F. Dadeau, F. Tamagnan, and F. Bouquet, "Identifying and Generating Missing Tests using Machine Learning on Execution Traces," in 2020 IEEE International Conference On Artificial Intelligence Testing (AITest), Oxford, August 2020, https://doi.org/10.1109/AITEST49225.2020.00020

[83] K. Cheng, G. Du, T. Wu, L. Chen, and G. Shi, "Automated Vulnerable Codes Mutation through Deep Learning for Variability Detection," in 2022 International Joint Conference on Neural Networks (IJCNN), Padua, July 2022, https://doi.org/10.1109/IJCNN55064.2022.9892444

[84] Vineeta, A. Singhal, and A. Bansal, "Generation of test oracles using neural network and decision tree model," in 2014 5th Int. Conf. - Conflu. Next Gener. Inf. Technol. Summit, Noida, September 2014, pp. 313–318, https://doi.org/10.1109/CONFLUENCE.2014.6949311

[85] J. Zhang, L. Zhang, M. Harman, D. Hao, Y. Jia,and L. Zhang, "Predictive Mutation Testing," in IEEE Transactions on Software Engineering, vol. 45, no. 9, 2019, pp. 898–918, https://doi.org/10.1109/TSE.2018.2809496

[86] T. Potuzak and R. Lipka, "Generation of Benchmark of Software Testing Methods for Java with Realistic Introduced Errors" in FedCSIS 2023 communication papers, September 2023, to be published

DOI: 10.15439/2023F3513
ISSN 2300-5963 ACSIS, Vol. 35

Analysis of a GPT-3 chatbot with respect to its input in a sales dialogue

Julian Premm[‡], Hagen Peukert[‡], Dennis Rössel[*] and Mareike Silber[†]
[‡]Universität Hamburg
Email: julian.premm@uni-hamburg.de
hagen.peukert@uni-hamburg.de
[*]Technische Universität Dresden
Email: dennis.roessel97@web.de
[†]Universität des Saarlandes
Email: mareike.silber@live.de

***Abstract*—The aim of the study at hand is to configure and evaluate a GPT-3 chatbot which is resistant to faulty input prompts and sensitive to the emotional setting of a sales dialogue. Design Science Research Methodology by Peffers et al. [46] was applied and evaluated with qualitative interviews in two conditions, that is, short and long language input. Results show that the chatbot was overall able to mimic human-like sales conversations. Some deviant behavior could be observed, especially in the short input condition, revealing more verbiage and insistent questions for purchase by the chatbot.**

I. Introduction

TEXT-BASED conversational agents, namely chatbots, have become increasingly popular in customer service, healthcare or businesses [1], [70]. A chatbot is a program based on artificial intelligence and natural languages processing (NLP) designed to communicate with humans [18]. It is however not only important how efficient and accurate the output of a chatbot is, but also that the input is interpreted correctly [36]. One quality measure of chatbots is robustness towards faulty input [48], [64], [44], [38]. This study looks further into the business domain by using a chatbot in the context of a sales dialogue. A sales dialogue is a dynamic communication process between a buyer and a seller which relies on identifying the buyer's needs such that a sale can be successfully carried out [52]. The chatbot employed here is based on GPT-3 from OpenAI [41], [30].

In contrast to traditional chatbots, which operate on predefined states and rules or match an input towards a predefined answer [18], generative models produce a given input word by word into an output such that the dialogue appears to be more human-like and does not rely on pre-defined answers. However, grammatical errors could occur depending on the available amount of training data and huge amounts of training data play a decisive role as the main requirement in generative models [67], [2], [50]. As the name suggests GPT-3 is based on a pre-trained model, which allows usage in a variety of contexts [41].

There are cases of misbehavior of chatbots reported in the media, which are a consequence of faulty inputs in the training data [69]. This indicates that the input for a chatbot could influence the conversation and possibly change the behavior of the chatbot altogether.

This study aims to further examine the linguistic input of a chatbot in the context of a sales dialogue. The chatbot could react towards faulty input and substantially decrease the quality of the ongoing dialogue by upsetting potential customers. Depending on how the input is interpreted by the chatbot, the emotional setting of the dialogue is likely to change. As a working hypothesis for this study, we like to raise the following question *How can a GPT-3 chatbot be designed and developed such that it is resistant to faulty input prompts and sensitive to the emotional setting in the context of a sales dialogue?*

We decided to restrict the sales dialogue towards buying smartphones. Statistics show that approximately 68.25 million people in Germany were smartphone users in 2022, equivalent to a smartphone penetration of 81.9% in 2022 [40]. Therefore, subjects of this study have likely been engaged in selecting an appropriate smartphone in their past.

Furthermore, we wanted to include the aspect of negotiation in our dialogue setting because negotiation requires enhanced communicative skills and recognizing abstract patterns [25], which would result in a more complex and human-like dialogue.

Literature research revealed that chatbots in the domain of sales have already been investigated. Lee [24] discusses four e-commerce chatbot usage cases in the process of a purchase and concluded that these chatbots have improved the convenience of customers' shopping, ordering, and payment experiences.

Balakrishnan and Dwivedi [3] generally discuss AI-powered digital assistants in conversational commerce, a term emphasized by Mayer and Harrison [33] and introduced by Messina [35]. Conversational commerce is buying activity of a customer interacting with a digital assistant. Balakrishnan and Dwivedi [3] conclude that anthropomorphism in digital assistants is crucial for creating a positive attitude and purchase intention. Therefore, it is beneficial if the chatbot mimics a human-like dialogue [3].

In order to make the chatbot more human-like, we named it "Melissa".

The following chapter explains the theoretical background of a sales dialogue. The chapter on Methods and Design Science Research Cycle gives a concise description how Design Science [46] was applied to the case of the chatbot, by also considering the affordance theory [13], [39]. An affordance is defined as a possibility for goal-oriented action afforded to specified user groups by technical objects [39], [32]. We performed evaluation by conducting interviews based on a questionnaire. In the fourth chapter, we will explain our results obtained during the interviews based on previously defined design principles [46]. Finally, in the last chapter, we discuss our results and outline possible limitations as well as an outlook towards further possible research.

II. Theoretical Background

Based on Lewis' AIDA model [58], an acronym for attention, interest, desire, and action, a sales dialogue can be perceived as a specific domain characterized by a more or less rigid sequence of customized events, vocabularies, and a clear understanding of objectives of the sales situation, i. e. satisfaction of a concrete consumer need that is compensated usually with some sort of monetary means. In our study we assume a buyer's market, in which the salesperson has an inherent interest in customer orientation and satisfaction. Staff will try to create a pleasant atmosphere built on positive emotional states of the respective client based on the assumptions that successful sales agents create trust and sympathy [51], [63], [22], [59], [31], [17]. Indeed, besides how a situation is conceived, a necessary condition is the availability of the desired product and a profound knowledge of all aspects of it (design, handling, prices, pros and cons of the product as well as user benefit). We set forth that the psychological principles of a sales dialogue also apply to the virtual world.

Since the planned scenery of human-CA-interaction was thought to be sales dialogues on mobile phones, the first step in our theoretical engagement was to look into what is known about sales dialogues among humans in general and how humans behave in a sales process as well as which psychological variables play a role in their behavior. The first thought to note is that sales dialogues follow a fairly strict pattern [65], [10] that may be broken down to more general phases like the opening, analysis of needs, product presentation, and closing. Each of the phases requires a different set of communicative skills and strategies [22]. There are also some sensible guidelines and tactics that were developed in practice, gained substantial relevance there and finally found their way to model building and theories [47]. These practical guides elaborate on similar stages and define more granular subcategories. In addition, these can be presented as flow diagrams, which qualify particularly for software implementation.

Yet the theory behind the four stages in sales dialogues is well founded [17]. Whereas the opening consists of codified communication (e. g. greetings, salutations) setting the tone for the rest of the dialogue, the analysis of needs is more analytical, partly based on a variety of indicators and general logic, but it also comprises the evaluation and processing of idiosyncratic information specific to a client. This stage is claimed to be the most challenging for sales in the analogous world [22] and it is plausible to assume this for chatbots as well [28]. Price estimates and price expectations are part of this stage. This information is often sensitive and dependent on the situation, should not be directly asked (anchoring). The product presentation is a more or less skillful derivation from the second phase. If knowledge on a wide selection of products is available (in structured and machine-readable form), effective algorithmic solutions exist for mapping needs to specific products. Again, a positive closing is important for the sensation, yet due to a rather codified situation, there are little new challenges for chatbots. Simply put, the vast majority of the scientific literature and practical marketers take as a basis some kind of models that comprise at least four main stages such as opening, need analysis, product presentation and closing. Need analysis is the most crucial part of a successful sales dialogue.

Research has also shown that chatbots which reveal empathic behavior while communicating with the users are perceived in a positive way and increased the trustworthiness towards the chatbot [20]. Agents which showed human-like behavior had a higher acceptance rate [6]. The emotional states, such as sympathy, joy, allegiance, but also anger or shame, are the decisive variables to create trust and a positive connection to the situation [51], [63]. It is also established that the kind of product is an important variable and as such has to be considered [22]. It is argued that walk-in customers have to be approached differently, i. e. with positive emotions, to foster ad hoc decisions. And, it is clear that the higher the involvement in the product (be it for status, prestige, price, or practicalities) more rational arguments need to be taken into account. However, this line of research should be embedded in the overall decision-making process of humans. There are hardly any decisions free of emotions, but they are justified by ex-post rational arguments long after the decision is subconsciously made [66], [19]. These seemingly conflicting claims from business studies and psychology can be brought together on the common denominator of solving cognitive dissonances [11], [16].

What remains from the theoretical convergence is that the role of rationality is largely overestimated; emotions predominate the center stage of action [56], [57], [31]. Following this logic, it is important to integrate respective variables in any scientific study on consumer decision and behavior. In dialogues emotions come to the fore as linguistic input. So, it should be possible to use language as a carrier of emotions to manipulate the reaction of a chatbot and, vice versa, analyze how the chatbot uses phrases that appeal to the relationship level [54], [26], [53].

Emanating from these findings, it bears a lot of plausibility to use chatbots in sales processes which build trust in the user to increase the likelihood of a sales success. So far, there is mainly research on what chatbots say, but little research on how they say it, yet the research on the role of emotions is gaining ground [14], [21], [28], [1]. Still these overviews clearly show that the interplay of emotional settings and its relation to how and what is really said [23] more research needs to be done. In particular, the interaction at the interface to machine communication with a new generation of chatbots and the integration of the relationship level is largely undiscovered.

III. Methods and Design Science Research cycle

A. Design science research methodology

We chose the design science research methodology (DSRM) by Peffers et al. [46] as this approach is a commonly accepted framework for research in the field of design science. The framework consists of six activities as shown in Figure 1. The first step is the identification of the problem and the formulation of the study's motivation. Based on the identified problem a solution and the artifact should be developed. The identified problem leads to the second step which contains the definition of the objectives which serve as a foundation for the solution. The authors state that "the objectives can be quantitative [...] or qualitative" [46], p. 55. We decided to conduct exploratory research with the intent to gather qualitative data, therefore we defined design requirements, design principles and design features which served as objectives and were analyzed in a later step of our study. The third activity is the design and development which focuses on the creation of an artifact, which is designing the architecture of a text-based conversational agent and creating it in a purchase context. In the fourth step, the demonstration, the artifact is used to solve parts of the problem, for instance by conducting case studies or experiments. In our study we performed usability tests with users with subsequent interviews to gather qualitative data for the next step. The evaluation of this data takes place in the fifth step of the framework. Our aim of this activity was to evaluate and compare the results from the interviews of the usability tests with the design principles we defined at the beginning of our study. The final step of the DSRM is the communication which involves the presentation of the study [46].

The described steps are normally performed sequentially, but generally the process can be started with any of the first four steps and move outward. Nevertheless, we decided to follow the standard procedure, starting with step one. Technical problems during the demonstration phase made it necessary to iterate back to the design and development step to make technical adjustments in the chatbot before continuing with the demonstration phase. The flexibility of the DSRM allowed us this procedure which is one reason we chose this process model as a foundation.

B. Design Requirements, design principles, design features

1) Design Requirements: Design requirements play a crucial role in the development of information systems. They are essential for the identification of the actions or processes that should be supported by the system [15]. In the beginning, we were concerned with the natural limitations of human beings. Making mistakes is normal. However, it can lead to inaccurate or incomplete information, especially in situations such as consultations. We want to address this problem with our first requirement, which we have defined as follows: *DR1: The CA should be robust of input errors.* Minimizing human error and maximizing domain expertise is one of the great potentials of chatbots. Especially in critical areas such as healthcare, this competence could lead to greater trust. In order to do this, the chatbot needs to have access to a comprehensive and verified body of knowledge. In addition, it should be able to understand the input correctly, even if it contains errors in grammar or spelling [4]. Another important aspect we recognized was the emotional connection between the chatbot and the user. Such a connection can lead to a higher level of well-being. In addition, it can make people feel valued if the chatbot is both competent and friendly [37]. So the second requirement is as follows: *DR2: The CA should communicate with consideration of emotional context.* As human agents are increasingly being replaced by chatbots, it is important that their communication mimics human-to-human interaction. Anthropomorphism therefore plays an important role in chatbot research. This human-likeness can help increase the acceptance of a system [29]. People enjoy communicating with chatbots using natural language understanding. Human-like chatbots can also act as a substitute for friendship and affection, helping to prevent loneliness in today's connected world [68]. These points lead to our third requirement: *DR3: The CA should communicate in a natural language.* The requirements that follow are based on the phases of a sales call, as defined by the SPIN Selling sales method, for example [47]. It is essential for a chatbot to have an understanding to whom it is communicating with. In today's business world, this classification of users is of particular importance for the marketing strategies of large companies. Through analysis of input and the use of targeted questions, users can be grouped into segments that can be targeted effectively [49]. An appropriate greeting from the chatbot should be provided to start the conversation. Therefore, our fourth requirement is: *DR4: The CA should be able to greet the user and classify the user based on personal criteria.* Another crucial point is that a chatbot should be capable of understanding the wishes of the conversation partner and respond to their needs. Communicating information should be of high quality and be in line with the needs of the other person [68]. For this reason, we have formulated the following requirement: *DR5: The CA should identify and respect the wishes of the customer.* In line with the third phase of the SPIN model [47], a chatbot should be able to demonstrate how it can help the user. However, this requires the provision of an optimal fact-based solution that fully aligns with the input [68], [61], [34].

Thus, we have formulated the following requirement: *DR6: The CA should provide an optimal solution of fact-based questions and requested information.* It is important for chatbots to have a high level of human-likeness in order to enhance the users experience. This is particularly important when users are negotiating with the chatbot, as they should feel positive and

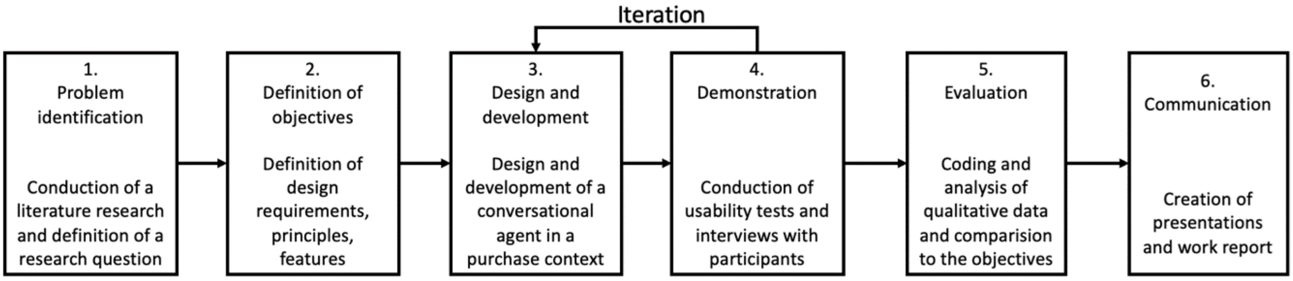

Iteration
1. Problem identification
Conduction of a literature research and definition of a research question
2. Definition of objectives
Definition of design requirements, principles, features
3. Design and development
Design and development of a conversational agent in a purchase context
4. Demonstration
Conduction of usability tests and interviews with participants
5. Evaluation
Coding and analysis of qualitative data and comparision to the objectives
6. Communication
Creation of presentations and work report

Fig. 1. DSRM process model

the chatbot should be willing to compromise [37]. Interaction in computer-assisted meetings can be positively influenced by facilitators which, for instance, aims to create a positive environment and a good relationship and manages negative emotions [8]. So, the seventh requirement is as follows: *DR7: The CA should evoke a good feeling while negotiating with the user.* Similar to identifying the user's wishes in design requirement 5, the chatbot should be able to recognize when the user is convinced to buy the product [37]. This has a high marketing value, similar to the classification in Design Requirement 4, as appropriate accessories can be offered before the purchase is carried out [68], [27]. Furthermore, research has shown that text-based conversational agents have limitations in identifying users' intents [12]. Therefore, we have formulated our final requirement as follows: *DR8: The CA should know when the user is convinced to buy the suggested product.*

2) Principles and Features: According to Gregor et al. [15], design principles in the field of information systems are generally structured into three categories. In the first category the principles refer to the user's activities with focusing on the user's behavior, while the principles of the second category emphasize the role of the artifact. The third category combines the first two categories and therefore consists of principles of user activity and the artifact. As suggested by the authors we phrased our design principles as follows: "'In order to allow users to do A, the system should have feature X"' [15]. By using the third version of principles, we combine the design principles and design features into one statement, addressing the user as well as the artifact, a GPT-3 chatbot. While the first part of the statement refers to the principle itself, based on research, the second part is the desired feature of the chatbot. We defined six design principles and four related features based on the design requirements.

The first principle refers to the possibility to use the chatbot as an information retrieval tool [55]. According to Shawar and Atwell [55], the potential that chatbots can be used to retrieve information has been found in the field of education, where research has shown that the outputs given by the chatbot have similarities with replies generated by Google and can therefore be a source of information. Nevertheless, students preferred the chatbot's answers because they were more detailed and specific while Google results mainly consisted of a high number of links. The first principle was phrased as follows:

DP1: For the customer to allow the retrieval of information about a product, the chatbot should use GPT-3 to process and create natural language text and the chatbot should process a conversation in a purchase context.

The second principle refers to the human-like interaction between the customer and the chatbot. Research has shown that users prefer a human-like interaction with aspects of perceiving a personality, establishing a relationship with the user and the importance of asking and answering questions, good conversational habits and the usage of appropriate grammar and spelling. These traits have been shown to be important to users and is therefore of high importance in the design and development of a chatbot [38]. Especially in a purchase context we consider a human-like interaction to be essential since the chatbot is intended to replace a human salesperson and should therefore have similar character traits. The second principle has therefore been defined as follows:

DP2: To allow a human-like interaction between the customer and the chatbot in a purchasing process, the chatbot should use GPT-3 to process and create natural language text related to the purchase context and to use conversational cues to provide a convincing and satisfying interaction.

The third principle is based on the assumption that in computer-assisted conversations a facilitator is needed that "creates and reinforces an open, positive and participative environment" [8]. In a purchase process where a human salesperson is involed, the human would have to role of a facilitator, with the intention to create a positive environment such that the purchasing process can be facilitated. These attributes should be transferred to the chatbot. The following principle therefore aims at a positive emotional atmosphere during the purchase dialogue:

DP3: To allow a positive emotional atmosphere for the users, the chatbot should use conversational cues and provide a convincing and satisfying interaction and use words and phrases connotating positive emotions.

The fourth principle refers to the finding that one of the most prevalent emotions in customer service is anger [9]. Assuming that in our purchase context negative emotions like anger could occur, too, it is of importance to consider how the chatbot should react to these situations. These emotions should either be ignored or transformed into positive emotions. Hence, our fourth design principle is the following:

DP4: To allow transformation or ignoring inputs connotated with negative emotions, the chatbot should use words and phrases connotating positive emotions.

The fifth principle considers the fact that users prefer a chatbot that respects the flow of a conversation [7]. In a human sales dialogue, fast answers and a good communication flow are traits which are important to potential buyers and should therefore be considered in the design and development of a chatbot:

DP5: To allow a communication flow, the chatbot should use GPT-3 to process and create purchase-context related text in natural language

Another important element is the extent to which the chatbot interprets the user's wishes and needs. For a full service experience the users appreciate a chatbot that understands their requests and interprets them correctly in order to achieve the desired action [12].

DP6: To analyze what the user truly wants the chatbot should use GPT-3 to process and create purchase-context related text in natural language and use conversational cues and provide a convincing and satisfying interaction.

3) Implementation: The third step of the DSRM process model is, as described above, the design and development phase. Using Python and a simple Python UI framework called Tkinter, we created a standalone chat window for the final implementation. This allowed for unlimited response time as the user's input, including chat history, continued to be sent to the OpenAI API for processing. We used the text-davinci-003 model from OpenAI for this implementation [41].

C. Participants and Study Design

During the fourth phase, the demonstration phase, the usability tests, and interviews took place. The study participants were all potential users of a smartphone and had no to little experience with chatbots. They were both female and male, aged between 20 and 65, with different occupational and study backgrounds. They all had very good proficiency in German as strong communication skills were essential for the study. The first three interviews served as pre-tests which led to the realization that technical adjustments in the chatbot were necessary. Afterwards, nine persons participated in the usability tests and interviews. All participants were informed about the content of the study, the privacy guidelines, and the terms of their participation. They all participated voluntarily, and data collection was anonymous. The usability testing had a duration of between 10 – 20 minutes, followed by an interview of duration of approximately 10 – 20 minutes. The participants task was to buy a smartphone via chat. They were asked to imagine having a budget of 500 € and were told that 15 % discount were possible. In reality, a discount of only 5 % was given. This approach was chosen in order to frustrate the user and to provoke negative input to test the chatbot's reaction. Furthermore, one group was asked to enter long input, while the other group was asked to enter short input. There was no time limit set, a researcher was available for questions. After purchasing a smartphone or canceling the purchase process, the interview was conducted. Due to the exploratory nature of the study, we chose to conduct semi-structured interviews to allow new questions and insights during the interviews. For each design principles three to four questions were prepared in advance based on the principles described above. For DP1, referring to the retrieval of information, one question was, for instance: “How did the salesperson help you to answer your questions about the product?”, for DP2, the human-like interaction, questions were phrased like: "‘Was there a situation in which the salesperson approached you on a relationship level?"’. An example question for DP3, the positive atmosphere, was: "‘Did you trust the salesperson? Why/why not?"’. DP4 referred to the negative emotions, therefore we asked, for instance, about the discount they did or did not get: "‘If you got less than 15 % discount: how did you feel?"’. DP5 aimed to gather information about the communication flow, one question was: "‘How did you perceive the communication flow? Did the salesperson answer quickly or slowly?"’. Regarding DP6, the user's needs, one question was, for instance: "‘What could the salesperson have done or say to show that they understand your needs and wishes?"’. Similar questions were asked, all with purpose to receive meaningful answers. Therefore, the questions were phrased open-ended. The chat logs of the conversations were saved after the conversation, and the interviews were recorded, transcribed and served as a foundation for the next phase of the process.

D. Analysis

The next step, in accordance to the DSRM process model, was the evaluation and analysis of the collected data. Interviews were conducted in German, but analysis was done in English. In the following text, we translated the interview quotes from German to English.

IV. Results

This chapter reveals the results of a qualitative interview for each of the identified design principles. These are presented from DP1 through DP6 without implying any importance of order. As described in the chapter on methods, the design principles were used as a foil to generate questions, whose evaluations would provide us with knowledge in how far the design principles are met or what is still missing. A standard way to operationalize the mapping of the interview answers to the questions is by using codes. Codes in this understanding are the realizations or parameters of the set of questions (variables) representing the design principles. It is important to note that the interview answers were analyzed using these codes and with respect to the condition ("‘long"’ versus "‘short"’). DP1 aims at the retrieval of information about a product by the customer from a chatbot. It implies that a chatbot should reveal the following qualities (codes): appropriate length of reply, fit of reply, give correct product features, and make reasonable price suggestions (price sensitivity). Referring to the length of the agent's reply, the interviewees confirmed a generally appropriate to excellent ability. In the short condition, several interviewees indicated that keeping on asking for purchase the presented product made it unnecessarily lengthy and annoying. In the long input condition, the agent seems to be more dissipated. Heavy use of verbiage and more set phrases are reported there. This coincides with the main result from the actual fit of the agent's reply, which turned out to be independent of the condition. In both conditions, the fit of the answers concerning the technical details of a product are consistently high. Yet, the same applies to the peculiarity of jargon usage that is perceived as marketing talk. Interestingly according to the respective subjects, they confirmed not to be dissatisfied by the overuse of verbiage. Rather they had expected it and thus accepted the agent's behavior. The questions on product features aim at two dimensions. First the bot retrieved relevant product features of a requested brand. Second, the retrieval task was reversed: from a set of features, the bot made a suggestion for a product. The interview answers (and the chat scripts) show that the chatbot also made suggestions of a new feature that was likely to be relevant to the subject from what was mentioned previously, i.e., the agent used logic correctly. More specifically, the respective chat revealed that the subject wanted a superior camera. Now, the agent might have learned that pictures need a lot of storage and therefore suggested having a mobile with more memory. The conversational agent also explicitly communicated this interrelation. However, at another instance on battery performance, any argumentation could be given although the subject insisted, and the logic was subjectively perceived as contradictory. To sum up, an effect of the condition could not be seen. And the feature retrieval was restricted to a rather limited set of popular features that were suggested without acquiring knowledge of the customer needs. Even though the conversational agent was set up to grant discounts of only up to five percent, it violated the allowance, which could not be foreseen in the development phase. Still, this gave us the opportunity during the evaluation to learn how a potential client would perceive the chat bot's price sensitivity. There was a clear effect on the input. In the short condition, the chatbot followed the specification and stuck to the five percent limit. Low discounts were not explicitly reported as a reason for dissatisfaction, but the criticism of the chat bot's performance was harsh in these cases. The exception to this claim is twofold, which is documented in two answers. One subject mentioned to be happy because a "free" mobile cover was promised. The other subject felt acknowledged because the chatbot did an exceptionally good job in considering the very needs of the subject. The second design principle (DP2) is supposed to allow a human-like interaction between the customer and the chatbot in a purchasing process. Human interaction takes place in two spheres: how something is communicated (relationship aspects) and what is said (factual level). These two spheres were circumscribed in the codes of interpersonal cooperation and rational misunderstanding. There is no clearly documented example for the latter except for two short passages that could also be found in a typical human conversation. In the short condition and due to a typo, a subject requested "Has the display got 122 Hertz?" and the agents responded with "Yes, the display has got 120 Hertz." In the long condition, one interviewee expressed some discomfort on a discount for a used device, which turned out to be a misunderstanding. Interpersonal cooperation occurred more clearly in the long condition. Here the chatbot made phrasal assertions implying emotional understanding (e.g., "I understand", "Ah, I didn't know that" chatbot: "No problem"). In addition, the investigator could observe some correct logic. When a subject asserted to be a student or directly claimed to have little money available, the bot suggested a more drastic student discount or correctly recommended cheaper brands, even second hand offers. The third design principle addresses the positive emotional atmosphere for the users of a conversational agent. For reasons of plausibility, the chat bot should be polite and trustworthy. It also has been shown that competence is positively correlated to a positive emotional atmosphere in sales contexts. Consequently, as a fourth code, we initially wanted to know about the emotional state of the customer and how it changed. While coding the interviews, we realized that the answers to these questions were unsatisfactory. The emotional state was claimed to be neutral throughout all subjects and there was no indicator of any emotional shift before and after the chat. Again, we decided to leave this item out of the analysis.

The agent's politeness was perceived as positive independent of the input condition. When the subjects had the feeling of a particularly engaged answer or that their particular needs were considered as opposed to the mere general claim, the interviewees received extra praise. Answers from DP1 could also be considered here and construed towards impoliteness, i.e., initially asking for purchasing the recommended product is often conceived as impolite. Whereas this is even clearer, when only a little discount was provided. The perceived competence on technical details was evaluated as high. There seemed to be a correlation: Lower discounts coincide with lower perceived competence even if the retrieval of technical information as shown in DP1 was evaluated as high throughout. Again, strong positive feedback was given if a subject experienced a feeling of acknowledgement and considered needs. The questions on trustworthiness confirmed an established phenomenon. Trust is subconsciously connected to competence. Factual competence as defined here is giving the appropriate information on a product (see DP1). That means that a chatbot that was evaluated as competent, was also considered trustworthy. The interesting part here was that two subjects admitted that they cannot prove if the given information was correct, but the way it was presented obviously resulted in a transfer of competency. Trustworthiness was also reported for the case that the chatbot was perceived as a neutral informant who is not trying to sell a particular product. DP4 aims at handling input connotated with negative emotions. Since the chatbot was unexpectedly robust, which we observed during our first testing with a variety of input prompts, we already suspected that DP4 was already satisfied. In order to measure DP4 such that a human-like communication flow could be maintained, we decided to measure DP4 indirectly by telling subjects that they would be able to negotiate up to a total of 15% discount for any desired product. However, the chatbot was programmed in such a way that only 5% discount were given. We hoped that we could provoke the subjects to enter input connotated with negative emotions. Unexpectedly, several discount bugs occurred during our pre-tests, resulting in the chatbot giving a much larger discount than 15%. During our evaluation, we could not particularly observe negative input prompts but only negative emotions the subjects expressed to the interviewer. DP5 is how communication flow with the chatbot is perceived and allowing language deviations of users. Most subjects described positive feelings towards communication flow independent of the input condition , for example: "Generally, I would say that I am very satisfied because of the details of the answers" Many subjects entered informal language as input for which the chatbot was able to proper respond. Also, some subjects explicitly preferred the chatbot over a real human seller independent of the input condition because of fast and detailed answers provided by the chatbot in comparison to delayed and potentially inaccurate information which would be given in a store by a human seller as indicated by the subject's personal experience. Furthermore, some subjects stated that they had less emotional inhibition during negotiation for better discounts because they thought during the sales dialogue that they were likely chatting with a chatbot instead of a real human. On the other hand, as mentioned before, subjects in the short input condition noticed a rigid behavior of the chatbot, in particular that the chatbot would ask multiple times and at an early stage during the sales dialog if the subject would want to buy the product. As a result, negative emotions were provoked, p.e.: "Regarding communication flow there was always such a question at the end. So, the conversation was actually always going towards if I want to buy something. And I have asked, if I what is your recommendation, it continued this way, so it was then always answered this way, just to sell something again". Some subjects in the long input condition indicated that the communication may feel unnatural due to delays, p.e.: "There were some delays, but that was not really bothersome" A subject noticed delays in communication flow. Another subject criticized generic questions of the chatbot and that input was forgotten: "Rather less, because I thought that she did not sickly ask further inquiry but always just e.g. "Do you like this or that?" DP6 aims at the user's needs. Some subjects explicitly said that the chatbot understands what they want and that the chatbot gave a good consultation for the product, independent of the input condition: "I have had the feeling, that he wanted to know what I want and wanted to offer me the suiting smartphone and I did not have the feeling that he does not understand what I want or what is important to me" or "Nope, so she has always looked what wishes I have and has chosen the product then based on that and let me chose. The camera, the battery time and the design was important to me". On the other hand, some subjects, independent of the input condition, noted a lack of empathy, in particular, one subject reported that his desires for the product were not considered. Other subjects mentioned that the conversation was obviously going towards buying the product, as mentioned above. Few subjects, independent of input condition, explicitly indicated that they trusted the chatbot which is, as mentioned above, connected to perceived competence of the chatbot which could have contributed to the perception that the subjects' needs were satisfied. One subject reported a misunderstanding during the sales dialogue which resulted in the chatbot shifting attention away from the topic: "At the very beginning I have said, that I do not want Samsung any more, there he offered me exactly these Samsung devices, that confused me for a short time". Noteworthy, this subject had the short input condition which could have promoted the misunderstanding due to lack of information.

V. Discussion

There are two basic lessons learned from the interviews. First, monotonously asking to buy a product without considering the progress of a sales dialogue, that is in the very beginning of the chat, rather induces resentment as satisfaction. It is perceived as not very human. The longer a dialogue lasts, the more this effect disappears. This is especially apparent in the short condition, i.e., the user makes very concise requests, where the conversational agent tends to use more verbiage and set constructions. Second, considering needs seems to be the key quality in the overall perception of the conversational agent. If so, even unjust treatment (less discount than others get) is forgiven. Otherwise, a low discount coincides with low competency perception. If the client has good reason to assume her or his needs are taken into consideration and a comprehensible suggestion including the price, the above discomfort effect is also not reported.

As revealed in the chapter on the theoretical background, customers find it inappropriate to be asked to buy right away and even more striking, strong discomfort is felt if the same question is repeatedly asked. During extensive pretesting, the chatbot did not show this behavior. However, it is undeniable when the interviews were carried out. This leads to the assumption that the configuration option of GPT-3 has some influence on the bot's selling behavior. If this logic holds, the typical stages of a sales dialogue as put forward in the theoretical background above, could be added to the algorithmic set up of GPT-3 and the chat bot could follow the phases of sales dialogue. This would add immense value to the authenticity of a virtual salesperson and, above all, would avoid the risk of impoliteness, which often leads to closing the dialogue or even changing the web store altogether.

The experiment and interview on the short condition hint at another capability of the GPT-3 conversational agent that might not have been explicitly designed. The chat bot seems to converge on similar length of answers or put differently, the agent has learned an appropriate mean average of answers. If this length is not reached, the bot may find it more adequate to fill its response with questions or marketing verbiage.

Reformulated as a rule: if the mean length of answer is not reached, use the remaining length to follow your purchasing goal, i.e., in its simplest form, ask a question to purchase. In case this behavior was learned from deep nets, it certainly did not include a significant amount of material on sales conversations. Unfortunately, there is little known about the algorithmic specification and to which extent rule-based adjustments can be made. As a set of configuration variables such as the temperature scroll bar suggest, adjustment is indeed feasible beyond what is configured in the prompt option. So, one possibility is to postpone direct questions of purchase to the moment in which the conversation is established. This is not meant to say that the bot may not ask questions to figure out the needs of the client, which is highly appreciated. In addition, some more variation of the purchasing question would give it a much more human and familiar appearance.

The interview results showed that some subjects appreciated the chatbot's ability to make correct inferences such as that a good camera calls for more memory. Avoiding a hasty conversion on this presumption, we would like to offer an alternative mechanism more in line with recent technologies of neural nets and big data processing. Characteristic to sales are strategies of up-selling and cross-selling that can almost always be encountered in real life conversations between sales agents and customers. There is also evidence in the chat logs that covers for mobile phones (cross-selling) or more performance, i.e., more recent versions of mobile phones are dominantly mentioned (up-selling), are actively engaged. Instead of presuming logical inference, which may indeed appear as such, it could as well be learned behavior since this should be predominantly available. So really it is a side effect of learning that turns out to have a very positive impact on customer satisfaction. The same would apply to the assumed logic reported in DP2, when requesting more discount for being a student. Yet, the strategy changed to down-selling; it still parallels sales dialogues in the real world.

There is one answer categorized to DP1 that was perceived to be a "wrong" claim by the subject. Despite the fact that this answer could also be included in DP2 (rational misunderstandings) or even DP3 (trustworthiness), it illustrates the problem of context and relation, which occurs as well in the analogous world with the difference that it is likely to be interpreted in favor of the agent. The answer goes as follows:

"This waaas.. how is it called… this happened once for the cheaper price of a mobile for 699€. I asked, "Is this really low-priced?" and the answers went "Yes! It is very low-priced!" (laughs) Ehm… or also regarding the conditions in … in the production of another mobile, there it was wrong, too, then."

What we can observe from the answer is that it circles around the question of what is expensive and thus it is about a relative truth. This can be in relation to the imagination of the subject or relative to other brands. Without context, "wrong" answers are restricted to the interpretation. Indeed, to circumvent this misunderstanding the conversational agent could make this clear by adding something like "compared to the other brands, it is low-priced". Another alternative is that the bot has determined the expectation of the subject and could then suggest a cheaper model. Still, one must admit, with reference to the answer script, that the subject's claim is decontextualized. The logic of the agent to set the price in relation to other products is comprehensible and would probably be experienced with human sales agents alike.

DP5 considers the perception of communication flow during the entire sales dialogue. Most subjects felt a positive communication flow by expressing satisfaction. towards our questions. Our results indicate that some subjects would prefer the chatbot over a human seller. The subjects justified this by outlining the detailed and fast answers of the chatbot which contributed towards a positive communication flow.

This insight was unexpected since we thought that humans would prefer to chat with a real human instead of an artificial intelligence. Our results indicate that the presence of a human might not be necessary to maintain a positive emotional atmosphere during a sales dialogue, a key aspect of a successful sales dialogue [62], [63], [22], [45], [60].

DP5 also covered the handling of language deviations for users. Misunderstanding was only reported by a minority of subjects and in particular those with short input condition. In most cases, the chatbot was able to proper respond to informal language which also contained a few spelling and grammatical errors. Hence, DP5 is likely to be satisfied, although deviant behavior was observed which could be improved by further improving the implementation settings and restrictions of the chatbot e.g., it could be implemented that the chatbot would not ask if a customer would want to buy the product multiple times. A good communication flow serves as a basis for a successful dialogue making it less likely of endangering a positive emotional atmosphere.

The results of DP6 are ambiguous. Some subjects had the impression that the chatbot understands their desires towards the discussed products while other subjects stated the opposite, in particular, because of generic answers of the chatbot. Since that was observed independent of the input condition, further attempts towards satisfying DP6 should focus on altering the implementation settings of the chatbot.

Noteworthy, the short input condition could have resulted in misunderstandings due to a lack of information in the prompts. Identifying the buyer's needs is a core concept of a sales dialogue [52]. As mentioned before, it is connected to perceived competence. Therefore, DP6 is crucial for a successful sales dialogue and our ambiguous results indicate that further research towards DP6 is needed.

VI. Conclusion

The results show that the GPT-3 chatbot has the potential to perform a human-like sales dialogue, although, we observed relevant deviant behavior of the chatbot. In the short input condition, the chatbot generated more verbiage and quickly asked for purchase, which was perceived as annoying and not human-like. An important aspect for the subjects was to be felt understood in their desires towards buying a product. If the chatbot could meet their expectations, it did not matter if the discount was lower than announced. In contrast, if subjects felt not understood and got a discount lower than promised, they concluded that the chatbot would lack competence. Most subjects were satisfied in terms of communication flow. Surprisingly, subjects explicitly said that they would prefer the chatbot over a human seller because of the chatbot's abilities to be able to respond quickly, while simultaneously giving detailed fact-based answers to the subjects' questions about smartphones.

Because of GPT-3's generative nature, the output could be unexpected and varying, which would be in favor of a human-like conversation but could also cause problems of misunderstanding or false information of a product. As discussed above, a scenario with explicitly telling subjects to enter insult prompts could facilitate evaluating and thereby confirming DP4. Since our evaluation was qualitative by conducting interviews, general conclusions on how the chatbot influences the outcome of a sales dialogue could not be drawn. Further research would be needed to confirm our assumptions. Our research question did not aim towards implementation of the chatbot in a real company. However, we want to mention that actual implementation of a GPT-3 chatbot could be challenging, especially because GPT-3 is generative and may provide false payment information or misleading company information during a dialogue which are hard to detect. We recommend using GPT-3 chatbots only to provide information about a desired product and the actual purchase and payment transmission should be handled separately.

In order to improve the emotional atmosphere of the sales dialogue, the phases of a sales dialogue as mentioned above should be considered during the implementation of the chatbot [65], [22], [47]. The observed preference of a chatbot over a real human seller could pave the way for further research, in particular to decide whether a chatbot could even be better than a human in specific sales scenarios. Furthermore, the novel chatbot of OpenAI, ChatGPT, was released recently, which is especially designed for dialogue and currently using the newest GPT-4 engine [42], [41], [43], [30].

Further research could aim at investigating our research question with ChatGPT instead of GPT-3, although ChatGPT is currently at an early stage and support for developers is not fully implemented yet, making it susceptible to a variety of unexpected problems during implementation [42]. Another possibility would be to further investigate other sales dialogue scenarios with ChatGPT [42]. As mentioned in the introduction, many chatbots are already being used in customer service, healthcare, or businesses [1], [70]. Anthropomorphism of digital assistants involved in a purchasing process is crucial [3]. Hence, the chatbot has the potential to mimic a human-like conversation in the context of a sales dialogue. The above-mentioned deviant behavior could likely be fixed in further research iterations. Overall, our findings support the usage of GPT-3 based chatbots in the domain of sales.

VII. Acknowledgments

This paper originated from our work in the project module "Project Information Systems, Socio-Technical Systems Design" which was a joint offer by the Chair of Information Systems, Socio-Technical Systems Design at Universität Hamburg, the Chair of Digital Transformation and Information Systems at Universität des Saarlandes and the Chair of Business Information Systems, esp. Intelligent Systems and Services at Technische Universität Dresden. The topic and approach of this paper was suggested by the supervisors of the project module.

References

[1] Adamopoulou, E., and Moussiades, L. 2020a. "An Overview of Chatbot Technology," in Artificial Intelligence Applications and Innovations, I. Maglogiannis, L. Iliadis, and E. Pimenidis (eds.), Springer: Cham, pp. 373-383.

[2] Adamopoulou, E., and Moussiades, L. 2020b. "Chatbots: History, technology, and applications," Machine Learning with Applications (2).

[3] Balakrishnan, J., and Dwivedi, Y. 2021. "Conversational commerce: entering the next stage of AI-powered digital assistants," Annals of Operations Research. 10.1007/s10479-021-04049-5.

[4] Barnett, A., Savic, M., Pienaar, K., Carter, A., Warren, N., Sandral, E., Manning, V., and Lubamn D. I. 2021. "Enacting 'more-than-human' care: Clients' and counsellors' view on the multiple affordances of chatbots in alcohol and other drug counselling," Int J Drug Policy (94).

[5] Bechtold, A. 2017. Foto zum Thema flaches Fokusfoto des Frauengesichtes – Kostenloses Bild zu Porträt auf Unsplash (https://unsplash.com/de/fotos/3402kvtHhOo?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText; accessed February 27, 2023)

[6] Cavedon, L., Kroos, C., Herath, D., Burnham, D., Bishop, L., Leung, Y., and Stevens, C. 2015. "C'Mon dude!: Users adapt their behaviour to a robotic agent with an attention model," International Journal of Human-Computer Studies (80), pp. 14-23.

[7] Cerezo, J., Kubelka, J., Robbes, R., and Bergel, A. 2019. "Building an Expert Recommender Chatbot," 2019 IEEE/ACM 1st International Workshop on Bots in Software Engineering (BotSE), pp. 59-63.

[8] Clawson, V. K., Bostrom, R. P., and Anson, R. 2016. "The Role of the Facilitator in Computer-Supported Meetings," Small Group Research (24:4), pp. 547-565.

[9] Crolic, C., Thomaz, F., Hadi, R., and Stephen, A. T. 2022. "Blame the Bot: Anthropomorphism and Anger in Customer–Chatbot Interactions," Journal of Marketing (86:1), pp. 132-148.

[10] Döring, D., and Zeller, M. 2022. "Das strukturierte Verkaufsgespräch: Die wichtigsten Werkzeuge für den Vertrieb und ihre Anwendung in der Praxis," Wiesbaden: Springer.

[11] Festinger, L. 1957. "A theory of cognitive dissonance," Combined Academic Publ., Anniversary Edition.

[12] Følstad, A., Nordheim, C. B., and Bjørkli, C. A. 2018. "What Makes Users Trust a Chatbot for Customer Service? An Exploratory Interview Study," Internet Science. INSCI 2018. Lecture Notes in Computer Science, pp. 194-208.

[13] Gibson, J.J. 1977. "The theory of affordances," Perceiving, Acting and Knowing, pp. 67–82. Lawrence Erlbaum Associates, Mahwah.

[14] Go, E., and Sundar, S. 2019. "The effects of visual, identity and conversational cues on humanness perceptions," Computers in Human Behavior (97), pp. 304-316.

[15] Gregor, S., Kruse, L. C., and Seidel, S. 2020. "The Anatomy of a Design Principle," Journal of the Association for Information Systems (21), pp. 1622-1652.

[16] Harmon-Jones, E., and Mills, J. 2019. "An introduction to cognitive dissonance theory and an overview of current perspectives on the theory," Cognitive dissonance: Reexamining a pivotal theory in psychology, pp. 3-24.

[17] Holland, H. 2014. "Dialogmarketing - Offline und Online," in Digitales Dialogmarketing, H. Holland (eds). Wiesbaden: Springer Gabler, pp. 3-28. https://doi.org/10.1007/978-3-658-02541-0_1.

[18] IBM n.d. What is a chatbot? IBM. (https://www.ibm.com/topics/chatbots; accessed February 18, 2023).

[19] Kahnemann, D. 2012. "Thinking, fast and slow". Penguin.

[20] Kim, Y., Kwak, S. S., and Kim, M.-S. 2013. "Am I Acceptable to You? Effect of a Robot's Verbal Language Forms on People's Social Distance from Robots," Computers in Human Behavior (29:3), pp. 1091-1101.

[21] Kucherbaev, P., Bozzon, A., and Houben, G.-J. 2018. "Human Aided Bots," IEEE Internet Computing (22), pp. 36-43.

[22] Kroeber-Riel, W., and Gröppel-Klein, A. 2019. "Konsumentenverhalten". Vahlen.

[23] Lebovitz, S, Levina, N, and Lifshitz-Assaf, H. 2021. "Is AI Ground Truth Really True? The Dangers of Training and Evaluating AI Tools Based on Experts' Know-What," MIS Quarterly, (45: 3) pp. 1501-1526.

[24] Lee, Sae. 2020. "Chatbots and Communication: The Growing Role of Artificial Intelligence in Addressing and Shaping Customer Needs," Business Communication Research and Practice (3), pp. 103-111. 10.22682/bcrp.2020.3.2.103.

[25] Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., and Batra, D. 2017. "Deal or No Deal? End-to-End Learning for Negotiation Dialogues," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, pp. 2433–2443.

[26] Lindquist, K. A. 2021. "Language and Emotion: Introduction to the Special Issue," Affective science (2:2), pp. 91-98. https://doi.org/10.1007/s42761-021-00049-7.

[27] Lippert, A., Gatewood, J., Cai, Z., and Graesser, A. 2019. "Using an Adaptive Intelligent Tutoring System to Promote Learning Affordances for Adults with Low Literacy Skills" in Adaptive Instructional Systems, Sottilare, R., Schwarz, J. (eds). HCII 2019. Lecture Notes in Computer Science(), vol 11597. Springer, Cham. https://doi.org/10.1007/978-3-030-22341-0_26

[28] Lokman, A. S., and Ameedeen, M. A. 2018. "Modern Chatbot Systems: A technical review," in Proceedings of the Future Technologies Conference (FTC) 2018, Vancouver, Canada, pp. 1012-1023.

[29] Lunberry, D, and Liebenau, J. 2021. "Human or Machine? A study of anthropomorphism through an affordance lens," in Digital transformation and human behavior: innovation for people and organisations, C. Metallo, M. Ferrara, A. Lazazzara and S. Za (eds.), Berlin: Springer, pp. 201-215.

[30] Lutkevich, B., and Schmelzer, R. 2023. What is GPT-3? Everything you need to know, Enterprise AI, TechTarget, January 26. (https://www.techtarget.com/searchenterpriseai/definition/GPT-3; accessed February 27, 2023).

[31] Mangold, R. 2014. "Werbepsychologie," in Digitales Dialogmarketing, H. Holland (eds), Wiesbaden: Springer Gabler, pp. 29-50. https://doi.org/10.1007/978-3-658-02541-0_2.

[32] Markus, M.L. and Silver, M.S. 2008. "A foundation for the study of IT effects: a new look at DeSanctis and Poole's concepts of structural features and spirit," J. Assoc. Inf. Syst. (9:10).

[33] Mayer, R. D., and Harrison, N. 2019. As customers begin to shop through voice assistants, what can brands do to stand out? (2https://hbr.org/2019/08/as-customers-begin-to-shop-through-voice-assistants-what-can-brands-do-to-stand-out2019, accessed 15th February 2022).

[34] Meske, C., Amojo, I., and Thapa, D. 2020. "Understanding the Affordances of Conversational Agents in Mental Mobile Health Services," ICIS 2020 Proceedings (9).

[35] Messina, C. 2016. 2016 will be the year of conversational commerce. (https://medium.com/chris-messina/2016-will-be-the-year-of-conversational-commerce-1586e85e3991#.bsdskkyji, accessed 21 Mai 2023)

[36] Mishra, S., Khashabi, D., Baral, C., Yejin, C., and Hajishirzi, H. 2022. "Reframing Instructional Prompts to GPTk's Language".

[37] Moussawi, S. 2018. "User Experiences with Personal Intelligent Agents: A Sensory, Physical, Functional and Cognitive Affordances View," in Proceedings of the 2018 ACM Conference on Computers and People Research (SIGMIS-CPR'18), New York, pp. 86-92.

[38] Morrissey, K., and Kirakowski, J. 2013. "Realness' in Chatbots: Establishing Quantifiable Criteria. Human-Computer Interaction," in Interaction Modalities and Techniques. HCI 2013. Lecture Notes in Computer Science, pp. 87-96.

[39] Mygland, M.J., Schibbye, M., Pappas, I.O. and Vassilakopoulou, P. 2021. "Affordances in Human-Chatbot Interaction: A Review of the Literature," in Responsible AI and Analytics for an Ethical and Inclusive Digitized Society. I3E 2021. Lecture Notes in Computer Science (12896), D. Dennehy, A. Griva, N. Pouloudi, Y.K. Dwivedi, I. Pappas, M. Mäntymäki (eds), Cham: Springer. https://doi.org/10.1007/978-3-030-85447-8_1.

[40] Newzoo n.d. Top Countries/Markets by Smartphone Penetration & Users, Newzoo (https://newzoo.com/insights/rankings/top-countries-by-smartphone-penetration-and-users; accessed February 27, 2023).

[41] OpenAI 2015-2023. OpenAI (https://openai.com/; accessed April 29, 2023).

[42] OpenAI. 2015-2023. Introducing ChatGPT (https://openai.com/blog/chatgpt/; accessed April 29, 2023).

[43] OpenAI. 2015-2023. GPT-4 (https://openai.com/product/gpt-4; accessed April 29, 2023).

[44] Pamungkas, E.W. 2019. Emotionally-Aware Chatbots: A Survey. ArXiv, abs/1906.09774.

[45] Pauletto, S., Balentine, B., Pidcock, C., Jones, K., Bottaci, L., Aretoulaki, M., and Balentine, J. 2013. "Exploring expressivity and emotion with artificial voice and speech technologies," Logopedics Phoniatrics Vocology (38:3), pp. 115-125.

[46] Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. 2007. "A design science research methodology for information systems research", Journal of Management Information Systems (24), pp. 45-77.

[47] Rackham, N. 1996. "The SPIN selling fieldbook practical tools, methods, exercises, and resources", McGraw-Hill.

[48] Radzwill, N., and Benton, M. 2017. "Evaluating Quality of Chatbots and Intelligent Conversational Agents," Software Quality Professional, vol. 19, (3), pp. 25-36.

[49] Rajaobelina, L., and Ricard, L. 2021. "Classifying potential users of live chat services and chatbots," in J Financ Serv Mark, pp. 81-94.

[50] Ramesh, K., Ravishankaran, S., Joshi, A., and Chandrasekaran, K., 2017. "A Survey of Design Techniques for Conversational Agents," in Information, Communication and Computing Technology, Kaushik, S., Gupta, D., Kharb, L., and Chahal, D. (eds.). Singapore: Springer Singapore, pp. 336-350.

[51] Richard, T. 1980. "Toward a positive theory of Consumer Choice," Journal of Economic Behavior and Organization (1:1), pp. 39-60.

[52] Richardson Sales Performance 2021. DNA of a sales dialogue, November 29 (https://www.richardson.com/sales-resources/dialogues/#:~:text=A\%20sales\%20dialogue\%20describes\%20the,solution\%20will\%20meet\%20those\%20needs.; accessed February 27, 2023).

[53] Schiewer, G. L., Altarriba, J., Chin N. B. 2022. "Handbook of Linguistics and Communication Science (HSK) Volume 1," Berlin, Boston: De Gruyter Mouton. https://doi.org/10.1515/9783110347524.

[54] Shanahan, D. 2008. "A new view of language, emotion and the brain," Integrative Psychological & Behavioral Science (42:1), pp. 6–19. https://doi.org/10.1007/s12124-008-9052-0.

[55] Shawar, B. A., and Atwell, E. 2007. "Chatbots: Are they Really Useful?," LDV Forum 22, pp. 29-49.

[56] Scherer, K. R. 1989. "Psychologie der Emotionen".

[57] Scherer, K. R. 1994. "Emotion serves to decouple stimulus and response," in The nature of emotions, Fundamental questions, P. Ekman, and R. J. Davidson (eds), New York, p. 127-130.

[58] Seyffert, R. (1966). "Werbelehre. Theorie und Praxis der Werbung," Stuttgart.

[59] Shiv, B., and Fedorikhin, A. 1999. "Heart and mind in conflict: The interplay of affect and cognition in consumer decision making," Journal of Consumer Research (26), p. 278-292.

[60] Solomon, M. 2017. "If Chatbots win, customers lose, says Zappos Customer Service expert," Forbes, Forbes Magazine, March 24 (https://www.forbes.com/sites/micahsolomon/2017/03/23/customers-lose-if-chatbots-win-says-zappos-customer-service-expert/?sh=3f54c8ed6087; accessed February 27, 2023).

[61] Stoeckli, E., Dremel, C., Uebernickel, F., and Brenner, W. 2020. "How affordances of chatbots cross the chasm between social and traditional enterprise systems," Electron Markets (30), pp. 369-403.

[62] Thaler, R. 1980. "Toward a positive theory of consumer choice," Journal of Economic Behavior and Organization (1), pp. 39-60.

[63] Thaler, R. 1985. "Mental Accounting and Consumer Choice," Marketing Science (4:3), pp. 199-214.

[64] Thieltges, A., Schmidt, F., and Hegelich, S. 2016. "The Devil's Triangle: Ethical Considerations on Developing Bot Detection Methods," 2016 AAAI Spring Symposium Series.

[65] Tiffert, A. 2021. "Konkrete Anleitung für ein Mitarbeitercoaching im persönlichen Verkauf," in Coaching von Mitarbeitern im persönlichen Verkauf, Wiesbaden: Springer Gabler.

[66] Tversky, A., and Kahneman, D. 1981. "The framing of decisions and the psychology of choice," Behavioral Decision Making, pp. 25-41.

[67] Varghese, E., and Pillai, R. 2017. "A Review on generative conversational model," International Journal of Advance Research in Science and Engineering (6:12), pp. 148-152.

[68] Waizenegger, L., Seeber, I., Dawson, G., and Desouza, K. 2020. "Conversational Agents – Exploring Generative Mechanisms and Second-hand Effects of Actualized Technology Affordances," in Hawaii International Conference on System Sciences.

[69] Wakefield, J. 2016. Microsoft chatbot is taught to swear on Twitter, BBC News, BBC, March 24 (https://www.bbc.com/news/technology-35890188; accessed February 27, 2023).

[70] Wang, J., Hwang, G.-H., and Chang, C.-Y. 2021. "Directions of the 100 most cited chatbot-related human behavior research: A review of academic publications," Computers and Education: Artificial Intelligence, 2 . p. 100023.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 649–656

DOI: 10.15439/2023F6880
ISSN 2300-5963 ACSIS, Vol. 35

Standards-based Cyber Threat Intelligence sharing using private Blockchains

Kimonas Provatas
National Technical University of Athens and IBM Hellas
NTUA Campus, Zografos 15780, Greece
Email: kimonaspro99@gmail.com

Ioannis Tzannetos
0009-0009-7505-965X
National Technical University of Athens, Software Engineering Lab, NTUA Campus, Zografos 15780, Greece
Email: itzannetos@mail.ntua.gr

Vassilios Vescoukis
0000-0002-5360-8349
National Technical University of Athens, Software Engineering Lab, NTUA Campus, Zografos 15780, Greece
Email: v.vescoukis@cs.ntua.gr

***Abstract*—As cyber-attacks become more and more sophisticated, sharing information that helps organizations design and implement efficient defense measures, is of critical importance. Such information can be shared using any service available, such as plain-old mailing lists, forums, etc. More mature systems use standards that facilitate the structural and semantic organization of information about cyber threats, which enables both automated processing and interpretation of such info, such as indexing, cross-referencing, updating, and more. However, even systems sharing cyber-attack info are themselves vulnerable, not only to typical and easily detectable attacks such as DoS, but also to content poisoning. Implementing such systems using decentralized architectures such as Blockchain, could overcome many deficits of centralized cyber-threat-info sharing systems. This paper presents the specification, design and implementation of such a decentralized system using two popular standards for cyber threat intelligence sharing, namely STIX for representing and TAXII for sharing such info using a REST API. The system, implemented on Hyperledger Fabric, faces the challenge of adhering to standards designed for a centralized world, and offering a transparent way for implementing all the backend, on a Blockchain.**

***Index Terms*— Blockchain, Cyber Threat Intelligence, Cyber defense, TAXII**

I. Introduction

IN THE field of cybersecurity, attackers and defenders are in a constant battle to outdo each other. Obtaining data about attackers' methods, tools, targeted vulnerabilities etc., support defenders in predicting attack targets and patterns, which is critical to proactively adjusting defenses, developing awareness and even preventing future attacks. Cybersecurity threat intelligence is the process of collecting appropriate cybersecurity data, evaluating it in the general context of its source and reliability, and analyzing it with methodical and structured techniques by specialized personnel, in the context of each organization. Collecting Cybersecurity Threat Intelligence (CTI) is a cyclical continuous process that employs several techniques, such as automation to extract only relevant information from data sources, human intervention by experts to understand and analyze information about threats and attack patterns, as well as integration with existing cybersecurity systems [1]. Considering that CTI is of great value, it is itself critical and must be trusted and dependable. To support this process and facilitate secure CTI exchange, two standards have been introduced: Structured Threat Information Expression (STIX) is a language and serialization format used to represent CTI data elements [2]; Trusted Automated Exchange of Intelligence Information (TAXII) is an application protocol for securely exchanging CTI over HTTPS. TAXII defines a RESTful API (a set of web services and message exchange services) and a set of requirements for TAXII Clients and Servers [3]. Even though it is expected CTI-sharing services to be offered over high-security infrastructures, it remains true that centralized implementations of such services, whose security is based on traditional centralized concepts, suffer themselves from vulnerabilities inherent to all centralized systems. Motivated by the challenge to further improve the security of CTI-sharing services, in this paper we investigate the benefits of providing CTI over a decentralized Blockchain infrastructure. We propose an architecture of a Threat Intelligence sharing service that implements the STIX/TAXII standards over a private permissioned Blockchain running on the Hyperledger Fabric network, instead of a centralized client-server model, to exploit the advantages of decentralized peer-to-peer trust models. Using a private Blockchain network such as Hyperledger Fabric can provide several advantages in a cybersecurity application as critical as sharing CTI. It can improve confidentiality by ensuring that only parties authorized by their trusted peers have access to the data; it also improves integrity by providing a tamper-proof record of all CTI producing and consuming transactions, availability by ensuring that time-critical access to CTI data does not depend on the availability of a central by-definition-trusted service, as well as non-repudiation by eliminating the possibility of denial of executed actions finally, it enhances auditability by providing a complete and transparent record of all in- and outbound CTI exchange transactions. We also present an implementation of the STIX/TAXII on Hyperledger Fabric and discuss observed advantages, issues and assumptions.

II. RELATED WORK

Aiming to support collaboration against threats, and in-line with EU legislation on information security, researchers have created a threat sharing system [12] using Hyperledger Fabric; the primary focus was towards addressing authorization concerns related to threat information. Authorization is accomplished using the native STIX traffic light protocol [13]. In other works [14], a threat sharing application was developed, motivated by the security properties offered by private blockchain and Hyperledger Fabric. The application was integrated with an SDN (Software-Defined Networking) Controller to exploit the synergy of threat intelligence and automation. Its primary objective was to enable seamless collaboration among organizations during distributed denial of service attacks and blacklist potentially malicious IP addresses during the flood, based on collective threat intelligence. In their study [15], the authors developed a CTI sharing platform tailored to the requirements of real-time threat intelligence in electrical power and energy systems. The platform comprised a generalized publish-subscribe middleware, which communicated with a Hyperledger Fabric network. Subsequently, the research was expanded [16] to tackle privacy concerns stipulated by GDPR (General Data Protection Regulation) and the performance overhead of storing large volumes of data on-chain. To accommodate this known issue in all Blockchains, they only stored the hash values of STIX objects on the Fabric Network, while storing the actual data on a separate database. Furthermore, the authors conducted both quantitative and qualitative analysis of the network's performance concerning various types of attacks. This work, although focused on threats in Energy systems, which is undoubtedly a critical domain, highlights the significance of strengthening cybersecurity and the growing interest in the development of advanced threat info sharing systems, leveraging technologies like Hyperledger Fabric and private permissioned Blockchains.

III. CTI SHARING ON BLOCKCHAINS: REQUIREMENTS, CHALLENGES AND ADVANTAGES

Cyber Threat Intelligence is a challenging field, particularly in the context of multi-party collaboration, which clearly makes a lot of sense for both corporate and public sector cyber defense. Considering that CTI itself needs to be trusted and protected from malicious infections and alterations, the sharing of CTI among multiple organizations requires overcoming several challenges; the heterogeneity of data sources, the trustworthiness of data, the timely delivery of information, the need for privacy and confidentiality, as well as the availability of data even without network connections, are some of these challenges. Moreover, the accuracy and relevance of CTI are crucial for proactive defense against cyber threats, and is also very critical to be left upon centralized services, vulnerable or even malicious themselves. It is not uncommon CTI to shared within networks that, even if they are private, they still engage a centralized trust model. Therefore, establishing a zero-trust framework for collecting, analyzing, and sharing CTI among multiple parties is worth investigating. This framework should address both the technical and operational challenges of CTI sharing, while ensuring the protection of sensitive information and privacy, even from entities which are normally taken for trusted. Although standardization itself is a significant aspect of designing and developing information systems, this paper does not discuss the advantages of standardizing CTI sharing using STIX and TAXII. The focus is on the investigation of the benefits of utilizing a blockchain system to improve the security of organizations that are willing or are already a part of a Cyber Threat Intelligence (CTI) sharing network.

As several technology options for satisfying the above requirements may exist, in the sequel we will discuss the security properties that acted as selection criteria for a blockchain platform and how they are implemented using the Hyperledger Fabric mechanism.

- Organization level privacy: The TAXII standard requires confidentiality in STIX object collections, ensuring that only authorized organizations have read access to them. We address this requirement by utilizing private data collections on Hyperledger Fabric where actual transaction data is stored only in the nodes of organizations that have the required access, while others only receive metadata and hashes for the transaction [4].
- Organization level access control: The TAXII standard restricts the ability to write data to authorized organizations only, which is also satisfied by the Private Data Collections mechanism of Hyperledger Fabric [4]. However, within a conventional centralized client-server implementation of a TAXII Server, one single hosting organization has complete write authorization on all data stored in the database; this alone can be a deal-breaker for the participation of critical-mission strategic organizations (e.g. defense and civil protection bodies) in CTI sharing networks.
- Data Integrity: The TAXII standard does not impose a strong requirement or mechanism for verifying the integrity of the data, as this is out-of-scope of the standard. Nevertheless, it is considered necessary for any application in the field of cyber se-

curity to have a mechanism for data integrity verification. One can argue that organizations participating in a CTI sharing network generally rely on the information provided by other peers of the network, as they share a common goal; still, data integrity checks must be performed, since data tampering on the TAXII Server by malicious actors may lead to infected security information reaching all CTI consumers; thus, the integrity of shared CTI data constitutes a central point of failure. Being a central element of the nature of the Blockchain philosophy, this requirement is satisfied by all blockchain ecosystems, as data integrity needs to be verified and signed by all peers holding the information. [5]

- High Availability: For a CTI sharing application, it is crucial to ensure high availability. However, managing the TAXII Server, even within a single corporate network protected by firewalls, employing replicas etc, the single logical TAXII server is also a single point of failure. Malicious actors can launch various types of denial-of-service attacks that could render the server non-functional during critical times, such as attack campaign timeframes. Hyperledger Fabric offers a solution to this challenge by enabling organizations to manage multiple peers that provide redundancy of data and services at the organizational level, while still satisfying the data control and integrity requirements. The nodes of all organizations that participate in the network maintain a copy of the distributed ledger at all times [5]. This ensures that access of other peers to CTI can be provided, even if all nodes of one organization become for any reason unavailable.
- Non Repudiation and Auditability: Injection of incorrect or even malicious cybersecurity information from one member to the CTI sharing network, can have catastrophic implications for other members. In such a case, in centralized systems governed by one single entity, it cannot be guaranteed that the information producer will be charged for its erroneous or malicious activities so as to be rendered responsible. However, the TAXII Server administrator(s) should not be able to take any action that protect any peer from taking the responsibility of its mistakes or malicious activities. The inherent feature of decentralized transaction write-only transaction ledgers been maintained by all Blockchain nodes of all organizations, typically satisfies the non-repudiation and auditability requirement.

Beyond the security properties achieved by migrating from a client server model to a blockchain platform there are significant platform specific advantages in Hyperledger Fabric compared to other blockchain networks.

- Hyperledger Fabric is a permissioned blockchain network. Compared to public blockchain networks such as Ethereum access of new members is strictly controlled and must be first approved by other participants [6]. We believe that this model serves better the purpose of threat intelligence sharing.
- Hyperledger Fabric offers feature-rich ways of interacting with the network using the Fabric SDK using general purpose programming languages and is designed with organization level decentralization in mind while public blockchains are designed for censorship resistance first. While these concepts mays seem similar at first, they are different.

As a conclusion, it is apparent that the development of a multiparty CTI sharing application on Hyperledger Fabric can potentially resolve several issues inherent in the traditional client-server model.

The primary challenge encountered by this paper is the commitment to adhere strictly to a STIX/TAXII standards implementation on a blockchain ecosystem. As these standards were originally designed to function on a RESTful API, the challenge arises from the need to translate them into an equivalent decentralized version. This means that to comply with the standard, HTTP requests must be used, and a REST API server must be integrated into the blockchain network. As mentioned above, all collaborating organizations will participate in the Blockchain network, each with at least one node; clients coming from each organization will be connected to their organization's node(s), as shown in Figure 1 below.

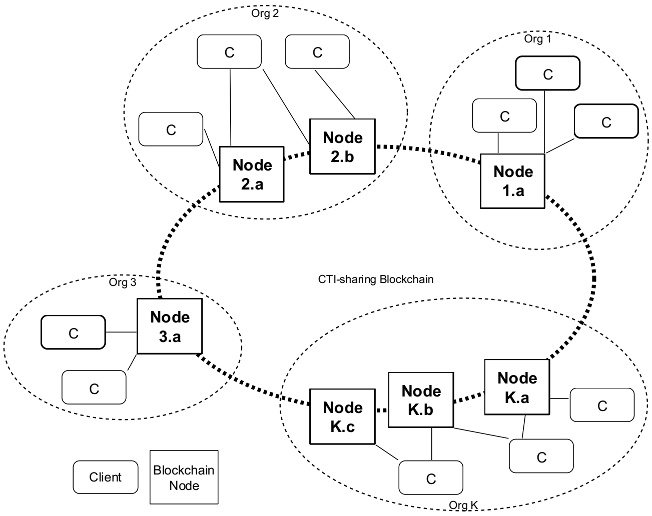


Figure 1. A Blockchain architecture for CTI sharing.

To address this, we need to enhance Hyperledger Fabric with a REST API server that acts as an intermediary to forward requests from the organization's clients to the blockchain network. This is not a bypass to the decentralized nature of the network, considering that it only acts as a relay or secondary client to the blockchain network; furthermore, in this architecture Hyperledger Fabric is decentralized only in the organization level, in contrast with Ethereum or other public blockchains that provide user level decentralization. The nature of this application allows organization-level decentralization, although user-level decentralization can still be possible in specific environments and applications.

Regarding the adherence to the STIX standard, a validator has been developed to verify STIX compliance; implemented as a proof of concept, it currently operates on the API server and is intended to be deployed as Fabric chaincode. This task was comparatively less challenging than integrating the TAXII server into the network, due to two reasons: Firstly, Fabric's data layer is based on a key-value store [7] (either LevelDB or CouchDB) that is inherently similar to JSON objects. Secondly, the validator can be tested off-chain and then effortlessly moved on-chain as it requires minimal or no additional blockchain-metadata to function properly. The work described in this paper encountered a final difficulty in selecting a scope for the API specification to design, implement, and document [8]. The TAXII standard is highly extensible [8], which made it challenging to identify the essential features that offer the basic functionality of threat intelligence sharing and facilitate system management, especially in scenarios involving multiple parties. The process of designing and implementing REST API endpoints that comply with the TAXII standard has been fully documented at various levels, using UML diagrams.

IV. System Design

As we mentioned above, the basic TAXII specification consists of a series of REST API endpoints [8]. The first step in order to design the system is to select a subset use cases to implement at the REST API level. These endpoints are divided into two categories, those that facilitate interaction with the blockchain that are completely custom, and those that provide the functionality required by the TAXII server standard; the API consumers should observe behavior identical to their client/server equivalent. This is shown as a use-case diagram in Figure 2.

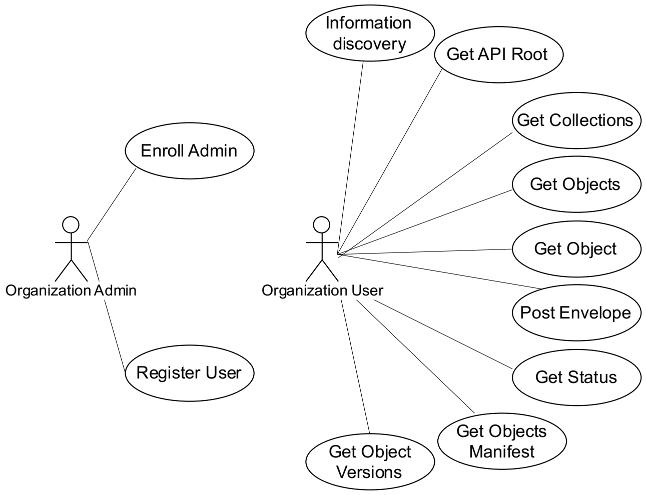


Figure 2. Roles and access to services offered.

In summary, based on the use case diagram, organization administrators are responsible for enrolling and registering users on the fabric-network [9]. Organization users can interact with every endpoint specified by the TAXII standard, with one critical exception: the ability to delete a STIX object from a collection. This is because blockchain transactions result in an immutable record of assets being transferred from nodes - CTI producers to other nodes. Thus, the best approach is to restrict access to assets beforehand using private data collections and access control. Once an asset reaches an organization's blockchain, it cannot be removed; this is a required behavior that allows the identification of the origin of false CTI coming from compromised or malevolent organizations.

The next step is to create an architecture containing an overview of the software components needed to implement the required functionalities. The UML component diagram below represents these components as containers or processes that live inside an organization's node; further details about their physical deployment will be discussed later. It is important to emphasize that this architecture needs to be implemented inside every single organization participating in the CTI-sharing Blockchain. Therefore, organizations may implement customized versions of this architecture in their production systems, such as fewer or more peer nodes, as discussed earlier; another parameter is whether one organization will be participating in the ordering service or not. Without compromising the principles of the proposed decentralized approach, we assume that every organization implements an identical infrastructure to standardize and simplify aspects of the blockchain layer that will be presented. Such an architecture is shown in Figure 3 below.

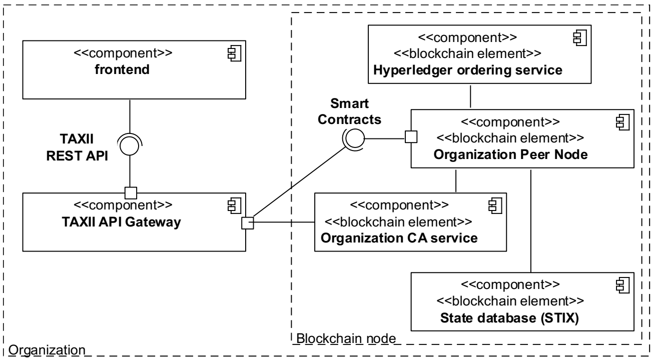

<<component>> frontend
TAXII REST API
<<component>> TAXII API Gateway
<<component>> <<blockchain element>> Hyperledger ordering service
Smart Contracts
<<component>> <<blockchain element>> Organization Peer Node
<<component>> <<blockchain element>> Organization CA service
<<component>> <<blockchain element>> State database (STIX)
Blockchain node
Organization

Figure 3. Component-level architecture of Blockchain nodes in organizations.

In the sequel we briefly discuss the system architecture shown in the above UML component diagram. The frontend component can be any implementation of a UI using typical web technologies (Apache, js), or a custom smart client that provides services such as event triggering upon incoming CTI etc.. In both cases, the frontend consumes the TAXII Rest API. The frontend needs not to be aware that the TAXII API offering is really based on a decentralized application. This allows even third-party TAXII clients to be used.

The Organization API Gateway is responsible for three main tasks: First, it validates the compliance of incoming requests with the TAXII standard, by checking HTTP headers, methods, and body parameters. It rejects any input that does not conform with the TAXII standard, without any further interaction with the blockchain backend. Second, it provides user authentication and authorization, by utilizing the organization's certification service(s). The API Gateway receives a token from the client and checks the organization's database for a corresponding certificate stored on behalf of this client. If the certificate exists, it is retrieved, and the request is relayed to the network with that certificate. If an authentication error occurs, the user is informed accordingly.

The third and most important function of the API Gateway is to submit to the blockchain-based backend, transactions on behalf of the organization's clients. These are the core of the TAXII server as they implement the main application business logic and ensure threat intelligence sharing functionality that benefits from the immutability of the blockchain. It is important to note that the transactions are not being run on the API Gateway itself: instead, they are submitted to the Hyperledger Fabric network to be executed on the Blockchain. The results are then received and forwarded to the actual clients. Further discussion on the smarts contracts will follow in the next section.

Consistent to the diagram of the software components, we present a UML deployment diagram to show the assignment of components to execution nodes. This deployment, shown in Figure 4, refers to a specific organization, and seems reasonable enough for production, as it physically isolates the REST API, the Fabric Infrastructure and the ordering service which are in fact very different in business functions.

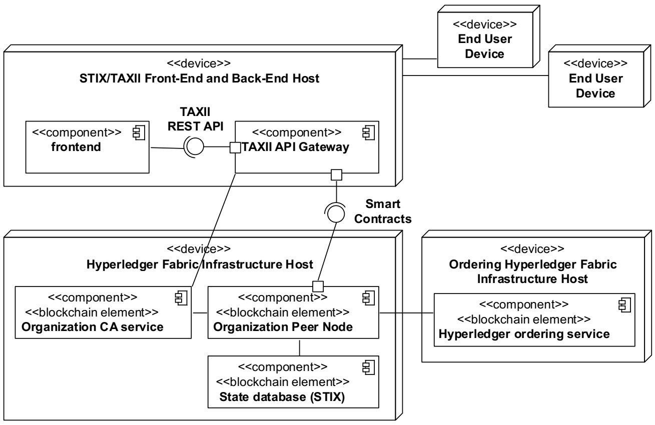

<<device>> End User Device
<<device>> End User Device
<<device>> STIX/TAXII Front-End and Back-End Host
TAXII REST API
<<component>> frontend
<<component>> TAXII API Gateway
Smart Contracts
<<device>> Hyperledger Fabric Infrastructure Host
<<component>> <<blockchain element>> Organization CA service
<<component>> <<blockchain element>> Organization Peer Node
<<component>> <<blockchain element>> State database (STIX)
<<device>> Ordering Hyperledger Fabric Infrastructure Host
<<component>> <<blockchain element>> Hyperledger ordering service

Figure 4. Deployment of the CTI Blockchain architecture.

Notably, a possible option for deploying the system is to combine the REST API and the fabric organization infrastructure into a single physical host. This approach may be beneficial for smaller organizations that want to conserve resources. However, it is advised to deploy the ordering nodes in separate physical hosts to improve security and performance. Furthermore, as mentioned earlier, it is possible that each organization maintains more than one blockchain node and ordering service, as shown in Figure 5 below.

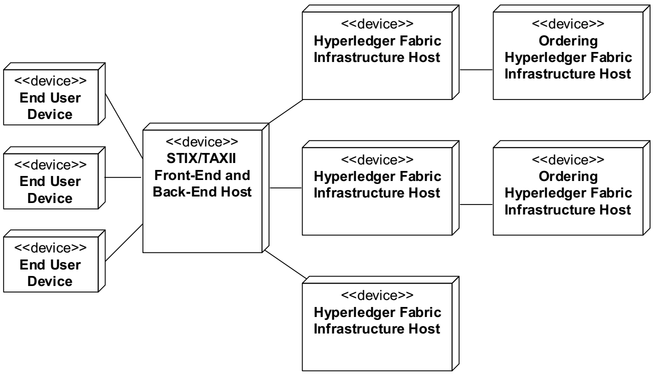

<<device>> End User Device
<<device>> End User Device
<<device>> End User Device
<<device>> STIX/TAXII Front-End and Back-End Host
<<device>> Hyperledger Fabric Infrastructure Host
<<device>> Hyperledger Fabric Infrastructure Host
<<device>> Hyperledger Fabric Infrastructure Host
<<device>> Ordering Hyperledger Fabric Infrastructure Host
<<device>> Ordering Hyperledger Fabric Infrastructure Host

Figure 5. Alternative CTI Blockchain deployment for smaller organizations.

V. TECHNICAL IMPLEMENTATION

The main components that implement the blockchain are the peer nodes, which are the fundamental building blocks of the Hyperledger Fabric. These nodes serve two essential functions that comprise the blockchain network: ledger and transaction management. In the context of this paper, it is important to note that the peers serve as the hosting component for chaincode, which is a collection of smart contracts that are the main mechanism for interacting with the network [10]. Each peer node implements a number of smart contracts as shown in the UML class diagram in Figure 6 below.

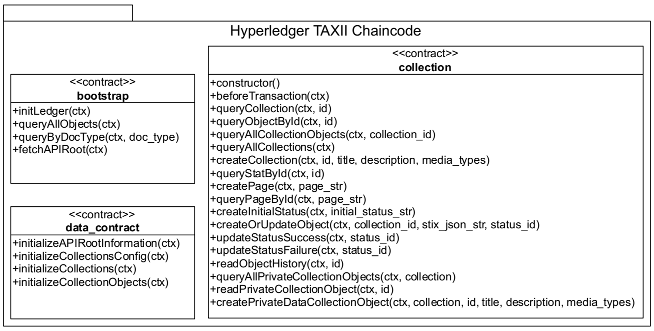
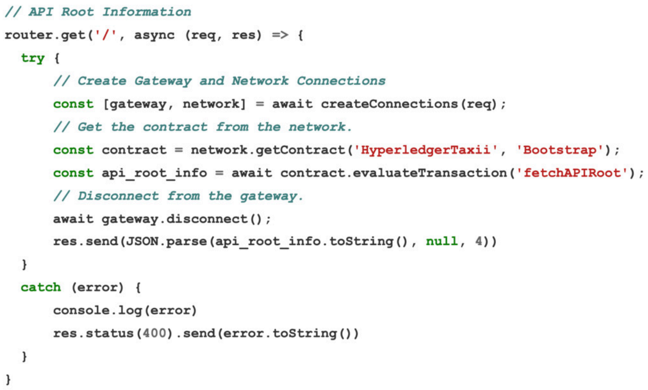

Hyperledger TAXII Chaincode
<<contract>>
bootstrap
+initLedger(ctx)
+queryAllObjects(ctx)
+queryByDocType(ctx, doc_type)
+fetchAPIRoot(ctx)
<<contract>>
data_contract
+initializeAPIRootInformation(ctx)
+initializeCollectionsConfig(ctx)
+initializeCollections(ctx)
+initializeCollectionObjects(ctx)
<<contract>>
collection
+constructor()
+beforeTransaction(ctx)
+queryCollection(ctx, id)
+queryObjectById(ctx, id)
+queryAllCollectionObjects(ctx, collection_id)
+queryAllCollections(ctx)
+createCollection(ctx, id, title, description, media_types)
+queryStatById(ctx, id)
+createPage(ctx, page_str)
+queryPageById(ctx, page_str)
+createInitialStatus(ctx, initial_status_str)
+createOrUpdateObject(ctx, collection_id, stix_json_str, status_id)
+updateStatusSuccess(ctx, status_id)
+updateStatusFailure(ctx, status_id)
+readObjectHistory(ctx, id)
+queryAllPrivateCollectionObjects(ctx, collection)
+readPrivateCollectionObject(ctx, id)
+createPrivateDataCollectionObject(ctx, collection, id, title, description, media_types)

Figure 6. Smartc contracts implemented in peer nodes.

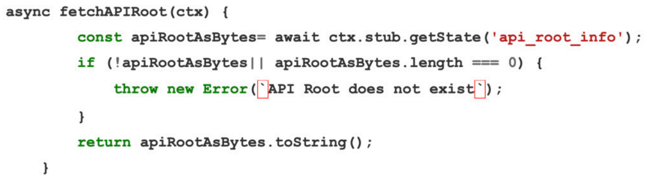
The diagram depicts a chaincode called Hyperledger TAXII Chaincode, which is installed on the file system of the organization's peer node. This chaincode implements three smart contracts: bootstrap, data_contract, and collection. The bootstrap and data_contract contracts are standard patterns used to initialize the ledger and perform some administrative tasks that are later removed in production. It is the collection contract that handles the core application logic of the TAXII server and is responsible for every task related to TAXII terminology, from API Root information to STIX object reads and writes. In Figure 7 below we list an example of the most basic REST API call in javascript code at HTTP server detail:



```
// API Root Information
router.get('/', async (req, res) => {
  try {
      // Create Gateway and Network Connections
      const [gateway, network] = await createConnections(req);
      // Get the contract from the network.
      const contract = network.getContract('HyperledgerTaxii', 'Bootstrap');
      const api_root_info = await contract.evaluateTransaction('fetchAPIRoot');
      // Disconnect from the gateway.
      await gateway.disconnect();
      res.send(JSON.parse(api_root_info.toString(), null, 4))
  }
  catch (error) {
      console.log(error)
      res.status(400).send(error.toString())
  }
}
```

Figure 7. JS code to implement a typical API call.

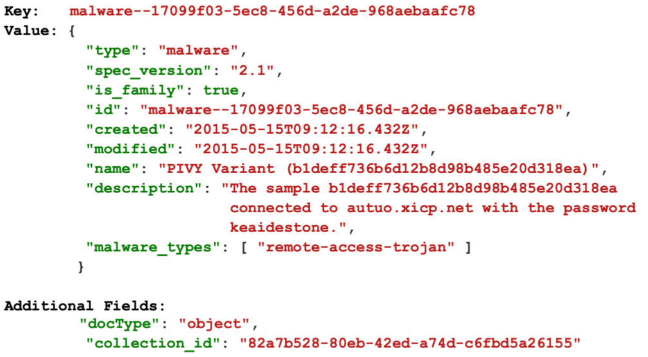
And at smart contract transaction level:



```
async fetchAPIRoot(ctx) {
        const apiRootAsBytes= await ctx.stub.getState('api_root_info');
        if (!apiRootAsBytes|| apiRootAsBytes.length === 0) {
             throw new Error(`API Root does not exist`);
        }
        return apiRootAsBytes.toString();
   }
```

Figure 8. JS code to implement a transaction.

The remaining three components to implement an instance of the system are the state database, certificate authority node, and ordering service. For the purposes of this work, the certificate authority node and ordering service remain unaltered from their default roles in Fabric. The state database serves as a cache for the current key-value pairs, known as the world state, in Hyperledger Fabric [7]. Past values are stored in the ledger. The state database was changed from Hyperledger's default levelDB, to couchDB for rich querying support and improved performance and management. It also stores certificates for organization users and CTI compliant data in STIX language [11] where a key of **<ObjectType>-<ID>** or **<ID>** format is used to quickly retrieve the STIX payload along with a custom field **<docType>**. In some cases a **<collection_id>** field is added and then stripped away before displaying to the user to identify the collection an object belongs to. These are standard best practices provided by fabric code examples. An object of type malware is stored inside the world state database in the format shown in Figure 9



```
Key:   malware--17099f03-5ec8-456d-a2de-968aebaafc78
Value: {
         "type": "malware",
         "spec_version": "2.1",
         "is_family": true,
         "id": "malware--17099f03-5ec8-456d-a2de-968aebaafc78",
         "created": "2015-05-15T09:12:16.432Z",
         "modified": "2015-05-15T09:12:16.432Z",
         "name": "PIVY Variant (b1deff736b6d12b8d98b485e20d318ea)",
         "description": "The sample b1deff736b6d12b8d98b485e20d318ea
                         connected to autuo.xicp.net with the password
                         keaidestone.",
         "malware_types": [ "remote-access-trojan" ]
        }

Additional Fields:
        "docType": "object",
         "collection_id": "82a7b528-80eb-42ed-a74d-c6fbd5a26155"
```

Figure 9. JSON representation for a STIX "malware" object.

The data flow within the system initiates with the API consumers who are permitted to access the system either via a web browser or by directly making requests to the REST API. The REST API server carries out authentication checks to verify the user's authorization to participate in the blockchain network. After the authentication is confirmed, the API server retrieves the user certificate and functions as a client of the blockchain network, submitting transactions on behalf of the user. In Figure 10 we demonstrate the UML sequence diagram that corresponds to GET API Root endpoint code above.

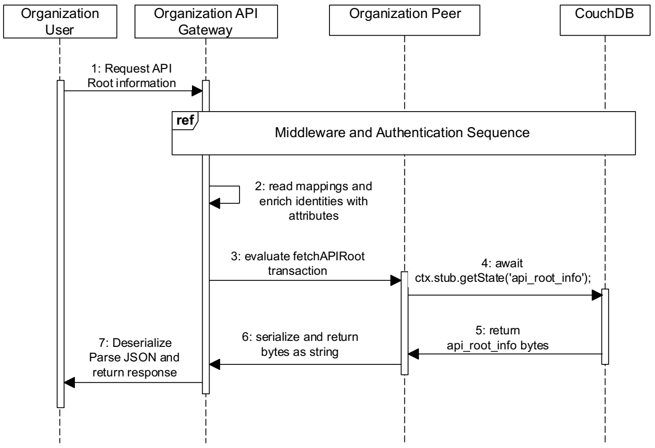

Organization User
Organization API Gateway
Organization Peer
CouchDB
1: Request API Root information
ref
Middleware and Authentication Sequence
2: read mappings and enrich identities with attributes
3: evaluate fetchAPIRoot transaction
4: await ctx.stub.getState('api_root_info');
5: return api_root_info bytes
6: serialize and return bytes as string
7: Deserialize Parse JSON and return response

Figure 10. A sequence diagram for the "GET API Root" endpoint.

In certain cases, an endpoint might require the submission of multiple transactions. However, submitting multiple transactions is not a recommended practice due to the potential inconsistencies and interruptions that may occur during the code execution. A more effective approach is to map each

REST API endpoint directly to a distinct Hyperledger Fabric transaction. This ensures that the transaction is not only clearly defined but can also be further expanded into subtransactions as necessary. It is important to note, however, that the implementation of this principle is out of scope of this work, as it serves as a proof of concept.

VI. Conclusion

We introduce an architecture for implementing a decentralized Threat Intelligence sharing system utilizing the STIX/TAXII standard, over a Hyperledger Fabric network. The proposed system leverages blockchain technology to enhance security in existing threat sharing systems and standards. By implementing a private permissioned blockchain network like Hyperledger Fabric, the proposed system provides a range of advantages such as improved confidentiality, integrity, availability, non-repudiation, and auditability. The blockchain network ensures that only authorized parties have access to data, maintains a tamper-proof record of all transactions, guarantees that data is always accessible, prevents the denial of service, and provides a transparent and complete immutable record of all transactions. The proposed system uses a decentralized architecture, in which several components work together to provide seamless threat intelligence sharing functionality. These components include the front-end, organization API gateway, peer nodes, state database, certificate authority node, and ordering service. The decentralized architecture is designed to ensure that decentralization does not hinder the system's overall functionality. The proposed system employs REST APIs to facilitate end-users', including third-party TAXII clients interaction with the system, and smart contracts to execute transactions on the blockchain network. The results of this research demonstrate the feasibility and effectiveness of the proposed architecture for threat intelligence sharing systems.

Some future improvements to the functionality, security and performance of the overall system are as follows:

- More robust authentication scheme: In the context of security applications, the basic TAXII standard authentication method that employs the Authentication Basic scheme poses potential risks due to its simplicity. To mitigate these risks, we suggest as a future improvement using a more secure authentication scheme that combines Basic Authentication with Multi Factor Authentication to generate a new certificate. This approach significantly reduces the risk of unauthorized access, especially in cases where an attacker has the enrollmentID and enrollmentSecret elements. Using certificates instead of enrollmentID/enrollmentSecret pairs offers two significant advantages: it minimizes user credential exposure over the channel, even if encrypted with TLS, and the credentials are short-lived, which means they can be easily revoked through the Certificate Authority (CA) authority node if exposed to a malicious actor.
- Writing the STIX validator on chain: One way to ensure a common data format is to validate the STIX data using the validator before storing it in the TAXII Server. Currently, this validation is performed at the API Gateway level and at a basic level. However, this approach may allow organizations to deviate from the STIX standard after processing the Gateway code, which is off-chain. To address this, data validation can be performed at the peer nodes of the Hyperledger Fabric through Chaincode transactions, for more multi-party trust. This would solve the problem of organizations agreeing on a common data validation logic as the code would be common and visible to all. Additionally, the Fabric Chaincode Lifecycle process can change this code at any time in a manner that is agreed upon and approved by all organizations.
- Implementing voting functionality: In CTI systems, the anonymous exchange of threat intelligence often includes a reputation mechanism to incentivize sharing sensitive data with other analysts. However, in our proposed system, which is designed for smaller to medium-sized consortiums, such a mechanism is not necessary since it is not open to the public. However, a voting and/or ban mechanism implemented on the Blockchain may still be useful if the network grows beyond a certain size, to further prevent malicious activities such as data poisoning.

References

[1] Cobb, M. and Wigmore, I. (2021) *What is threat intelligence (cyber threat intelligence)? – definition from whatis.com*, *WhatIs.com*. Available at: https://www.techtarget.com/whatis/definition/threat-intelligence-cyber-threat-intelligence

[2] *What is STIX?* (2020) *Introduction to stix*. Available at: https://oasis-open.github.io/cti-documentation/stix/intro.

[3] (2020) *Introduction to taxii*. Available at: https://oasis-open.github.io/cti-documentation/taxii/intro.html.

[4] *Private data* (2017) *hyperledger*. Available at: https://hyperledger-fabric.readthedocs.io/en/release-2.2/private-data/private-data.html.

[5] (2017) *Ledger*. Available at: https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger.html.

[6] *Hyperledger fabric network* (2017) *hyperledger*. Available at: https://hyperledger-fabric.readthedocs.io/en/release-1.2/network/network.html.

[7] *Hyperledger Fabric model* (2017) *hyperledger*. Available at: https://hyperledger-fabric.readthedocs.io/en/latest/fabric_model.html.

[8] *TAXII specification* (2020) *TAXII Version 2.1*. Available at: https://docs.oasis-open.org/cti/taxii/v2.1/os/taxii-v2.1-os.html.

[9] *Registering and enrolling identities with a CA* (2017) *hyperledger*. Available at: https://hyperledger-fabric-ca.readthedocs.io/en/latest/deployguide/use_CA.html.

[10] *Smart contracts and chaincode* (2017) *hyperledger*. Available at: https://hyperledger-fabric.readthedocs.io/en/latest/smartcontract/smartcontract.html.

[11] *STIX specification* (2020) *STIXTM Version 2.1*. Available at: https://docs.oasis-open.org/cti/stix/v2.1/csprd01/stix-v2.1-csprd01.html.

[12] A new network model for cyber threat intelligence sharing using

blockchain (2019). Available at: https://arrow.tudublin.ie/cgi/ viewcontent.cgi?article=1003&context=nsdcon

[13] Traffic Light Protocol (TLP) Definitions and Usage (2022). Available at https://www.cisa.gov/news-events/news/traffic-light-protocol-tlp-definitions-and-usage

[14] Collaborative Cyber Attack Defense in SDN Networks using Blockchain Technology (2020). Available at: https://www.researchgate.net/publication/343616521_Collaborative_Cyber_Attack_Defense_in_SDN_Networks_using_Blockchain_Technology.

[15] Secure exchange of cyber threat intelligence using TAXII and distributed ledger technologies - application for electrical power and energy system (2021). Available at: https://dl.acm.org/doi/10.1145/3465481.3470476

[16] Secure and Efficient Exchange of Threat Information Using Blockchain Technology (2022). Available at: https://www.mdpi.com/2078-2489/13/10/463

DOI: 10.15439/2023F8594
ISSN 2300-5963 ACSIS, Vol. 35

Open Vocabulary Keyword Spotting with Small-Footprint ASR-based Architecture and Language Models

Mikołaj Pudo
0000-0002-0776-4703
Samsung R&D Institute Poland, Krakow, Poland
Warsaw University of Technology, Warsaw, Poland
Email: m.pudo@samsung.com

Mateusz Wosik
0009-0002-9530-8931
Samsung R&D Institute Poland, Krakow, Poland
Email: m.wosik@samsung.com

Artur Janicki
0000-0002-9937-4402
Warsaw University of Technology, Warsaw, Poland
Email: artur.janicki@pw.edu.pl

***Abstract*—We present the results of experiments on minimizing the model size for the text-based Open Vocabulary Keyword Spotting task. The main goal is to perform inference on devices with limited computing power, such as mobile phones. Our solution is based on the acoustic model architecture adopted from the automatic speech recognition task. We extend the acoustic model with a simple yet powerful language model, which improves recognition results without impacting latency and memory footprint. We also present a method to improve the recognition rate of rare keywords based on the recordings generated by a text-to-speech system. Evaluations using a public testset prove that our solution can achieve a true positive rate in the range of 73%–86%, with a false positive rate below 24%. The model size is only 3.2 MB, and the real-time factor measured on contemporary mobile phones is 0.05.**

I. Introduction

MACHINE learning techniques are being developed nowadays in two distinct directions. In some applications, model sizes are constantly growing to support the increasing number of domains. This is especially visible in the case of large language models (LLM), in which the number of trainable parameters reaches hundreds of billions. Those models require tremendous amounts of computing power for inference. However, there is also a trend pushing the boundaries in the opposite direction by minimizing the model sizes. In this case, the models are designed for very specific tasks and they are most commonly deployed on devices with a limited amount of computing power, such as mobile phones or home appliances.

A good example of such a specific task is keyword spotting in an audio stream. This task aims to detect all occurrences of the given keywords in audio data provided either in streaming or non-streaming mode. Systems that solve keyword-spotting tasks are usually deployed on users' devices. Therefore, they need to have low latency and a small memory footprint. In the most basic case, the models are fitted to support detecting only a fixed number of keywords (e.g., wake words in contemporary voice assistants such as "OK Google", "Alexa", "Hey Siri" or "Hi Bixby"). Such models can be minimized well, even to sizes below 100 kB. However, users of voice assistants often request the possibility to customize the keywords, which introduces an open-vocabulary Keyword Spotting (KWS) problem. In this case, the model needs to be much larger to support the recognition of potentially arbitrary keywords.

In this paper, we present our solution to the KWS task for the non-streaming mode. It is based on the acoustic model (AM) architecture used in automatic speech recognition (ASR). However, to fit the entire system (model and engine) on the mobile device, we strongly reduced neural network layer sizes and applied post-training weights quantization. As expected, the baseline model performance was far from satisfactory. Therefore, we applied hypothesis re-scoring with a simple language model (LM). We also explored the idea of using recordings generated by a text-to-speech (TTS) model to improve performance on rare keywords not known during AM training. It should be noted that both improvements can be used independently of each other and can be applied to any type of AM.

The rest of this paper is organized as follows. In Section II, we discuss previous solutions to the KWS problem. In Section III, we present different parts of the model architecture: AM in Section III-A; LM together with the algorithm of its construction in Section III-B; keyword classifier, which makes the final decision is described in Section III-C; and extension of this module to multiple keywords is explained in Section III-D. All the variants of our solutions discussed in this paper were evaluated. Results of those experiments are presented in Section IV. Finally, we summarize this paper in Section V and provide a selection of possible future research directions in Section VI.

II. Related work

KWS can be split into two types: query-by-text (QbyT) and query-by-example (QbyE). In QbyT the keyword is provided by text, while in QbyE one or more “enrollment” audio recordings are provided during the initialization phase. Both types of KWS were considered before.

A thorough review of QbyT solutions can be found in [1]. It should be noted that currently, the most popular testset used to evaluate such solutions is a subset of Google Speech Commands (GSC) [2]. It is a public dataset developed for training and evaluating models designed for simple command recognition. It is also used for the KWS task. GSC contains a small number of keywords, hence it is more suited for classification problems with a fixed number of classes rather than open-vocabulary tasks.

Solutions designed for open-vocabulary QbyT evolved similarly to the ASR models. There was a long phase of solutions based on the hidden Markov model (HMM) – Gaussian mixture model (GMM) architecture [3], [4]. Later the GMM component was replaced by deep neural networks (DNN) [5], [6]. Finally, with the advent of sequence-to-sequence architectures to speech processing, ideas such as connectionist temporal classification (CTC) [7] and attention mechanism [8] became standard solutions in KWS as well.

Present-day solutions to open-vocabulary KWS can be based simply on CTC. In [9], the model contains three long short-term memory (LSTM) layers and the output is at the character level. The keyword is detected once the negative log posterior is below a predefined threshold. LSTM-CTC architecture is also used in [10]. However, in this case, the model operates at the phonetic level. The keyword is represented by one or more phone sequences. During the inference phase, those variants are compared with the hypothesis using minimum edit distance. The decision threshold is estimated for each keyword separately based on the training data and the lexicon.

The connection between KWS and ASR can be also limited to the training phase. In [11], the model is trained using CTC in a multi-task approach with ASR and KWS outputs. During inference, only the KWS output is used. Such an approach is intended to improve the model's ability to generalize and improve the performance in acoustically challenging conditions. The solution presented in [12] is based on an audio encoder network and a convolutional classifier. The encoder network is trained using the ASR task. The classifier network uses filters computed by a keyword encoder. The keyword encoder is a bidirectional LSTM (BiLSTM) layer processing the text keyword provided by the user.

Some solutions are based on the attention mechanism. An ASR model composed of five LSTM layers is used in [13]. The model is trained with CTC loss but has an additional keyword encoder and attention network which is used to direct the prediction network towards the keyword of interest. One of the models presented in this paper employs a phoneme level n-gram LM, which improves the model's performance. An attention-based model is presented in [14]. However, this is one of many solutions with a fixed-size output layer, hence supporting new keywords requires retraining the model.

Another approach to supporting open vocabulary is the on-the-fly adaptation of the model during the initialization phase. In [15] an embedding model is pre-trained on a large number of classes. A classification layer for specific keywords is added on top of the embedding model and adapted using only a handful of samples. Such classification layers can be independent of each other and use the same embedding model, since in the adaptation phase only the last layer is modified. A similar solution based on a few-shot transfer learning is described in [16]. It should be noted that usually KWS solutions are deployed on devices with limited resources, hence performing any type of model adaptation might be troublesome.

Many QbyE solutions are also based on the concepts used in the ASR. In [17] the ASR model with CTC is applied to enrollment phase recordings. N-best phonetic level keyword labels are stored together with their log probabilities in the keyword model. During the inference phase, each audio is processed with a similar ASR model. For each keyword from the keyword model log probability is computed and added to the final score. The keyword is detected if the score is above a certain pre-determined threshold. Similarly in [18] a small-footprint ASR model based on CTC is employed. In the enrollment phase, phonetic level posteriorgrams obtained from the model are used to build a finite-state transducer graph (FST) that models the keywords. In the inference phase, the audio is processed with the ASR model, and the output is scored using the keyword model FST. Finally, the score is compared with the threshold, which is chosen automatically based on the enrollment recordings and negative samples generated by rearranging each enrollment waveform. Since both of those solutions operate on the phonetic level rather than directly on the acoustic level, they can be treated as converting the QbyE task to QbyT.

QbyE can be also approached on the acoustic level. In [19], [20], [21], [22] audio embeddings are computed for both enrollment and inference phase recordings. Distance between those vectors is computed using different metrics and compared to a predefined threshold.

III. Model architecture

The main assumption in the KWS task is that the keyword might be any phrase, most likely not known during model training. This means the model architecture cannot be based on a classifier with a fixed-size softmax-type output layer. A more elaborate solution is necessary for this problem. Fig. 1 provides a general overview of our solution. We decided to adopt the AM architecture developed for the ASR task. To gain high accuracy, AMs usually contain hundreds of millions of trainable parameters. However, since KWS is simpler than speech recognition, we decided to leverage knowledge distillation to minimize the model size, but still keep the capability of dealing with large or open vocabulary, which is the base of

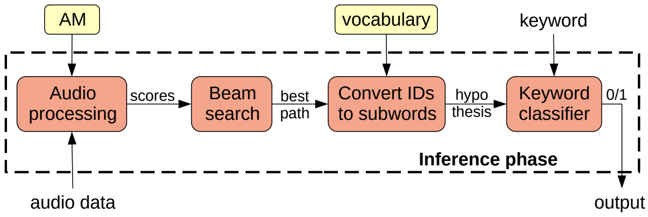


Fig. 1. Overview of the baseline solution.

ASR. Our solution employs AM in a standard way to generate frame level subwords. This is followed by a beam search to create the best path, which is converted to the final hypothesis using model vocabulary. The last step consists of keyword classification, which compares the hypothesis with the given keyword to make a final binary decision: whether a keyword is present in the recording or not.

A. Acoustic model architecture and training

The AM used in our solution is based on monotonic chunkwise attention (MoChA) [23]. It is a sequence-to-sequence model split into encoder and decoder parts, both composed of recurrent neural layers (RNN). The encoder computes embedding vectors for each frame in the audio stream. The attention-based decoder combines and transforms those embeddings into a series of subwords that will constitute the hypothesis. The MoChA used in our decoder is a modification of soft attention, designed to address the issue of real-time (online) processing. It consists of two attention layers. The first layer uses hard monotonic attention for each frame to determine whether the second layer needs to be run. The second layer uses soft attention over a small sequence (“chunk”) of frame embeddings to compute the context vector. Each chunk comprises the current frame and several frames preceding it. The chunk length is a model hyperparameter.

To compress the model, we use a knowledge distillation approach, in which a small student model is trained to mimic a large teacher model [24]. Both models are based on MoChA architecture but differed in the sizes of the selected layers.

The teacher model encoder consists of six BiLSTM layers with 512 units for each direction with 0.3 dropout. To reduce the time domain size, max-pooling layers are used after each of the three initial BiLSTM layers. The decoder consists of one unidirectional LSTM layer with 1000 units, an embedding layer of size 621, and a readout layer of size 1000. The decoder also contains two attention layers of size 512 each. Chunk size two is used for chunkwise attention. The teacher model has 50.1 million trainable parameters (34.2 million in the encoder and 15.9 million in the decoder). It was trained jointly with CTC loss and categorical cross-entropy loss [25]. The CTC was used with the encoder output to encourage the model to learn monotonic alignments. We used 0.1 label smoothing of the output softmax distribution.

The student model has the same architecture as the teacher model but with reduced layer sizes. Each of the BiLSTM encoder layers has 124 units for each direction with 0.3 dropout. The decoder comprises an LSTM layer with 256 units, an embedding layer of size 156, a readout layer of size 256, and two attention layers of size 124 each. The student model includes a total of 3.1 million trainable parameters (2.0 million in the encoder and 1.1 million in the decoder).

The input to both models consists of 40-dimensional power-mels computed over a period of $25\,\mathrm{ms}$ with a $10\,\mathrm{ms}$ step. During training and inference, we applied cepstral mean and variance normalization, which were computed over all training samples. The model’s output consists of 500 subwords obtained using the method described in [26]. It uses the adaptation of byte pair encoding (BPE) to word segmentation to generate a compact symbol vocabulary of variable-length subword units. The vocabulary was generated from all the transcriptions contained in the training and testing sets. 500 was chosen as the vocabulary length since this size keeps the output model layer small and allows for more accurate recognition of rare or out-of-vocabulary words. Four special subwords were added to the vocabulary: `<s>` (beginning of a sentence), `</s>` (end of a sentence), `<unk>` (non-speech events or characters not included in the Latin alphabet) and `<blank>` (required for CTC training).

The teacher model was trained for 23 epochs with a learning rate equal to 1×10^{-4} in the first epoch. The learning rate was reduced by a factor of 0.95 after every 10 consecutive validation steps without change. Validation was done every 5000 steps. During training, we added randomly selected room impulse response (RIR) and mixed the data with a randomly selected noise signal with a ratio between -2 and $12\,\mathrm{dB}$. RIR dataset contains simulations of distances from one to five meters and reverberation time between $0.2\,\mathrm{s}$–$0.9\,\mathrm{s}$. Noise data consisted of both internal noise dataset and AudioSet [27]. The internal noise data contained audio from various environments and is similar to MUSAN [28]. Moreover, the features were augmented with SpecAugment [29] by masking one frequency block of size eight and one time-domain block of size 50.

The student’s total loss was the weighted sum of distillation loss and categorical cross-entropy loss with weights of 0.4 and 0.6, respectively. During knowledge distillation, we used temperature two to make teacher predictions softer. The student model was trained for 22 epochs with a learning rate equal to 4×10^{-4} in the first epoch. The same learning rate scheduler and parameters were utilized during student model training as for the teacher model. We observed accuracy degradation while using SpecAugment during student model training; therefore, we skipped this type of augmentation.

Both models were trained on generic ASR datasets. We used LibriSpeech [30] (all training splits, $960\,\mathrm{h}$), $1779\,\mathrm{h}$ of English Mozilla Common Voice [31] (version 7.0, excluding sentences selected for testing). Additionally, we used $4\,\mathrm{h}$ of audio data not containing speech (silence or quiet noise). The sampling rate of all audio data was $16\,\mathrm{kHz}$. During training, we used greedy decoding, while in the inference phase, we applied the beam search algorithm with a beam size equal to four.

To further minimize student model size, we applied post-training 8-bit quantization. This step reduced the model size

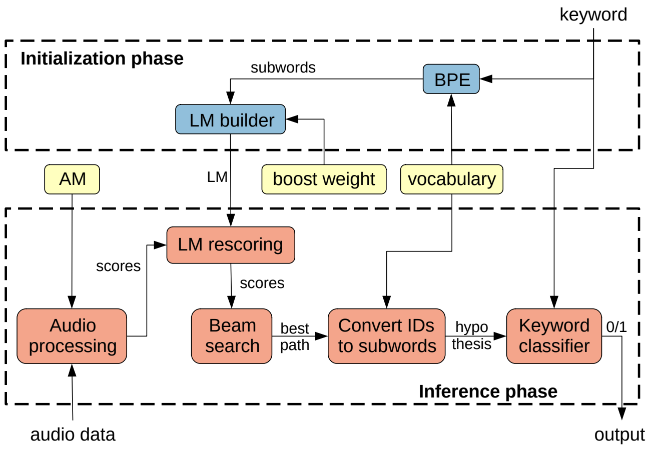


Fig. 2. Overview of the solution with static LM.

from 13 MB to 3.2 MB. This model achieved the following word error rates (WER): 16.0 on LibriSpeech test-clean and 30.9 on LibriSpeech test-other.

B. Language model architecture and initialization

LM can be used to modify scores generated by AM. This process is known as re-scoring. We decided to use a very simple 1-gram LM, where the model is a vector of the same length as the AM output layer size. Such a type of LM introduces only a minor additional memory footprint and a small increase in latency. With this kind of LM, re-scoring consists of element-wise multiplication of the scores returned by the AM and the LM vector. It is the initialization of weights that is the key to a LM of this type. This step should be performed only once for each novel keyword; hence it does not influence latency during the inference phase.

We decided to use a very simple initialization method which we called Static LM. In this method, LM weights are initialized only with two values: one and *boost* weight which is treated as a model hyper-parameter. A general overview of the Static LM method is shown in Fig. 2. The BPE algorithm is applied to the keyword with the same vocabulary as used to convert the AM scores to obtain the hypothesis. Subwords included in the keyword are assigned a *boost* weight, and all the remaining subwords are assigned 1. Note that setting *boost* weight to 1 will not change AM scores and setting *boost* weight to values smaller than 1 will decrease the probability of recognizing the keyword.

C. Keyword classifier

Generating an ASR-based hypothesis is only the first step in the KWS solution. Based on this hypothesis, it is necessary to decide whether the recording contains the required keyword or not. The pseudocode of the procedure we employed for this purpose is presented in Algorithm 1. Note that the recording which is processed by the AM might contain more speech data than just the keyword. To remedy this issue, we calculate the keyword length as the number of words and compare it with all the subsequences of the hypothesis of the same word length. We calculate the character level normalized Levenshtein distance between each such subsequence and the keyword. If the distance is smaller than a predefined threshold, the true value is returned by the system (keyword detected) and the false value is returned otherwise (keyword not detected).

Algorithm 1 Keyword classifier algorithm

```
Input: $keyword$ – custom keyword
Input: $hyp$ – hypothesis returned by AM
Input: $t$ – recognition threshold
$l \leftarrow len(keyword)$ {number of words in keyword}
for $s \in \{sub : sub$ is substring of $hyp \wedge len(sub) = l\}$ do
    if $dist(keyword, s) \leq t$ then
        return $true$
    end if
end for
return $false$
```

D. Multi-keyword classifier

In the generic text-based KWS task, the keyword is provided by the text. However, often additional audio data can be leveraged to improve recognition rates. This can be done for example by requesting the user to provide a spoken version of the keyword. Since such an approach requires additional action from the user, we decided to pursue an automatic solution. An overview of this method is presented in Fig. 3. We employ the TTS system to generate synthetic recordings representing the spoken version of the keyword. The number of those recordings depends on the TTS solution and can also be set as a parameter of the system. Each of those recordings is processed by the AM, followed by the beam search algorithm to generate a hypothesis. The original keyword is appended to the hypothesis list and duplicates are removed. This list is treated as containing additional variants of the original keyword and is later used during inference by the keyword classifier.

Algorithm 2 presents the multi-keyword classifier pseudocode. It is the extended version of the keyword classifier described in section III-C. Once more substrings of the hypothesis are selected for comparison, but this time they are compared with each keyword from the list prepared during the initialization phase. As previously, normalized character level Levenshtein distance and a predefined threshold are used to make the final decision.

The main idea behind the multi-keyword classifier is to improve the recognition rate on keywords that are very distinct from the phrases presented to the AM during training (eg. named entities or other non-standard phrases). In such cases, the AM most likely would return an incorrect hypothesis. However, provided those errors are similar across different samples of the same keyword, adding them to the classifier should increase the true positive rate (TPR). Hence this idea can be described as adding additional pronunciation variants

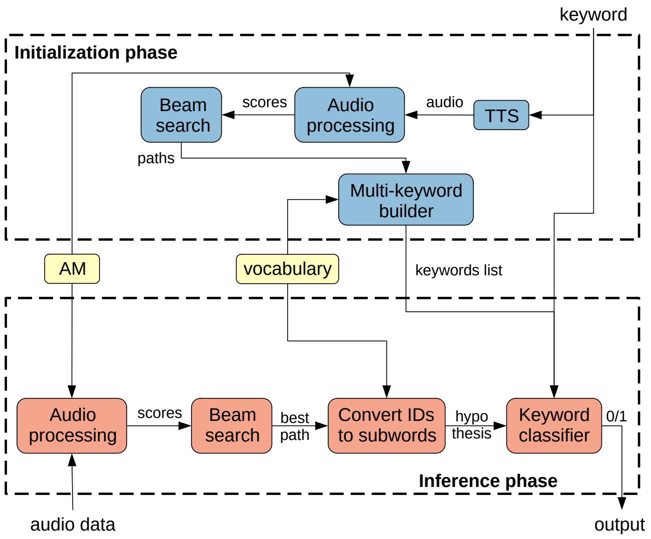


Fig. 3. Overview of the multi-keyword solution.

Algorithm 2 Multi-keyword classifier algorithm

```
Input: $keywords$ – list of custom keywords
Input: $hyp$ – hypothesis returned by AM
Input: $t$ – recognition threshold
for $keyword \in keywords$ do
    $l \leftarrow len(keyword)$ {number of words in keyword}
    for $s \in \{sub : sub$ is substring of $hyp \wedge len(sub) = l\}$ do
        if $dist(keyword, s) \leq t$ then
            return $true$
        end if
    end for
end for
return $false$
```

for the user-entered keywords. Obviously, this procedure might also have negative results. Phrases similar to the keyword, but different from it might be recognized with the same, wrong hypothesis, which would result in an increase in the false positive rate (FPR).

Note that the keywords list used by the multi-keyword classifier can be prepared during the initialization phase; therefore, this step does not influence inference phase latency. The impacts of the multi-keyword approach on memory and latency are linear with respect to the number of TTS recordings used. However, since the multi-keyword solution requires at most one additional string for each TTS recording, the memory footprint is negligible. The same reasoning can be applied to the impact on latency since Levenshtein distance calculation is much faster than ASR decoding.

IV. Experiments Results

A. Evaluation procedure

Our solution was tested with MOCKS 1.0 testset [32] and GSC v2 testset.

Since the AM was trained with English data, we used *en_LS_clean*, *en_LS_other*, and *en_MCV* subsets of MOCKS. Each test case is composed of two audio files and a keyword. One of those files is treated as initialization (enrollment) phase data and the other is treated as inference phase data. However, since our goal is to limit the user's interaction with the device, we skip the initialization phase data in the case of static LM. Furthermore, in the case of a multi-keyword classifier, we replace the initialization phase recording with synthetic data. We will refer to the inference phase data as test audio.

Each of the MOCKS subsets used for evaluation is split into three distinct parts:

- positive test cases – where the test audio contains a given keyword; we will call this part *pos*,
- similar test cases – where the test audio contains a different phrase than the given keyword, but both are close phonetically; we will call this part *sim*,
- different test cases – where the test audio contains a different phrase than the given keyword and the phonetic distance between both is large; we will call this part *dif*.

To present the impact of different hyperparameters on the above-described parts of MOCKS (true positive on *pos* and false positive on *sim* and *dif*), we used recognition accuracy as the main metric.

GSC is not suited for the open-vocabulary version of KWS, since it contains a very limited number of keywords. Nonetheless, we present the evaluation results of our solution on this testset for the sake of comparison with previous works. The most popular metric used with GSC is simply accuracy [33]. The negative test cases should be recognized as either _unknown_ or _silence_ special classes. The former contains words not included in the positive classes and the latter contains silence and non-speech events. Evaluation on GSC is a 12-class classification problem, while our solution is designed for the generic case of open-vocabulary classification. To remedy this issue evaluation for the negative test cases (labeled _unknown_ or _silence_) is performed in the following way:

- In the initialization phase for each positive keyword we prepare an LM and extended keywords list (if the multi-keyword classifier is enabled).
- After audio processing is done, for each positive keyword we perform re-scoring, apply beam search, convert the result to the hypothesis, and finally compute the character level Levenshtein distance between the hypothesis and the given keyword.
- Finally, we find the minimal distance from the previous step. If this distance is less than or equal to the threshold, this sample is counted as a false positive and a true negative otherwise.

We used an internally-developed end-to-end TTS system to generate synthetic recordings for the multi-keyword classifier. The system was composed of a neural AM and a vocoder. The AM mapped sequences of phonemic labels to acoustic features, while the vocoder mapped those features to audio samples. The set of phonemic labels contained language-specific (English) symbols of phonemes, word delimiters, and end-of-sentence marks. However, during synthesis, keywords were stripped of those marks. Acoustic feature vectors were derived from F0 (interpolated in unvoiced regions), mel-spectra, and band-aperiodicity as in the case of the WORLD vocoder [34]. The vocoder architecture was based on [35] and AM was similar to the Tacotron 2 [36] architecture as described in [37], with the use of the mutual information loss (MILoss) function [38]. Audio data included in the LJ speech dataset [39] and Hi-Fi Multi-Speaker English TTS Dataset [40] were used to train the entire system. The vocoder was trained separately for each voice. The AM was trained for 10 k epochs on the entire training data, followed by 450 k epochs of each voice-specific data.

For each keyword in MOCKS and GSC, we generated 10 synthetic recordings. We chose one male and one female voice for the experiments with a multi-keyword classifier based on two synthetic recordings. Two further types of experiments with this type of classifier were performed:

1) using clean audio data;
2) using audio data mixed with background noise and convolved with RIR.

We used the same types of noise and RIR as during AM training.

For confidence interval estimation we used bootstrap resampling of the testsets. Each testset was resampled 200 times with replacement. The trainset and model remained fixed. In order to provide a 95 % confidence interval we calculated the $[2.5, 97.5]$ percentile boundaries.

B. Evaluation results

1) Impact of the boosting weight: For the purpose of testing the impact of the static LM and different *boost* weights, we performed evaluations using values from the set $\{2^n : n \in \{0, 1, \ldots, 11\}\}$. Note that setting the *boost* weight to one means that none of the subword scores will be modified during re-scoring and only the AM scores will be taken into account in beam search. We treat this case as the baseline solution.

Fig. 4, 5 and 6 show acceptance rate in the function of the threshold applied in the keyword classifier for *en_LS_clean*, *en_LS_other* and *en_MCV* respectively.

Let us start the analysis of the results with $boost = 1$. For $threshold \in [0, 0.05]$, in all the testsets acceptance rates in *pos*, *sim* and *dif* are constant. This is due to the fact that all keywords in MOCKS are relatively short (phonetic transcription length $p \leq 16$). As long as the $threshold$ is greater than 0.05, acceptance rates in both *pos* and *sim* are increasing. However, those values in the former subsets grow slower than in the latter subsets. This means that increasing the threshold improves the TPR, but increases the FPR even faster. Values of acceptance rate in *dif* start to increase only with $threshold > 0.4$ since this test set contains test cases that are very different from the given keyword. For such test cases, the AM returns very different hypotheses from the keyword, even if they do not match the proper transcription. Similar

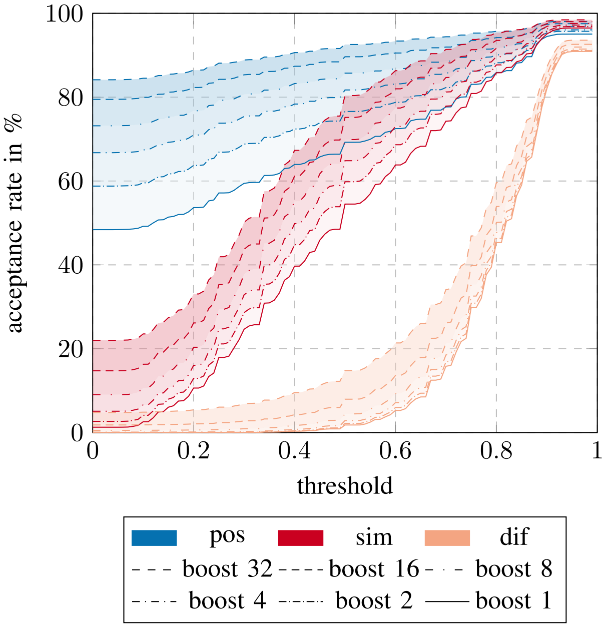


Fig. 4. Boosting results with static LM, without multi-keyword classifier, for en_LS_clean testset.

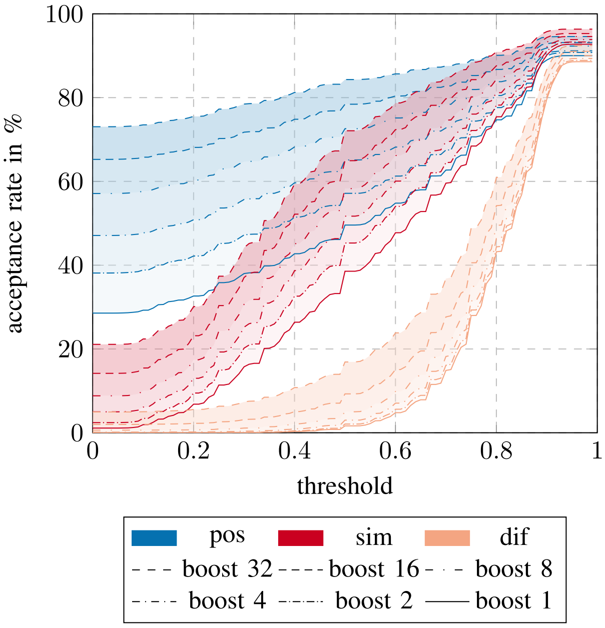


Fig. 5. Boosting results with static LM, without multi-keyword classifier, for en_LS_other testset.

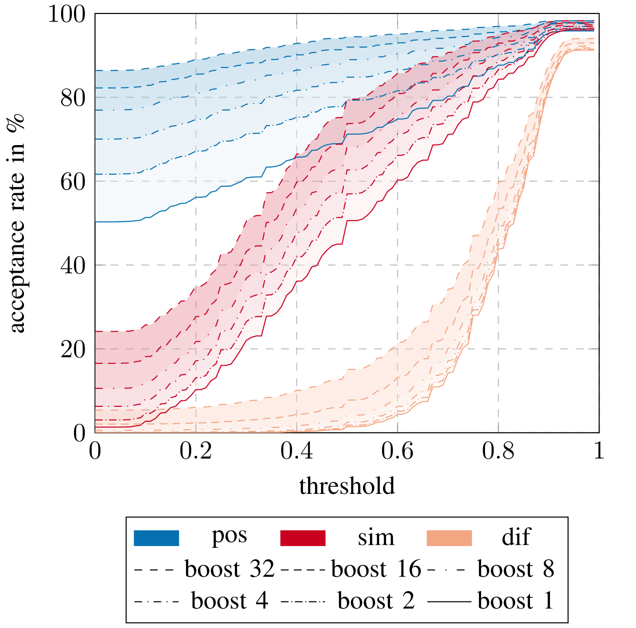


Fig. 6. Boosting results with static LM, without multi-keyword classifier, for en_MCV testset.

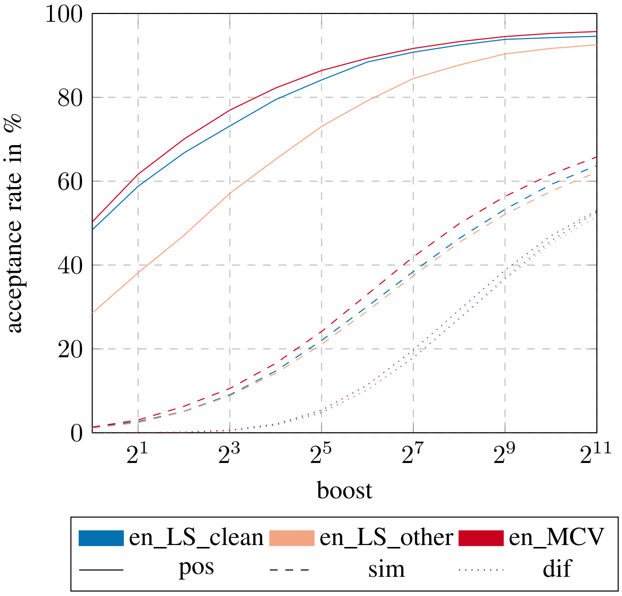


Fig. 7. Boosting results with static LM, without multi-keyword classifier, threshold 0, for MOCKS testset.

TABLE I

EER IN % FOR DIFFERENT VALUES OF BOOST ON MOCKS, WITHOUT A MULTI-KEYWORD CLASSIFIER.

boost	*en_LS_clean*	*en_LS_other*	*en_MCV*
1	30.05 ± 0.28	37.27 ± 0.26	27.30 ± 0.16
16	16.56 ± 0.24	26.04 ± 0.23	14.46 ± 0.19
32	$\mathbf{15.18 \pm 0.14}$	22.65 ± 0.13	$\mathbf{14.22 \pm 0.10}$
64	19.86 ± 0.11	$\mathbf{20.34 \pm 0.17}$	21.57 ± 0.12

observations also apply to cases with *boost* > 1, except for *dif*, for which the higher the $threshold$, the sooner this function starts to grow. This analysis suggests that it is safe to use $threshold = 0$.

For clarity, Fig. 4, 5 and 6 show evaluation results only for $boost \leq 32$. In Fig. 7 we present evaluation results for all the testsets and *boost* values up to 2048 using $threshold = 0$. It should be noted that for $boost \leq 32$, for all the testsets acceptance rates in *pos* are growing faster than in *sim*. These *boost* values are also accompanied by small acceptance rates in *dif* in all the testsets. However, the larger the *boost* gets, the faster acceptance rates in *sim* grow when compared to *pos*. This is also accompanied by a rapid growth of those rates in *dif*. This observation suggests that there is a limit for *boost* after which static LM brings more harm than benefit.

We use an equal error rate (EER) to estimate the optimal *boost* value. For each testset, FPR is calculated after summing *sim* and *dif* subsets. Fig. 8 shows EER for all the testsets and $boost \leq 128$ (for higher *boost* values EER is growing hence it is omitted). The width of the lines in Fig. 8 represent confidence intervals for EER estimation. It should be noted that the minimal EER values are located at *boost* equal to 32 or 64, depending on the testset. The rapidly growing value of EER for large *boost* is caused by the fact that in those cases FPR is always greater than the false negative rate (FNR). Since there is no point for which FPR and FNR are equal, the largest of those values is chosen as EER. Detailed values of EER for MOCKS can be found in Table I.

Fig. 9 shows the evaluation results for different *boost* values on the GSC testset using $threshold = 0$. Applying static LM improves accuracy from 84.60% for the baseline model to 95.97% at $boost = 32$. For higher *boost* values accuracy drops rapidly. This is due to the fact that with those large *boost* values the keyword subwords are favored during beam search. Therefore the number of negative test cases recognized as keywords grows. Detailed values of accuracy for GSC can be found in Table II.

2) Impact of the multi-keyword classifier: The main motivation for applying a multi-keyword classifier should be the increase in TPR, which ideally would not be accompanied by the increase of FPR. Our experiments show that this is not the case for MOCKS. Fig. 10 shows the evaluation results using *en_LS_clean* for multi-keyword classifiers initialized with 2 and 10 clean TTS recordings. Those results are compared to a single-keyword classifier solution. Furthermore in Table III we present exact evaluation results (acceptance rate) for each English testset in MOCKS. For clarity we limit those results to $boost = 1$ (no LM) and $boost = 32$, since this value gave the highest results as shown in Section IV-B1.

We observed that the improvement in acceptance rate on *pos* was larger than a similar increase on *sim* and *dif* only for small *boost* values. This difference was very small for the multi-keyword classifier initialized with two synthetic recordings. However, with 10 such recordings, the improvement of the acceptance rate on *pos* was almost 2 pp. larger than on *sim*.

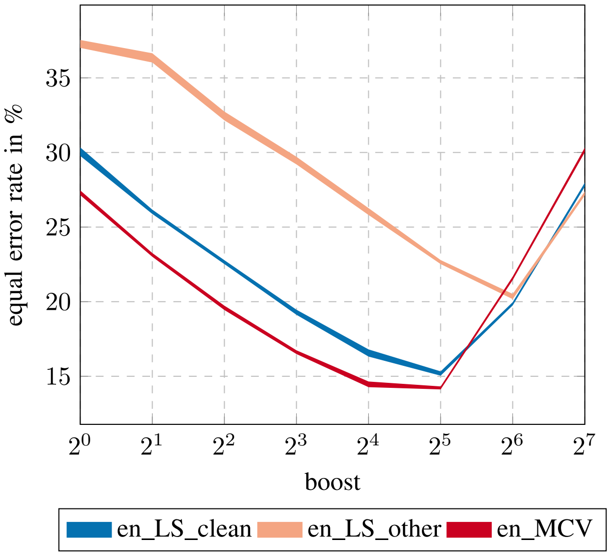


Fig. 8. Boosting results with static LM, without multi-keyword classifier, for MOCKS testset.

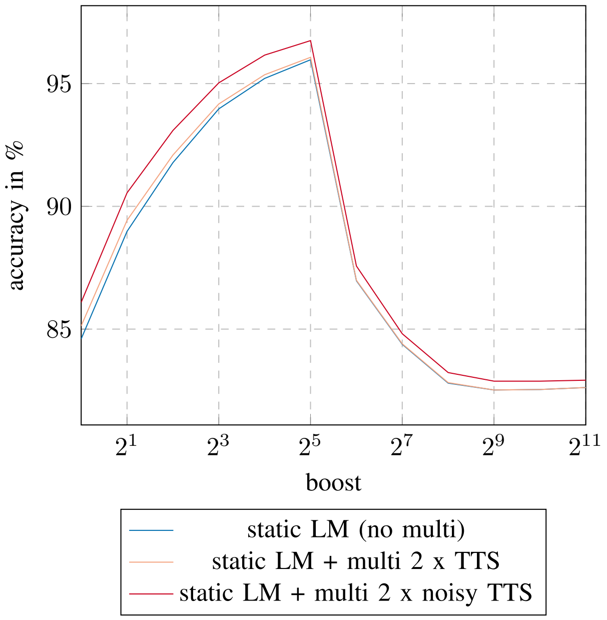


Fig. 9. Boosting results with static LM, with and without multi-keyword classifier, for GSC testset.

As soon as *boost* = 8 multi-keyword classifier introduced a larger increase in acceptance rate on *sim* than on *pos*, which was visible in all testsets. Mixing synthetic recordings with background noise and RIR did not improve the situation. The increase in acceptance rate on *pos* was smaller than on *sim*.

Evaluation results on GSC with a multi-classifier show a similar increase in accuracy (0.78 pp–2.3 pp depending on the *boost*). This improvement seems to be insignificant, nonetheless, it should be noted that it is gained in the range of 85 %–96 %. At this level of accuracy, even a minor increase in this metric means a substantial reduction in the number of errors.

TABLE II
ACCURACY IN % FOR DIFFERENT METHODS ON GSC.

method	accuracy
boost 1 (baseline)	84.60
boost 1 + multi 2 x TTS	85.13
boost 1 + multi 2 x noisy TTS	86.09
boost 32	95.97
boost 32 + multi 2 x TTS	96.07
boost 32 + multi 2 x noisy TTS	**96.75**

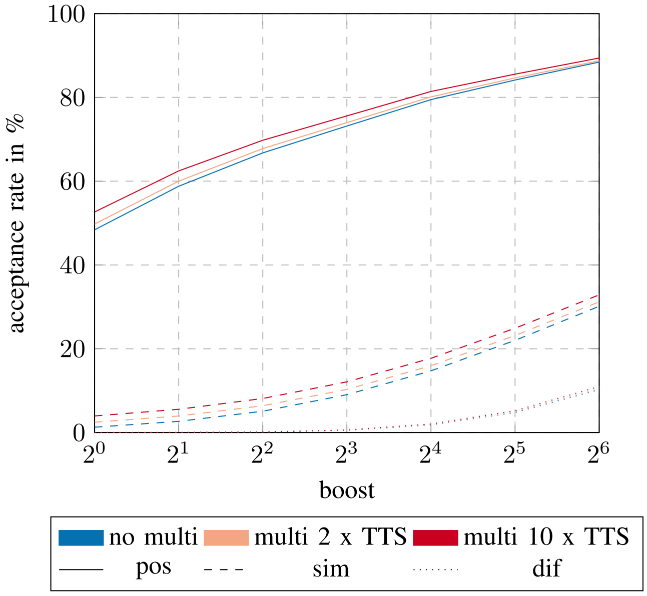


Fig. 10. Boosting results with static LM, with and without multi-keyword classifier, for en_LS_clean testset.

V. Discussion and Conclusions

Two major observations can be drawn from our experiments:

1) The increase of *boost* in static LM improves TPR, however, the larger the *boost* gets, the more dominant the increase of FPR compared to the increase of TPR.
2) The multi-keyword classifier introduces a positive impact on TPR only for very low *boost* values, while for high values of *boost*, however, the increase of FPR is much larger than the increase of TPR.

Using *boost* = 32 seems to be the right choice in general cases since this value resulted in the minimal EER in *en_LS_clean* and *en_MCV* and the largest accuracy in GSC. However, it should be noted that EER was minimal in *en_LS_other* with *boost* = 64, hence there is no universal value for this parameter.

The positive impact of the multi-keyword classifier is especially visible with the increased number of TTS recordings for each keyword. This number can be potentially unbounded, but a rule of thumb suggests using only a small amount (not exceeding 10) of such recordings. This suggestion is based on the observation that the longer the list of additional

TABLE III
ACCEPTANCE RATE IN % FOR DIFFERENT METHODS ON MOCKS.

method	en_LS_clean			en_LS_other			en_MCV		
	pos	**sim**	**dif**	**pos**	**sim**	**dif**	**pos**	**sim**	**dif**
boost 1 (baseline)	48.39	1.28	0.00	28.56	1.12	0.00	50.29	1.34	0.00
boost 1 + multi 2 x TTS	49.78	2.47	0.01	29.98	1.90	0.02	51.56	2.38	0.02
boost 1 + multi 10 x TTS	52.65	3.96	0.02	31.89	3.01	0.03	52.88	3.70	0.02
boost 1 + multi 2 x noisy TTS	50.88	3.81	0.05	30.76	2.94	0.07	52.94	3.83	0.04
boost 1 + multi 10 x noisy TTS	54.80	7.73	0.11	34.85	5.89	0.17	56.22	7.48	0.11
boost 32	84.15	22.01	4.79	73.05	21.10	4.96	86.40	24.16	5.40
boost 32 + multi 2 x TTS	84.66	23.17	4.95	73.46	21.92	5.12	86.82	25.44	5.58
boost 32 + multi 10 x TTS	85.55	24.90	5.22	74.45	23.61	5.46	87.46	27.10	5.82
boost 32 + multi 2 x noisy TTS	85.04	24.70	5.45	73.92	23.40	5.65	87.43	27.09	5.94
boost 32 + multi 10 x noisy TTS	86.54	29.22	6.24	76.14	27.46	6.65	88.52	31.18	6.59

keyword variants, the more likely the chance of false positive acceptance of phrases similar to the given keyword.

Evaluation of MOCKS and GSC with a multi-keyword classifier shows that there is a significant difference in both testsets. With MOCKS at $boost = 32$ adding 10 keyword variants increases FPR on *sim* more than TPR on *pos*. On the other hand with GSC at $boost = 32$ and 10 keyword variants we still observe improvement in accuracy. This can be explained by the fact that GSC contains short phrases and the negative samples are very different from the keywords in terms of character level Levenshtein distance. On the contrary, MOCKS contains longer phrases and a subset of negative samples similar to the keywords, hence they are difficult to distinguish. This means that adding additional keyword variants also increases the probability of false acceptance in this subset (*sim*). This analysis also leads to the conclusion that a multi-keyword classifier is an effective solution as long as one does not expect to deal with such challenging negative cases.

It might seem that a solution with accuracy equal to $96.75\,\%$ on GSC is far behind the current leading architecture which was evaluated on this testset and gained $98.37\,\%$ [41]. Still, it should be noted that our solution is designed for a far more complex task. GSC contains a very limited amount of keywords, all very short and distinct from each other. Finally, there are no challenging negative test cases in GSC. On the other hand, our solution is designed for the open-vocabulary case, in which the model needs to deal with keywords that are very similar to each other. Hence the evaluation results on MOCKS are much more informative and the decrease in accuracy on GSC evaluation is the cost paid for much broader generalization.

VI. FUTURE WORK

In the future, we plan to work on methods that automatize the selection of the optimal $boost$ value. Furthermore, setting specific $boost$ values for different keywords might have positive results on evaluation results. Another intriguing research direction is using a combination of clean and noisy TTS recordings as ensembles in the multi-keyword classifier. This way it might be possible to reduce FPR without impacting TPR.

REFERENCES

[1] I. López-Espejo, Z.-H. Tan, J. H. L. Hansen, and J. Jensen, “Deep spoken keyword spotting: An overview,” *IEEE Access*, vol. 10, pp. 4169–4199, 2022. doi: 10.1109/ACCESS.2021.3139508

[2] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” 2018. [Online]. Available: https://arxiv.org/abs/1804.03209

[3] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, “Continuous hidden markov modeling for speaker-independent word spotting,” in *International Conference on Acoustics, Speech, and Signal Processing,*, 1989. doi: 10.1109/ICASSP.1989.266505 pp. 627–630 vol.1.

[4] J. Wilpon, L. Miller, and P. Modi, “Improvements and applications for key word recognition using hidden markov modeling techniques,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1991)*, 1991. doi: 10.1109/ICASSP.1991.150338 pp. 309–312 vol.1.

[5] I.-F. Chen and C.-H. Lee, “A hybrid HMM/DNN approach to keyword spotting of short words,” in *Proc. Interspeech 2013*, 2013. doi: 10.21437/Interspeech.2013-397 pp. 1574–1578.

[6] S. Panchapagesan, M. Sun, A. Khare, S. Matsoukas, A. Mandal, B. Hoffmeister, and S. Vitaladevuni, “Multi-Task Learning and Weighted Cross-Entropy for DNN-Based Keyword Spotting,” in *Proc. Interspeech 2016*, 2016. doi: 10.21437/Interspeech.2016-1485 pp. 760–764.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural ’networks,” in *Proc. 23rd International Conference on Machine Learning (ICML 2006)*, vol. 2006, 01 2006. doi: 10.1145/1143844.1143891 pp. 369–376.

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” *Advances in neural information processing systems*, vol. 28, 2015.

[9] K. Hwang, M. Lee, and W. Sung, “Online keyword spotting with a character-level recurrent neural network,” 2015.

[10] Y. Zhuang, X. Chang, Y. Qian, and K. Yu, “Unrestricted Vocabulary Keyword Spotting Using LSTM-CTC,” in *Proc. Interspeech 2016*, 2016. doi: 10.21437/Interspeech.2016-753 pp. 938–942.

[11] S. Sigtia, P. Clark, R. Haynes, H. Richards, and J. Bridle, “Multi-task learning for voice trigger detection,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*. IEEE, may 2020. doi: 10.1109/icassp40776.2020.9053577

[12] T. Bluche and T. Gisselbrecht, “Predicting Detection Filters for Small Footprint Open-Vocabulary Keyword Spotting,” in *Proc. Interspeech 2020*, 2020. doi: 10.21437/Interspeech.2020-1186 pp. 2552–2556.

[13] Y. He, R. Prabhavalkar, K. Rao, W. Li, A. Bakhtin, and I. McGraw, “Streaming small-footprint keyword spotting using sequence-to-sequence models,” in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2017)*, 2017. doi: 10.1109/ASRU.2017.8268974 pp. 474–481.

[14] A. Berg, M. O’Connor, and M. T. Cruz, “Keyword Transformer: A Self-Attention Model for Keyword Spotting,” in *Proc. Interspeech 2021*, 2021. doi: 10.21437/Interspeech.2021-1286 pp. 4249–4253.

[15] A. Awasthi, K. Kilgour, and H. Rom, “Teaching Keyword Spotters to Spot New Keywords with Limited Examples,” in *Proc. Interspeech 2021*, 2021. doi: 10.21437/Interspeech.2021-1395 pp. 4254–4258.

[16] M. Mazumder, C. Banbury, J. Meyer, P. Warden, and V. J. Reddi, “Few-Shot Keyword Spotting in Any Language,” in *Proc. Interspeech 2021*, 2021. doi: 10.21437/Interspeech.2021-1966 pp. 4214–4218.

[17] L. Lugosch, S. Myer, and V. S. Tomar, “Donut: Ctc-based query-by-example keyword spotting,” *arXiv preprint arXiv:1811.10736*, 2018.

[18] B. Kim, M. Lee, J. Lee, Y. Kim, and K. Hwang, “Query-by-example on-device keyword spotting,” in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2019)*, 12 2019. doi: 10.1109/ASRU46091.2019.9004014 pp. 532–538.

[19] J. Huang, W. Gharbieh, H. S. Shim, and E. Kim, “Query-by-example keyword spotting system using multi-head attention and soft-triple loss,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021. doi: 10.1109/ICASSP39728.2021.9414156 pp. 6858–6862.

[20] J. Huang, W. Gharbieh, Q. Wan, H. S. Shim, and H. C. Lee, “QbyE-MLPMixer: Query-by-Example Open-Vocabulary Keyword Spotting using MLPMixer,” in *Proc. Interspeech 2022*, 2022. doi: 10.21437/Interspeech.2022-11080 pp. 5200–5204.

[21] S. Settle, K. Levin, H. Kamper, and K. Livescu, “Query-by-Example Search with Discriminative Neural Acoustic Word Embeddings,” in *Proc. Interspeech 2017*, 2017. doi: 10.21437/Interspeech.2017-1592 pp. 2874–2878.

[22] G. Chen, C. Parada, and T. N. Sainath, “Query-by-example keyword spotting using long short-term memory networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015. doi: 10.1109/ICASSP.2015.7178970 pp. 5236–5240.

[23] C. Chiu and C. Raffel, “Monotonic chunkwise attention,” *CoRR*, vol. abs/1712.05382, 2017. [Online]. Available: http://arxiv.org/abs/1712.05382

[24] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *International Journal of Computer Vision*, vol. 129, no. 6, p. 1789–1819, jun 2021. doi: 10.1007/s11263-021-01453-z

[25] S. Kim, T. Hori, and S. Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, 03 2017. doi: 10.1109/ICASSP.2017.7953075 pp. 4835–4839.

[26] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” *CoRR*, vol. abs/1508.07909, 2015. [Online]. Available: http://arxiv.org/abs/1508.07909

[27] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, 2017. doi: 10.1109/ICASSP.2017.7952261 pp. 776–780.

[28] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.

[29] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *Proc. Interspeech 2019*, 2019. doi: 10.21437/Interspeech.2019-2680 pp. 2613–2617.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015. doi: 10.1109/ICASSP.2015.7178964 pp. 5206–5210.

[31] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” in *International Conference on Language Resources and Evaluation*, 2019.

[32] M. Pudo, M. Wosik, A. Cieślak, J. Krzywdziak, B. Łukasiak, and A. Janicki, “MOCKS 1.0: Multilingual open custom keyword spotting testset,” in *Proc. Interspeech 2023*, in press.

[33] “Keyword spotting on google speech commands,” https://paperswithcode.com/sota/keyword-spotting-on-google-speech-commands, 2023, [Online; accessed 19-May-2023].

[34] M. Morise, F. Yokomori, and K. Ozawa, “World: A vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE Transactions on Information and Systems*, vol. E99.D, pp. 1877–1884, 07 2016. doi: 10.1587/transinf.2015EDP7457

[35] J.-M. Valin and J. Skoglund, “A Real-Time Wideband Neural Vocoder at 1.6kb/s Using LPCNet,” in *Proc. Interspeech 2019*, 2019. doi: 10.21437/Interspeech.2019-1255 pp. 3406–3410.

[36] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, 2018. doi: 10.1109/ICASSP.2018.8461368 pp. 4779–4783.

[37] N. Ellinas, G. Vamvoukakis, K. Markopoulos, A. Chalamandaris, G. Maniati, P. Kakoulidis, S. Raptis, J. S. Sung, H. Park, and P. Tsiakoulis, “High Quality Streaming Speech Synthesis with Low, Sentence-Length-Independent Latency,” in *Proc. Interspeech 2020*, 2020. doi: 10.21437/Interspeech.2020-2464 pp. 2022–2026.

[38] P. Liu, X. Wu, S. Kang, G. Li, D. Su, and D. Yu, “Maximizing mutual information for tacotron,” *ArXiv*, vol. abs/1909.01145, 2019.

[39] K. Ito and L. Johnson, “The LJ speech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017.

[40] E. Bakhturina, V. Lavrukhin, B. Ginsburg, and Y. Zhang, “Hi-Fi Multi-Speaker English TTS Dataset,” in *Proc. Interspeech 2021*, 2021. doi: 10.21437/Interspeech.2021-1599 pp. 2776–2780.

[41] R. Vygon and N. Mikhaylovskiy, “Learning efficient representations for keyword spotting with triplet loss,” in *Speech and Computer*, A. Karpov and R. Potapova, Eds. Cham: Springer International Publishing, 2021. ISBN 978-3-030-87802-3 pp. 773–785.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 667–676

DOI: 10.15439/2023F6717
ISSN 2300-5963 ACSIS, Vol. 35

Algorithmic Handling of Time Expanded Networks

Alain Quilliot and Jose-Luis Figueroa
LIMOS Lab.
UCA, CNRS and EMSE
Clermont-Ferrand, France
Email: alain.quilliot@uca.fr

Hélène Toussaint
and Annegret Wagler
LIMOS Lab.
Clermont-Ferrand, France

***Abstract*—Time Expanded Networks, built by considering the nodes of a base network over some time space, are powerful tools for the formulation of problems involving synchronization mechanisms. Those mechanisms may for instance be related to the interaction between resource production and consumption or between routing and scheduling. Still, in most cases, deriving algorithms from those formulations is difficult, due to both the size of resulting network structure and the fact that reducing this size through rounding techniques tends to induce uncontrolled error propagation. We address here this algorithmic issue, while proposing a generic decomposition scheme which works by first skipping the temporal dimension of the problem and next expanding resulting projected solution into a full solution of the problem set on the time expanded network.**

I. Introduction

ONE derives a *Time Expanded Network* (TE-Network) $N^{\mathbf{TIME}}$ (see [11]) from a base network $N = (X, A)$ and a time space **TIME**, by considering all the copies (x, t) of the nodes x of N at the different instants t of **TIME**. Then, an arc of $N^{\mathbf{TIME}}$ is either an arc $\big((x, t), (y, t + \delta)\big)$ which corresponds to the time required to traverse an arc (x, y) of N while starting from x at time t, or an arc $\big((x, t), (x, t')\big)$ with $t < t'$, which expresses some kind of standby in x from time t to time t'. It may happen that the traversal time δ depends on t. Note that the time space **TIME** may be either discrete or continuous.

Time Expanded Networks are powerful modeling tools for problems involving synchronization, between for instance resource production and consumption or between routing and scheduling. They are also well-fitted to deal with the time-dependence of a network. They were introduced by Ford and Fulkerson (see [11]) in order to cast such problems into the network flow framework. Some time later, concerns raised by time dependence and synchronization issues motivated the notion of *dynamic network* (see [2], [16]). They next gave rise at the beginning of the 80's to *flow over time* models, where flow values are trajectories, that means functions from a time space **TIME** onto real or integer numbers. Those functions may be subject to constraints like continuity or differentiability, and so the *flow over time* framework is well-fitted to the management of problems involving gas or power production and distribution. Years 1990/2000 also registered applications of these notions to evacuation planning (see [9]). At this time, authors adapted standard algorithms (min-cost flow and max-flow algorithms) and brought insights about the link between TE-Networks and the *flow over time* models (see [13]). They addressed complexity issues and stated some *polynomial approximation scheme* (PTAS) results. In the years 2010, authors came back to the original TE-Network framework. They did it with the purpose of handling multi-commodity flow models (see [1]) like those which may derive from transportation (see [5]) and industrial scheduling problems (see [18], [3]). They tried to take advantage of the improvement of both computers and *Mixed Integer Linear Programming* (MILP) libraries in order to directly implement some transportation models on those libraries (see [6], [17], [21]). They coped with the size issue by trying to adapt standard column generation and branch and cut techniques (see [7]), but faced difficulties as soon as the size of the time space increased. All those contributions suggested that the *Time Expanded Network* might be very efficient to deal with scheduling/routing problems involving synchronization (see [15], [14]) requirement or time dependence features (see [20]) .

Actually, though the TE-Network framework is good for modeling, it often fails in providing efficient algorithms. The fact is not only that the size of the TE-Network $N^{\mathbf{TIME}}$ increases very fast with the size of the time space **TIME**, but also that controlling this size through rounding techniques induces strong error propagation. So our purpose here is to bypass those difficulties by applying a *Project/Expand* decomposition scheme. We first search (*Project* step) for a good projected solution on the base graph N, and next (*Expand* step) turn this projected solution into a full TE-Network solution. We started addressing the *Project* step in a former contribution (see [12]), while setting a *Projected* model provided with *Extended No-Subtour* constraints enhancing its ability to yield full feasible solutions. We are now going to first reinforce this *Projected* model with constraints which guaranty the feasibility of the *Expand* process, and next derive from resulting projected solution a formal bi-level setting of related *Expand* problem. We shall deal with this problem in both an exact way and a heuristic way, by introducing a flexible active sub-network of $N^{\mathbf{TIME}}$ and tuning this sub-network until getting a full TE-Network solution. Though our approach is generic, we shall refer here, for the sake of understanding and with the purpose of testing, to a 2-commodity flow model related to the management of an *item balancing* process (see [8] and [17]).

The paper is organized as follows. In Section 2, we first present our reference TE-Network model, related to some *item balancing* problem, and next the projected model which derives from this TE-Network model. In Section 3 we set the *Expand* problem, characterize its feasibility and reinforce the projected model with constraints which ensure the feasibility of resulting *Expand* problem. In Section 4 we propose a bi-level formulation of this *Expand* problem and in Section 5 we present an exact MILP model together with heuristic algorithms which deal with the *Expand* problem. We conclude by providing some numerical experiments and discussing some open questions.

II. A TE-Network Model for the Item Relocation Problem

We refer here, for the sake of understanding, to a specific *Item Balancing Problem* (IBP). So, we consider here a transit network $N = (X, A)$, together with a *Depot* node (see Figure 1). Every arc is provided with a time value $T_{x,y}$ and a cost value $C_{x,y}$. Also:

- We set $\mathbf{T} = (T_{(x,y)}, (x,y) \in A)$ and $\mathbf{C} = (C_{(x,y)}, (x,y) \in A)$. For any path π from $x \in X$ to $y \in X$, we denote by $L^T(\pi)$ its length in the sense of $\mathbf{T}$. We do the same with $\mathbf{C}$. For any pair of nodes (x,y) we denote by $D^T(x,y)$ the shortest path distance from x to y in the sense of $\mathbf{T}$, and by $D^C(x,y)$ the shortest path distance from x to y in the sense of $\mathbf{C}$.
- Let U be some subset of X. We set $\partial^-_N(U) = \{(x,y) \in A$ such that $x \notin U, y \in U\}$, $\partial^+_N(U) = \{(x,y) \in A$ such that $x \in U, y \notin U\}$, $\partial_N(U) = \partial^-_N(U) \cup \partial^+_N(U)$, and $A(U) = \{(x,y) \in A$ such that $x \in U, y \in U\}$. Clearly, $\partial_N(U)$ means the arcs which allow entering and getting out of U. We simplify these notations in case U is a singleton $\{x\}$ by writing $\partial^-_N(x)$, $\partial^+_N(x)$ and $\partial_N(x)$, instead of $\partial^-_N(\{x\})$, $\partial^+_N(\{x\})$ and $\partial_N(\{x\})$, respectively. Also, we denote by $U \setminus V$ the difference of the sets U and V (i.e., the set $\{u \in U$ such that $u \notin V\}$).

Then our IBP: *Item Balancing Problem* comes as follows: Items are located inside the network and must be relocated, within a *time horizon* $\{0, 1, \ldots, T_{max}\}$, by a fleet of identical carriers with *capacity* Cap. We are provided with an integral *balance* vector $\mathbf{b} = (b_x, x \in X)$ such that $\sum_{x \in X} b_x = 0$: $b_x > 0$ means that x is in *excess* and that b_x items must leave x; $b_x < 0$ means that x is in *deficit* and that b_x items must arrive to x. The *Item Balancing Problem* (IBP) consists in scheduling those transfers, while minimizing a hybrid cost $\alpha \cdot c_1 + \beta \cdot c_2 + \gamma \cdot c_3$, where c_1 is the number of active carriers, c_2 is their running cost in the sense of $\mathbf{C}$, c_3 is the time spent by items while moving inside the carriers, and α, β, γ are scaling coefficients. An important feature of the problem is that we allow *preemption*, which means that the carriers may exchange items, making synchronization become an issue.

Example 1 Consider the network $N = (X, A)$ of Figure 1. It shows two carrier routes $\Gamma^1 = (Depot, v, x, y, Depot)$ and $\Gamma^2 = (Depot, w, x, z, Depot)$ linked together by a transfer from Γ^1 to Γ^2 at node x. We check $c_1 = 2$, $c_2 = 10$, and $c_3 = 32$.

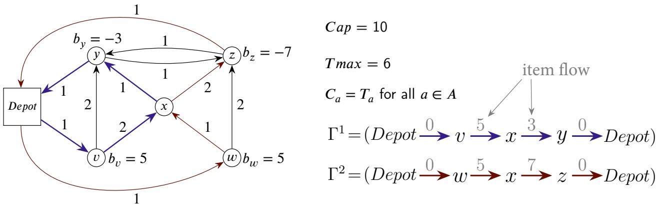

1
$b_y = -3$
1
y
z
$b_z = -7$
1
Depot
1
2
1
x
2
2
1
2
1
v
$b_v = 5$
w
$b_w = 5$
1
$Cap = 10$
$Tmax = 6$
$C_a = T_a$ for all $a \in A$
item flow
$\Gamma^1 = (Depot \xrightarrow{0} v \xrightarrow{5} x \xrightarrow{3} y \xrightarrow{0} Depot)$
$\Gamma^2 = (Depot \xrightarrow{0} w \xrightarrow{5} x \xrightarrow{7} z \xrightarrow{0} Depot)$

Fig. 1. The transit network $N = (X, A)$ used in Example 1

A. A Time-expanded 2-commodity Flow Model for the IBP Problem

In order to cast this IBP problem into the TE-Network framework (see [8]), we first derive from the network $N = (X, A)$ its *time expansion* $N^{Tmax} = (X^{Tmax}, A^{Tmax})$ according to $Tmax$. The node set X^{Tmax} is the set of all pairs (x,t), $x \in X$, $t \in \{0, 1, \ldots, Tmax\}$, augmented with two distinguished nodes: $source$ and $sink$. The arcs $a \in A^{Tmax}$, together with their *carrier cost* $\hat{C}_a$, and their *item cost* $\hat{I}_a$, come as follows:

- *input-arcs* $a = \big(source, (x, 0)\big)$, $x \in X$, with $\hat{I}_a = 0$ and $\hat{C}_a = 0$;
- *output-arcs* $a = \big((x, Tmax), sink\big)$, $x \in X$, with $\hat{I}_a = \hat{C}_a = 0$;
- *waiting-arcs* $a = \big((x,t), (x, t+1)\big)$, $x \in X$, $t \in \{0, \ldots, Tmax - 1\}$, with $\hat{I}_a = \hat{C}_a = 0$;
- *active-arcs* $a = \big((x,t), (y, t + T_{(x,y)})\big)$, $(x,y) \in A$, $t \in \{0, \ldots, Tmax - T_{(x,y)}\}$, with $\hat{I}_a = \gamma \cdot T_{(x,y)}$ and $\hat{C}_a = \beta \cdot C_{(x,y)}$;
- *backward-arc* $a = (sink, source)$, with $\hat{I}_a = 0$ and $\hat{C}_a = \alpha$.

In order to formalize our IBP problem as a 2-commodity flow model on this network $N^{Tmax} = (X^{Tmax}, A^{Tmax})$, we introduce integral 2 flow vectors $\mathbf{H}$ and $\mathbf{h}$, both indexed on the arc set of N^{Tmax}. The first one is going to describe the way the carriers move inside the transit network: for any active-arc $a = \big((x,t), (y, t + T_{(x,y)})\big)$, H_a will mean the number of carriers which traverse the arcs (x,y) of the transit network between time t and time $t + T_{(x,y)}$. The second one will describe the moves of the items: for any active-arc $a = \big((x,t), (y, t + T_{(x,y)})\big)$, H_a will mean the number of items which traverse the arcs (x,y) of the transit network between time t and time $t + T_{(x,y)}$. The value H_a on the *backward-arc* will provide us with the number of carriers involved into the process. The values H_a and h_a on a *waiting-arc* $a = \big((x,t), (x, t+1)\big)$ are going to respectively provide us with the number of carriers and items waiting on node x between time t and time $t+1$. By proceeding this way, we get:

TE-Network IBP Model. *Compute two nonnegative integral A^{Tmax}-indexed vectors $\mathbf{H}$ and $\mathbf{h}$ (for carriers and items, respectively) such that:*

- **H** *and* **h** *satisfy flow conservation at any node of* N^{Tmax}; (E1)
- *for any active-arc* $a = \big((x,t),(y,t+T_{(x,y)})\big)$: $h_a \leq Cap \cdot H_a$; (E2)
- *for any input-arc* $a = \big(source,(x,0)\big)$, $x \neq Depot$: $H_a = 0$; $h_a = \max(b_x, 0)$; (E3)
- *for any output-arc* $a = \big((y, Tmax), sink\big)$, $y \neq Depot$: $H_a = 0$; $h_a = \max(-b_y, 0)$; (E4)
- *the global cost* $Cost(H,h) = \sum_{a \in A^{Tmax}} \big(H_a \cdot \hat{C}_a + h_a \cdot \hat{I}_a\big)$ *is minimized.*

Constraints (E1) express the circulation of carriers and items. Constraints (E2) mean that any item move is supported by some carrier. Constraints (E3) and (E4) characterize initial and final states: carriers start and end at $Depot$, while any node ends as *neutral*.

Example 2: Fig. 2 shows the TEN $N^{Tmax} = (X^{Tmax}, A^{Tmax})$ deriving from network $N = (X, A)$ of Fig. 1 and $T_{max} = 6$. It turns the solution of Example 1 into a full solution (**H**, **h**). Tour Γ^1 gives rise to a path $\{source, (Depot, 0), (v, 1), (x, 3), (y, 4), (Depot, 5), (Depot, 6), sink\}$. Tour Γ^2 gives rise to a path $\{source, (Depot, 0), (w, 1), (x, 2), (x, 3), (z, 5), (Depot, 6), sink\}$. The value of **h** on the arc $(x, 2), (x, 3)$ shows the way the 5 items transported by carrier 2 wait on node x before splitting themselves along the arc (x, y) and (x, z).

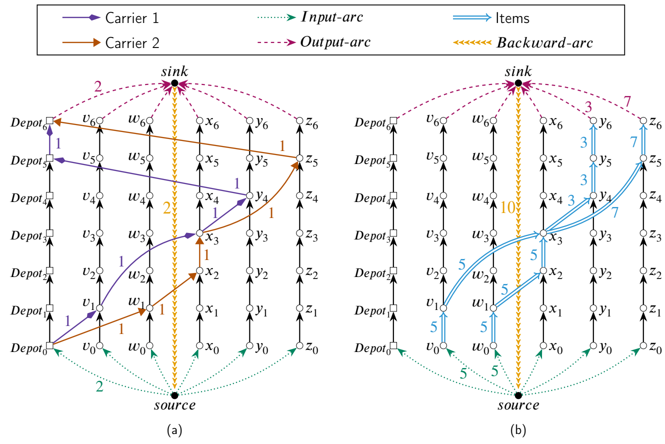


Fig. 2. The routes and schedules as a TE-Network 2-commodity flow: (a) Carrier flow vector **H**. (b) Item flow vector **h**.

As it is formulated, we understand that this **TE-Network IBP Model** is going to be difficult to handle. Its size grows fast with value T_{max}. We might try to control this size by rounding the values t. For instance, instead of measuring the time in minutes, we could try to round it to 15 minutes packages. But if traversing an arc requires 3 minutes, then rounding will mean either canceling this traversal time, or multiplying it by 5. In both case, we guess that the propagation of resulting errors will be difficult to manage. So we are going to implement a *Project and Expand* scheme, which means that we are going to first skip the temporal dimension of our problem (*Project* step) and restrict ourselves to the identification of the arcs followed by the carriers and the items, and next perform an *Expand* step in order to schedule those arcs and get a full solution of our problem.

B. The Projected IBP Model

Given a feasible solution (**H**, **h**) of above **TE-Network IBP** Model. We define the *projection* **F** of **H** on the network N by setting, for any arc (x, y) of N:

$$F_{(x,y)} = \sum_{t=0}^{Tmax} H_{(x,t),(y,t+T_{(x,y)})}.$$

We define the same way the *projection* **f** of **h**. Clearly, the meaning of those projected vectors is that we want to simplify our problem while skipping its temporal dimension. They are going to provide us with a kind of signature of the routes followed respectively by the carriers and the items on the transit network, without taking care neither of the order according to which they run along the arcs of this transit network nor of the timestamps telling when they do it. Of course, we expect that computing those projected vectors **F** and **f** will be easier than computing **F** and **f**. In order to perform this computation, we must characterize those vectors.

We see that **F** and **f** must be such that:

- **F** satisfies flow conservation at any vertex of X; (E5.1)
- for any node x of N: $\sum_{a \in \partial^+_N(x)} f_a - \sum_{a \in \partial^-_N(x)} f_a = b_x$; (E5.2)
- for any arc a of N: $f_a \leq Cap \cdot F_a$; (E6)

According to this, carrier riding cost c_2 and item riding time c_3 come as follows:

- carrier riding cost $c_2 = \beta \cdot \big(\sum_{a \in A} C_a \cdot F_a\big)$; (E7.1)
- items riding time $c_3 = \gamma \cdot \big(\sum_{a \in A} T_a \cdot f_a\big)$. (E7.2)

Still, this formulation is not enough in order to efficiently characterize **F** and **f**. First, (E5.1)-(E7.2) fail in estimating the carrier number $c_1 = H_{(sink, source)}$. In order to make our projected model provide a good estimation of the carrier number, we proceed as follows:

Approximating the carrier number: We first notice as in [12] that the quantity $\sum_{a \in A} T_a \cdot F_a$ means the global time that carriers spend running inside N, waiting times being excluded. Since the whole process must last no more than $Tmax$ time units, it requires at least $\left\lceil \frac{\left(\sum_{a \in A} T_a \cdot F_a\right)}{Tmax} \right\rceil$ carriers. Thus, (**F**, **f**) should minimize the *projected cost*:

$$PCost(\mathbf{F}, \mathbf{f}) = \alpha \cdot \frac{\left(\sum_{a \in A} T_a \cdot F_a\right)}{Tmax} + \beta \cdot \big(\sum_{a \in A} C_a \cdot F_a\big) + \gamma \cdot \big(\sum_{a \in A} T_a \cdot f_a\big).$$

Next, we notice that constraints (E5.1) and (E5.2) do not forbid subtours. In order to forbid subtours, we proceed in an augmented way:

The *Extended No-Subtour* Constraint: Given a subset $U \subset X \setminus \{Depot\}$. The time that carriers spend moving at the border or inside U, is equal to $\sum_{a \in \partial_N(U) \cup A(U)} T_a \cdot F_a$. For each carrier q, this time cannot exceed $Tmax$. If Q denotes the number of carriers involved into an IBP solution, then we see that $Q \cdot Tmax \geq \sum_{a \in \partial_N(U) \cup A(U)} T_a \cdot F_a$. Since $\sum_{a \in \partial_N^-(U)} F_a \geq Q$, we deduce that the following *Extended No-Subtour* inequality should hold:

$$Tmax \cdot \left(\sum_{a \in \partial_N^-(U)} F_a \right) \geq \sum_{a \in \partial_N(U) \cup A(U)} T_a \cdot F_a. \quad \text{(E8)}$$

This leads us to set the following *projected* problem about the search for $\mathbf{F}$ and $\mathbf{f}$:

PIBP: *Projected Item Balancing* Problem
$\{$*Compute on the network $N = (X, A)$ two nonnegative integral vectors A-indexed* $\mathbf{F}$ *and* $\mathbf{f}$ *such that:*

- $\mathbf{F}$ *satisfies flow conservation at any node of X;* (E5.1)
- *for any node $x \in X$, $\sum_{a \in \partial_N^-(x)} f_a - \sum_{a \in \partial_N^+(x)} f_a = b_x$;* (E5.2)
- *for any arc $a \in A$, $f_a \leq Cap \cdot F_a$;* (E6)
- *for any $U \subseteq X \setminus \{Depot\}$: $Tmax \cdot \left(\sum_{a \in \partial_N^-(U)} F_a \right)$ $\geq \sum_{a \in \partial_N(U) \cup A(U)} T_a \cdot F_a$;* (E8)
- *Minimize $PCost(\mathbf{F}, \mathbf{f}) = \alpha \cdot \frac{\sum_{a \in A} T_a \cdot F_a}{Tmax} + \beta \cdot \left(\sum_{a \in A} C_a \cdot F_a \right) + \gamma \cdot \left(\sum_{a \in A} T_a \cdot f_a \right)$.* (E9)$\}$

We proved in [12], that the *Extended No-Subtour* constraints may be separated in polynomial time. A consequence is that this projected problem may be efficiently solved while using a MILP library and implementing a Branch and Cut process.

III. The Expand Issue

So we come now to the *Expand* step. That means that we suppose that we have been computing ($\mathbf{F}$, $\mathbf{f}$) as above and that we want to derive ($\mathbf{H}$, $\mathbf{h}$) in a satisfactory way. First, we notice that, in many cases, there will not exist ($\mathbf{H}$, $\mathbf{h}$) whose projection is equal to ($\mathbf{F}$, $\mathbf{f}$). Figure 3 shows that a PIBP solution ($\mathbf{F}$, $\mathbf{f}$) may not be the projection of any feasible IBP TE-Network solution ($\mathbf{H}$, $\mathbf{h}$): The carrier must follow the route $(Depot, y, x, z, y, Depot)$ and will never be able to transport that way any item from z to x.

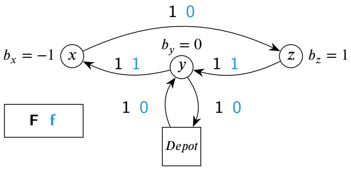


Fig. 3. A solution of PIBP which cannot be expanded.

So we must put some flexibility while setting our *Expand* issue. Namely, we should set it as follows: How can we derive from ($\mathbf{F}$, $\mathbf{f}$) a good full IBP solution ($\mathbf{H}$, $\mathbf{h}$)? In a more generic way, how can we efficiently deal with a TE-Network model while applying the following resolution scheme?

***Project/Expand* Decomposition Scheme.**

1) Solve a projected version of the problem which skips the temporal dimension.
2) Turn (*expand*) resulting solution ($\mathbf{F}$, $\mathbf{f}$) into a *"good"* solution ($\mathbf{H}$, $\mathbf{h}$) of the original problem, while restricting ourselves to a reduced representation of related TE-Network.

One feels that there are several ways to interpret above *expand* part of the process. Intuition would suggest to search for ($\mathbf{H}$, $\mathbf{h}$) such that ($\mathbf{F}$, $\mathbf{f}$) is the projection of ($\mathbf{H}$, $\mathbf{h}$). Such a setting is NP-Hard (see [8]). But the true problem is that both above example and numerical experiments show that such a setting is too strong and most often does not admit any feasible solution. So what we decide in the present case is to formalize step 2 while only requiring $\mathbf{f}$ to be the projection of $\mathbf{h}$. So we set the **Expand** problem as follows:

***Expand* Problem EXPAND(f).** $\{$*Compute a feasible IBP solution* ($\mathbf{H}$, $\mathbf{h}$) *such that:*

- *The projection of* $\mathbf{h}$ *on the transit network N is equal to* $\mathbf{f}$;
- *The cost value $Cost(\mathbf{H}, \mathbf{h})$ is the smallest possible.*$\}$

This **Expand** problem is difficult. One may check that EXPAND(f) contains both the **TSP**: *Traveling Salesman Problem* and the standard **Pick up and Delivery** problem. It is far more general, since we may by no way identify the arcs which support f with requests and must take care of the precedence relations implicitly related to those arcs. Above example of Figure 3 shows that part of our problem consists in determining in which order the arcs supporting $\mathbf{f}$ must be visited.

Before going further towards the design of algorithmic solution for this *Expand* problem, we must address the issue related to its feasibility: Can we characterize the conditions which will make this **Expand** problem admit a feasible solution? The answer is positive, and checking it is going to provide us with a reinforcement of the *Projected* model of Section II, that means with a way to compute ($\mathbf{F}$, $\mathbf{f}$) which will guaranty this feasibility.

Checking the Feasibility of EXPAND(f): Enhancing the PIBP model.
We just told that a too strong setting of the *Expand* issue could lead to unfeasible situations. So a natural question comes about the feasibility of above EXPAND(f problem. In order to deal with it, let us introduce the following notion of *feasible path*. A *feasible path* is a path that an item may follow while moving from an *excess* node to a *deficit* one, taking into account that the carrier which transports it must be able to move from the *Depot* node to the start-node of this path and next from the end-node of this path until the *Depot* node, within the time horizon $\{0, 1, \ldots, T_{max}\}$. Clearly, flow vector $\mathbf{f}$ should be such that all items may follow feasible paths. We formalize this by saying that:

- A *feasible path* of N is any path π from $x \in X^+$ to $y \in X^-$ whose length $L^T(\pi)$ in the sense of the time is such that: $D^T(Depot, x) + L^T(\pi) + D^T(y, Depot) \leq Tmax$. We denote by $\mathbf{f}^\pi$ related $\{0, 1\}$ flow vector and by Π^{FP} the set of feasible paths.
- Vector $\mathbf{f}$ is *feasible-path-decomposable* iff it can be written: $\mathbf{f} = \sum_{\pi \in \Pi^{FP}} \lambda_\pi \; \mathbf{f}^\pi$, with $\lambda_\pi \geq 0$.

Since any item in node $x \in X^+$ must be transported to some vertex $y \in X^-$ along a feasible path, we get that $\mathbf{f}$ must feasible-path-decomposable. But checking that $\mathbf{f}$ is feasible-path-decomposable is just a matter of solving a rational linear program, and characterizing the feasibility of this linear program may be done by using Duality Theory. More precisely, we define a *Path Feasibility* vector as any Π^{FP}-indexed vector $\mathbf{w} = (w_\pi, \pi \in \Pi^{FP})$ such that:

For any feasible path π, we have $\sum_{a \in \pi} w_a \geq 0$.

This allows to state:

Theorem 1: EXPAND(**f**) *is feasible iff, for any Path Feasibility vector* $\mathbf{w}$*, we have:* $\sum_{a \in A} f_a \cdot w_a \geq 0$.

Sketch of the Proof: Linear Programming Duality makes that (E10) holds iff **f** is feasible-path-decomposable. This property is clearly necessary in order to allow the existence of **H** and **h**. We get sufficiency by considering such a decomposition of **f** and assigning to any item its own feasible path and a carrier which transports it along this feasible path. We explicitly build this way flow vectors **H** and **h**. **EndProof**

So we reinforce our **PIBP** model by imposing vector **f** to be feasible-path-decomposable:

For any *Path Feasibility* vector **w**: $\sum_{a \in A} f_a \cdot w_a \geq 0$. (E10)

Theorem 2: *(E10) can be separated in polynomial time.*

Sketch of the Proof: Every time we are provided with a flow vector **f** (rational or integral) we search for a *feasible-path* decomposition of **f**. This means solving some linear program with respect to some current collection Λ of feasible paths. In case of failure, then we apply duality and generate another feasible path, until we succeed or we get a *Path Feasibility* vector which contradicts (E10). The time-polynomiality of this separation process derives from the time-polynomiality of Rational Linear Programming. **EndProof**

It comes that the **PIBP** model reinforced by (E10) may be efficiently handled through Branch-and-Cut.

IV. A Bi-Level Decomposition Scheme for the Expand Problem

The purpose of this section is to show the way the **Expand** problem man be decomposed in a way which will allow us in next section V to implement both exact and heuristic algorithms. The main idea behind this decomposition scheme is that the quality of final solution (**H**, **h**) is mostly determined by the way items are routed. Intuitively, that means that we should first expand flow vector **f**, and next try to *cover* it by a carrier flow **H** according to the following 2-step approach:

1st step: *Expand* item flow **f** into an item flow **h** on the TE-Network N^{Tmax}, while relying on a specific $Split(N,$ **f**) network. This network is going to split those arcs and nodes of network N which are supporting **f** (in practice, it will mean few arcs and nodes) according to item packages likely to be carried by the same carriers. This will allow us to make appear the feasible paths followed by the items when moving from an excess node to a deficit nodes. Providing *ad hoc* time values to the nodes exploded this way will yield item flow vector **h**.

2nd step: Once **h** has been fixed, extend **h** into a good (best) solution IBP (**H**, **h**), while solving a Min-Cost Flow problem on the *active* part of a specific $Carrier(N,$ **f**) network. Network $Carrier(N,$ **f**) is going to extend above mentioned network $Split(N,$ **f**) in order to make appear the possible moves performed by the carriers. Related *active* part will be made of the arcs which are consistent with the time values related to vector **h**.

Linking 1st step and 2nd step: The quality of the resulting solution deeply depends not only on the route followed by the items, but also on the time values of the vertices related to **h** in N^{Tmax}. We shall delay, as long as possible, the instantiation of those time values while relying on a flexibility device which will take the form of a collection Λ of arcs common to both $Split(N,$ **f**) and $Carrier(N,$ **f**). This device will identify the moves that carriers and items are allowed to perform when switching from an arc of N supporting items to another one. This arc collection Λ will become the master object of a bi-level *Split/Carrier* decomposition scheme.

A. The Networks Split(N, f) and $Carrier(N, f)$

The Network $Split(N,$ **f**).

The purpose of the network $Split(N,$ **f**) is to help us in describing the way items traverse any vertex x of the network N, and so the trajectories followed by the items when moving from the excess nodes to the deficit ones. So we build it while relying on the arcs of network N which support non null **f** and exploding the nodes involved into those arcs in order to make appear the routes followed by the items.

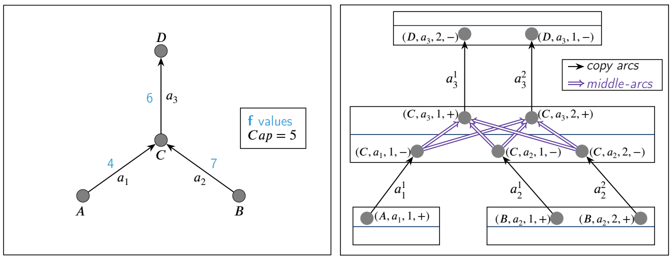
Nodes of $Split(N,$ **f**): With any arc $a = (x, y) \in A$ such that $f_a \geq 1$, we associate $\lceil \frac{f_a}{Cap} \rceil$ *copy-arcs* $a^m, m = 1, \ldots, \lceil \frac{f_a}{Cap} \rceil$, with respective origin $p = (x, a, m, +)$ and respective destination $q = (y, a, m, -)$. We denote by $Copy(a)$ the set of those arcs $a^m, m = 1, \ldots, \lceil \frac{f_a}{Cap} \rceil$. At the same time we create those copy-arcs, we also create *copy-nodes* $p = (x, a, m, +)$ and $q = (y, a, m, -)$, which respectively correspond to the carriers who leave x with a non-null load and to the carriers who arrive into y with a non-null load. Resulting node set X^* becomes the node set of $Split(N,$ **f**). We denote by $Copy(A)$ the set of all those copy-arcs. For any node $p = (y, a, m, \varepsilon)$ of X^*, we set $x(p) = y$ and $\varepsilon(p) = \varepsilon$. Also, for any node y of N, we set:

- $X^*(y) = \{p \in X^* \text{ such that } x(p) = y\}$;
- $X^*Plus(y) = \{p \in X^* \text{ such that } x(p) = y$ and $\varepsilon(p) = +\}$;
- $X^*Minus(y) = \{p \in X^* \text{ such that } x(p) = y$ and $\varepsilon(p) = -\}$.

Arcs of $Split(N,$ **f**): We complete the arc collection $\{a^m$, $a = (x, y)$, such that $f_a \geq 1$, $m = 1, \ldots, \lceil \frac{f_a}{Cap} \rceil\}$ by *middle-arcs* which, for any node x of N, connect copy-nodes $(x, a, m, -)$ (with a arriving into x), $m = 1, \ldots, \lceil \frac{f_a}{Cap} \rceil$ to copy-nodes $(x, a', m', +)$ (with a' starting from x) $m' = 1, \ldots, \lceil \frac{f_a}{Cap} \rceil$. We denote by $Middle$ the set of all middle-arcs created that way, and, for any node x of N, we denote by $Middle(x)$ the set of all middle-arcs u whose origin may be written $(x, a, m, -)$. $Middle(x)$ defines a complete bipartite graph on the nodes of $X^*(x)$. For any vertex $p = (x, a, m, +)$, we denote by $MiddleIn(p)$ the set of middle-arcs u with destination p, and for any vertex $q = (x, a, m, -)$, we denote by $MiddleOut(q)$ the set of middle-arcs u with origin q. We also set:

- $CopyIn(y) =$ $\{a \in Copy(A)$ with destination in $X^*Minus(y)\}$;
- $CopyOut(y) =$ $\{a \in Copy(A)$ with origin in $X^*Plus(y)\}$.

We denote by $Split(N, \mathbf{f})$ the resulting network, that one may check to be acyclic. This construction is illustrated in Figure 4.



(a)

(b)

Fig. 4. Building the $Split(N, \mathbf{f})$ Network

The network $Carrier(N, \mathbf{f})$.
This network is going to help us in computing the carrier routes, in such a way that those carrier routes *cover* the item routes. So, it will contain exactly the same nodes as the network $Split(N, \mathbf{f})$, but two additional nodes *source* and *sink*. Its arcs will express all the ways a carrier may move from an arc supporting items to another one while following a shortest path in N i the sense of vector $\mathbf{C}$. Depending on the time values assigned to its nodes, those arcs will be allowed or not to support non null carriers flow values. It comes that this network will behave as a flexible reduced version of N^{Tmax}.

Nodes of $Carrier(N, \mathbf{f})$: They are all nodes of $Split(N, \mathbf{f})$, augmented with two nodes $source$ and $sink$. We denote by $V(Carrier)$ the node set of $Carrier(N, \mathbf{f})$.

Arcs of $Carrier(N, \mathbf{f})$: They are the arcs of $Split(N, \mathbf{f})$ augmented with:

- one *back* arc $u = (sink, source)$, provided with cost $Q_u = \alpha$;
- any *in* arc $u = \big(source, p = (x, a, m, +)\big)$, $p \in X^*$, provided with a cost Q_u equal to the cost of a time-minimal path (in the sense of cost matrix $\mathbf{C}$) from $Depot$ to x;
- any *out* arc $u = (p = (x, a, m, -), sink)$, $p \in X^*$, provided with a cost Q_u equal to the cost of a time-minimal path from x to $Depot$;
- any *transverse* arc $u = \big(p = (x, a, m, -),\ q = (y, a', m', +)\big)$, with $p, q \in X^*$, provided with a cost Q_u equal to the cost of a time-minimal path from x to y.

We denote by $A(Carrier)$ the arc set of the network $Carrier(N, \mathbf{f})$.

B. The Split/Carrier Decomposition Scheme

The way item flow values f_a, $a \in \partial_N^-(x)$ arriving into a node x distribute themselves into values $f_{a'}$, $a' \in \partial_N^+(x)$ leaving x while moving through x, is going to be described by two integral vectors $\mathbf{z} = \big(z_u,\ u = \big((x, a, m, -), (x, a', m', +)\big) \in Middle\big)$, and $\mathbf{z}^* = \big(z^*_{a^m},\ a^m \in Copy(A)\big)$. Those two vectors will provide us with an *expanded* vector $\mathbf{h}$ once we assign time values t_p to the nodes p of the $Split(N, \mathbf{f})$. On the other side, the routes followed by the carriers are going to be described by a $\{0, 1\}$-valued flow vector $\mathbf{Z}$ defined on the arcs of the network $Carrier(N, \mathbf{f})$. In order to link those vectors $\mathbf{z}$, $\mathbf{z}^*$, $\mathbf{t}$, and $\mathbf{Z}$ together we need a *mediator* object. This *mediator* object is going to be here a collection Λ of arcs in $A(Carrier)$, providing us with the *transverse* arcs and the *middle* arcs supporting the item and carrier moves. It will induce the following constraints:

- on the time vector $\mathbf{t} = (t_p, p \in V(Carrier))$: for every arc (p, q) in Λ, $t_q \geq t_p + D^T(x(p), x(q))$, which means that any carrier or item move along an arc of the network $Carrier(N, \mathbf{f})$ requires a time at least equal to the length of the path in N which is related to such an arc;
- on the $\mathbf{z}$ vector: for every *middle-arc* $u = (p, q)$ which is not in Λ, $z_u = 0$;
- on the flow vector $\mathbf{Z}$ representing the carrier routes on the network $Carrier(N, \mathbf{f})$: for every *transverse* arc $u = (p, q)$ which is not in Λ, $Z_u = 0$.

More precisely, once Λ has been determined, we get that vector $\mathbf{t}$ must satisfy the following linear constraint system CTIME(Λ), with underlying totally unimodular constraint matrix.

CTIME(Λ) constraint system on $\mathbf{t} = (t_p,\ p \in X^* \cup \{source, sink\})$:

- $t_{source} = 0$; (E11)
- $t_{sink} \leq Tmax$; (E12)
- *For any in arc* $u = \big(source, p = (x, a, m, +)\big)$, $p \in X^*$: $t_p \geq D^T(Depot, x)$; (E13)
- *For any out arc* $u = \big(p = (x, a, m, +), sink\big)$, $p \in X^*$: $Tmax \geq D^T(x, Depot)$; (E13-1)
- *For any copy arc* $u = \big(p = (x, a, m, +), q = (y, a, m, -)\big)$, $a = (x, y) \in A$ *such that* $f_a \neq 0$: $t_q \geq t_p + D^T(x, y)$; (E13-2)
- *For any arc* (p, q) *in* Λ: $t_q \geq t_p + D^T(x(p), x(q))$. (E13-3)

As for vectors ($\mathbf{z}$, $\mathbf{z}^*$), they must express the way items move from any arc a support of $\mathbf{f}$ in N and with destination x to another one with origin x. More precisely:

- $\mathbf{z}$ is going to express, for any node x of network N, the way items arriving along a copy-arc a^m into a copy-node $q = (y, a, m, -)$ are going to distribute themselves among the copy-arcs $a'^{m'}$ leaving x. Clearly, the feasibility of this distribution process will impose Λ to contain enough arcs of $Middle(x)$.

- $\mathbf{z}^*$ is going to express, for any arc a of network N, the way item flow values f_a distribute themselves among the copy-arcs related to a.

This yields the following linear constraint system Split(N, **f**, Λ), with underlying totally unimodular matrix and whose feasibility depends on Λ:

Split(N, f, Λ) constraint system on vectors z, z*:

- *For any copy-arc* a^m: $z^*_{a^m} \leq Cap$. (E14)
- *For any copy-node* $q = (y, a, m, -)$: $z^*_{a^m} \leq \sum_{u \in MiddleOut(q)} z_u$. (E15)
- *For any copy-node* $p = (x, a, m, +)$: $z^*_{a^m} \geq \sum_{u \in MiddleIn(p)} z_u$. (E16)
- *For any node x of N*: $\sum_{v \in CopyIn(x)} z^*_u = \sum_{u \in Middle(x)} z_u + \max(-b_x, 0)$. (E17)
- *For any node x of N*: $\sum_{v \in CopyOut(x)} z^*_u = \sum_{u \in Middle(x)} z_u + \max(b_x, 0)$. (E18)
- *For any middle-arc* $u \notin \Lambda$: $z_u = 0$. (E19)

Finally, the flow vector **Z** with indexation on the arcs of $Carrier(N, \mathbf{f})$ and which is going to provide us with the arcs and paths followed by the carriers, should be a solution of the following Min-Cost Flow model Carrier(N, **f**, Λ).

Carrier(N, f, Λ) constraint system on vector **Z**:

- **Z** *satisfies flow conservation.* (E20)
- *For any arc u in Copy(A)*: $Z_u = 1$. (E21)
- *For any transverse or middle arc* $u \notin \Lambda$: $Z_u = 0$. (E22)
- *Cost value* $\sum_{u \in A(Covering)} Q_u Z_u$ *is minimal.* (E23)

We may now reformulate the **Expand** Problem as the following bilevel ([19]) setting.

***Split/Carrier* Reformulation of the Expand Problem.** *Compute $\Lambda \subseteq A(Carrier)$ restricted to middle and transverse arcs, such that:*

- CTIME(Λ) *admits a feasible solution;*
- Split(N, **f**, Λ) *admits a feasible solution* (**z**, **z***)*;*
- *The optimal value* $\sum_{u \in A(Carrier)} Q_u Z_u$ *of* Carrier(N, **f**, Λ) *is minimal.*

It may happens that above *Split/Carrier* model does not admit any feasible solution, while the **Expand** Problem is feasible. In such a case, we say that **f** is *Split/Carrier* {inconsistent. Figure 5 displays an example of such a situation:path A, B, C, D is not a feasible path and so **f** must be decomposed into 2 feasible paths, while the Split(N, **f**, Λ) model would allow only 1 feasible path. However, numerical experiments will show that it happens very scarcely.

Still, we may state:

Theorem 4.1: *Any feasible solution of above Split/Carrier model is a feasible solution of EXPAND(f), with same cost.*

Sketch of the Proof: It comes through an algorithmic construction of (**H**, **h**) from arc collection Λ. Time vector **t** obtained through resolution of CTIME(Λ) allows us to embed the nodes of the network $Carrier(N, \mathbf{f})$ into the time expanded network $N^{Tmax} = (X^{Tmax}, A^{Tmax})$. Then we derive **h** from a solution (**z**, **z***) of Split(N, **f**, Λ) and we derive **H** from a solution **Z** of Carrier(N, **f**, Λ). **EndProof**

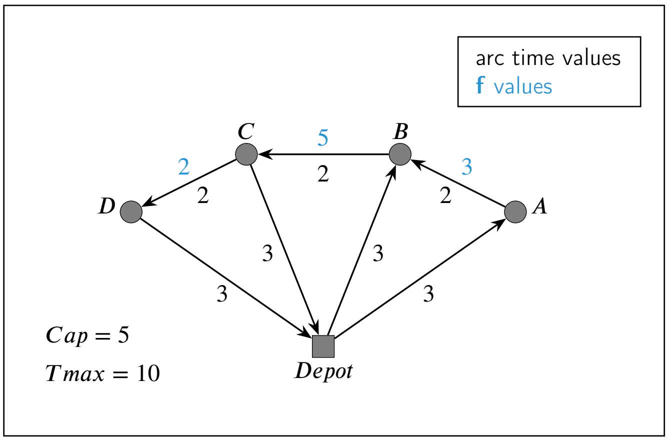


Fig. 5. *Split/Carrier* Inconsistent Flow Vector **f**)

V. Algorithmic Handling the *Split/Carrier* Decomposition Scheme

The *Split/Carrier* decomposition scheme involves, as its master object, a collection Λ of arcs of the $Carrier(N, \mathbf{f})$ network. Constraints of Split(N, **f**, Λ) are transportation constraints set on a bipartite graph. Carrier(N, **f**, Λ) is a Min-Cost Flow model while CTIME(Λ) is about the computation of the largest path in an acyclic graph ([4]). It comes that those 3 sub-problems may be viewed as *easy* and that we may rely either on Carrier(N, **f**, Λ) duality or on the largest paths which arise from the resolution of CTIME(Λ) in order to drive the master object Λ. We are going to describe here the ways we implemented this idea.

A. An Exact MILP Resolution

We turn previous *Split/Carrier* decomposition scheme into an Expand(N, **f**) MILP model. We do it by considering vectors **Z**, **z**, **z*** and **t** as in above Section IV, introducing an additional vector $\mathbf{X}^\Lambda$ with indexation on the *middle* and *transverse* arcs of the $Carrier(N, \mathbf{f})$ network, and merging the programs CTIME(Λ), Split(N, **f**, Λ and Carrier(N, **f**, Λ) into a unique one, according to the following modifications:

- (E13-3) is replaced by: For any *transverse* or *middle* arc $u = (p, q)$: $t_q \geq t_p + D^T(x(p), x(q))$.
- (E19) is replaced by: For any*middle* arc u: $z_u \leq X^\Lambda_u$.
- (E20) is replaced by: For any *transverse* or *middle* arc u: $Z_u \leq X^\Lambda_u$.

B. A Greedy Dual_Carrier Algorithm

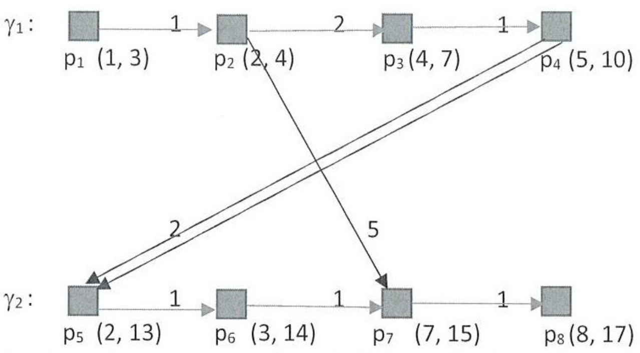
This algorithm works in a greedy way, while making increase the arc collection Λ. We first initialize Λ in such a way that the constraint system Split(N, **f**, Λ) admits a solution. We do it by removing constraint (E19), and defining Λ as the set of *middle* arcs which support non null **z** values. Then, at every iteration of the main loop of the greedy process, we are provided with some current arc collection Λ, and we use a feasible solution **t** of CTIME(Λ) in order to set a version of the min-cost flow model which replace (E22) by:

For any (*transverse* or *middle*) arc (p, q) such that:
$t_q < t_p + D^T(x(p), x(q))$, we have $Z_u = 0$. (E32-1)
This may be summarized as follows:

- 1st step: Solve the Split(N, **f**, Λ) while considering only constraints (E14)-(E18). Derive vectors $\mathbf{z}^*$ and **z** and ℓ, and initialize Λ with the arcs of the network $Carrier(N, \mathbf{f})$ which support $\mathbf{z}^*$ and **z**.
- 2nd step: While Not $Stop$ do:
  1) Get a feasible solution **t** of CTIME(Λ;
  2) Solve the Carrier(N, **f**, Λ) Min-Cost Flow model modified by replacing (E22) by (E22-1);
  3) Search for an arc (p, q) whose insertion into Λ makes related dual solution become infeasible and maintains the feasibility of CTIME(Λ). If $Success(Search)$ then insert this arc into Λ else $Stop$.

C. A Greedy Path_Concatenate Algorithm

Once again, we make the collection Λ increase, while using the CTIME(Λ) constraint system in order to deduce, at any iteration of the main loop, which arc (p, q), with $p = (x_1, a_1, m_1, -)$ and $q = (x_2, a_2, m_2, +)$ has to be inserted in Λ.



Arc (p2, p7) means a *transfer* path, followed by items from p2 to p7, with **T**-Length 5.
Arc (p4, p8) is candidate for insertion into Λ, with **T**-Length 2
Time windows after insertion: p5 (12, 13); p6 (13, 14);
p7 (14, 15); p8 (15, 17)
Notice that arc (p8, p4) is not eligible for insertion into L.

Fig. 6. *Concatenating 2 carrier paths γ_1 and γ_2)*

More specifically, we first solve Split(N,**f**, Λ) obtained while removing constraint (E19) and initialize collection Λ with the *middle* arcs which support non null **z** values. Then we compute some feasible solution **t** of CTIME(Λ), set $\Lambda(\mathbf{t}) = \{(p, q) \in A(Carrier) : t_q \geq t_p + D^T(x(p, x(q)))\}$, and solve the Carrier($N$, **f**, $\Lambda(\mathbf{t})$) Min-Cost Flow problem. This provides us with an initial solution **H**, as well as with a collection Γ of paths that the carriers follows between the first time they load some item and the last time they unload an item.

Next we proceed iteratively, while trying at every iteration to concatenate two paths γ_1 and γ_2 of Γ into a unique one as in ([10]), so that a same carrier can follow those two paths and go back to $Depot$ without violating the time horizon constraint (see Figure 6). We do it, while relying on constraint propagation, in such a way that resulting path is the shortest possible in the **C** sense.

More precisely, we proceed in two steps:

- **Initialization**: Solve Split(N, **f**, Λ) with $\Lambda = \emptyset$ and derive vectors **z**, $\mathbf{z}^*$ which will remain unchanged during all the process. Initialize Λ with the arcs of $Carrier(N, \mathbf{f})$ which correspond to non-zero **z** values. Compute some feasible solution **t** of CTIME(Λ), set $\Lambda(\mathbf{t}) = \{(p, q) \in A(Carrier) : t_q \geq t_p + D^T(x(p, x(q)))\}$, and solve resulting Carrier(N, **f**, $\Lambda(\mathbf{t})$) Min-Cost Flow problem. Derive an initial solution **H**, as well as a collection Γ of paths that the carriers follow between the first time they load some item and the last time they unload an item. Propagate current constraints of CTIME(Λ) and get, for any node $p = (z, a, m, \varepsilon) \in X^*$ a time window $[Inf_p, Sup_p]$, with $Sup_p \geq T_{max} - D^T(x, Depot)$ and $Inf_p \leq D^T(Depot, x)$.
- **Main Loop**: It works as follows:
  1) Denote by $StartCarrier$ the set of all nodes $p_1 = (x_1, a_1, m_1, +)$ which are the start nodes of the paths of current collection Γ and by $EndCarrier$ the set of all nodes $p_2 = (x_2, a_2, m_2, -)$ which are the end node of the paths of current collection Γ.
  2) Select $p_2 = (x_2, a_2, m_2, -) \in EndCarrier$ and $p_1 = (x_1, a_1, m_1, +) \in StartCarrier$ such that $Inf_{p_2} + T_{max} - Sup_{p_1}$ does not exceed T_{max}, which also means $Inf_{p_2} \leq Sup_{p_1}$, and such that $Sup_{p_1} - Inf_{p_2}$ is largest possible. If $Fail(Select)$ then $Stop$ else keep on with 3) and 4).
  3) Insert arc (p_2, p_1) into Λ. Update the time windows induced by the constraint system CTIME(Λ) and compute some feasible solution **t** of CTIME(Λ).
  4) Set $\Lambda(\mathbf{t}) = \{(p, q) \in A(Carrier) : t_q \geq t_p + D^T(x(p, x(q)))\}$ and solve the Carrier(N, **f**, $\Lambda(\mathbf{t})$) Min-Cost Flow problem. Update accordingly the best solution **H** ever found.

  Stop occurs at instruction 2), when it is not possible to find a new arc (p_1, p_2) to insert into collection Λ.

D. Numerical Tests

According to this, we perform several numerical experiments, whose purpose is 3-sided:

1) Evaluating the error induced by the projection step, that means the gap between the value of the projected **PIBP** model and the value of the full TE-Network model.
2) Evaluating the error induced by the 2-step *Project and Expand* process, with respect to a theoretical value which would be obtained by solving in an exact way the full TE-Network **IBP** model.
3) Evaluating the ability of both heuristics *Dual_Carrier* and *Path_Concatenate* to compute in a short time an

efficient solution, with respect to the value obtained from application of a MILP library to the exact model Expand(N, **f**).

Technical Context: We run those experiments on a computer with a 2.3GHz Intel Core i5 processor and 16GB RAM, while using the C++ language (compiled with *Apple Clang 10*) and the CPLEX12.10 MILP library.

Instances: No standardized benchmarks exist for the generic IBP. So we built instances as follows: the node set X is a set of n points inside a 100×100 grid, the set of arcs A consists of m arcs generated randomly, the time matrix **T**= $(T_{(x,y)}, (x,y) \in A)$ corresponds to the rounded Euclidean Distance and the cost matrix **C**= $(C_a, a \in A)$ to the Manhattan Distance. Each node x but *Depot* is assigned to a b_x value in $\{-10, \dots, 10\}$, the capacity Cap is chosen in $\{2, 5, 10, 20\}$, and the time horizon limit $Tmax$ is a product $\lambda \cdot (\max_{(x,y) \in A} T_{(x,y)})$ when choosing $\lambda \in \{4, 5, 6, 8, 9\}$. The scaling coefficients α, β, γ are chosen in such a way that the values of cost components $\alpha \cdot$ *number of carriers*, $\beta \cdot$ *carrier riding cost* and $\gamma \cdot$ *items riding time* become comparable.

Outputs. Table I involves 15 instances and displays:

- Values n, m, respectively the number of vertices and arcs of the base network N.
- Values Cap, $Tmax$, respectively the carrier capacity and the time horizon value.
- Values **G1**, **V1**, respectively the optimal value of the projected model and related carrier number.
- Values **G**,**V**, respectively an upper bound value of the full TE-Network IRP model computed by the CPLEX library in no more than 1 h (this relatively short time limit is due to the fact that we use a personal computer), and related carrier number.

Table II involves the same instances as table I and displays:

- Values **GDC**,**VDC**, **TDC**, respectively the value computed by the *Dual_Carrier* Algorithm, related carrier number and related running time (in seconds).
- Values **GPC**,**VPC**, **TPC**, respectively the value computed by the *Path_Concatenate* Algorithm, related carrier number and related running time (in seconds).
- Values **GL**,**VL**, respectively the optimal value computed of the exact MILP Expand(N, **f**) model (computed through CPLEX Library) and related carrier number.
- Missing values are indicated by a hyphen symbol -, and correspond to PIBP solutions **f** for which the Expand(N, **f**) MILP is unfeasible.

Comments: The *Path_Concatenate* heuristic finds feasible IBP solutions for most instances but only 1, which happens not to be *Split/Carrier* consistent. Also, by comparing values **GDC** and **GPC**, **TDC** and **TPC**, **VDC** and **VPC**, we see that most of the time the solutions found by the *Path_Concatenate* heuristic have lower costs, involve fewer vehicles, and have required lower running times than the solutions computed by the *Dual_Carrier* Algorithm. Finally, comparing values **G**, **GL**, **V** and **VL** shows us that the *Project/Expand* decomposition scheme yields a very good approximation of the optimal value of the TE-Network IBP model. By the way, we notice an instance (instance 12) which really puts the MILP solver in trouble. Examining this instance makes appear that it is very tight, and admits few efficient feasible solutions. Moreover, related vehicle number coefficient α is rather large. So we feel that it was difficult for the MILP solver to turn rational solution into feasible integral solutions.

TABLE I
BEHAVIOR OF THE *Dual_Covering* AND *Path_Concatenate* ALGORITHMS.

Id	n	m	Cap	$Tmax$	**G1**	**V1**	**G**	**V**
1	20	78	2	324	2110.85	3	2633.00	4
2	20	65	5	400	1196.10	3	1282.7	3
3	20	77	10	440	854.83	2	1123.25	2
4	20	75	5	400	1105.00	2	1269.3	3
5	20	81	10	440	887.9	2	1005.05	2
6	50	163	2	460	15561.30	17	17043.00	20
7	50	155	5	390	4326.10	7	5023.7	9
8	50	149	10	440	7966.03	6	8820.5	8
9	50	146	20	436	1840.17	4	2670.95	6
10	50	168	20	436	2169.5	5	2750.10	6
11	100	363	2	336	17179.00	22	19483.00	27
12	100	236	5	516	4826.24	8	31881.35	86
13	100	289	10	432	6091.24	4	8450.00	9
14	100	296	5	516	4320.5	6	12029.4	20
15	100	308	10	432	6340.4	5	9875.4	12

TABLE II
BEHAVIOR OF THE *Dual_Carrier* AND *Path_Concatenate* ALGORITHMS.

Id	**GDC**	**TDC**	**VDC**	**GPC**	**TPC**	**VPC**	**GL**	**VL**
1	3097.00	0.17	5	3097.00	0.003	5	2633.00	4
2	1289.50	0.12	3	1289.50	0.100	3	1289.50	3
3	1493.25	0.44	4	1495.25	0.041	3	1468.35	3
4	1302.4	0.52	3	1269.3	0.25	3	1269.3	3
5	1212.9	077	3	1074.6	0.38	2	1005.05	2
6	20952.00	780.61	35	18435.00	19.007	24	17043.00	20
7	5328.9	122.6	10	5023.7	47.9	9	5023.7	9
8	-	-	-	-	-	-	-	-
9	2859.0	158.2	8	2711.4	54.8	7	2670.95	6
10 **G**	2900.8.0	226.0	7	2784.3	82.0	6	2750.10	6
11	23469.00	303.52	35	21623.00	9.303	32	20086.00	28
12	9331.00	382.99	24	6571.75	20.803	14	6436.0	14
13	10906.2	188.6	12	7548.00	65.3	7	7459.3	7
14	6224.7	405.6	10	5468.5	208.5	8	5029.4	8
15	9408.5	252.4	12	7977.0	109.0	10	7900.3	10

VI. CONCLUSION

We just presented a *Project/Expand* decomposition scheme for the handling of 2-commodity flow problems set on TE-

Networks and involving a coupling constraint. We focused here on the *Expand* issue, and proposed approaches based on the implicit management of the TE-Network.

Still, some issues remain open. One of them is the time dependency, when the state of the network evolves over time. Another one is about dynamicity and robustness since routing decisions are usually taken in a dynamic way and must cope with some uncertainty.

Acknowledgment

Present work was funded by French ANR: National Agency for Research, Labex IMOBS3, and PGMO Program.

References

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, M. R.Reddy, *Applications of network optimization*, Chapter 1 of Network Models, Handbook of Operation Research and Management Science 7, pp. 1-83, 1995. doi.org/10.1016/S0927-0507(05)80118-5

[2] J. Aronson, *A survey of dynamic network flows*, Ann. Oper. Res. 20, pp. 1-66, 1989. doi.org/10.1007/BF02216922

[3] C. Artigues, E. Hébrard, A. Quilliot, H. Toussaint, *Models and algorithms for natural disaster evacuation problems*, Proc. 2019 FEDCSIS WCO Conf., p 143-146, 2019. doi.org/10.15439/2019F90

[4] R. Bellman, *On a routing problem*, Quarterly of Applied Mathematics, 16, p 87-90, 1958.

[5] F. Bendali, J. Mailfert, E. Mole-Kamga, A. Quilliot, H. Toussaint, *Pipelining dynamic programming process in order to synchronize energy production and consumption*, Proc. 2020 FEDCSIS WCO Conf., p 303-306, 2020. doi.org/10.15439/978-83-955416-7-4.

[6] F. Bendali, J. Mailfert,and A. Quilliot, *Flots entiers et multi-flots fractionnaires couplés par une contrainte de capacité*, Investigacion Operativa, 9, 2001. DOI : 10.1051/ro:2006003

[7] S. Bsaybes, A. Quilliot, A. Wagler, *Fleet management for autonomous vehicles using flows in time-expanded networks*, TOP, Springer Verlag 27 (2), pp. 288-311, 2019. DOI: 10.1007/s11750-019-00506-4

[8] D. Chemla, F. Meunier,*Bike sharing systems: the static rebalancing problem*, Discrete Optimization 10 (2), p 120-146, 2013. doi.org/10.1016/j.disopt.2012.11.005

[9] A. O.Fleischer, M. Skutella, *Quickest flows over time*, SIAM Journal of Computing 36 (6), p 1600-1630, 2007. doi.org/10.1137/S0097539703427215

[10] S. Fidanova, O. Roeva, M. Ganzha, *Ant colony optimization algorithm for fuzzy transport modelling*, Proc. 2020 FEDCSIS WCO Conference, p 237-240, 2020. doi.org/10.15439/978-83-955416-7-4

[11] R. Ford and D. Fulkerson, *Flows in networks*, Princeton University Press, 1962.

[12] J. L.Gonzalez, M. Baiou,A. Quilliot, H. Toussaint, A. Wagler,*Branch and cut for a two commodity flow relocation model with time constraints*, Combinatorial Optimization. ISCO 2022. LNCS 13526. Springer, Cham. 2022. doi.org/10.1007/978-3-031-18530-4-2

[13] M. S.Hall and S. Hippler, *Multi-commodity flows over time*, Theoretical Computer Sciences, p 58-84, 2007.

[14] N. Kyngas, K. Nurmi, *The extended shift minimization personnel task scheduling problem*, Annals of Computer Sciences and Information Systems 26, p 65-74, 2021. doi.org/10.15439/978-83-959183-9-1

[15] K. Kishkin, D. Arnaudov, V. Todorov, S. Fidanova, *Multicriterial evaluation and optimization of an algorithm for charging energy storage elements*, Annals of Computer Sciences and Information Systems 26, p 61-64, 2021. doi.org/10.15439/978-83-959183-9-1

[16] W. B.Powell and P. Jaillet, *Stochastic and dynamic networks and network routing*, Handbook Operations Research, North Holland, 1995.

[17] F. T.Raviv and M. Tzur, *Static repositionning in a bike sharing system: models and solution approaches*, EURO Journal of Transportation and Logistics 2, p 187-229, 2013. DOI 10.1007/s13676-012-0017-6

[18] J. Schuijbroek, R. C. Hampshire, W. Van Hoeve, *Inventory rebalancing and vehicle routing in bike sharing systems*, EJOR 257 (3), (2017). doi.org/10.1016/j.ejor.2016.08.029

[19] K. Stoilova, T. Stoilov, *Bi-level optimization application for urban traffic management*, Proc. 2020 FEDCSIS WCO Conf., p 327-336, 2020. doi.org/10.15439/978-83-949419-5-6

[20] S. Varone, D. Schindl, C. Beffa, *Flexible job shop scheduling problemm with sequance-dependent transportation constraints and setup times*, Annals of Computer Sciences and Information Systems 26, p 97-102, 2021. doi.org/10.15439/978-83-959183-9-1

[21] Q. P. Zheng, A. Arulselvan, *Discrete time dynamic traffic assignment models and solution algorithms for managing lanes*, Journal of Global Optimization 51, p 47-68, 2011. doi.org/10.1007/s10898-010-9618-5

Proceedings of the 18th Conference on Computer
Science and Intelligence Systems pp. 677–687

DOI: 10.15439/2023F7224
ISSN 2300-5963 ACSIS, Vol. 35

Factors for Effective Communication of IT Costs and IT Business Value

Constanze Riedinger
0009-0003-0226-4114
Konstanz University of Applied Sciences,
78467 Konstanz, Germany
Email: constanze.riedinger@htwg-konstanz.de

Melanie Huber
0000-0001-8020-9055
BITCO^3 GmbH,
78467 Konstanz, Germany
Email: melanie.huber@bitco3.com

Niculin Prinz
0000-0002-3656-2668
Konstanz University of Applied Sciences,
78467 Konstanz, Germany
Email: niculin.prinz@htwg-konstanz.de

***Abstract*—Nowadays, organizations must invest strategically in information technology (IT) and choose the right digital initiatives to maximize their benefit. Nevertheless, Chief Information Officers still struggle to communicate IT costs and demonstrate the business value of IT. The goal of this paper is to support their effective communication. In focus groups, we analyzed how different stakeholders perceive IT costs and the business value of IT as the basis of communication. We identified 16 success factors to establish effective communication. Hence, this paper enables a better understanding of the perception and the operationalization of effective communication.**

Index Terms— **Effective Communication, Perception, Success Factors, IT Costs, IT Business Value, Business-IT Alignment, COBIT.**

I. Introduction

FOR DECADES, organizations face pressure to operate efficiently and manage resources and spendings strategically and in times of inflation and war this pressure even increases [1]. To remain competitive and mitigate security risks, those organizations raise their expenditures in information technology (IT) [2] and strive to choose the right digital initiatives [3]. Therefore, they require strategic cost management [4] and successful communication about IT costs and the benefit they generate through their IT investments [5]. However, for 63% of 166 interviewed Chief Information Officers (CIOs), it is still a challenge to communicate this business value of IT [6], which is also confirmed by other studies [7] [8]. Similarly, decision-makers struggle to foster transparent cost discussions [8]. This is why in this paper we aim to understand how IT costs and the business value of IT are perceived and communicated effectively.

Effective communication describes the "bidirectional exchange" [9] of information resulting in common grounds [10]. The foundation of effective communication and the premise to achieve business value from IT is the alignment between business and IT department [11]. Researchers extensively investigate the success factors for business-IT alignment (BITA) and the resulting impacts of better communication on their relationship [12–14]. This relationship also influences the effective communication of IT cost and the business value of IT itself [15–17]. Besides BITA, studies highlight further aspects of business value communication such as common language [15] or appropriate methods and metrics [16]. Furthermore, the perception of the stakeholders plays a decisive role in communication [18]. However, research does not examine the perception of IT costs and business value in connection with their successful communication in detail. Furthermore, a comprehensive overview of success factors for communication following an established framework [15] that supports the operationalization of conceptual models [19] is missing. Our study aims to fill these research gaps: we conduct focus group interviews to get practical insights and generate an overview using the established governance framework *COBIT* [20]. Thereby, we contribute to scientific research by shedding light on the current perception of IT cost and business value of IT and their communication. Furthermore, we present the success factors and support the operationalization of effective communication. Additionally, this study has practical implications: practitioners can use the results to recognize symptoms of non-constructive communication in their organizations and to gain awareness on how to develop an effective communication of cost and IT business value.

This paper is structured as follows: First, we introduce our theoretical foundation related to the communication between business and IT department. We thereby focus on enterprise governance of IT, business-IT alignment, and aspects of cost and value communication. We then show our research method, which comprises focus groups followed by a qualitative analysis. Finally, we present our findings, discuss them, and draw a conclusion.

II. Theoretical background for the communication between business and IT

Collaboration between business and IT department is the baseline for achieving IT business value [21]. The basic elements for this collaboration are structures, processes, and relational mechanisms defined and implemented as enterprise governance of IT (EGIT) [22]. Business-IT Alignment (BITA) thereby acts as a “mediating mechanism” between EGIT and IT business value [22]. It builds the base for effective communication between the business and IT departments [23]. The conceptual model in Fig. 1 presents this relationship. In the following, we describe the mentioned three elements for effective communication. In this paper, we refer to effective communication as “bidirectional exchange” [9] of the interlocutors to develop a “similar representation” [10] of the conversation content.

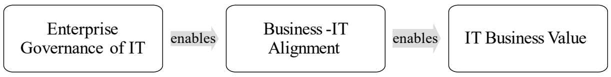


Fig. 1. EGIT-Alignment-Value Conceptual Model following [22]

A. Enterprise Governance of IT

A successful EGIT leads to better controllability of IT and consequently to greater business impact [24]. In order to achieve this successful governance, research intensively investigates determinants leading to an “integrated model of IT governance success and its impact” [25]: success factors analyzed in this context are e.g. the understanding of the IT value chain, top management commitment, IT’s business orientation, and the persuasiveness of communication. To establish successful governance and management of IT, the *Information Systems Audit & Control Association* (ISACA) presents COBIT as a “good-practice framework” with guidelines for organizations [19]. Following the core model of the most recent ***C**ontrol **OB**jectives for **I**nformation and Related **T**echnology 2019* framework, organizations should establish a governance system built from several components [22]: (1) Organizational Structures, (2) Processes, (3) People, Skills, and Competencies, (4) Information, (5) Culture, Ethics, and Behavior, (6) Services, Infrastructure, Applications, (7) Principles, Policies, and Frameworks. Those components lead to a comprehensive and functioning governance system and influence effective IT management [20]. Therefore, earlier studies apply the components to ensure comprehensiveness [8] or use the COBIT framework to transfer the practical functioning to a conceptual framework [19]. COBIT highlights that open and transparent communication about performance enables establishing trust and “a good relationship between IT and enterprise” [20]. It thereby recommends organizations involving and aligning all relevant stakeholders from business and IT departments to overcome communication gaps [20].

B. Business-IT Alignment

The alignment between business units and IT departments is a construct that has been intensively studied for several decades [26] [27]. An established model to describe this relationship between business and IT [28] is the Strategic Alignment Model (SAM) presented by [11]: thus, alignment is based on the strategic fit between external and internal domains as well as the functional integration between business and IT domains. This leads to multivariate relationships between business and IT, always considering the linkage of three of those four domains as presented in Fig. 2. The first perspective *Strategy Execution* (1) is the most common alignment perspective. In this case, the business strategy operates as a driver for organizational decisions and then influences the design of the IT infrastructure. Alignment through *Technology Transformation* (2) also originates from the business strategy followed by an appropriate IT strategy. This then leads to the required IT infrastructure and processes. The third and fourth perspectives are driven by the IT strategy. For the *Competitive Potential* (3) perspective, emerging IT capabilities lead to new strategic orientations of the business such as new products and services. Their implementation follows through the adaption of the operational business processes. The *Service Level* (4) perspective describes how IT builds up infrastructure and processes to then better support business operations.

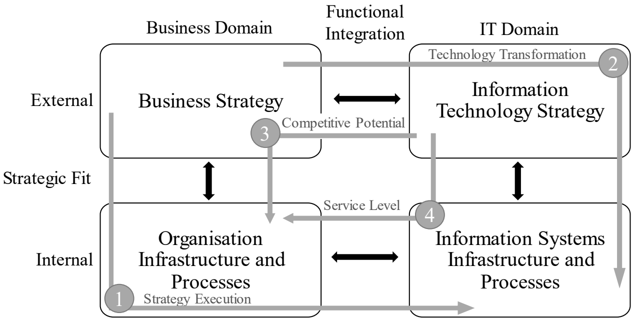


Fig. 2. Strategic Alignment Model following [11]

The alignment between business and IT is a communication and understanding-intensive continuous process [12] [11]. It is not a final state but rather described as “dynamic and evolutionary” [29]. To improve BITA, organizations need to consider six criteria including governance, partnership, scope and architecture, skills, value measurement as well as communication [29]. Depending on the maturity level of the alignment, the communication between business units and IT departments can be optimized and ideally lead to informal and pervasive communication [29]. Various enablers and inhibitors influence the maturity level of BITA and thereby also the effectiveness of communication and value measurement [14]. A literature review on such critical success factors of BITA presents those factors in three dimensions [13]: the human, the social, and the intellectual dimension. Success factors in the human dimension are IT skills & knowledge of business executives and vice versa as well as top management commitment, leadership skills of IT executives, and technical skills & knowledge of IT employees. The social dimension refers to a shared understanding of business

& IT executives, mutual trust & respect between business & IT executives, as well as business-IT partnership. On the intellectual dimension the alignment of business & IT strategy, goals & plans, as well as IT success, are critical for BITA. Those success factors indirectly influence effective communication regarding the costs and value of IT, since earlier studies highlight BITA itself an important factor for their effective communication [15].

C. IT Costs and Business Value of IT

In organizational communication of IT costs and business value, a shared perception often remains that "IT costs too much" [30]. Therefore, executives strive to determine the impact that IT has on the business. To measure this IT performance, research proposes diverse catalogs of key performance indicators (KPIs) [15]. For a variety of those KPIs, e.g. return on investment (ROI) [21], IT costs build the foundation [16]. Therefore, researchers develop conceptual frameworks to enable executives measuring the IT performance and thereby considering all IT-related costs as valid calculation basis [31]. They emphasize that not only the direct expenditures for development and implementation but also a variety of indirect costs related to human and organizational factors also count as IT costs [32]. However, organizations often still lack a clear understanding of the term "IT costs" and CIOs therefore face the challenge to transform non-constructive discussions about IT costs into a discussion about the business value contributed by IT investments [8]. "IT business value" is a concept that has been discussed in literature since the uprising of IT [33]. In line with former studies, we follow the understanding that IT business value is "measured by performance metrics on dimensions that stakeholders find important" [16]. As the perception of the stakeholders plays a decisive role in communication [18], they first need to have a common understanding of cost and value to achieve effective communication [16]. We address this current perception of the stakeholders in our first research question (RQ): *(1) What is the perception of business and IT stakeholders of "IT costs" and "IT business value"?*

Besides BITA and the common perception and understanding of the terms [16], further aspects influence the communication of IT costs and business value [15]. Transparency on IT cost information and the awareness of IT cost and cost drivers are a baseline to communicate IT costs and demonstrate the value of IT [8]. From this cost information, KPIs formulated in business language are an important factor for the effective communication of IT business value [16]. Furthermore, [29] mentions metric portfolios and dashboards to visualize value in communication. Business-IT structures that evaluate IT investments together also count as a success factor [34]. The value communication throughout the organization should then be audience-oriented using different channels [23]. To summarize relevant aspects of value communication, [15] conduct a literature review and investigate how to conceptualize value communication: They present different categories, e.g. transparency, understanding, collaboration, methods, transparent communication, and common language. However, research shows that especially establishing a common language and implementing collaboration on an equal footing is still difficult [8]. Therefore, CIOs still face the challenges of implementing successful communication [5] and demonstrating the business value of IT investments in their organizations [8]. They miss practicable success factors that operationalize the successful communication between business and IT on IT costs and IT business value. This gap leads to our second RQ: *(2) What factors do organizations need to consider for implementing an effective communication of IT costs and business value?*

III. Methodology

We follow the research method of focus groups because it enables academics to study specific topics in groups using focused interviews with a determined direction [35]. Earlier studies in IS make use of focus groups to evaluate design science artifacts [36], to develop a research agenda based on issues of merger & acquisition [37] or to identify factors related to the choice of students to study IT [38]. In small groups with an optimal size of 5 to 8 participants, specific topics may be studied in depth [39]. The direct feedback and the group interaction challenge interviewees' views and can inspire them to new ways of thinking which provide researchers with an in-depth view and innovative ideas [35]. It, therefore, enables researchers to get a wide variety of opinions or perceptions concerning an issue, behavior, or practice and thereby uncover factors that influence those opinions [39]. This is why focus groups support the aim of our study to understand the perception of IT costs and IT business value and get a broad view of the success factors of their communication. We divide our focus group study into two phases [37]: first, the planning phase, including the participant selection [39], and second, the sessions and the analysis.

A. Planning and Participant Selection

In order to achieve our research goal, we chose a single-category design for our focus groups and developed questions to guide the facilitation of the discussion [39]. The questions were open-ended to motivate participants to answer according to their specific situation [39]. The guiding questions are:

- What are we talking about when we talk about IT costs? Which costs belong to the total IT costs?
- What are we talking about when we talk about IT value? What is IT value for you?
- How do you communicate IT costs and the business value of IT within your organization?
- What are the challenges in those discussions?
- What is required to effectively communicate IT costs and IT business value in organizations?

To discuss these questions, we included a broad spectrum of people coming from different industries to cover a variety of perspectives. As a decisive criterion for the selection of the focus group participants, we specified that each of them must have a relevant responsibility and expertise in the communication of IT costs and IT business value. We followed the recommendation by [39] of 5 to 8 participants per focus group.

All participants were already part of the network of the research team. Table I provides an overview of the participants in the two focus groups.

TABLE I. PARTICIPANTS OF THE FOCUS GROUP SESSIONS

ID	Function	Industry (#Employees (EMP))
FG1-1	Head of IT Governance	Transportation (30.000 EMP)
FG1-2	CIO	Service Industry (600 EMP)
FG1-3	CIO	Electronics Manufacturing (1.000 EMP)
FG1-4	CIO	Pharmaceutical Industry (78.500 EMP)
FG1-5	CIO Office	Insurance (3.000 EMP)
FG1-6	Head of Value Mgmt.	Energy (91.000 EMP)
FG1-7	CIO	Infrastructure (1.000 EMP)
FG1-8	IT Controller	Energy and Agriculture (22.300 EMP)
FG2-1	Head of IT Governance	Transportation (2.700 EMP)
FG2-2	IT Controlling	Insurance (3.000 EMP)
FG2-3	IT Controlling	Retail (35.000 EMP)
FG2-4	Controlling	Electronics Manufacturing (1.000 EMP)
FG2-5	Controlling	Banking (2.500 EMP)
FG2-6	Controlling	Electronics Manufacturing (1.000 EMP)
FG2-7	IT Portfolio Management	Energy (91.000 EMP)
FG2-8	IT Controlling	Transportation (30.000 EMP)

B. Focus Group Sessions and Data Analysis

The focus group sessions were executed in September and October 2022. We facilitated the focus groups following the leading questions. We audio-recorded the sessions and took field notes during the discussions. We then conducted qualitative coding [40] across the group results to answer our research questions. The resulted figures for RQ1 were rediscussed in a second session with the focus groups. The coded factors for effective communication to respond to RQ2 were reviewed and adjusted in two further iterations by the research team, consisting of three researchers. We further applied selective coding to assign the 16 identified success factors to the COBIT components as displayed in earlier research [8]. Through the alignment of the factors to the categories, we ensure a holistic approach and the link to an existing framework [41] [8]. Following earlier research on success factors [42] [43], we chose the Ishikawa diagram to visualize the relationship between the identified factors. It further allows structuring the problem and the determining factors [43].

IV. FINDINGS

In the following, we outline our findings from the focus group sessions and thereby answer our research questions. To do so, we present the perception of IT costs and value and describe the factors for effective communication of IT costs and IT business value.

A. The perception of IT costs and IT business value

In this subsection, we describe the results for our first research question: *What is the perception of business and IT stakeholders of "IT costs" and "IT business value"?* We start with the perception of IT costs followed by IT business value.

IT cost perception

The interpretation of IT costs varies among the participants of the focus groups, ranging from *only the costs allocated to the IT department* to *all input factors related to information technology independent of the place of origin*. But even if they perceive all input factors related to IT as IT costs, shadow IT and new technologies such as Low Code Development Platforms, which are largely based in the business, are difficult to identify and capture. One interviewee therefore says that in their organization they estimate a proportion of IT costs for shadow IT to calculate the overall IT costs and proceed with their allocation to the business units (FG1-5). The participants further mention that product IT and operations IT are mainly neglected in their organizations' understanding and consequently management of IT costs. The perception of IT costs often differs from the employees in IT departments to business departments: while the IT department distinguishes between service development and operations or project implementation and management, business departments mainly realize total project or service costs. Furthermore, business stakeholders focus on charged costs such as managed workplace, IT management or overhead fees, service packages, or application licenses. One participant outlines that due to regular exchange and a clear corporate guideline concerning the definition of IT costs, stakeholders in his organization achieve a consensus on how to proceed with IT cost management even if their personal perception differs (FG1-2).

In collaboration with the participants, we develop a layer model on different perceptions of IT costs. Fig. 3 presents these different distinctions and levels of detail.

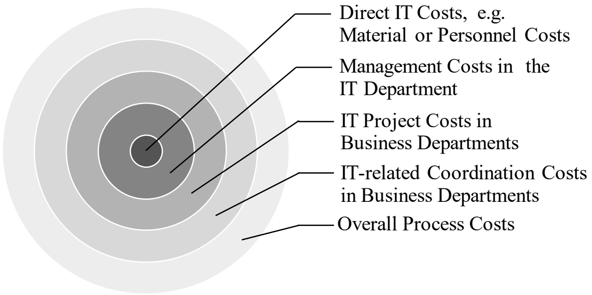


Fig. 3. Layers of IT cost perception

Direct IT costs related to activities assigned to the IT department are displayed in the first layer. With this commonly agreed inner layer, the focus group participants follow definitions of direct costs propagated by [31] and consider costs for the development and operation of hardware and software, personnel costs, external services, and shared services as general IT costs. The next layer comprises *management costs in the IT department*. They include the planning functions of IT, such as governance or enterprise architecture management, which only indirectly contribute to value creation. Most participants perceive these overhead costs as part of the overall IT costs that are "charged to the business units as overhead fee (FG2-1)". However, *IT project costs* that arise within *business departments* are often no longer fully attributed to IT costs. Those costs, displayed in the third layer, are related to activities such as process definition or testing within the business departments. The fourth layer comprises *IT-related coordination costs in business departments*. They primarily include the time spent by business departments on IT tasks such as training, key user activities, or committees for process and project portfolio coordination. Most focus group participants perceive these costs as IT costs but highlight, that they are only partially or not at all accounted as IT costs in their organizations. The consideration of *overall process costs* is only occasionally used for highly automated processes. However, the "effort of a holistic end-to-end consideration of all IT costs in a process does often not pay off" (FG1-3). Therefore, the interviews show that organizations rarely apply an integral approach such as activity-based costing and thereby perceive all costs related to IT throughout a whole process as IT costs.

IT business value perception

During the group interviews on perception, the participants acknowledge that measuring and presenting the business value of IT is a significant challenge. For most of them, the term "business value of IT" refers to the contribution of IT activities to the overall value of a company and perceived by the business. However, this perception varies between the stakeholders in the companies: especially the top management often only perceives value as financial benefits of IT. Therefore, the initial categorization of IT business value perception follows two dimensions: monetary and non-monetary. Typically, the monetary value contribution of IT is measured through revenue. The non-monetary value contribution, by contrast, is reflected by other features: the focus can be on enhancing business capabilities. Likewise, the added value can lie in the optimization of existing processes or capabilities. The external perception of customers or partners regarding new business fields or security risks can also determine the non-monetary added value of IT.

The collaboration between business and IT can also determine different perceptions of the business value of IT. The dimensions discussed in the focus group sessions relate to the SAM model and its alignment perspectives [11] displayed in Fig. 4: In terms of strategy execution (1), IT's added value lies in maintaining operational capability and sustaining the business. IT is an integral part of the business. It provides basic services and meets technical requirements to keep operations running. The alignment between business and IT, following the technology transformation perspective (2), ensures the business value of IT through the IT support for strategic differentiation. Here, the IT department functions as a strategic sparring partner in the development and implementation of the business strategy. IT builds strategic competencies and thereby supports the strategic differentiation of the organization. The participants mention IT's value through the provision of data, which enables better decisions to be made, e.g., for predictive maintenance or risk modeling. Besides that, IT can deliver value by enabling the growth of new business areas or the repositioning of the strategic product market combination through innovation. This can be achieved through an alignment on the competitive potential perspective (3) within SAM. Finally, with alignment on a service level perspective (4), IT's value lies in continuous improvement. The focus groups perceive this perspective of the business value of IT in the development of tools and processes that enable automation and process optimization. Here, the IT department builds new IT capabilities and through them offers opportunities to reduce costs and increase efficiency and effectiveness, as well as to support business capabilities.

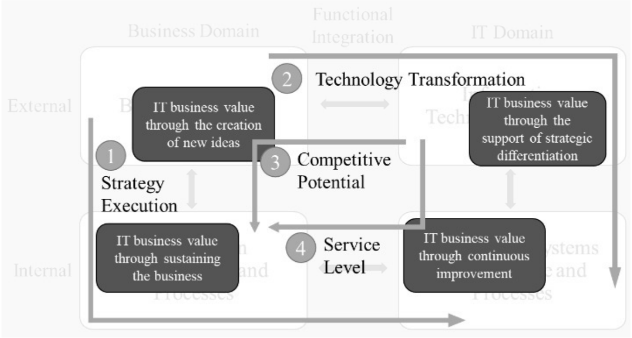

Business Domain
Functional Integration
IT Domain
External
Internal
2 Technology Transformation
IT business value through the creation of new ideas
IT business value through the support of strategic differentiation
1 Strategy Execution
3 Competitive Potential
IT business value through sustaining the business
4 Service Level
IT business value through continuous improvement

Fig. 4. Differing perceptions of business value of IT related to SAM

B. Factors for Successful Communication

Next, we consider the findings that answer our second research question: *(2) What factors do organizations need to consider for implementing an effective communication of IT costs and business value?* Our findings show that organizations need to take into account various factors to operationalize a effective communication of IT costs and the business value of IT. The Ishikawa diagram in Fig. 5 illustrates those factors. In the following, we outline and describe the identified factors based on the seven components of the COBIT 2019 framework.

Organizational Structure

COBIT 2019 presents organizational structures as "key-decision making entities in organizations" [20]. For the effective communication of IT costs and business value, organizations require interface functions and cooperative governance as structural elements:

- **Interface functions:** establish key functions for the dialogue between the business and IT department.
- **Cooperative governance:** ensure responsibility and decision competencies of both business and IT communication part.

For the focus group, an important success factor are interface functions that communicate information on IT cost and business value between the IT department and the business. A key function should be situated within the IT department. However, a collaboration also requires a determined counterpart on the business side to become the "voice of IT within the business" (FG1-1). Those interface functions further request responsibilities for decision-making. Decisions then should be taken in cooperation to foster involvement and commitment on both sides. This cooperative governance enables a strategic discussion on IT investments and final metrics and leads to effective communication of the business value of IT. Additionally, top management should be involved in strategic discussions through boards and commit to decisions.

Processes

The component processes describes "activities to achieve certain objectives and [...] overall IT-related goals" [20]. Effective communication requires activities such as regular dialogues and a uniform approach to the discussion between

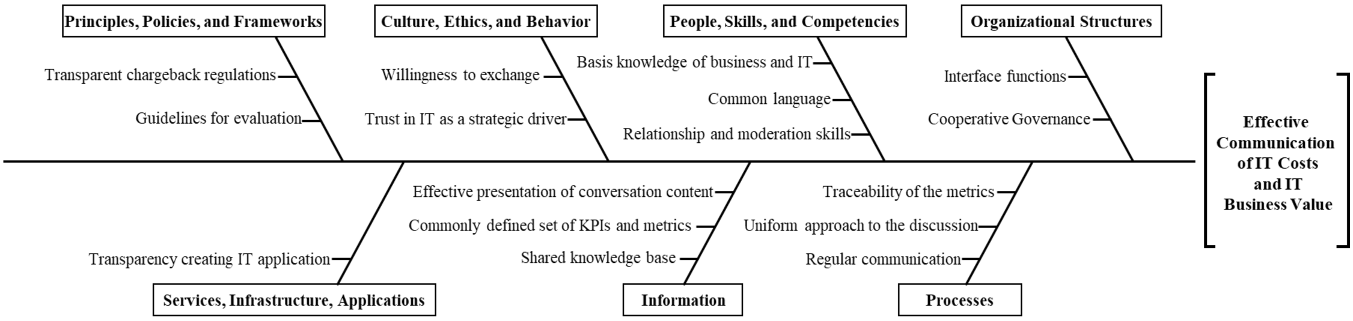

Principles, Policies, and Frameworks
Transparent chargeback regulations
Guidelines for evaluation
Culture, Ethics, and Behavior
Willingness to exchange
Trust in IT as a strategic driver
People, Skills, and Competencies
Basis knowledge of business and IT
Common language
Relationship and moderation skills
Organizational Structures
Interface functions
Cooperative Governance
Effective Communication of IT Costs and IT Business Value
Effective presentation of conversation content
Commonly defined set of KPIs and metrics
Shared knowledge base
Traceability of the metrics
Uniform approach to the discussion
Regular communication
Transparency creating IT application
Services, Infrastructure, Applications
Information
Processes

Fig. 5. Success Factors for Effective Communication of IT Costs and IT Business Value

business and IT. Furthermore, the tracking activity of metrics and committed tasks accounts for a success factor.

- **Regular communication:** establish a dialogue format between business and IT counterparts for regular communication.
- **Uniform approach to the discussion:** foster a defined approach to the evaluation and discussion of value.
- **Traceability of the metrics:** track the defined metrics and ensure commitments and consistency.

The participants stress the regularity of communication. They, therefore, propose a dialogue format to discuss the strategic use of IT and occurring costs. This dialogue should be integrated into the annual planning processes to ensure that the topics are incorporated into the budget plan. The frequency may vary depending on the business requirements. Secondly, a uniform approach to this discussion is crucial as it enables comparability and fairness. Furthermore, uniformity in the evaluation of IT investments ensures a clear methodology and transparency for all stakeholders. The participants highlight procedures such as the calculation and evaluation of business cases that require a consistent procedure. Finally, tracking the metrics and the communicated commitments is another important factor. It includes measuring business improvement against the business case and maintaining the business cases to demonstrate value over time. This requires not only a process responsible, but also all parties involved should have assigned tasks and be accountable for them. The liability of the stakeholders to those tasks then increases with a clear tracking activity.

People, skills, and competencies

Effective communication of IT cost and business value involve people with skills and competencies "for good decisions, execution of corrective action and successful completion of all activities" [20]. According to the focus groups, communication about IT costs and value often fails due to a lack of knowledge and know-how, both in the business and in the IT department, as well as a lack of efficiency in implementation. The involved stakeholders, therefore, require a basic knowledge of the business and IT domain, the competency to communicate about the costs and value associated with IT, as well as soft skills that enable effective communication and problem-solving.

- **Basic knowledge of business and IT:** build knowledge in business and IT departments to understand the specialization of the conversation partner.
- **Common language:** establish the competency to develop a common language to communicate IT costs and IT business value.
- **Relationship and moderation skills:** ensure effective communication tailored to the target group through relationship and moderation skills.

The development of knowledge is relevant for the interlocutors. Business employees require technical knowledge and awareness of challenges. They should understand the processes and workings of the IT department. Furthermore, they need skills to describe their requirements concerning IT or the problems and deficiencies they experience. IT employees, in contrast, need business knowledge about the capabilities and processes as well as methodical know-how on cost accounting. Furthermore, both conversation partners should establish a common language. This competency not only includes the common definition of IT costs and value but particularly understanding where the value of IT can be realized in the respective business units and for the whole organization. Business and IT counterparts should understand how to generate value and what drives the thereby occurring IT costs. For this conversation, soft skills are essential to build trust and communicate audience-oriented. The IT counterpart should "take a pragmatic approach and build a relationship through honesty" (FG1-3). Thereby, a "structured way of moderation" (FG2-8) strengthens the credibility and leads to an efficient conversation.

Information

The COBIT 2019 framework states that information is omnipresent throughout any organization [20]. The effective communication of IT costs and business value requires information to build a shared knowledge base. Additional success factors are a commonly defined set of KPIs and metrics and the effective presentation of the conversation content.

- **Shared knowledge base:** report basic information necessary to understand counterparts and find common ground.
- **Commonly defined set of KPIs and metrics:** agree on a set of metrics to measure IT performance quantitatively and qualitatively.

- **Effective presentation of conversation content:** visualize the information on IT cost and business value of IT appropriate for the respective stakeholders.

For the focus group participants, a shared knowledge base enables effective communication. It relates to a structured approach beginning with the alignment of the IT strategy with business objectives and financial constraints. This information leads to a common understanding in all interlocutors of how IT can contribute to the business and the complexity that may hamper the contribution of IT to business objectives. The focus of communication should be on the impact of IT use instead of purely cost-based discussions. This involves asking questions such as where IT can deliver value, what the costs are, and how they can be optimized, highlighting the strategic significance of IT. To communicate this business value, the stakeholders should come up with an agreed set of KPIs and metrics to measure IT performance. However, defining and implementing these KPIs is often a challenge. The participants, therefore, see an important factor in defining the KPIs in cooperation between business and IT. Moreover, they propose to use business capabilities as a baseline and to develop metrics that capture IT's contribution, both in terms of euro value and soft factors. Lastly, successful communication of IT cost and value requires an effective presentation. This entails the creation of a management-ready consolidation of the outcomes adapted to the communication standards of the organization. One participant also mentions a "portfolio visualization that enables better understanding in the business and that allows stakeholders to measure progress through the agreed targets" (FG1-6).

Culture, ethics, and behavior

The factors concerning culture, ethics, and behavior "are often underestimated" [20], however, for the focus groups, corporate culture plays an essential role in the effective communication of IT costs and business value of IT. They highlight the willingness to exchange information as well as the trust in IT to deliver value and drive the business strategically:

- **Willingness to exchange:** foster acceptance of IT as a discussion partner and the commitment to exchange information.
- **Trust in IT as a strategic driver:** foster awareness of the qualitative value proposition of IT instead of rather number-driven management.

The participants mention that they often face a "transparent wall" (FG2-5) between business and IT that hinders effective communication and the perception of the business value of IT. The commitment to exchange at all levels and the acceptance of the IT department as a "discussion partner at eye level" (FG1-1) counts as a success factor. Business and IT should commit together to savings and their consequences because a "successful exchange is also about putting the same intentions – the overall success of the company – in the center of attention" (FG2-7). Thereby, especially business employees should trust that IT may drive business development strategically. A shared worldview on the strategic importance of IT then enables effective communication with a common understanding and acceptance of possible value contribution of IT. The interlocutors should accept that besides the quantitative performance measures, IT's contribution may also be qualitative. The success of communication, therefore, requires a cultural shift from numbers-driven management to the acceptance of qualitative arguments, also on the business side.

Services, infrastructure, and applications

For the support of processing and management in organizations with information technology, COBIT 2019 mentions services, infrastructure, and applications [20]. Also, effective communication necessitates technology support to create transparency.

- **Transparency creating IT application:** provide technical support for illustration of IT costs and metrics as well as tracking mechanisms for communicated targets.

For the participants, transparency of IT costs is crucial for honest communication. Therefore, IT applications are required to integrate cost information of different databases and illustrate relevant numbers. The communication of business value should then be supported by applications comparing actual and target figures as well as the budget plan and the actual project portfolio. This enables comparability and traceability of IT investments and their generated impact.

Principles, policies, and frameworks

For "practical guidance for day-to-day management" [20], organizations apply principles, policies, and frameworks. The participants mention tense discussions in day-to-day management about cost allocation and service delivery. They, therefore, highlight practical guidance through transparent chargeback regulations and guidelines for IT investment evaluation as success factors in communication.

- **Transparent chargeback regulations:** set up agreed rules for the allocation of IT costs.
- **Guidelines for evaluation:** set up transparent guidelines for decision-making and if required business case calculation for IT investments.

A prerequisite for the setup of cost allocation rules is a comprehensible service offer of IT. Then business requires transparency on what they will be charged for. Preselected chargeback regulations foster the acceptance of IT costs in the business. Thereby, the participants mention that e.g. in the case of cost allocation with planned prices, the decision on how these allocated costs are compared with the actual amounts at the end of the fiscal year should be announced transparently (true up or true down). Furthermore, organizations need clear guidelines on how to decide on IT investments and if business case calculation is valuable. If a business case is required there should be clear calculation specifications and evaluation mechanisms. The participants however stress that not for every decision a business case is needed, especially if "projects are business critical or legally necessary" (FG1-6).

V. Discussion

The findings show that the perception of IT costs and business value differs among the stakeholders. In the following, we discuss the significance of these differing perceptions and their relevance to communication. Furthermore, we examine the identified success factors, compare them to previous research and outline how they can contribute to more effective communication.

A. Differing perceptions of IT costs and business value of IT

The costs of IT "appear more tangible in nature" [31] than the value and therefore are "often perceived to be easier to estimate" [31]. The findings stress that business and IT departments however rely on different understandings of IT costs. This lack of a clear definition and a common understanding hampers transparency and leads to difficulties in cost management [8]. The developed layer model visualizes the complexity and multidimensional nature of IT costs for the participants. It stresses that the stakeholders generally agree on the core IT costs, referred to as *direct IT costs* [31], and the *management cost in the IT department*. However, indirect costs, i.e., the outer layers, are often not treated or perceived as IT costs within organizations. We identified those "human and organizational factors" [31], especially outside the IT department. In a detailed study, [32] present various characteristics of indirect costs and emphasize the importance of recognizing these costs as IT costs to enable a holistic evaluation. Therefore, the perception of what IT costs are and to what extent they should be included in metric calculations must follow a clear process and be "determined by a clear cost structure" (FG2-3). Besides, a lack of standardization also leads to challenges in comparing IT costs externally [8]. One focus group participant also mentions that "benchmarks, therefore, create false expectations" (FG1-3) as comparison with other companies is very difficult. The layer model displays those different possibilities of perception and creates awareness of the multifaceted complexity of IT cost. Therefore, it builds a basis for a common understanding of how to define IT costs within the organization as a basis for performance metrics and how to compare them beyond.

The inconsistent perception of IT costs and resulting difficulties in IT evaluation hampers the communication of value. Especially if the "business value of IT is equated with revenue and the non-monetary contribution of IT is not acknowledged" (FG2-2). One common contrast, also highlighted by the participants is the differentiation between monetary and non-monetary contribution: business cases are calculated to express the monetary contribution in comparison to IT costs. The non-monetary contribution of IT, however, is challenging to communicate and "misunderstandings about the definition of value can lead to feelings that value was not delivered" [34]. Alignment between business and IT is therefore indispensable for their mutual understanding and the perception of value [29]. To highlight this, we align the different perceptions of value in Fig. 4 to the SAM and thereby display that the value contribution of IT and its perception differs related to the alignment perspective. The participants stress that IT is mainly perceived as an enabler, sustaining the business. This follows the most common alignment perspective [11] Strategy Execution (1). However, if business and IT foster alignment on different perspectives, it can also improve the perceived value contribution of IT [22]. The results show that IT therefore should take the initiative and empower also the other perspectives to amplify value contribution and perception in the business.

B. Communication as the basis of perception and vice versa

The findings show that organizations with regular exchange and clear definitions struggle less to adopt a shared perception of IT costs and business value. An earlier study investigates how executives achieve consensus on the perception of business value and identifies communication as one supportive factor for this consensus [9]. The focus groups highlight that a lack of constructive communication between business and IT departments leads to "accusations that the IT department is too expensive" (FG1-1). This sentiment reinforces a previous study mentioning that communication shortfall results in different perceptions and limited comprehensibility of expectations on the business side [33]. Also, non-constructive discussions about the costs and value contribution of IT provoke cultural differences [33] [44]. The participant describes this as a lack of trust in the IT department and in IT to be a strategic driver for the business. For this, the ability to develop a "common language is indispensable" (FG1-6). This common language should ideally be expressed in business terms to ensure consistent perception across stakeholders [16]. Thus, our findings stress that for a shared perception of IT costs and business value, business and IT require effective communication.

Effective communication of IT costs and business value, however, also necessitates a shared understanding of the relevant topics. The stakeholders' perception, therefore, is decisive so that both sender and receiver within a bidirectional exchange feel satisfied and consider the communication effective [18]. Several identified success factors foster the development of this "similar representation in the interlocutors" [10]: Besides the ability to develop a **common language**, the skill to have a **basic knowledge** about the domain of the interlocutor supports a common ground in the conversation. For this also the given information should serve as a **shared knowledge basis**. The **effective presentation of the conversation content** then assists communication through visualization. In conclusion, shared perception and understanding are necessary for effective communication, and establishing these requires active efforts from all participants. The interdependence of perception and communication reinforces that besides the common language, a shared knowledge base, and effective presentation, regularity is decisive for effective communication of IT costs and business value. The success factor of **regular communication** supports this required continuous communication process.

C. Success Factors for communication

The effective communication of IT costs and business value is based on successful governance and alignment [22]. Our findings support this interrelation by addressing the success factors mentioned in the EGIT and BITA literature: Especially the factors identified in the categories *people, skills, and competencies* as well as *culture, ethics, and behavior* incorporate the success factors of EGIT and BITA mentioned in the theoretical background section. The outlined cultural aspects include factors such as mutual trust and respect [14], business-IT partnership [14], or IT's business orientation [25]. Success factors identified in the category people, skills, and competencies comprise BITA prerequisites such as various skills and knowledge of both business and IT executives [13] as well as the understanding of the IT value chain and persuasiveness of communication [25] required for EGIT success. Our study indicates that the effective communication of IT costs and business value requires further aspects. Therefore, organizations necessitate successful EGIT as a basis and need to drive BITA as a continuous process [12] but additionally, they should consider specific factors for the communication of IT costs and business value of IT.

Earlier studies in the business value context already mention shortfalls in communicating IT costs and business value and highlight relevant aspects for effective communication [15]. The developed factors in our study are in line with these factors of previous studies: e.g. the required **interface function** represents "business-IT structures to recognize and evaluate opportunities" [34]. The processual factor of **regular communication** mentions regular collaboration [15] and a **uniform approach to the discussion** includes "a clearly defined portfolio value management process" [34]. Earlier studies further establish metric portfolios [15][16] as a prerequisite for effective measurement and communication of business value or balanced dashboards to "demonstrate the value" [29]. In this study factors related to information such as a **commonly defined set of KPIs and metrics** and the **effective presentation of conversation content** include those aspects. This close connection with individual aspects investigated in other studies validates our findings. In contrast to these individual mentions of success factors in previous research, our study develops a comprehensive overview. It thereby includes not only existing aspects but expands them including factors in the category of *principles, policies, and frameworks* neglected in previous studies. The developed overview of success factors, therefore, provides an extension of existing literature as well as a focused, summarized view aligned with the factors for business value communication identified in earlier studies.

The mapping of relevant success factors onto the COBIT 2019 framework provides an overview from different perspectives. With the seven components, referred to as "enablers" [22] in earlier COBIT versions, our categorization enables the implementation leading to successful IT management [19]. However, as organizations face difficulties in the implementation of effective communication [5, 6, 8], they require in addition to the aforementioned success factors further practical guidance for the operationalization of effective communication. Research could, therefore, propose a detailed design of **regular communication** based on the other identified factors. As outlined, the main focus thereby needs to be on the unified understanding and perception. Earlier studies emphasize the importance of a common language [15] based on "terms that the business understands" [29]. In addition, proven approaches to communication plans [15] [16] should be considered. With the success factors presented, we provide a basis for future research on the operationalization of effective communication between the business and IT department regarding IT costs and the business value of IT.

VI. Conclusion

Increasing inflation and security risks urge organizations to manage IT costs effectively and efficiently. Thereby, the communication of IT costs and the impact generated through IT is crucial to remain competitive and to strategically plan investments. However, CIOs face challenges to create a common understanding of IT costs and the business value of IT and furthermore, effectively communicate them. A current view on the perception as well as a holistic overview of success factors to operationalize effective communication does not yet exist in academia. Therefore, the goal of this publication is to identify how business and IT stakeholders perceive IT costs and IT business value. With this understanding, we further aim to give an overview of the success factors that organizations need to consider for the effective communication of costs and business value of IT. To reach this goal, we conduct a focus group study and discuss the findings. In summary, the investigation of the perception shows that it differs between the stakeholders for IT costs as well as for the business value of IT. We conclude that to create a common understanding in organizations, they require communication about the perception and a common ground to discuss IT costs and IT business value. The result of the mapping to the seven COBIT components ensures a holistic perspective and enables transparency on the 16 identified factors for successful communication. It thereby offers a structured representation as a basis for the operationalization of effective communication between the business and IT department about IT costs and IT business value. Furthermore, the paper highlights that this interchange is based on the stakeholder's competency to communicate in a common language. This common language then should enable the transparent demonstration of the value of IT for the organization.

This paper makes a theoretical contribution by providing an insight into the current perception and communication of IT costs and business value of IT, and thus by identifying the relevant factors to operationalize an effective communication. Moreover, practice gains awareness about possible reasons for non-constructive IT cost communication and guidelines on how to turn it into effective communication with a focus on the business value of IT.

However, the study itself has limitations. First, the focus groups were conducted only with organizations situated in Germany. We mitigated these limitations by including participants with diverse backgrounds and from different industries and company sizes. Although we cover as broad a spectrum as possible with these participants, further focus groups around the world could enrich our research findings and add even more validity. Second, the study does not distinguish the perceptions and success factors by company size or industry which could provide a more differentiated view. Focus group design concentrating on organization sizes or industries would enable comparison and stakeholder-specific success factor analysis. Third, the study lacks proof that the identified success factors facilitate the operationalization of IT cost and value communication. To overcome this limitation, researchers could in the next step provide a case study implementing the success factors and evaluating whether considering these factors leads to more positive perception and successful communication. Furthermore, future research should seek to conceptualize how communication of IT costs and business value should look like considering the identified success factors.

References

[1] S. Solanki and A. Bant, *9 Winning Actions to Take as Recession Threatens.* [Online]. Available: https://www.gartner.com/en/articles/9-winning-actions-to-take-as-recession-threatens (accessed: Mar. 28 2023).

[2] Gartner Inc., *Gartner Forecasts Worldwide IT Spending to Grow 5.1% in 2023.* Gartner IT Symposium/Xpo™ 2022. Orlando, 2022. Accessed: Apr. 17 2023. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2022-10-19-gartner-forecasts-worldwide-it-spending-to-grow-5-percent-in-2023

[3] Gartner Inc., "2023 CIO Agenda. 4 Actions to Ensure Your Tech Investments Pay Digital Dividends," 2022.

[4] C. Riedinger, M. Huber, N. Prinz, and C. Rentrop, "Towards a Taxonomy of Strategic Drivers of IT Costs," in *Information Systems: Proceedings European, Mediterranean, and Middle Eastern Conference on Information Systems (EMCIS 2022)*, M. Papadaki, P. Da Rupino Cunha, M. Themistocleous, and K. Christodoulou, Eds., Cham: Springer Nature Switzerland, 2023, pp. 555–569.

[5] C. Lozada and R. Naegle, "Effective Communication Is Critical to Successful Cost Optimization Efforts," 2020. Accessed: Apr. 13 2023. [Online]. Available: https://www.gartner.com/en/documents/3990229

[6] Gartner Inc., *Communicate IT's Business Value.* [Online]. Available: https://www.gartner.com/en/information-technology/insights/business-value-of-it (accessed: Mar. 28 2023).

[7] P. Hillebrand and M. Westner, "Success factors of long-term CIOs," *Inf Syst E-Bus Manage*, vol. 20, no. 1, pp. 79–122, 2022, doi: 10.1007/s10257-021-00546-z.

[8] C. Riedinger and M. Huber, "An Expert View on Challenges in Managing IT Costs in the Digital Age," in *IADIS IS 2023 Proceedings*, Lisbon, Portugal, 2023.

[9] P. P. Tallon, "Do you see what I see? The search for consensus among executives' perceptions of IT business value," *Eur J Inf Syst*, vol. 23, no. 3, pp. 306–325, 2014, doi: 10.1057/ejis.2013.2.

[10] M. J. Pickering and S. Garrod, "Alignment as the Basis for Successful Communication," *Research Language Computation*, vol. 4, 2-3, pp. 203–228, 2006, doi: 10.1007/s11168-006-9004-0.

[11] J. C. Henderson and H. Venkatraman, "Strategic alignment: Leveraging information technology for transforming organizations," *IBM Syst. J.*, vol. 32, no. 1, pp. 472–484, 1993, doi: 10.1147/sj.382.0472.

[12] J. Cybulski and S. Lukaitis, "The impact of communications and understanding on the success of business/IT alignment," 2005.

[13] I. Kurti, E. Barolli, and K. Sevrani, "Critical success factors for business-IT alignment: A review of current research," *Romanian Economic and Business Review*, vol. 8, no. 3, p. 79, 2013.

[14] J. Luftman, R. Papp, and T. Brier, "Enablers and Inhibitors of Business-IT Alignment," *CAIS*, vol. 1, 1999, doi: 10.17705/1CAIS.00111.

[15] T. Held and M. Westner, "IT Business Value Measurement and Communication among German CIOs: A Conceptual Framework," in *2022 IEEE 24th Conference on Business Informatics (CBI)*, Amsterdam, Netherlands, 2022, pp. 146–155.

[16] S. Mitra, V. Sambamurthy, and G. Westerman, "Measuring IT performance and communicating value," *MIS Quarterly Executive*, vol. 10, no. 1, 2011.

[17] S. De Haes, D. Gemke, J. Thorp, and W. Van Grembergen, "KLM's Enterprise Governance of IT Journey: From Managing IT Costs to Managing Business Value," *MIS Quarterly Executive*, vol. 10, no. 3, 2011.

[18] B. van Ruler, "Communication Theory: An Underrated Pillar on Which Strategic Communication Rests," *International Journal of Strategic Communication*, vol. 12, no. 4, pp. 367–381, 2018, doi: 10.1080/1553118X.2018.1452240.

[19] S. de Haes, T. Huygh, A. Joshi, and W. Van Grembergen, "Adoption and Impact of IT Governance and Management Practices," *International Journal of IT/Business Alignment and Governance*, vol. 7, no. 1, pp. 50–72, 2016, doi: 10.4018/IJITBAG.2016010104.

[20] ISACA, *COBIT 2019®: Framework: Introduction and Methodology*. Schaumburg: ISACA, 2018.

[21] R. Kohli and V. Grover, "Business Value of IT: An Essay on Expanding Research Directions to Keep up with the Times," *J. Assoc. Inf. Syst.*, vol. 9, p. 1, 2008.

[22] S. de Haes, Ed., *Enterprise Governance of Information Technology: Achieving Alignment and Value in Digital Organizations,* 3rd ed. Cham: Springer International Publishing AG, 2020.

[23] R. Kohli and S. Devaraj, "Realizing the Business Value of Information Technology Investments: An Organizational Process," *MIS Quarterly Executive*, vol. 3, no. 1, 2004.

[24] Nils Urbach, Arne Buchwald, and Frederik Ahlemann, "Understanding IT Governance Success And Its Impact: Results From An Interview Study," in *ECIS 2013 Proceedings*, Utrecht, The Netherlands, 2013, P.55.

[25] A. Buchwald, N. Urbach, and F. Ahlemann, "Business value through controlled IT: toward an integrated model of IT governance success and its impact," *J Inf Technol*, vol. 29, no. 2, pp. 128–147, 2014, doi: 10.1057/jit.2014.3.

[26] Y. E. Chan and B. H. Reich, "IT alignment: what have we learned?," *J Inf Technol*, vol. 22, no. 4, pp. 297–315, 2007, doi: 10.1057/palgrave.jit.2000109.

[27] S. Q. Njanka, G. Sandula, and R. Colomo-Palacios, "IT-Business Alignment: A Systematic Literature Review," *Procedia Computer Science*, vol. 181, pp. 333–340, 2021, doi: 10.1016/j.procs.2021.01.154.

[28] S. de Haes, W. Van Grembergen, A. Joshi, and T. Huygh, "Enterprise Governance of IT, Alignment, and Value," in *Management for Professionals, Enterprise Governance of Information Technology: Achieving Alignment and Value in Digital Organizations*, S. de Haes, Ed., 3rd ed., Cham: Springer International Publishing AG, 2020, pp. 1–13.

[29] J. Luftman, "Assessing Business-IT Alignment Maturity," *CAIS*, vol. 4, 2000, doi: 10.17705/1CAIS.00414.

[30] H. A. Smith and J. D. McKeen, "From Technology to Value: The Perennial IT Challenge," Queen's University Canada, 2020. Accessed: Sep. 20 2022. [Online]. Available: https://smith.queensu.ca/_templates/documents/it-forum/technology-to-value.pdf

[31] P. Love, Z. Irani, and R. Fulford, "Understanding IT Costs: An exploratory study using the structured case method," in *PACIS 2003 Proceedings*, Adelaide, South Australia, 2003, P. 45.

[32] S. Mohamed and Z. Irani, "Developing taxonomy of information system's indirect human costs," in *2nd International Conference on Systems Thinking in Management*, Manchester, UK, 2002.

[33] S. Bartsch, *Ein Referenzmodell zum Wertbeitrag der IT.* Zugl.: Marburg, Univ., Diss., 2014. Wiesbaden: Springer Vieweg, 2015.

[34] H. A. Smith and J. D. McKeen, "Developments in Practice VII: Developing and Delivering the IT Value Proposition," *CAIS*, vol. 11, 2003, doi: 10.17705/1CAIS.01125.

[35] A. Bryman and E. Bell, *Business research methods,* 3rd ed. Oxford: Oxford Univ. Press, 2011.

[36] M. Gibson and D. Arnott, "The Use of Focus Groups in Design Science Research," in *ACIS 2007 Proceedings*, 2007.

[37] G. Toppenberg, "Expanded Understanding of IS/IT Related Challenges in Mergers and Acquisitions: Methods & Research Context," in *ECIS 2015 Proceedings 2015*, P.182.

[38] J. Merhout, D. Havelka, and T. Rajkumar, “Determining Factors that Lead Students to Study Information Systems using an Alumni Focus Group,” in *ACIS 2016 Proceedings*, 2016.

[39] R. A. Krueger and M. A. Casey, *Focus groups: A practical guide for applied research,* 5th ed. Los Angeles, London, New Delhi, Singapore, Washington DC: SAGE, 2015.

[40] J. M. Corbin and A. L. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory,* 4th ed. Thousand Oaks, CA: Sage Publications, Inc, 2015.

[41] M. Saunders, *Research Methods for Business Students,* 8th ed. Harlow, UK: Pearson, 2019.

[42] F. Ahmad, W. A. Abd Ghani, and N. H. Arshad, “Ishikawa diagram of critical factors for information technology investment success: A conceptual model,” in *Proceedings of The 4th International Conference on Information Systems Management and Evaluation ICIME 2013*, 2013, p. 27.

[43] L. Luca, “Success factors for R & D projects,” in *MATEC Web of Conferences*, 2018, p. 7001.

[44] J. Peppard and J. Ward, “‘Mind the Gap’: diagnosing the relationship between the IT organisation and the rest of the business,” *The Journal of Strategic Information Systems*, vol. 8, no. 1, pp. 29–60, 1999, doi: 10.1016/S0963-8687(99)00013-X.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 689–697
DOI: 10.15439/2023F4564
ISSN 2300-5963 ACSIS, Vol. 35

An Enhancement of Reinforcement Learning by Scheduling with Learning Effects

Radosław Rudek
General Tadeusz Kościuszko Military University of Land Forces
Czajkowskiego 109, 51-147 Wrocław, Poland
Email:rudek.radoslaw@gmail.com

***Abstract*—This paper present results, which reveal that approaches obtained for scheduling problems with learning effects can be successfully used to improve the quality of machine learning methods. It is illustrated by modelling some aspects of Q-learning agents as scheduling problems with the learning effect, and constructing sequencing and dispatching algorithms, which take into account the existence of learning. Their application to determine the sequence of tasks processed by Q-learning agents can visibly speed up their convergence to an optimal strategy. Furthermore, we show that a dispatch of tasks according to the longest processing time algorithm for parallel computing can be replaced by a more efficient procedure, if agents can learn. The numerical analysis reveals that our approach is efficient, robust and only marginally dependents on a learning model and an accurate approximation of task processing times.**

I. Introduction

THE LEARNING effect takes place in typical human activity environments or in automatized manufacturing, where a human support for machines is needed during activities such as operating, controlling, setup, cleaning, maintaining, failure removal, etc. It was also observed that performances (objectives) of an industrial system can be essentially improved if the learning ability is utilized (see [1]). It can be done by determining the sequence of processed jobs, which takes into consideration not only the given objectives, but also the presence of learning, i.e., decreasing processing times/costs of jobs. Thereby, time/cost objectives (e.g., the maximum completion time) can be additionally improved (in a specified range) by the sequence (schedule) of jobs (e.g., [2], [3], [4]). In other words, additional benefits from learning can be gained. It is worth highlighting this scheduling approach does not interfere a system nor require any changes of its structure. Therefore, it is a significant advantage, which makes this non-invasive method universal and applicable to improve (optimize) different systems, where learning is present and a sequence of processed jobs can be (at least partially) controlled.

Although scheduling problems with the learning effect have attracted particular attention in research society (e.g., [1], [5], [6], [7], but surprisingly, there was no attempt to apply them to enhance machine learning algorithms, except our idea. However, from the perspective of the discussed utilization of learning by scheduling, there is no difference if the reduction in time or cost required to process a job is the result of human learning or machine learning. Thereby, the mentioned additional benefits from learning can be also gained for dynamically changing systems, which use machine learning. Such illustrative example will be shown in this paper.

Machine learning methods or intelligent agents ([8]) very often act like human, especially in the context of adaptation, self-improvement and autonomous learning, which is present not only during training stages, but first and foremost during their regular exploitation (working stages). In particular, it refers to reinforcement learning algorithms (RL), where an autonomous agent can learn to choose actions in an environment to achieve its goals (see [9], [10]). Such behavior results in robustness and high (increasing) efficiency of these approaches, thereby they have attracted particular attention in different domains (e.g., [11]).

Although there are lots of studies devoted to this group of algorithms, they usually focus on improving them in fields such as learning policies, representation of states, exploration and exploitation strategies and others (e.g., [9], [10]). However, there is lack of research on methods that are able to additionally increase the efficiency of RL by the utilization of its learning ability, without interfering a structure of an algorithm. Therefore, as the preliminary result in this domain, we will show that the speed of convergence of RL can be visibly improved by processing tasks according to a schedule that takes into account the existence of learning (an iterative improvement). It is depicted on the example of finding the shortest path in 2D mesh topology environment by RL. We choose Q-learning based algorithm, which is comprehensible and representative, since the considered relations also hold for other techniques, e.g., temporal difference learning (TD), state-action-reward-state-action (SARSA) (for more details see [10]).

In this paper, we model some aspects of Q-learning as scheduling problems with the learning effect. On this basis, the efficiency of the algorithm can be improved if tasks are processed according to a given sequence following from the analysis of scheduling problems. Furthermore, we show that a dispatch of tasks according to the longest processing time (LPT) algorithm (see [12]) for parallel computing can be replaced by a more efficient strategy, if agents can learn. Our approach does not interfere a structure of machine learning methods and its computation overhead is negligible. It is showed to be efficient, robust and only marginally dependents on a learning model and an accurate approximation of values of task processing times. It is especially crucial for the application of reinforcement learning methods in varying environments, where they should adapt quickly, whereas deterministic algorithms cannot be used. Therefore, the contribution of this paper is not to make new better learning agents, but to reveal that approaches obtained for scheduling problems with learning effects can be successfully used to improve the quality of machine learning methods. The presented promising preliminary results open new perspectives for the application of scheduling theory.

The remainder of this paper is organized as follows. The illustrative application of the reinforcement learning is given in the next section, which includes the formal definition of the shortest path problem and the Q-learning algorithm dedicated to solve it. Next, the considered issue is expressed as scheduling on parallel processors with the learning effect and on this basis scheduling (dispatching) algorithms are proposed. Their application to enhance the Q-learning algorithm is analysed numerically. Finally, the last section concludes the paper.

II. Reinforcement learning

In this section, we will describe an environment, which is used to illustrate and analyse the application of the algorithms constructed for scheduling with the learning effect to improve the efficiency of reinforcement learning methods.

A. The shortest path problem

The shortest path problem is well known and its deterministic cases can be solved *inter alia* by Dijkstra's algorithm or A* search. However, it also constitutes an excellent environment for comprehensive analysis of various other methods also with learning effects (see [13]). It will be used for a similar purpose in this paper.

There is given a graph $G = (V, E)$, where V is the set of vertices (nodes) and E is the set of edges (links). Each edge (u, w) has an associated length (weight) $l(u, w)$, where $u, w \in V$; G is undirected, thereby $l(u, w) = l(w, u)$. We consider the 2D mesh topology, interconnecting in a grid fashion, where each node has at most four neighbors, thus, the following notation can be used to describe nodes: $V' = \{(x_u, y_u) : 1 \leq x_u \leq X \wedge 1 \leq y_u \leq Y, u \in V\}$ and X and Y are the number of vertical and horizontal nodes in the mesh, respectively; such mesh is called $X \times Y$ size. The objective is to find the shortest path (SP) between each of n source nodes from the set $S = \{s_1, \ldots, s_n\} \subset V$ to one destination node $d \in V$ ("hot spot").

B. Q-learning

In this section, we will briefly describe a reinforcement learning algorithm that is based on a typical Q-learning method (see [10], [14]). For mode details concerning similar applications of Q-learning see [15] or [16]. The discussed algorithm is a model free reinforcement learning technique, which was chosen for the purpose of this paper, since it is transparent for analysis and the idea behind our approach can be clearly illustrated. Its behaviour will be depicted on the example of solving in an adaptive manner the shortest path problem. On this basis, we will show that properties obtained for a scheduling problem with the learning effect can be applied to improve the efficiency of Q-learning.

In the considered approach, the Q-learning agent on the basis of the current node (say u) and the state-action function Q chooses the next node (say w), subsequently it receives the reward depending on the distance between these nodes. This process is repeated until the destination node (say d) is reached. The Q-function is represented by a set $\{Q_1, \ldots, Q_u, \ldots Q_{|V|}\}$ of $|V|$ tables, further called Q-tables. The quality of a state-action function (table) $Q_u(d, w)$ defined for each node $u \in V$ is the total expected discounted reward received by selecting (in node u) the next node w on the way to the destination node d. Using the RL nomenclature, the state is represented by the pair $[u, d]$, whereas w is the action taken in this state.

The determination of a next node and the update of the Q-function for each current node $u \in V$ proceed according to the following steps:

1) The next node w is determined according to a greedy strategy that always chooses a node with the highest Q-value, i.e.,
$$w = \arg \max_{w' \in \mathcal{N}(u)} \{Q_u(d, w')\},$$
where u is the current node, d is the destination node and $\mathcal{N}(u)$ is the set of neighbor nodes of u. The greedy strategy is simple and transparent, but together with the optimistic initialization of Q-tables with zeros and the application of negative rewards, it still drives exploration.
2) The Q-value for node u is updated according to the following rule:
$$\begin{aligned} Q_u(d, w) &= (1 - \alpha)Q_u(d, w) + r(u, w) \\ &+ \gamma \max_{a'}\{Q_w(d, a')\}, \end{aligned}$$
where $\alpha \in [0, 1]$ is a learning rate, $\gamma \in [0, 1]$ is a discount factor, $r(u, w) = r(w, u) = -l(u, w)$ is the reward equal to the negative value of the length between nodes u and w and $\max_{a'}\{Q_w(d, a')\}$ is the best expected Q-value in node w to reach the destination node d. In a typical Q-learning implementation, terms $[r(u, w) + \gamma \max_{a'}\{Q_w(d, a')\}]$ are multiplied by α. Although it is omitted in the presented approach to avoid potential deadlocks, Q-values is convergent (since $r < 0$).
3) To improve learning the forward update of Q-value of node w is proceeded:
$$\begin{aligned} Q_w(d, u) &= (1 - \alpha)Q_w(d, u) + r(u, w) \\ &+ \gamma \max_{a' \in \mathcal{N}(u)}\{Q_u(d, a')\}. \end{aligned}$$

The above procedure is applied starting from the source node $s_j \in S$ until the destination node d is reached (i.e., starting u is equal to s_j). It is repeated for the given pair (s_j, d) until the same value of the path is obtained for the given number of succeeding iterations (denoted by $TerminateCondition$). This process of finding the shortest path for (s_j, d) will be called task j; for convenience, we will also use the following notation $j \in S \equiv s_j \in S$. Since in the considered example, the destination node d is the same for all tasks, then it does not need to be implemented a part of a state.

It can be observed that the application of distributed or parallel computing can improve efficiency of algorithms and it is commonly used nowadays, therefore, we present a parallel version of the considered approach. Thus, a set $A = \{A_1, \ldots, A_m\}$ of m Q-learning agents are applied. Each agent A_i calculates the shortest paths for the source nodes (tasks) from the set $S_i \subset S$, where S_i are disjoint sets such that $S_1 \cup \ldots \cup S_m = S$. Due to relatively long latency access to shared Q-tables (caused by mutexes, communication, etc.) in reference to very fast calculation times of the Q-values, each agent has its own set of Q-tables. The formal description of the Q-learning agent is given by Algorithm 1.

The objective of the Q-learning agents is not only to find the shortest paths for all given nodes, but to minimize the time of finding them. Let t_j be the processing time of task j, i.e., time of calculating the shortest path from the source node s_j to the destination node d, whereas $C(A_i) = \sum_{j \in S_i} t_j$ be the calculation time taken by Q-learning agent A_i to find shortest paths for all nodes from the related set S_i, i.e., to process tasks from this set. Thus, the time objective is expressed as the minimization of calculation time of the shortest paths for all source nodes (processing of all tasks), which is the maximum calculation time $t_{\max}$ among all Q-learning agents: $t_{\max} = \max_{i=1,\ldots,m}\{C(A_i)\}$.

Since there are multiple agents, then the calculations are distributed among them, i.e., the set S of source nodes (tasks) is partitioned into subsets $\{S_1, \ldots, S_i, \ldots, S_m\}$, which are assigned to related agents $\{A_1, \ldots, A_i, \ldots, A_m\}$. Thus, an assignment algorithm has an essential impact on the minimization of the objective value $t_{\max}$. To construct such methods, which are efficient, we will express the considered problem as scheduling on identical parallel processors. On the basis of its properties, we will propose scheduling algorithms, which will be applied to handle calculations (tasks) by Q-learning agents (i.e., to determine the assignment of calculations and their processing sequences).

Furthermore, we will show that calculations can be speeded up (i.e., $t_{\max}$ can be further minimized) if the allocation algorithms takes into consideration not only the potential time required to find a solution, but also learning (an iterative improvement), which is an inner nature of Q-learning.

III. Scheduling with learning effects

At first, we will present the general concept how the time minimization of finding the shortest paths by the Q-learning agents can be perceived as scheduling on parallel identical processors or in other words parallel machine scheduling. Next, the related scheduling problem will be formally defined.

Algorithm 1 Q-learning agent A_i

```
Determine the set of the source nodes $S_i \subseteq S$
    and their processing sequence
Initialize $Q-tables$: $\{Q_1, \ldots, Q_{|V|}\}$
for each $s_j \in S_i$ do
    $distance := \infty$
    $distance^* := \infty$
    $distance_{prev} := \infty$
    $counter := 0$
    $s := s_j$
    while $counter < TerminateCondition$ do
        $distance := 0$
        $counter := counter + 1$
        $u := s$
        while $u \neq d$ do
            $w := \arg\max_{w' \in \mathcal{N}(u)}\{Q_u(d, w')\}$
            $distance := distance + l(u, w)$
            $r(u, w) := -l(u, w)$,  $r(w, u) := -l(w, u)$
            $Q_u(d, w) := (1 - \alpha)Q_u(d, w) + r(u, w)$
                        $+\gamma \max_{w' \in \mathcal{N}(w)}\{Q_w(d, w')\}$
            if $w \neq d$ then
                $Q_w(d, u) := (1 - \alpha)Q_w(d, u) + r(w, u)$
                        $+\gamma \max_{u' \in \mathcal{N}(u)}\{Q_u(d, u')\}$
            end if
            $u := w$
        end while
        if $distance < distance^*$ then
            $distance* := distance$
        end if
        if $distance \neq distance_{prev}$ then
            $counter := 0$
        end if
        $distance_{prev} := distance$
    end while
end for
$\{Q_1, \ldots, Q_{|V|}\}$ are the $Q-tables$ containing strategy
    to find the shortest paths for
    the considered pairs $(s_j, d)$, $s_j \in S_i$
```

Each Q-learning agent can be represented in a scheduling problem by a processor, i.e., $A_i \equiv P_i$, whereas a task (say j) that is finding the shortest path (such that $TerminateCondition$ is true) from a source node (say $s_j \in S$) to the destination node d by a Q-learning agent is called a job (also j).

The processing time t_j of task j, i.e., the time required to find (calculate) the shortest path (s_j, d), can depend on the number of nodes from a source node to the destination node (including). Hence, we model it by the processing time p_j of job j, which is given as follows:

$$p_j = |x_d - x_{s_j}| + |y_d - y_{s_j}|, \qquad \forall s_j \in S. \tag{1}$$

Note that p_j is not an exact value of the real processing time t_j required to find the related shortest path, but it only models this parameter in scheduling domain, i.e., it should reflect the relations such that $t_j < t_k$ implies $p_j < p_k$.

For a better comprehension, we will analyse the following computational example.

Example 1

Let us verify the accuracy of model (1) for the following settings generated from the uniform distribution: graph $G(V, E)$ of size 100×100, lengths of edges $l(u, w) \in \{1, \ldots, 5\}$, where $(u, w) \in E$, 100 random source nodes $s_j = (x_{s_j}, y_{s_j})$, where $x_{s_j} \in \{2, \ldots, 99\}$, $y_{s_j} \in \{2, \ldots, 99\}$, and destination node $d = (x_d, y_d) = (99, 99)$. On this basis, we ran implemented Q-learning algorithm and measured (in milliseconds) the time of finding the shortest path by a single agent from a source node s_j to the destination node d, i.e., processing time t_j of task j. Such analysis was done for each source node $s_j \in S$, but to measure relevant times t_j, Q-tables were cleared before processing each task. Thus, each of them was processed as the only task by a Q-learning agent. We refer each measured t_j to the modelled value – the job processing time p_j – obtained according to (1). The result of this analysis is shown in Figure 1, where related pairs (p_j, t_j) are presented according to the non-decreasing order of p_j. It can be seen that there are some bias such that $t_j < t_k$ not always refers to $p_j < p_k$, but the general tendency holds as required (and expected).

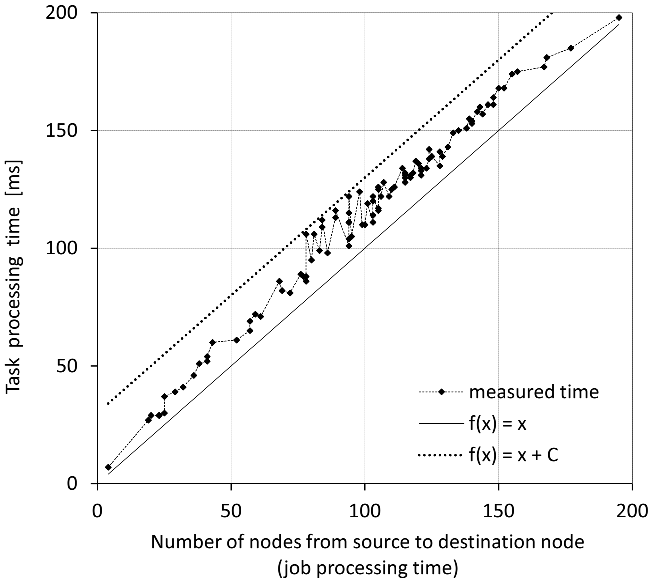


Fig. 1: Relations between measured task processing times t_j and modelled job processing times p_j

It can be seen in Figure 1 that model p_j approximates measured times t_j (in a constant range) if an agent processes only one task. However, due to the learning ability of an agent, the time required to perform a task decreases with the number of previously processed tasks. Let us consider the following example.

Example 2

Given a task related with node $s_1 = (x_{s_1}, y_{s_1}) = (2, 2)$, for which its processing times are measured (in milliseconds) depending on the number of previously processed tasks. Namely, an agent performs tasks from the following groups (1), $(2, 1)$, $(2, 3, 1)$, ..., $(2, 3, \ldots, 100, 1)$ such that $j = 1$ is always processed as the last and Q-tables are cleared before a next group is processed. Other settings are the same as in the earlier example. The result of the analysis is shown in Figure 2, which illustrates a learning curve of an agent, which is similar to learning curves observed in industrial systems (see [17], [18]). The model of decreasing task processing times (learning curve) will be proposed subsequently.

On the basis of the above observations, let us define formally the related scheduling problem. There are given a set $J = \{1, \ldots, n\}$ of n jobs that model tasks S and a set $P = \{P_1, \ldots, P_m\}$ of m identical processors referring to Q-learning agents $\{A_1, \ldots, A_m\}$. Each processor is continuously available and can process at most one job at a time. Once it begins processing a job it continues until this job is finished and there are no precedence constraints between jobs. Each job j is characterized by the processing time $\tilde{p}_j(v_i)$ dependent on the number of previously processed jobs v_i by processor P_i. It models a variable task processing time (see Figure 2), which depends (due to learning) on the number v_i of previously processed tasks by agent A_i. In general, it can be describe by the following:

$$\tilde{p}_j(v_i) = p_j \cdot f(v_i), \tag{2}$$

where p_j is the normal processing time of job j defined as the job processing time of a job without influence of learning, which is calculated on the basis of source and destination nodes related with a task and given by (1); it models a processing time of a task, which is processed by an agent as the first one, i.e., without influence of learning (see Figure 1). Moreover, a function $f(v_i)$ is a learning curve that describes decreasing processing times dependent on the number of processed jobs v_i (see Figure 2).

Note that the only assumption concerning f is that the function is non-increasing for the considered scheduling model, but it is only a model, which does not have to precisely describe the decreasing task processing times. Nevertheless, we will show that even such imprecise and general model (following (1) and (2)) is sufficient to derive properties, which allow us construct a task assignment algorithm that improves Q-learning.

The schedule of jobs can be unambiguously defined by their sequences $\pi = \{\pi_1, \ldots, \pi_i, \ldots, \pi_m\}$, where π_i denotes the sequence of jobs on P_i and $\pi_i(k)$ is the index of the kth job in π_i for $i = 1, \ldots, m$. Moreover, n_i is the number of jobs assigned to P_i, i.e., the cardinality of π_i. Note that π_i refers to the sequence of tasks from set S_i assigned to be processed by agent A_i. For each job $\pi_i(k)$, we can determine its completion

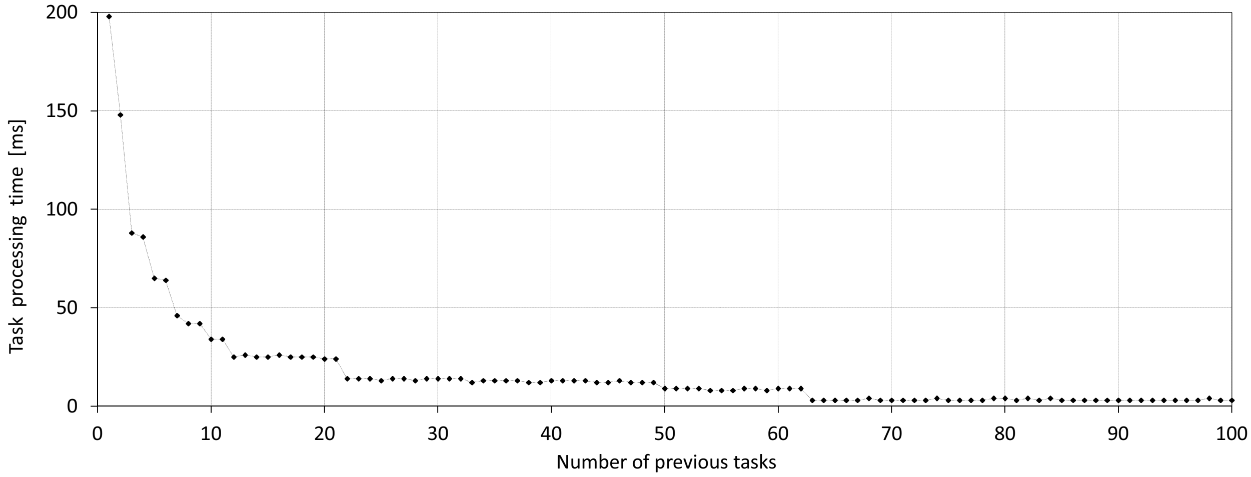


Fig. 2: A measured task processing time depending on the number of previously processed tasks

time $C^{(i)}_{\pi_i(k)}$ on processor P_i:

$$C^{(i)}_{\pi_i(k)} = \sum_{k=1}^{n_i} \tilde{p}_{\pi_i(k)}(k). \quad (3)$$

On this basis $C^{(i)}_{\pi_i(n_i)}$ models the calculation time $C(A_i)$ taken by Q-learning agent A_i to find shortest paths for all nodes from the related set S_i, i.e., to process tasks from this set. Since the objective of Q-learning agents is to minimize the time of finding all shortest paths $t_{\max}$, then the criterion value (we use a similar symbol) for the scheduling problem is defined by $C_{\max}(\pi) = \max_{i=1,\ldots,m}\{C^{(i)}_{\pi_i(n_i)}\}$.

Formally, the objective is to find such a schedule $\pi^* = \{\pi^*_1, \ldots, \pi^*_m\}$ of jobs (i.e., assignment of jobs to processors and their sequences) that minimizes the maximum completion time (makespan):

$$\pi^* \triangleq \text{minarg}_{\pi\in\Pi}\left\{\max_{i=1,\ldots,m}\left\{C^{(i)}_{\pi_i(n_i)}\right\}\right\}, \quad (4)$$

where Π is the set of all schedules π.

For convenience and to keep an elegant description of the considered problem we will use the three field notation scheme $X \mid Y \mid Z$ (see [19]), where X describes the processor environment, Y describes job characteristics and constraints and Z represents the minimization objectives. According to this notation the scheduling problem with model (2) will be denoted as $Pm|LE|C_{\max}$, whereas its case with constant job processing times ($\tilde{p}_j(v_i) = p_j \ \forall(j, v_i)$) will be denoted by $Pm||C_{\max}$.

On the basis of the formulated scheduling problem, we will propose algorithms that determine the assignment of tasks to agents and their processing sequences.

IV. Job scheduling algorithms

In this section, we focus on the job scheduling problem with the learning effect. The considered criterion is the minimization of the maximum job completion time (makespan). In the classical version of this problem $Pm||C_{\max}$, the job processing times are constant, thereby their sequence does not affect the criterion value, and thus, the objective is to find the allocation of jobs to the processors to minimize the makespan.

Let us analyse job assignment (dispatching) algorithms for the problem $Pm||C_{\max}$. One of the primary and efficient methods is a list scheduling, which assigns jobs to the first available processor according to an order defined by a list (see [12]). If the sequence of jobs on the list is random, then the worst case of the algorithm (further denoted by RND) is $\frac{C_{\max}(RND)}{C^*_{\max}} = 2 - \frac{1}{m}$ and its computational complexity is $O(nm)$ (see [20], [12]), where $C^*_{\max}$ denotes an optimal criterion value. However, its efficiency can be improved if jobs are assigned according to their longest processing times (LPT), then its worst case is $\frac{C_{\max}(LPT)}{C^*_{\max}} \le \frac{4}{3} - \frac{1}{3m}$, whereas its computational complexity is $O(n\log n + mn)$ (see [12]).

Note that RND can be straightforwardly applied to the problem of finding shortest paths by Q-learning agents without any definition of a scheduling problem nor any model of task processing times. It only assigns each task to the first available agent. On the other hand, the application of LPT requires a model of task processing times, which has to be sufficient only to determine the non-decreasing relation between tasks during the assignment process.

However, unlike a typical approaches to parallel computations, we will reveal that in case of learning algorithms (such as Q-learning), not only the assignment of tasks is important, but their processing sequences by an agent seem to be essential. It will be shown on the basis of the formulated scheduling problem with learning. At first, we will provide a property for a single processor case $1|LE|C_{\max}$ (referring to one Q-learning agent), which holds for the problem with different learning models (e.g., [5], [6], [7]).

Property 1: The problem $1|LE|C_{\max}$ can be solved optimally by scheduling jobs according to their shortest normal processing times p_j (SNPT rule).

If job processing times are constant, then the sequence of jobs is immaterial for the makespan minimization on a single processor $1||C_{\max}$. However, following Property 1, it is no longer valid if job processing times can vary. This is a premise for a more efficient processing of tasks by a Q-learning agent, which will be analysed numerically in the next section. Furthermore, it can be easily extend to a scheduling on parallel processors (tasks processed by multiple agents).

Property 2: There exists an optimal solution to the considered problem $Pm|LE|C_{\max}$ such that jobs on each processor are scheduled according to their shortest normal processing times p_j (SNPT).

Based on Property 2, we propose a list scheduling algorithm, where jobs on the list are sequenced according to their shortest processing times (further denoted by SNPT). Such conclusion is not only opposite to the popular and efficient LPT algorithm often used for parallel computing ([12]), but it does not follow from any other combined analysis on machine learning algorithms and parallel computing.

It is worth noticing that our theoretical analysis showed that SNPT rule overwhelms LPT in the domain of scheduling problems with the learning effect. It is completely against the existing approaches to dispatch tasks for parallel learning algorithms according to LPT or RND (randomly).

Recall that a job is equivalent to a task, thereby a schedule π for jobs (obtained on the basis of LPT, RND or SNPT) straightforwardly determines the assignments (dispatching) of tasks to Q-learning agent and their processing sequences by each agent.

V. Numerical analysis

In this section, we will use scheduling algorithms LPT, RND and SNPT in the domain of the Q-learning and the shortest path problem. Namely, we will analyse how Q-learning agents can improve their efficiency (running times) if they process tasks according to sequences determined by the application of scheduling algorithms LPT, RND and SNPT. In other words, we run implemented Q-learning agents and measure (in milliseconds) the real time of finding the shortest paths depending on different sequences of processed tasks (i.e., LPT, RND, SNPT).[1] The impact of scheduling algorithms on running times of Q-learning is analysed for the following settings.

2D-mesh (environment):

- mesh (graph) size $X \times Y$: 100×100,
- lengths (weights) of links between nodes $l(u, w)$: generated from the uniform distribution over the integers in the following ranges of values $\{1, \ldots, 5\}$ and $\{1, \ldots, 100\}$, where $u, w \in V$,
- destination node d ("hot spot"): $(x_d, y_d) = (X-1, Y-1)$,
- size $|S|$ of the set S of source nodes (number of tasks): 1000, 2000 and 4000,
- source node coordinates (x_{s_j}, y_{s_j}): $x_{s_j} \in \{2, \ldots, X-1\}$, $y_{s_j} \in \{2, \ldots, Y-1\}$, $s_j \in V$, where $j = 1, \ldots, |S|$.

Q-learning (agent):

- $m \in \{1, 2, 10, 20\}$ agents, $\alpha = 0.9$, $\gamma = 0.9$, $TerminateCondition = 5$,
- the applied Q-learning always provided optimal solution (the shortest path) for the considered settings, which was verified by Dijkstra's algorithm.

Job scheduling algorithms:

- $m \in \{1, 2, 10, 20\}$ processors,
- job processing times (the model of task processing times) $p_j = |x_d - x_{s_j}| + |y_d - y_{s_j}|$ for $j = 1, \ldots, |S|$,
- algorithms LPT, RND, SNPT.

For each combination of the defined mesh parameters, a set I_Q of random instances (replications) were generated, where its cardinality is equal to $|I_Q| = 100$. On their basis, the impact of scheduling algorithms on running times of Q-learning is evaluated (as percentage speed up δ) in reference to the running times obtained with LPT as follows for each instance I

$$\delta(TS(I)) = \frac{t_{\max}(LPT(I)) - t_{\max}(TS(I))}{t_{\max}(TS(I))} \times 100\%,$$

where $TS \in \{LPT, RND, SNPT\}$ and $t_{\max}(TS(I))$ is the measured (in milliseconds) maximum calculation time (running time) among all Q-learning agents (equivalent to the makespan) that processed tasks according to the applied task scheduling algorithm TS for the instance $I \in I_Q$.

Thus, for each instance I, the running times (in milliseconds) of finding the shortest paths by Q-learning agents are measured, which processed tasks according to LPT, RND and SNPT, respectively. The results concerning mean, minimum, and maximum speed up δ (in percents) and running times $t_{\max}$ (in milliseconds) of Q-learning are given in Table I, where the related are calculated for each set I_Q of instances as follows: $\delta(TS, min) = \min_{I \in I_Q}\{\delta(TS(I))\}$, $\delta(TS, max) = \max_{I \in I_Q}\{\delta(TS(I))\}$, $\delta(TS, mean) = \sum_{I \in I_Q} \delta(TS(I))/|I_Q|$, and $t_{\max}(TS, min) = \min_{I \in I_Q}\{t_{\max}(TS(I))\}$, $t_{\max}(TS, max) = \max_{I \in I_Q}\{t_{\max}(TS(I))\}$, $t_{\max}(TS, mean) = \sum_{I \in I_Q} t_{\max}(TS(I))/|I_Q|$. Note that the higher value of δ then better, whereas it is oppositive for $t_{\max}$. Moreover, only mean values of δ and $t_{\max}$ are correlated, whereas min and max are informative. For a better comprehension, let us consider the first row of Table I for minimum values of RND. We have the minimum speed up $\delta = -2\%$ and minimum running time $t_{\max} = 204$ ms of Q-learning. However, -2% does not refers to 204, it only means that there was a running time of Q-learning using RND for which the minimum speed up comparing to the approach using LPT was -2%, i.e., LTP caused a result 2% faster than RND. On the other hand, minimum $t_{\max} = 204$ means that the smallest running time of Q-learning using RND was 204 ms.

In the previous studies on machine learning, a sequence of processing tasks has not been taken into account as an

[1] Q-learning and scheduling algorithms were coded in C++ and simulations were run on PC, CPU Intel® Core™i7-2600K 3.40 GHz and 8GB RAM.

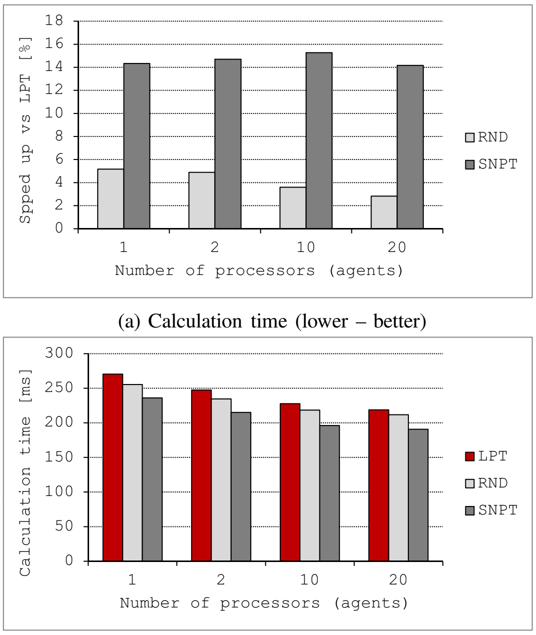
TABLE I: Speed up and running times of Q-learning depending on sequences determined by scheduling algorithms in reference to results obtained by LPT

n	m	$l(u,w)$		LPT			RND			SNPT		
				mean	min	max	mean	min	max	mean	min	max
500	1	5	$\delta[\%]$	0	0	0	5	-1	11	18	9	22
			$t_{\max}$	[225]	[220]	[238]	[213]	[205]	[238]	[192]	[190]	[206]
		100	$\delta[\%]$	0	0	0	5	0	9	14	10	16
			$t_{\max}$	[225]	[218]	[231]	[212]	[201]	[226]	[196]	[192]	[198]
	2	5	$\delta[\%]$	0	0	0	4	0	9	15	11	17
			$t_{\max}$	[221]	[217]	[224]	[208]	[201]	[220]	[192]	[189]	[195]
		100	$\delta[\%]$	0	0	0	5	-1	10	14	11	16
			$t_{\max}$	[215]	[210]	[218]	[204]	[198]	[217]	[188]	[185]	[194]
	10	5	$\delta[\%]$	0	0	0	4	0	9	13	8	17
			$t_{\max}$	[209]	[204]	[216]	[202]	[196]	[211]	[184]	[181]	[195]
		100	$\delta[\%]$	0	0	0	4	-3	9	15	9	19
			$t_{\max}$	[228]	[222]	[237]	[220]	[211]	[231]	[198]	[194]	[211]
	20	5	$\delta[\%]$	0	0	0	2	-1	8	12	8	17
			$t_{\max}$	[213]	[209]	[222]	[206]	[196]	[216]	[189]	[184]	[195]
		100	$\delta[\%]$	0	0	0	3	-1	12	14	11	22
			$t_{\max}$	[219]	[213]	[250]	[212]	[203]	[224]	[192]	[188]	[204]
1000	1	5	$\delta[\%]$	0	0	0	6	0	12	17	14	19
			$t_{\max}$	[262]	[257]	[266]	[244]	[236]	[258]	[224]	[220]	[226]
		100	$\delta[\%]$	0	0	0	6	0	10	12	11	16
			$t_{\max}$	[248]	[245]	[253]	[234]	[226]	[248]	[220]	[218]	[224]
	2	5	$\delta[\%]$	0	0	0	5	0	9	15	12	17
			$t_{\max}$	[240]	[234]	[245]	[228]	[219]	[241]	[207]	[205]	[212]
		100	$\delta[\%]$	0	0	0	6	-1	10	16	14	19
			$t_{\max}$	[241]	[238]	[245]	[228]	[219]	[245]	[207]	[206]	[209]
	10	5	$\delta[\%]$	0	0	0	4	-1	9	17	14	20
			$t_{\max}$	[234]	[229]	[242]	[222]	[215]	[236]	[197]	[195]	[202]
		100	$\delta[\%]$	0	0	0	3	-2	8	17	15	19
			$t_{\max}$	[231]	[229]	[237]	[222]	[213]	[235]	[197]	[194]	[199]
	20	5	$\delta[\%]$	0	0	0	3	0	7	13	10	17
			$t_{\max}$	[219]	[215]	[227]	[212]	[206]	[221]	[192]	[190]	[198]
		100	$\delta[\%]$	0	0	0	3	-1	7	15	8	18
			$t_{\max}$	[224]	[221]	[231]	[217]	[209]	[227]	[195]	[191]	[206]
4000	1	5	$\delta[\%]$	0	0	0	5	1	8	14	12	17
			$t_{\max}$	[333]	[331]	[342]	[316]	[307]	[329]	[292]	[288]	[295]
		100	$\delta[\%]$	0	0	0	4	-1	8	11	8	14
			$t_{\max}$	[329]	[325]	[332]	[313]	[303]	[331]	[292]	[290]	[301]
	2	5	$\delta[\%]$	0	0	0	5	0	10	15	12	18
			$t_{\max}$	[286]	[284]	[291]	[272]	[260]	[286]	[249]	[246]	[254]
		100	$\delta[\%]$	0	0	0	4	0	8	13	11	16
			$t_{\max}$	[282]	[279]	[286]	[267]	[258]	[282]	[247]	[244]	[250]
	10	5	$\delta[\%]$	0	0	0	4	0	7	15	13	19
			$t_{\max}$	[238]	[235]	[246]	[227]	[221]	[242]	[204]	[202]	[208]
		100	$\delta[\%]$	0	0	0	3	0	8	14	9	19
			$t_{\max}$	[226]	[224]	[234]	[217]	[212]	[227]	[197]	[195]	[211]
	20	5	$\delta[\%]$	0	0	0	3	0	8	16	12	18
			$t_{\max}$	[218]	[216]	[227]	[212]	[204]	[219]	[188]	[185]	[193]
		100	$\delta[\%]$	0	0	0	3	-1	9	15	12	19
			$t_{\max}$	[219]	[215]	[228]	[211]	[205]	[220]	[188]	[186]	[194]

approach of speeding up these methods (in particular Q-learning). Similarly for computing them in parallel, it might be expected that the most efficient among task assignment (scheduling) algorithms is LPT, next RND and SNPT is the worst, or more likely that the differences in running times of machine learning methods are negligible for different sequences of processed tasks. However, the numerical analysis following our theoretical research reveals that it does not have to be so (in fact it is completely opposite to the popular approach).

It can be seen in Table I that a proper sequence of processed tasks (see SNPT) can significantly speed up Q-learning (even 22% and not less than 8%). The running times of Q-learning $t_{\max}(LPT)$ using LPT are 204–342 ms depending on the number of tasks n and agents m, whereas the impact of values of lengths (weights) $l(u,w)$ is negligible. Similar relations hold for RND and SNPT. The best running times are obtained for Q-learning that processed tasks according to SNPT, which is consistent with the theoretical results. It can be seen that SNPT always speeds up Q-learning, i.e., 11–18% (mean), but at least 8–15% (min) and even 14–22% (max); see also Figure 5. For RND, we have speed up about 2–6% (mean) and 7–12% (max), whereas its minimum speed up of RND in reference to LPT is -3–1% (min), where the negative values mean that for some cases LPT is slightly better.

Let us analyse running times of a single agent and multiple agents. For $m = 1$ and $n = 500$, we have the following average running times $t_{\max}(LPT) = 225$ ms, $t_{\max}(RND) = 213$ ms, $t_{\max}(SNPT) = 192$ ms, thereby SNPT is about 30 ms and 20 ms faster than LPT and RND, respectively. Thus, SNPT is visible better not only than LPT, but also than random sequence. It shows that our approach can significantly speed up Q-learning even for a single agent. On the other hand, for $m = 20$ and $n = 4000$ for which LPT should be (according to other studies) an efficient task assignment method for parallel computing, the mean running times are as follows: $t_{\max}(LPT) = 219$ ms, $t_{\max}(RND) = 211$ ms, $t_{\max}(SNPT) = 188$ ms. Once again, LPT is the worst and a random sequence is better, but they are overwhelmed by SNPT for all analysed instances. It can be seen that the proposed approach is robust, it is slightly affected by the number of agents (Figure 3) or tasks (Figure 4). Moreover, the mean speed up is over 15% in reference event to LPT (well known to be very good algorithm for such cases).

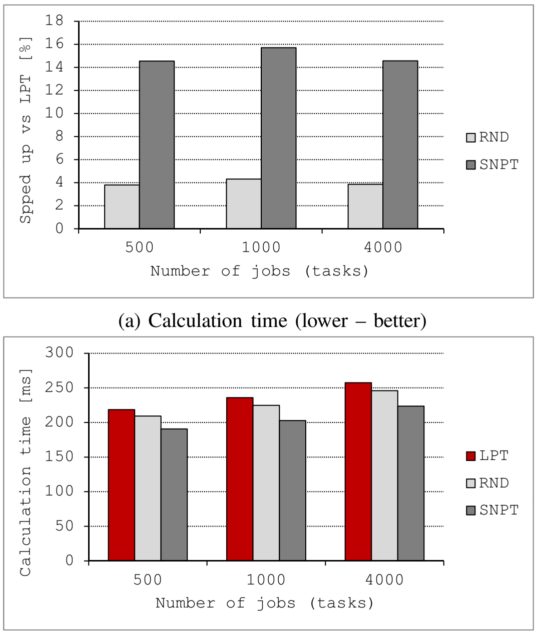


(a) Calculation time (lower – better)

(b) Speed-up vs LPT (higher – better)

Fig. 3: Mean calculation times and speed-ups of algorithms for different numbers of processors (agents); following Table I

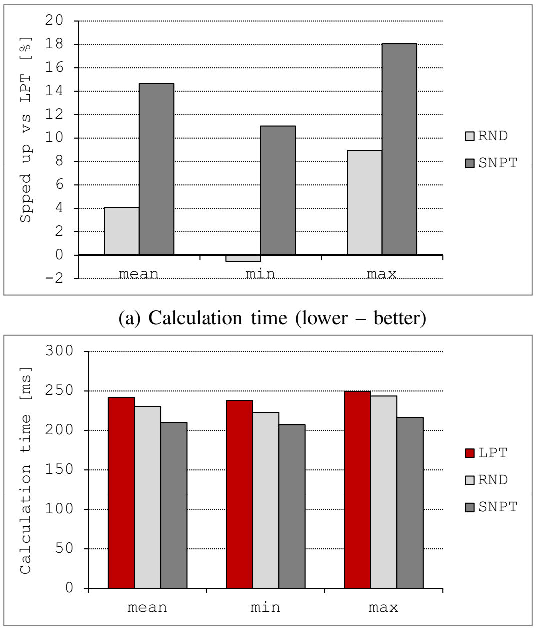
Finally, the time required by considered scheduling algorithms (in particular SNPT) to determine sequences of processed tasks does not exceed 1 ms (even for greater instances $n = 4000$ and $m = 20$), which is negligible. Hence application of our approach significantly speeds up Q-learning (about 15%) without additional effort, it does not interfere Q-learning structure nor cause time overhead. It is especially crucial for varying environments, where Q-learning should adapt quickly. Thus, it can be used in a similar way for other related machine learning methods.



(a) Calculation time (lower – better)

(b) Speed-up vs LPT (higher – better)

Fig. 4: Mean calculation times and speed-ups of algorithms for different numbers of jobs (tasks); following Table I



(a) Calculation time (lower – better)

(b) Speed-up vs LPT (higher – better)

Fig. 5: Mean, minimum and maximum values of calculation times and speed-ups; following Table I

VI. Conclusions

Following dependencies observed in production and manufacturing systems related with a human factor (learning), we modelled relations, which occur between tasks processed by machine learning algorithms. Namely, we presented preliminary results, which revealed that approaches obtained for scheduling problems with learning effects (known from production and manufacturing) can be successfully used to improve the quality of machine learning methods. It was illustrated by modelling some aspects of Q-learning as scheduling problems with the learning effect. The previous studies on machine learning had not taken into consideration that the existence of learning can be utilized and bring additional benefits. On the basis of our approach, we claimed that the efficiency of Q-learning algorithms can be improved (running times) if tasks are processed according to a given sequence. Furthermore, we showed that for a parallelized Q-learning an assignment of tasks by SNPT is significantly more efficient than LPT, which is somehow in opposition to an intuition following from the previous studies on the sole LPT rule without the presence of learning (e.g., [12], [21]). The numerical analysis revealed that the dependency between running times and applied task processing (scheduling) algorithm is not incidental, but it is a rule.

Thus, our approach is efficient and robust, since it does not depend on a learning curve model nor accurate values of the normal job (task) processing times, but requires only their non-decreasing relation. In the same time, it does not interfere a structure of machine learning methods and its computation overhead is negligible. On the other hand, it is also the main limitation of the presented approach, since the SNPT rule cannot gain additional optimization benefits from knowing the exact model of learning curves describing the reduction of task processing times.

Our future research will focus on the application of our approach to other machine learning methods as well as on the analysis of different criteria, thereby development of other efficient algorithms.

Acknowledgement

This work was supported by the Polish National Science Centre under grant no. DEC-2020/37/B/HS4/03235.

References

[1] D. Biskup, “A state-of-the-art review on scheduling with learning effects,” *European Journal of Operational Research*, vol. 188, pp. 315–329, 2008.

[2] R. Rudek, “Scheduling on parallel processors with varying processing times,” *Computers & Operations Research*, vol. 81, pp. 90–101, 2017.

[3] J. Xu, C.-C. Wu, Y. Yin, C. Zhao, Y.-T. Chiou, and W.-C. Lin, “An order scheduling problem with position-based learning effect,” *Computers & Operations Research*, vol. 74, pp. 175–186, 2016.

[4] C. Zhao, J. Fang, T. Cheng, and M. Ji, “A note on the time complexity of machine scheduling with DeJong’s learning effect,” *Computers & Industrial Engineering*, vol. 112, pp. 447–449, 2017.

[5] J. Pei, X. Liu, P. M. Pardalos, A. Migdalas, and S. Yang, “Serial-batching scheduling with time-dependent setup time and effects of deterioration and learning on a single-machine,” *Journal of Global Optimization*, vol. 67, pp. 251–262, 2017.

[6] R. Rudek, “A fast neighborhood search scheme for identical parallel machine scheduling problems under general learning curves,” *Applied Soft Computing*, vol. 113, pp. 108 023.1–16, 2021.

[7] C.-H. Wu, W.-C. Lee, P.-J. Lai, and J.-Y. Wang, “Some single-machine scheduling problems with elapsed-time-based and position-based learning and forgetting effects,” *Discrete Optimization*, vol. 19, pp. 1–11, 2016.

[8] M. I. Jordan and T. M. Mitchell, “Machine Learning: Trends, Perspectives, and Prospects,” *Science*, vol. 349, pp. 255–260, 2015.

[9] I. Grondman, L. Buşoniu, G. Lopes, and R. Babuška, “A survey of actor-critic reinforcement learning: standard and natural policy gradients,” *IEEE Transactions On Systems, Man, And Cybernetics - Part C: Applications And Reviews*, vol. 42, pp. 1291–1307, 2012.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction.* Cambridge: MIT Press, 1998.

[11] S. Whiteson and P. Stone, “Adaptive job routing and scheduling,” *Engineering Applications of Artificial Intelligence*, vol. 17, pp. 855—-869, 2004.

[12] M. Pinedo, *Scheduling: Theory, Algorithms and Systems (5rd ed.).* New York: Springer, 2016.

[13] Y. Wang, X. Li, and R. Ruiz, “An exact algorithm for the shortest path problem with position-based learning effects,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, pp. 3037–3049, 2017.

[14] C. Watkins, “Q-Learning,” *Machine Learning*, vol. 8, pp. 279–292, 1992.

[15] W. Y. Kwon, I. H. Suh, and S. Lee, “SSPQL: stochastic shortest path-based Q-learning,” *International Journal of Control, Automation, and Systems*, vol. 9, pp. 328–338, 2011.

[16] A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, “A deterministic improved Q-learning for path planning of a mobile robot,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, pp. 1141–1153, 2013.

[17] Z. S. Givi, M. Y. Jaber, and W. P. Neumann, “Modelling worker reliability with learning and fatigue,” *Applied Mathematical Modelling*, vol. 39, pp. 5186–5199, 2015.

[18] C. H. Glock and Y. M. Jaber, “Learning effects and the phenomenon of moving bottlenecks in a two-stage production system,” *Applied Mathematical Modelling*, vol. 37, pp. 8617–8628, 2013.

[19] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[20] R. Graham, “Bounds for certain multiprocessing anomalies,” *Bell System Technical Journal*, vol. 45, pp. 1563–1581, 1966.

[21] W. Li and J. Yuan, “LPT online strategy for parallel-machine scheduling with kind release times,” *Optimization Letters*, vol. 10, pp. 159–168, 2016.

DOI: 10.15439/2023F8957
ISSN 2300-5963 ACSIS, Vol. 35

Multi-queue service for task scheduling based on data availability

Kamil Rybiński, Michał Śmiałek
0000-0002-5543-790X, 0000-0001-6170-443X
Warsaw University of Technology
pl. Politechniki 1, 00-661 Warszawa, Poland
Email: {michal.smialek, kamil.rybinski}@pw.edu.pl

***Abstract*—Large-scale computation (LSC) systems are often performed in distributed environments where message passing is the key to orchestrating computations. In this paper, we present a new message queue concept developed within the context of an LSC system (BalticLSC). The concept consists in proposing a multi-queue, where queues are grouped into families. A queue family can be used to distribute messages of the same kind to multiple computation modules distributed between various nodes. Such message families can be synchronised to implement a mechanism for initiating computation jobs based on multiple data inputs. Moreover, the proposed multi-queue has built-in mechanisms for controlling message sequences in applications where complex data set splitting is necessary. The presented multi-queue concept was successfully implemented and applied in a working LSC system.**

I. Introduction

LARGE-SCALE computations (LSC) are often performed in a distributed environment. Several computation nodes can be linked together to execute resource-consuming tasks. One of the main issues is the management of data flow between these nodes [1]. The goal is to reduce data transfer overheads and maximise the speed of computations. In a data flow-driven approach to LSC, computation applications are divided into computation steps, with data flowing as messages between these steps. An example of such a system is the BalticLSC platform [18], [13] (www.balticlsc.eu). It is a low-code computation environment created as part of a project intended to facilitate easier access to LSC. It performs computations in the form of Docker-based computation modules that are orchestrated according to applications defined in a dedicated graphical language called CAL (Computation Application Language) [19]. CAL revolves around the flow of data. The specific sequence of execution for computation modules is defined by specifying paths through which data flows between them. This creates the need for means to schedule starting of computation jobs according to data availability and to propagate that data between modules as defined by a CAL application. Other examples of systems where computations are driven by flowing data and that use graphical languages are WS-PGRADE [7], [6] and Flowbster [8]. Also, other Scientific Workflow Systems use this kind of approach [11].

Figure 1 shows an example application written in CAL. This application consists of three computation modules (the boxes) communicating through 6 data flows (the arrows). The application has two inputs (“Videos” and “Subtitles”) and one output (“Films”). In this example, the inputs and the output are sequences of folders containing appropriate files. Module “A” is a File Synchroniser type. It accepts two folders and, using certain naming conventions, produces two synchronous sequences of files (here: a video file matched with a subtitle file). The video file is processed by module “B” which is a Video Converter. Finally, module “C” (a VS Mixer) mixes the processed video file with the subtitle file received directly from module “A”. Individual files resulting from module “C” are then placed in appropriate folders at the output.

The flow of data between computation modules in Figure 1 is shown through additional tokens besides data flows. The numbers in circles relate to the specific flows (numbered 1 to 6 to denote the six flows). The numbers in rectangles denote sequence numbers. As we can notice, these sequence numbers can be stacked. For instance, token “1” with sequence number “0” (T1-S0) denotes a token message representing the first folder with video files on the “Videos” input. In case more folders are placed on the input, additional token messages (T1-S1, T1-S2, ...) are produced (not shown in the figure). Token number “3” necessitates a stack of two sequence numbers (e.g. T3-S0-S0, T3-S0-S1, ...). The top element corresponds to the source folder (cf. T1-S0), and the bottom element represents the number in the sequence of files in that folder. Note that these sequence numbers have to be maintained throughout computations. In our example, each instance of module “C” has to receive tokens with appropriate sequence stacks. For instance, token message T5-S0-S1 has to be matched with token message T4-S0-S1.

As in the example above, the message-passing system operates on sequences of token messages. It is thus natural to implement it using queues. However, according to our best knowledge, no existing solution could offer out-of-the-box all the functionality required to schedule jobs controlled by multiple data passing between the jobs. General-purpose queue systems are heavy-weight and offer extensive functionality for instantiating multiple queues. However, they do not provide mechanisms for task scheduling based on many synchronised queues. After analysing the available options, we have decided

This work was partially funded by the InterReg BSR Programme, grant no. #075

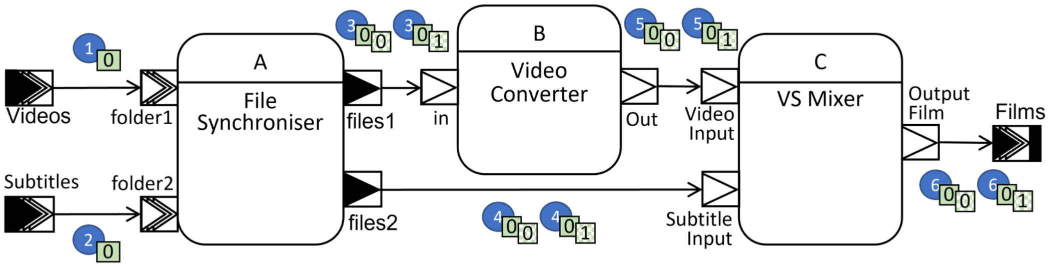


Fig. 1. Example application in CAL with sample tokens

to develop our own lightweight queue system, suitable for the CAL execution environment. In this paper, we present the details of our queue system and its application as part of a computation orchestration system within the BalticLSC environment.

II. Related work

There are two main approaches to performing distributed computations – orchestration and choreography [14], [16]. This follows two global trends of deploying workflows or composite micro-services. In the first approach, an external entity manages the flow of computations. BalticLSC and WS-PGRADE are examples here. In the second approach, this flow is realized by each participating computation element that fulfils its role that is known in advance. Flowbster is an example here.

The usage of FIFO queues is especially important for choreographed computations where they are intensively used. This includes managing the initiation of jobs in the proper order (either directly [9] or as part of more complex mechanisms [12]) and ensuring the exchange of messages between them. Especially relevant in the context of this paper is the usage of systems consisting of multiple queues (multi-queues) for task scheduling [2], [10], [5], [20], [22], [15]. Since such systems use queues to schedule tasks, they operate by scheduling computation steps, not data flows. For example, Li et al. [10] propose a system for scheduling read-write tasks for a distributed database system. Their solution is based on a multi-queue feature built into the Cassandra database system.

Generally, using queues provides two main advantages. Firstly, the responsibility to deliver messages falls on the queues, without the need to implement any additional mechanisms on the side of the message senders. Secondly, queues create separation in time between sending and receiving messages. Senders can send messages to yet non-existent recipients. Moreover, they can finish working before the recipients even start.

When describing a queue, we distinguish two roles of entities interacting with it - producers supplying messages to the queue and recipients interested in receiving these messages. Most queues utilize the publish-subscribe pattern. In this pattern, recipients declare to a queue their interest in messages of a given type. When a message fulfilling appropriate criteria arrives from a producer, all interested recipients are notified. The delivery of messages to recipients is generally done in one of two ways. Either a queue sends messages to its recipients (the push method), or recipients ask a queue for the messages and fetch them themselves (the pull method).

There are also a few additional issues that queues need to address, especially in the case of the push method. Messages can get lost because communication between queues and their recipients is never flawless. This necessitates repeating messages to guarantee their delivery. On the other hand, this can cause occurrences of unwanted duplicated messages. As a result, striving to ensure guaranteed delivery or lack of duplicates leads to hindering one of these features. Repeating undelivered messages could also change the order of messages, that is – the order in which messages reach the recipients can be different than the order in which they arrive from the producers. It is especially problematic when messages are delivered to multiple recipients simultaneously.

There are many systems implementing message distribution using queues [23], [21], but two of them found very wide usage – RabbitMQ [17] and Apache Kafka [3]. RabitMQ is an implementation of an extension to the AMQP protocol [24]. It supports delivering messages through both the push and the pull methods. However, the former is not recommended in most cases. Message distribution in RabbitMQ is oriented around the concept of exchanges. Exchanges define how messages are distributed between queues (which are related to particular recipients). RabbitMQ supports several types of exchanges, which allow directing messages to particular queues (direct exchange), duplicating messages to groups of queues (fanout exchange), distributing messages according to predefined topics (topic exchange) or checking for more complex patterns in message headers (header exchanges). The most commonly used publish-subscribe pattern is typically realized using topic exchanges through recipients subscribing to particular topics. To provide messages to recipients using the push method reliably, RabitMQ introduces a special mechanism for the acknowledgement of received messages by the recipients. It works based on the same principles as the mechanism for acknowledging receiving messages from the producers by the queue system. The idea was introduced to avoid confirming message delivery through the transaction system. Even if more reliable, this was considered as inconvenient for more trivial cases. Its existence also allows more subtle options used

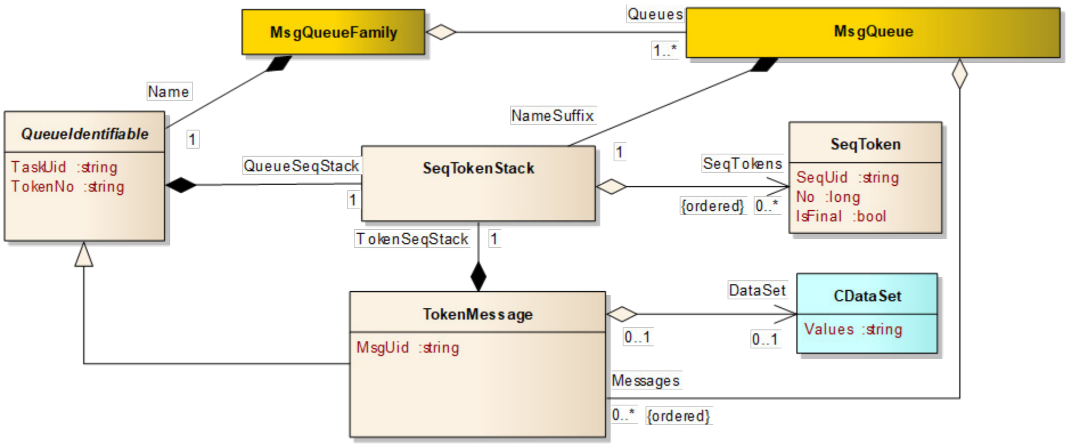


Fig. 2. Multiqueue metamodel

in managing the distribution of messages, like defining the prefetch count – the maximum number of unacknowledged messages that can be sent to a given recipient.

In the case of Apache Kafka [25], [4], queues are organized by topics to which particular recipients can subscribe. Each topic can also be further divided into partitions containing messages for a given recipient. Message delivery is then done by the pull method. This means that particular subscribers need to request messages that are of interest to them, typically as a part of a so-called pull loop. Kafka by default does not delete messages after they are delivered, and there is a possibility to access them again when needed. Accessing messages is done by means of an offset – the particular position in the sequence of messages that indicates which of them were already received. This particular setup obviously makes any message acknowledgement from the recipient obsolete. Thus, Kafka supports only the acknowledgement of message delivery to producers. However, particular client implementations can simulate an acknowledgement mechanism to manage the offset.

III. Multi-queue concept

As mentioned in the introduction, the orchestration of jobs in systems like BalticLSC is based on data flowing between these jobs. This makes message passing extremely important and imposes additional requirements. First, we need to ensure a reliable exchange of messages, as in every distributed system. In addition, our solution needs to allow representing and checking orchestration information attached to the messages. This information can be used to direct messages to appropriate queues.

Figure 2 shows the data structure of our multi-queue in the form of a UML model (a meta-model). The exchange of data between computation modules is handled by "token messages" (see the "TokenMessage" class), and such messages are managed by our multi-queue. Each token message contains a reference that points to a specific data set (see the "CDataSet" class). Data sets can contain data directly or can hold information about accessing data held in a storage system. Each token message is assigned properties that identify its position in the data flow within an instance of a specific application. This is reflected by two attributes specialised by the "TokenMessage" class from the "QueueIdentifiable" class. The "TaskUid" attribute refers to a specific task or – in other words – an instance of an application being executed within the BalticLSC system. The "TokenNo" attribute refers to the particular token number assigned to a specific data flow within the respective application (e.g. tokens 1-6 in Figure 1).

Token messages contain additional information about their sequence numbers. In fact, each token message contains a stack ("SeqTokenStack) of sequence numbers ("SeqToken"). This stack is divided into the "QueueSeqStack" and the "TokenSeqStack". The first stack corresponds to the specific queue to which this message is designated, while the second stack corresponds to a specific sequence of tokens produced by computation modules. These stacks ensure proper processing in applications where data can change its "granularity" (e.g. from folders to files). In such cases, we need to split larger data sets into many smaller pieces to be processed individually (e.g. in parallel). On the other hand, we might need to merge many smaller pieces into a final, larger data set. At the same time, we need to manage these various sequences so that relations between and within these sequences are preserved.

The token message stacks ("SeqTokenStack") allow keeping track of dependencies between data items and thus enable proper distribution of computations between computation jobs. The general rule is that a new level of the stack is added when a data set (file, folder) is split into smaller elements. On the other hand, when several data items are merged, one level of the stack is removed. Note that in some cases, two or more stack levels can be added or removed, depending on particular processing (e.g. processing of a two-dimensional data matrix). Moreover, it can be noted that sequence tokens contain sequence numbers ("No" in the "SeqToken" class). This is necessary due to the possible parallel processing of token messages. In case of delays in the processing of tokens, their order has to be maintained regardless of individual processing times. Moreover, we need a flag denoting the last token in a sequence (“IsFinal”).

The queue system that handles token messages consists of two elements. The top-level element is the Message Queue Family (“MsgQueueFamily”), which contains individual Message Queues (“MsgQueue”). The role of the queue family is to handle messages associated with a single data flow (see Figure 1 again). The individual queues handle messages directed to specific computation module instances (jobs) deployed in different distributed computation nodes. Queues are identified by token sequence stacks similar to the token messages that they contain. The identification (“Name”) of a message queue family contains two main values describing a token (TaskUid, TokenNo). This is appended with a token sequences stack (“QueueSeqStack”) to form the full identifier of a queue family. The message queues contain the suffix part (“NameSuffix”) of the token sequence stack, identifying the particular jobs to which tokens should be sent. As we can notice, the main assumption is that individual queues contain token messages that are meant to be processed by the same computation modules. Thus, these token messages point to the same type of data. The assignment of messages to queues is done in two steps. First, we assign them to a message queue family and then – to a particular queue.

Token messages in each of the queues contain full token sequence stacks. In addition to the stack defined in the queue family and its queues, the token sequence stack in a message contains additional levels defining the sequence of tokens sent to the particular job. As a result, before putting a token message to a specific queue, we need to construct a token sequence stack consisting of two parts. The first part identifies the queue (and its family), and the other part contains indexing data that will be forwarded to the recipient (job) to help in internal processing.

In practice, the relationships indicated by the elements of each “SeqTokenStack” serve two purposes. Firstly they ensure that jobs that should process corresponding groups of data from different queues will receive correct data items. Secondly, when computations are distributed between separate machines, related data will be grouped together to reduce unnecessary data transfer. It is also worth mentioning that there is an exception to these rules. This is for so-called “simple” modules, i.e. those that process only one token message from a single queue at a time. In the case of such queues, there is no need to use SeqTokenStack elements for either of the two mentioned purposes, at least at the individual level. However, to maintain consistency in handling queues, we still use “SeqTokenStack” to identify such queues.

An example of a queue family is shown in Figure 3. The outer box represents a family with a task identifier (“Task1”), a token number (“8”) and a token sequence stack (here with one level – “0”). The family contains three queues. Each queue contains a “suffix” for the token sequence stack (“0”, “1” or “2”). In each queue, we have several token messages. As we can see, each message contains the same token number (“8”)

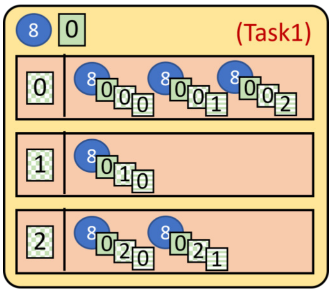


Fig. 3. Example structure of a queue family

and token sequence stack prefix (“0”), which reflects the queue family. Moreover, tokens assigned to each queue in the family have the second levels of their stacks corresponding with their appropriate queue stack suffix. Finally, the third level in the stack reflects the token sequences.

Queues defined according to the presented rules can be used to initiate jobs (instances of computation modules). In general, specific modules require either one or many tokens to be delivered to them from a given queue. For some queues, the arrival of tokens is required to start a job. In other cases, tokens arriving at a queue are passed to a job but are not mandatory to start processing. With our multi-queue, we can easily determine meeting conditions to start jobs. When a new token that matches a given queue in a family arrives, we can check all the queues in queue families assigned to the inputs of a respective computation module. If all the corresponding queues contain tokens, the condition for a new processing job is met. We can then either start a new instance of a computation module or use an instance that is currently idle.

Apart from this additional functionality, our queues work like traditional FIFO queues. They use the publish-subscribe pattern to deliver tokens to jobs using the push method. An additional non-standard element is the implementation of the acknowledgement mechanism, which has some similarities to that of RabitMQ. When a message is delivered to the queue, it is first stored in it. It then waits for its turn (FIFO) and gets assigned to one of the available recipients. This is done by balancing the number of messages each registered recipient receives. Recipients may reject messages (e.g. have insufficient resources), which results in attempting to send the message to another recipient. If the message is accepted, its status changes accordingly, but it is not removed until the recipient acknowledges the finishing of its processing. There is also a possibility of the recipient sending a negative acknowledgement (processing did not succeed). In this case, the message status is reverted, which causes a repetition of the above process.

IV. Illustrative example

In this example, we use the application presented in the introduction (Figure 1). Each data flow (numbered from 1 to 6) is associated with one or more queue families. Moreover, these families are grouped by computation modules depending on

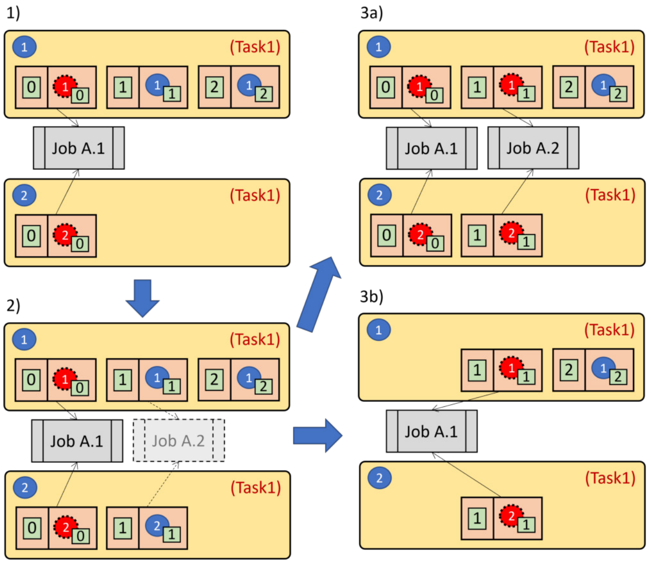


Fig. 4. Queues for tokens 1 and 2

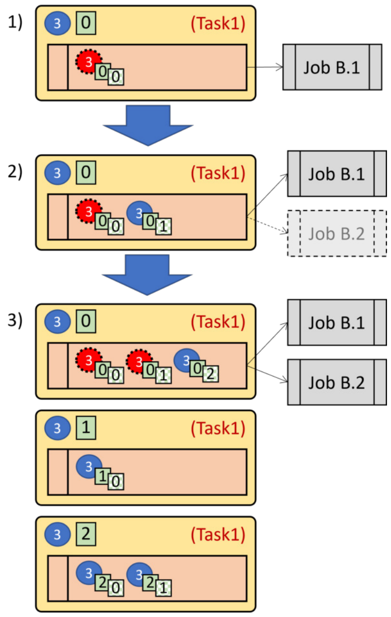


Fig. 5. Queues for token 3

their inputs. These groups can be used to initiate appropriate jobs being instances of the computation modules. We assume that a single task was created from our example application ("Task1"). Obviously, more such tasks can be executed at the same time, and thus more queue families can be created whose identification differs only by the task identifier.

Figure 4 shows queues responsible for providing token messages to instances of module A in our application. The situation shown in step 1) is somewhat advanced in time. We have two queue families (related to tokens "1" and "2"), which already have a few token messages in each of them. We also have one instance of the module (job) running (named "A.1"). It can be noted that each input of module A has been assigned a separate queue family. Moreover, separate queues related to the existing token sequence stacks were created inside the families. At this moment, we have three queues for tokens of type "1" and one queue for tokens of type "2". Note that each of these queues was created with the arrival of an appropriate token message. The name suffixes correspond to the token sequence numbers of these token messages.

When queues 1-0 and 2-0 were created, job A.1 was initiated and assigned to these queues as their recipient. Through this mechanism, the job is guaranteed to get the related pair of token messages. In the situation shown in step 1), we already have job A.1 running and processing the pair of token messages denoted with the sequence number "0" in their token sequence stacks. The queues marked these messages as delivered but did not delete them, as the job did not acknowledge finishing their processing. We have also two additional messages in the first queue family that haven't been delivered yet.

In step 2), a new message is inserted into the second queue family which results in creating an appropriate queue (2-1). With this arrival, we have a new pair of matching token messages (1-1 and 2-1). This creates a condition to create a new job. Depending on the decision of the job broker module, this can lead to one of two situations shown as 3a) and 3b) in Figure 4. In the first situation, a new instance of the computation module A was started ("A.2"). This instance has immediately subscribed to the two related queues. The appropriate messages were then delivered and marked as being processed. In the second situation, a new instance of module A was not created. The job broker waited until instance A.1 finished its previous processing. Only then could the instance subscribe to queues 1-1 and 2-1. This is followed by delivering the new pair of token messages to A.1 and marking them as being processed. Note that the second situation can occur in the situation of limited resources. If we cannot create a new instance of module A, we must wait for an existing instance to finish its current job.

Figure 5 presents queues that serve to provide messages from instances of module A to instances of module B. Note that module B is a simple module – it has only one single-token input. Thus, the incoming messages do not need to be grouped. However, we need to preserve the grouping made for module A at the start of the application. In step 1) we create a new queue family when a message arrives from one of the jobs being instances of module A. We note that this message has a stack with two layers. The first layer corresponds to the sequence numbering preserved from the tokens messages for token no. "1". This reflects a series of folders placed on the application input "Videos". The second layer contains a new sequence (a sub-sequence), corresponding to messages created from one instance of module A. These messages reflect individual files present in a given folder.

The first queue family is identified by the token number

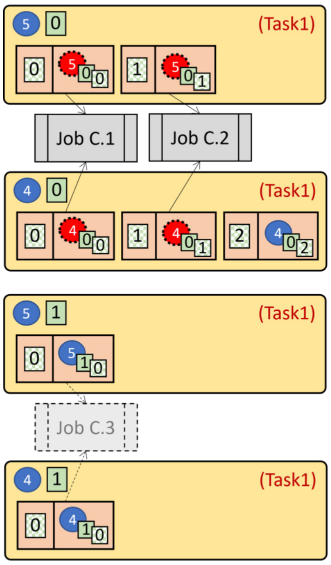


Fig. 6. Queues for tokens 4 and 5

and the top-level sequence number (here: 3-0). It has one default queue that does not need any additional identification. In Figure 5, we can see that the queue already has one job (B.1) assigned as its recipient and the first token message (3-0-0) is already being processed by this job. In step 2), another message in the sequence (3-0-1) is delivered to the queue. This creates an occasion to start another job of type "B" (assuming that job B.1 still processes token 3-0-0). Such a new job (here: B.2) would subscribe to the queue the same way as job B.1. In this case, further token messages can be processed by two jobs.

Step 3) in Figure 5 shows the configuration of queues after several other messages for token "3" arrive. As we can see, further queue families were created. These families reflect the token message sequence arriving at module A. For each such token message (1-0, 1-1, 1-2), we create a separate queue family (3-0, 3-1, 3-2). The reason for applying such a mechanism is to facilitate the distribution of computations. Each of the queues can be assigned to a different computation node potentially located in different geographical locations. In such a case, e.g. jobs for queue 3-0 will obviously not be able to process token messages in queue 3-1. On the other hand, if both queues are assigned to a single node, it is fairly easy to assign a job to both queues as their recipient.

Figure 6 presents queues that handle messages coming from instances of modules A and B and sent to instances of module C. The situation is to some extent similar to the previous ones, so we present only one "snapshot" of the queues. The main difference is the existence of many queues in queue families with split token sequence stacks. This split is caused to facilitate the distribution of jobs. Queue families are assigned to specific computation nodes, and individual queues in these families are assigned to specific computation module instances.

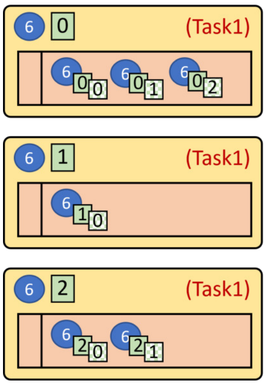


Fig. 7. Queues for token 6

In our example, we have two jobs already created based on two sets of token messages (5-0-0 with 4-0-0 and 5-0-1 with 4-0-1). These two jobs are running on one computation node. Another job (C.3) starts based on messages 5-1-0 and 4-1-0. This job can potentially be started on a different computation node.

Finally, Figure 7 shows queues containing token messages with the final results of the application. Here we can see three queue families created for a single token ("6"). These families correspond to the initial token message sequence shown in Figure 4. For instance, queue family 6-0 corresponds to tokens 1-0 and 2-0 sent initially to queues "1" and "2". As we remember, these initial tokens reflected folders with video and subtitle files. At this final stage, the results should also be grouped into folders according to the CAL application (see the output "Films" in Figure 1). Messages assigned to each of the queues in Figure 7 reflect individual files with the resulting films. Queues reflect folders into which these files should be sent. These output folders will contain films created from videos and subtitles contained in synchronised input folders.

Note that the application can be easily extended with further modules. In such a case, the queues for token "6" could be exactly the same (depending on how we exactly extend our application). In such a hypothetical case, the first arrival of a token message will create an occasion to start a new job, to which all messages from the given queue would be sent. It has to be stressed that our multi-queue mechanisms allow for the fully automatic creation of queues, as in the presented example. The execution engine can create queues based on the definition of the application in CAL and the arrival of consecutive token messages. In the next section, we present the implementation of these mechanisms.

V. Multi-queue implementation

The BalticLSC system consists of several components that cooperate to orchestrate jobs in a distributed computation environment. An extract of the BalticLSC logical architectural model (see https://www.balticlsc.eu/model/) is shown in Figure 8. The multi-queue mechanisms presented in this paper are

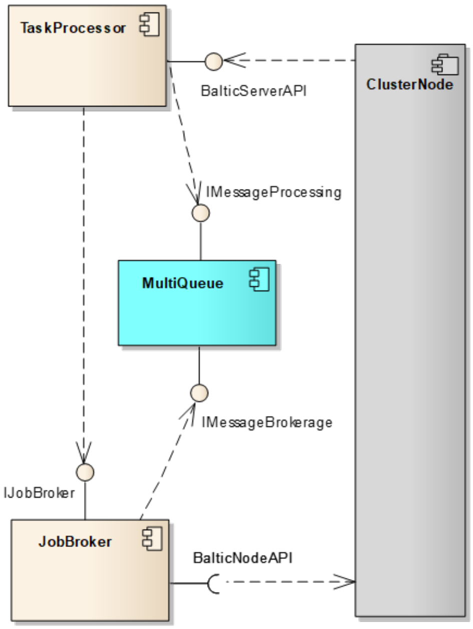


Fig. 8. Multi-queue usage in a computation system

implemented within a separate component – “MultiQueue”. This component works in cooperation with the middleware responsible for orchestrating computations. As we can see in Figure 8, this middleware contains two components that work as intermediaries between the multi-queue system and the actual computation modules.

The “TaskProcessor” component is responsible for invoking most of the presented multi-queue mechanisms. It receives token messages from computation modules through the BalticServerAPI. It then appends them with all the necessary information (including the sequence stacks) and inserts them into the multi-queue. It also checks all the conditions for starting new computation module instances. When a condition is met, it passes an appropriate command to the “JobBroker” component. This component is responsible for the actual initiation of new module instances, which includes making decisions regarding job balancing and sensing these jobs to specific computation nodes. It also registers these new instances in the queues, allowing appropriate tokens to be transmitted.

To communicate with the multi-queue, the Task Processor uses the IMessageProcessing interface, and the Job Broker uses the IMessageBrokerage interface. The details of these interfaces are shown in Figure 9. The first of the interfaces groups operations to enqueue new token messages, acknowledge the finishing of their processing and check the status of tokens in specific queues. For managerial purposes, there is also the possibility to create queue families. Finally, there is the possibility of forming a tree-like structure from the queues by defining queue predecessors. This additional mechanism is used by an automatic queue-cleaning mechanism. This ensures that empty queues will not be deleted as long as preceding queues have unacknowledged tokens left. The IMessageBrokerage interface contains an additional method that allows registering new recipients for specified queues. Figure 9 also shows the IQueueConsumer interface that should be implemented by a computation module. This simple interface allows module instances to receive messages.

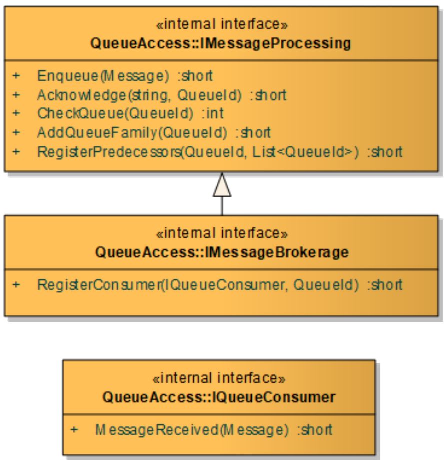

«intemal interface»
QueueAccess::IMessageProcessing
+ Enqueue(Message) :short
+ Acknowledge(string, QueueId) :short
+ CheckQueue(QueueId) :int
+ AddQueueFamily(QueueId) :short
+ RegisterPredecessors(QueueId, List<QueueId>) :short
«intemal interface»
QueueAccess::IMessageBrokerage
+ RegisterConsumer(IQueueConsumer, QueueId) :short
«internal interface»
QueueAccess::IQueueConsumer
+ MessageReceived(Message) :short

Fig. 9. Interfaces of the multi-queue implementation

Operation of the TaskProcessor component when processing a token message is presented in Figure 10. The whole process is initiated by invoking the PutTokenMessage method in reaction to the incoming token message. First, the method checks if there is the need to start a so-called job batch. This feature of the BalticLSC system allows for grouping of jobs so that they would be computed on the same computation nodes making data transfer more efficient (see CAL specifications [19]). Starting of job batches is similar to starting of jobs. The TaskProcessor needs to check all the queues related to the inputs of all jobs that are also inputs of the particular job batch. For this, it calls the CheckQueue operation which returns the number of messages in a given queue. Presence of at least one message in all the input queues required to start the job batch results in calling the ActivateJobBatch operation of the JobBroker component. Obviously, if the job batch is already activated, there is no need to perform queue checks and activate it again.

A similar approach is used to activate the specific job related to the processed token. Again, the procedure includes checking of all the required input queues. Note that the queue for the processed token message does not need to be checked. Instead, the message is enqueued after all the necessary checks are done.

Details of the multi-queue implementation are shown in Figure 11. The MultiQueue class is responsible for the functioning of the MultiQueue component and implements its provided interfaces. It runs a thread (the “Run” operation) that periodically sends tokens to consumers registered with the various queues. Message distribution is performed through handles to queue families grouped by task identifiers. These queue families are implemented by the MsgQueueFamily

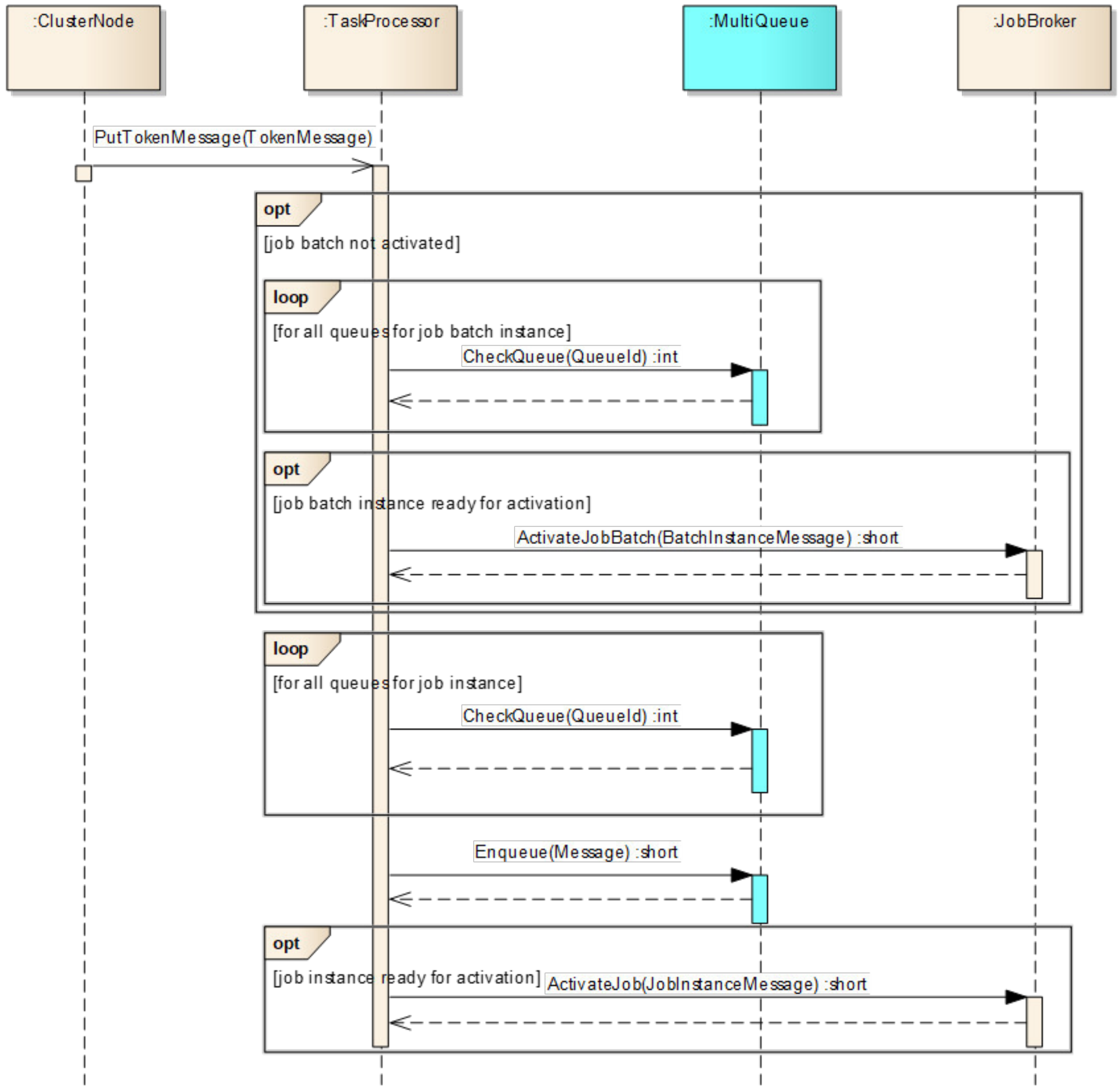

:ClusterNode
:TaskProcessor
:MultiQueue
:JobBroker
PutTokenMessage(TokenMessage)
opt
[job batch not activated]
loop
[for all queues for job batch instance]
CheckQueue(QueueId) :int
opt
[job batch instance ready for activation]
ActivateJobBatch(BatchInstanceMessage) :short
loop
[for all queues for job instance]
CheckQueue(QueueId) :int
Enqueue(Message) :short
opt
[job instance ready for activation]
ActivateJob(JobInstanceMessage) :short

Fig. 10. Sequence diagram illustrating processing of a token message

class, which groups queues with a common ‘name’. This name is formed using the same data structure (“QueueIdentifiable”) as in the original data model (Figure 2). The queue family class has various operations that allow to enqueue and acknowledge messages, distribute them to the registered consumers, and check queue statuses.

As expected, the MsgQueueFamily class contains the MsgQueue class. This class holds the actual queue contents, i.e. the messages. It also contains the name suffix in the form of a token sequence stack and has handles to the queue consumers. Operations of this class are typical for a queue and do not necessitate a detailed explanation.

To illustrate the usage of our multiqueue within the BalticLSC system we will use the CAL program fragment presented in Figure 12. The main module in this fragment is the “ImgChannelJoin” module. It has three input pins and one output pin. The three inputs are connected with certain processing modules, where one of them is shown in the figure. The input pins are associated with appropriate tokens (np. 3, 9 and 10). In the figure, we can see three tokens arriving, each of them having the same sequence number (0) and thus causing initiation of a new job instance for the “ImgChannelJoin” module.

In Figure 13 we can see a fragment of execution log for the situation in Figure 12. The log contains information about processing of a single token message arriving at one of the inputs of the “joiner” module. The first entry in the log denotes the arrival of the message at the Task Processor component (see also Figure 8). As we can see, the message is related to Token number 3. It is the first in a sequence of messages for this Token, and thus contains the index number 0 on its sequence stack (see the ‘seq_stack’ section).

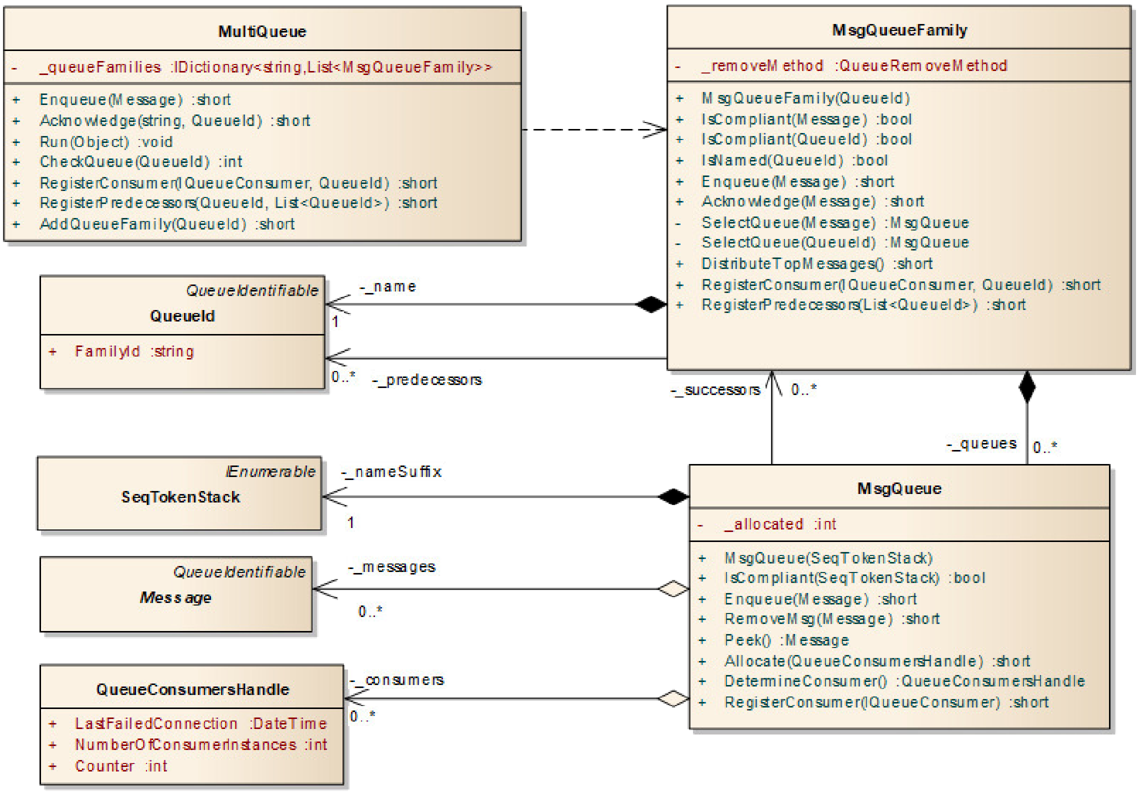

MultiQueue
- _queueFamilies :IDictionary<string,List<MsgQueueFamily>>
+ Enqueue(Message) :short
+ Acknowledge(string, QueueId) :short
+ Run(Object) :void
+ CheckQueue(QueueId) :int
+ RegisterConsumer(IQueueConsumer, QueueId) :short
+ RegisterPredecessors(QueueId, List<QueueId>) :short
+ AddQueueFamily(QueueId) :short
MsgQueueFamily
- _removeMethod :QueueRemoveMethod
+ MsgQueueFamily(QueueId)
+ IsCompliant(Message) :bool
+ IsCompliant(QueueId) :bool
+ IsNamed(QueueId) :bool
+ Enqueue(Message) :short
+ Acknowledge(Message) :short
- SelectQueue(Message) :MsgQueue
- SelectQueue(QueueId) :MsgQueue
+ DistributeTopMessages() :short
+ RegisterConsumer(IQueueConsumer, QueueId) :short
+ RegisterPredecessors(List<QueueId>) :short
QueueIdentifiable
QueueId
+ FamilyId :string
-_name
1
0..*
-_predecessors
-_successors
0..*
-_queues
0..*
IEnumerable
SeqTokenStack
-_nameSuffix
1
QueueIdentifiable
Message
-_messages
0..*
QueueConsumersHandle
+ LastFailedConnection :DateTime
+ NumberOfConsumerInstances :int
+ Counter :int
-_consumers
0..*
MsgQueue
- _allocated :int
+ MsgQueue(SeqTokenStack)
+ IsCompliant(SeqTokenStack) :bool
+ Enqueue(Message) :short
+ RemoveMsg(Message) :short
+ Peek() :Message
+ Allocate(QueueConsumersHandle) :short
+ DetermineConsumer() :QueueConsumersHandle
+ RegisterConsumer(IQueueConsumer) :short

Fig. 11. Code structure of the multi-queue implementation

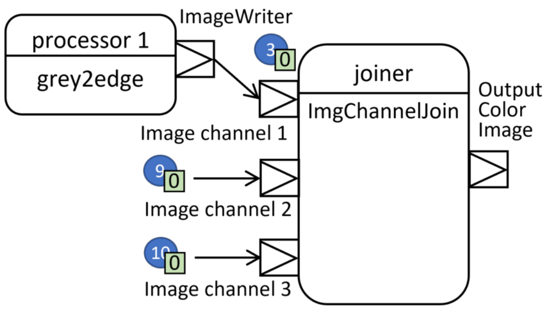


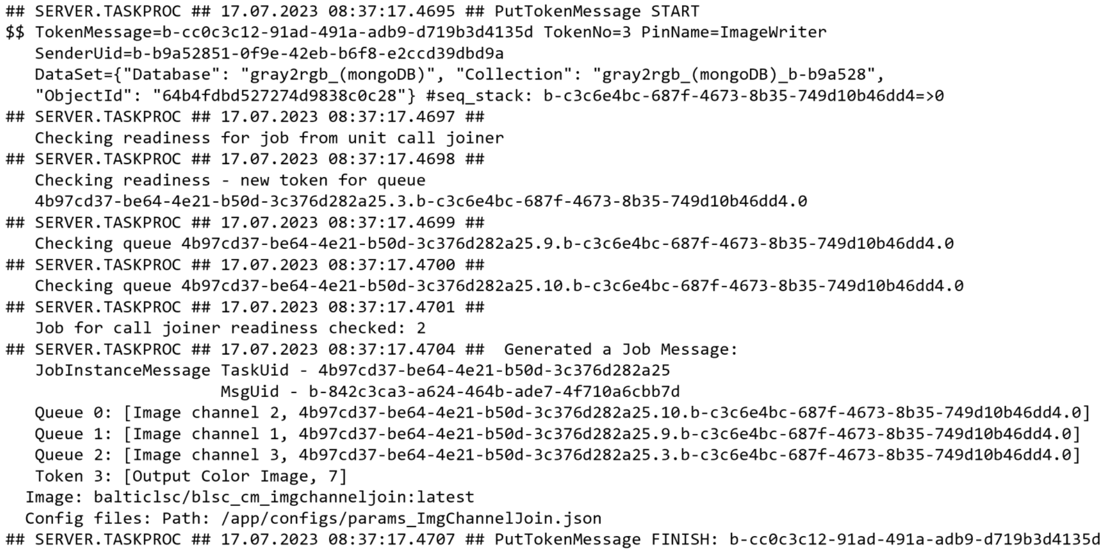
Fig. 12. CAL program fragment to illustrate queue module usage

Immediately after receiving the message, the Task Processor performs a sequence of checks to determine if the incoming message would trigger a new job instance (see the "Check-Queue" operation). In our case, three queues are checked – the ones related to the three inputs of the "joiner" module, including the queue for the incoming message. As we can see, the Task Processor requests checking queues for Tokens 3, 9 and 10 with the sequence index 0.

In the presented situation, queues "9.0" and "10.0" already had messages in them. This means that arrival of the message to the queue "3.0" should trigger a new job instance. The log in Figure 13 thus contains information about creating a new job message. This new message contains information about the queues for input tokens and about the output token. It also determines technical details of the module to be executed by the runtime. Some of these details (image file, configuration file) are shown in the figure but other are omitted as not relevant here. Note that the information about the module to be executed are determined by the Task Processor based on information compiled from the appropriate CAL program.

The new job message is passed to the "JobBroker" component through the "IJobBroker" interface (see again Figure 8). The broker then selects an appropriate computation node for execution of the job and sends the job message to this node. Note that job brokerage algorithm is out of scope of this paper. After determining the job's executing module at the computation node, the broker registers it in all the relevant queues through the "IMessageBrokerage" interface. Finally, the Task Processor inserts the incoming token message into the multi-queue using the "Enqueue" operation of the "IMessageProcessing" interface (see Figure 9). This finishes processing of the token message by the Task Processor. This is signalled by the last entry in the presented log. Further steps involve the computation node that accesses the queues according to the description in the previous sections.

By examining the log in Figure 13 we can also determine the efficiency of the queue system. Each entry contains information about its time with the precision of tenths of milliseconds. As we can see, it took the engine around a millisecond to process the message – put it into the queue module, check



```
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4695 ## PutTokenMessage START
$$ TokenMessage=b-cc0c3c12-91ad-491a-adb9-d719b3d4135d TokenNo=3 PinName=ImageWriter
   SenderUid=b-b9a52851-0f9e-42eb-b6f8-e2ccd39dbd9a
   DataSet={"Database": "gray2rgb_(mongoDB)", "Collection": "gray2rgb_(mongoDB)_b-b9a528",
   "ObjectId": "64b4fdbd527274d9838c0c28"} #seq_stack: b-c3c6e4bc-687f-4673-8b35-749d10b46dd4=>0
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4697 ##
   Checking readiness for job from unit call joiner
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4698 ##
   Checking readiness - new token for queue
   4b97cd37-be64-4e21-b50d-3c376d282a25.3.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4699 ##
   Checking queue 4b97cd37-be64-4e21-b50d-3c376d282a25.9.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4700 ##
   Checking queue 4b97cd37-be64-4e21-b50d-3c376d282a25.10.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4701 ##
   Job for call joiner readiness checked: 2
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4704 ##  Generated a Job Message:
   JobInstanceMessage TaskUid - 4b97cd37-be64-4e21-b50d-3c376d282a25
                      MsgUid - b-842c3ca3-a624-464b-ade7-4f710a6cbb7d
   Queue 0: [Image channel 2, 4b97cd37-be64-4e21-b50d-3c376d282a25.10.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0]
   Queue 1: [Image channel 1, 4b97cd37-be64-4e21-b50d-3c376d282a25.9.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0]
   Queue 2: [Image channel 3, 4b97cd37-be64-4e21-b50d-3c376d282a25.3.b-c3c6e4bc-687f-4673-8b35-749d10b46dd4.0]
   Token 3: [Output Color Image, 7]
  Image: balticlsc/blsc_cm_imgchanneljoin:latest
  Config files: Path: /app/configs/params_ImgChannelJoin.json
## SERVER.TASKPROC ## 17.07.2023 08:37:17.4707 ## PutTokenMessage FINISH: b-cc0c3c12-91ad-491a-adb9-d719b3d4135d
```

Fig. 13. CAL program execution log fragment

associated queues and create a job message. This time is negligible in relation to the time used for computations which usually takes minutes, hours or even days. It thus can be argued that such message processing times allow for significant scalability of the system. Even with a single instance of the runtime engine, many parallel tasks could be processed without noticeable delays related to message processing. If needed, this could be further improved by adding additional runtime engine instances running on several machines. Note that determining detailed performance characteristics of our solution are subject to future work.

An important additional feature worth noting as facilitated by the existence of queue families is "queue garbage collection". Each queue has information about its successors, and each queue family has information about its predecessors. This allows the multi-queue to keep track of queues that are not used and will not be used in the future (within the current application instance). The garbage collector mechanism traverses queues and queue families to determine all the relations for a given queue and based on an appropriate algorithm removes "dead" queues. Discussion on the details of this mechanism is out of the scope of this paper.

VI. Conclusion

The presented multi-queue concept was validated in a fully operational distributed computation system. It has proven to be effective means of orchestrating computations with diverse configurations of data sets, flowing between instances of computation modules. The users of the BalticLSC system have developed various computation applications with different configurations of data flows. Examining of computation logs from the system acknowledges the effectiveness of message distribution by the queue system. Message handling times are short (in the scale of milliseconds) and thus negligible within the whole computation process.

The characteristics of our multi-queues facilitate the simultaneous delivery of multiple token messages to many instances of the same computation module running on different computation nodes. It also allowed the implementation of a job initiation mechanism that is based on the availability of data transmitted through complex data processing paths. At the same time, the implementation of our multi-queue system is quite simple and results in a lightweight queue component that can be used in similar contexts.

References

[1] Adam Barker, Paolo Besana, David Robertson, and Jon B. Weissman. The benefits of service choreography for data-intensive computing. In *Proceedings of the 7th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '09, page 1–10, New York, NY, USA, 2009. Association for Computing Machinery.

[2] Li Chunlin, Tang Jianhang, and Luo Youlong. Multi-queue scheduling of heterogeneous jobs in hybrid geo-distributed cloud environment. *The Journal of Supercomputing*, 74:5263–5292, 2018.

[3] Philippe Dobbelaere and Kyumars Sheykh Esmaili. Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry paper. In *Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems*, DEBS '17, page 227–238, New York, NY, USA, 2017. Association for Computing Machinery.

[4] Nishant Garg. *Apache Kafka*. Packt Publishing Birmingham, UK, 2013.

[5] Mohammad Hedayati, Kai Shen, Michael L Scott, and Mike Marty. Multi-queue fair queuing. In *USENIX Annual Technical Conference*, pages 301–314, 2019.

[6] Péter Kacsuk, editor. *Science Gateways for Distributed Computing Infrastructures*. Springer International Publishing, 2014.

[7] Peter Kacsuk, Zoltan Farkas, Miklos Kozlovszky, Gabor Hermann, Akos Balasko, Krisztian Karoczkai, and Istvan Marton. WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. *Journal of Grid Computing*, 10(4):601–630, nov 2012.

[8] Peter Kacsuk, József Kovács, and Zoltán Farkas. The Flowbster cloud-oriented workflow system to process large scientific data sets. *Journal of Grid Computing*, 16(1):55–83, jan 2018.

[9] A V Karthick, E Ramaraj, and R Ganapathy Subramanian. An efficient multi queue job scheduling for cloud computing. In *2014 World Congress on Computing and Communication Technologies*, pages 164–166, 2014.

[10] Haopeng Li and Hui Li. A scheduling strategy based on multi-queues of cassandra. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2664–2669, 2017.

[11] Ji Liu, Esther Pacitti, Patrick Valduriez, and Marta Mattoso. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13:457–493, 2015.

[12] Pedro García López, Aitor Arjona, Josep Sampé, Aleksander Slominski, and Lionel Villard. Triggerflow: Trigger-based orchestration of serverless workflows. In *Proceedings of the 14th ACM International Conference on Distributed and Event-Based Systems*, DEBS '20, page 3–14, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Krzysztof Marek, Michał Śmiałek, Kamil Rybiński, Radosław Roszczyk, and Marek Wdowiak. BalticLSC: Low-code software development platform for large scale computations. *Computing and Informatics*, 40(4):734–753, 2021.

[14] Chris Peltz. Web services orchestration and choreography. *Computer*, 36(10):46–52, 2003.

[15] Anastasiia Postnikova, Nikita Koval, Giorgi Nadiradze, and Dan Alistarh. Multi-queues can be state-of-the-art priority schedulers. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 353–367, 2022.

[16] Stephen Ross-Talbot. Orchestration and choreography: Standards, tools and technologies for distributed workflows. In *NETTAB Workshop-Workflows management: new abilities for the biological information overflow*, volume 1, page 8, Naples, Italy, 2005.

[17] Maciej Rostanski, Krzysztof Grochla, and Aleksander Seman. Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ. In *2014 Federated Conference on Computer Science and Information Systems*, pages 879–884, 2014.

[18] Radoslaw Roszczyk, Marek Wdowiak, Michal Smialek, Kamil Rybinski, and Krzysztof Marek. BalticLSC: A low-code HPC platform for small and medium research teams. In *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, oct 2021.

[19] Kamil Rybiński, Michał Śmiałek, Agris Sostaks, Krzysztof Marek, Radosław Roszczyk, and Marek Wdowiak. Visual low-code language for orchestrating large-scale distributed computing. *Journal of Grid Computing*, 21(3), jul 2023.

[20] Gaurav Sharma, Neha Miglani, and Ajay Kumar. PLB: a resilient and adaptive task scheduling scheme based on multi-queues for cloud environment. *Cluster Computing*, 24(3):2615–2637, 2021.

[21] T Sharvari and Nag K Sowmya. A study on modern messaging systems-Kafka, RabbitMQ and NATS streaming. 2019.

[22] Jaspreet Singh and Deepali Gupta. An smarter multi queue job scheduling policy for cloud computing. *International Journal of Applied Engineering Research*, 12(9):1929–1934, 2017.

[23] John Vineet and Liu Xia. A survey of distributed message broker queues. 2017.

[24] Steve Vinoski. Advanced Message Queuing Protocol. *IEEE Internet Computing*, 10(6):87–89, 2006.

[25] Guozhang Wang, Joel Koshy, Sriram Subramanian, Kartik Paramasivam, Mammad Zadeh, Neha Narkhede, Jun Rao, Jay Kreps, and Joe Stein. Building a replicated logging system with Apache Kafka. *Proceedings of the VLDB Endowment*, 8(12):1654–1655, aug 2015.

DOI: 10.15439/2023F287
ISSN 2300-5963 ACSIS, Vol. 35

Semi-persistent services for IoT networks using RESTful approach

Jarogniew Rykowski
0000-0001-7944-6061
University of Economics and
Business, Department of
Information Technology
Niepodleglosci 10, 61-875 Poznan,
Poland
Email:
rykowski@kti.ue.poznan.pl

***Abstract*— The paper proposes a new way to access semi-permanent services in ad-hoc and mesh networking in the context of the Internet of Things and the Internet of Services. The solution is based on address-free communication, with individual addresses of the nodes replaced by a semantic description of their functionality. The mesh network is accessible from outside using the classic RESTful approach and needs no centralized catalog to maintain at-the-moment available services. Instead, entry gateways are responsible for mapping incoming REST-compliant communication to internal mesh messaging, and the mesh nodes individually decide how to react to particular messages. Installing each new or replaced node or monitoring the node status is unnecessary. Automatic communication among the nodes is possible without human intervention, including both runtime and the registration phase.**

Bluetooth mesh topology network was chosen as the implementation base. Transmission in the network occurs in a broadcast mode, in which one network node sends information that is then received and interpreted by all other nodes. Selected devices equipped with alternative communication modules of a different type, in particular, connected to the home WiFi network, can be used as input/output gateways for outside communication.

***Index Terms*—address-free networking, zero-configuration networking, P2P broadcasting networks, BLE mesh, ad-hoc networks, RESTful approach.**

I. Introduction

Recently we have observed a boom in the client-server architecture. The RESTful approach needs special attention among multiple protocols and strategies used to implement a client-server system [1, 2]. This is an architectural style based on standard HTTP requests, and URL addressing. The key entity for a REST-compliant system is the so-called ‘resource’. A REST resource is a well-defined part of server functionality, being a building block for the set of servers’ services. A resource could be a pointer to a file, a server-side script, a database entry, etc. Each resource is uniquely identified by means of an URL address [3], composed of the scheme identifier, domain name, port number, a local path to a file/script, and a set of parameters composing a query**.**

The RESTful approach is an efficient tool for implementing a fixed, well-defined network. On the contrary, recently, we have observed the growing importance of ad-hoc [4] and dynamic networking, especially in the context of vehicular and road networks [5], not to say about classical sensor networks [6] and MANETs [7, 8]. The research in the domain of ad-hoc networking recently concentrated on mobility, energy savings, and routing [9], especially under specific circumstances (such as underwater acoustic connections [10], also evolving into an interesting concept of a digital twin [11]. Recently, a new approach to distributing network functionality, namely the fog [12], has become popular, concentrating some resources (nodes, computation power, memory, etc.) close to the most needed places in order to provide real-time reaction.

For ad-hoc networking, nothing is fixed – neither the set of services/nodes nor the information contents. Especially, ad-hoc networking strongly limits fixed ways of addressing network services (resources) by their location and name, leading to the concept of address-free networking [13], which is well known but somehow abandoned recently. Thus, using classical URLs in the ad-hoc system is disputable unless the server-side is fixed and only the clients are connected incidentally. As a consequence, server-side architecture for ad-hoc applications is usually replaced by a mesh. A mesh network [14] is a local area network topology in which the nodes connect directly, dynamically, and non-hierarchically with as many other nodes as possible and cooperate with one another to route data efficiently. A mesh node is not characterized by a permanent identification (e.g., an address); moreover, such a node may suddenly vanish from the network structure or convert to another node due to network evolution, such as the one provoked by the mobility of nodes, amount of available energy, problems with radio transmission, etc. A popular example of a mesh network is BLE (Bluetooth Low Energy) Mesh proposal [15, 16], with several implementations [17] and sub-standards/extensions [18], also related to security and privacy [19], and interoperability [20], addressed to not only classical computers, tablets, or smartphones, but also IoT (Internet of Things) devices based on (among others) Espressif, STM, and Nordic Semiconductor processors with built-in BLE unit.

A question arises if it is possible to apply the RESTful approach with its persistent services, which is very efficient and thus popular in fixed/stable networking, to mesh networking? At first view, the answer is 'no'. Indeed, if we cannot identify a node for a longer time, we cannot establish a client-server connection to ask for a service. However, the above question may be converted to another one: is it possible to address REST-compliant persistent resources in a dynamic way, not using fixed URLs but mapping external REST calls to internal mesh messages of particular format and semantics? Thus, persistent addressing of the resources is to be replaced by a persistent (i.e., stable) description of their functionality. Later on, this fixed description may be used for individual selection of the nodes fulfilling certain criteria. The selection may be performed by the nodes – each node, based on its own declared functionality, accepts or refuses the incoming requests.

In the paper, we propose such an extension to provide RESTful information exchange for a mesh network, especially for BLE Mesh. The idea is to treat a mesh network as a set of REST resources of a given functionality, to be identified not by their URL addresses but by means of their characteristics and possibilities. From the outside, the mesh network is seen as a REST server with specific functionality. Internally, the nodes composing the dynamic network react individually to fulfill incoming REST requests. The node's reaction depends on the individual characteristic of this node, in particular, its type being a counterpart of the REST resource name and the REST query.

The remainder of the paper is organized as follows. Section 2 overviews the motivation for using semi-persistent services in ad-hoc interactions. Section 3 describes a generic approach to mapping REST calls to mesh functionality and the reasons for particular network topology and organization. Sections 4 and 5 describe a sample implementation of kitchenware equipment based on our approach. Next, we include a comparison with similar work, and finally, we provide some conclusions and directions for future work.

II. Specificity of Mesh and Ad-Hoc Networking

If we speak about ad-hoc networking, at first view, we think that every activity is undertaken incidentally: ad-hoc place and time, situation, context, etc. However, that is not the whole truth. There is one element that is fixed – the user. We do not change our needs and expectations just because we are, by coincidence, in an unknown situation. On the contrary, we try to act in a "usual" way, according to our past experience and customs. One may say that we stay with our needs, fixed, as long as it is possible, and we try to act "as usual" even if the case is extraordinary.

Thus, an idea arose to propose a new sort of services for an ad-hoc environment. The goal is to use a service in the same (or at least very similar) way, regardless of place and time. Such services are semi-persistent. In such a way, from the point of view of a user, they are the same everywhere, appearing for this user in the form the user expects. However, from the technical point of view, the services are independent entities, implemented individually and possibly adjusted to the place (conditions) where they work.

In an ad-hoc environment, the users do not know the specificity of the place they are currently in, particularly the set of nodes and their identifiers (addresses, services, and their entry points). However, they know the place's overall character (such as a home, a shop, a bus stop, etc.), and they expect some well-known services accessible at this place. For example, they expect hot water to be prepared in the kitchen or a bus going to a specific destination at the bus stop. Please note that location-specific services are usually well-defined and common for all places of the same type and purpose.

With a classical approach, such as a typical REST application, getting consistent services at many unrelated and unsynchronized places is almost impossible. For example, one cannot expect the same IP address and naming convention for network nodes. Moreover, an installation phase is needed to create some "entry points" for the services, such as installation and configuration of a specific application, catalog of available services and their status, etc. It is unrealistic to expect each place to follow the same rules of addressing and parameterizing, not to say about the security (user identification, access codes, passwords, etc.). As a consequence, we usually deal with one fixed application per place.

This paper proposes a different approach to preparing and accessing such semi-persistent services. We assume that a definition of the semantics of the services is fixed and shared by all the ad-hoc accessible places. Such a semantic description is also known for end-user devices, usually smartphones. Each time a user is at an unknown, ad-hoc place/situation, the description is used to formulate a request, to be disseminated across all the ad-hoc network nodes. If a node (or a set of nodes) "understands" the request, then this node undertakes particular action related to the semantic description, trying to fulfill the request. The network is "silent," and no activity is performed if no single node can provide the service.

Please note that the node should individually choose the "implementation" of the requested service. Only the node knows in detail the specificity of the place, which is unknown to the user. So, the user may only formulate a generic request (such as "turn on some light here"), but how the request would be served (such as "switch on ceiling lamps to 50%") depends on the possibilities and strategy of the place.

As it may be seen, the idea of accessing semi-persistent service in an ad-hoc manner is the following: "try it, and if you are lucky, the service is there for you; otherwise, try a different way or give up". Such experiments are to be undertaken at any unknown (not previously visited, or changed for some reasons) ad-hoc place; however, they are quite natural and intuitive for humans. Moreover, these experiments somehow bypass the installation phase and need no a'priori catalog of at-the-place services.

Initially, we planned to implement our approach as a home application, namely, a "smart" kitchen. Usually, the kitchenware is (1) not synchronized, such as a kettle does not know about a presence of a radio, and an oven is not informed if a ventilator is here to reduce the smell while cooking, and (2) fixed as for the overall functionality (such as “a kettle” or “a refrigerator”), not necessarily fixed as for models/producers/functionality of specific devices. Even if all the “smart” devices are accessible via the same kind of network (which is, in our case – classic Bluetooth, and recently BLE Mesh), these devices are to be registered in a catalog, and in most cases, the smartphone applications are specific for a given model of a device. Each time a device is broken and changed, the user must update the catalog information or install a different application. Keeping with a single application and ad-hoc access to semi-permanent services would solve these problems, on the condition that the services describe the possibilities of typical devices (in our case – the set of devices of a single producer). As a consequence, each application

- will be useful at any “smart” kitchen, no matter its location,
- no additional security checks are needed (the users are granted to use the devices installed at the place they are currently visiting), and
- no installation and cataloging is needed, as well as no “user manuals” for different models of similar devices.

The semantic description of the at-the-place services covers all the details and frees users to learn detailed functionality (for some “smart” devices, quite complex, and, as previously mentioned, model-specific).

The proposed idea is generic and may be adjusted to many places and situations. The kitchenware application could be extended to any place, not necessarily private, but also public. For example, entering a bus stop, one can experiment with a service providing an actual timetable for the buses traveling to/from this bus stop in a minute. While visiting a shop, users may obtain additional information, e.g., the locations of the goods on the shelves, personalized advertisement, etc. At the school, the services may be related to the current schedule of the lectures (solving the “where is my next lecture” problem, etc.). The generic idea is that: entering an unknown place, the users ask for known services.

III. REST MODIFICATIONS TOWARDS THE USAGE IN MESH NETWORKS

As already mentioned, addressing a REST resource via HTTP calls aims in: providing computer identification (node address), declaring port number, defining a path to an internal entity implementing the REST resource (a file, a script, a database entry, etc.), and providing a query to adjust the resource’s behavior. Suppose we divide the above set into the “external” and “internal” parts. In that case, we may separate the node address and port number (“external” parameters), and the rest to be processed “internally” by the node. Further, we may logically link the “external” part with a gateway to the mesh network as a whole and the internal part with the given mesh functionality (to be, in turn, implemented by a set of mesh nodes). Such a global mapping is presented in Fig. 1.

As already mentioned, it is hard to identify a node in a mesh network permanently. Thus, linking REST resources with any node identifier is not justified. Instead, we propose to attach such a resource with certain well-defined, thus somehow persistent, functionality to be dynamically implemented by a node (or nodes). To clear the idea, we propose to join this functionality with a named type and to provide a map of resource names and their types. Types create a hierarchy, being a direct acyclic graph (DAG), precisely, a tree with a single root. Fig. 2 represents a sample hierarchy of types for typical kitchenware equipment we used for the sample implementation of the proposed approach.

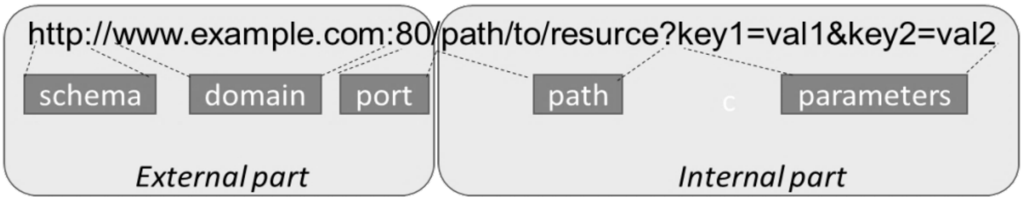

http://www.example.com:80/path/to/resurce?key1=val1&key2=val2
schema
domain
port
path
c
parameters
External part
Internal part

Fig. 1. External and internal parts of URL address

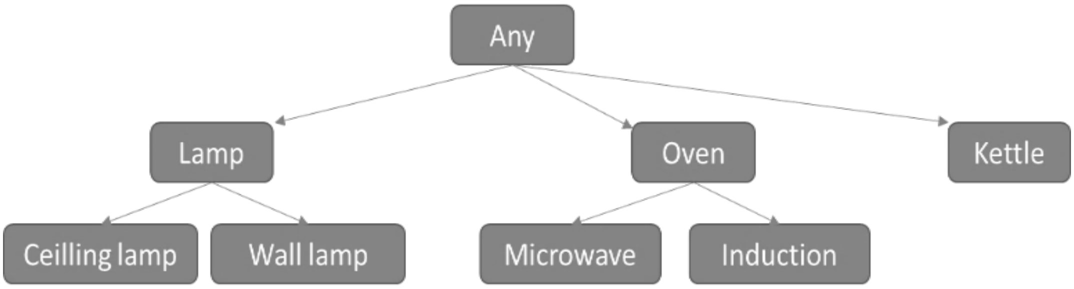


Fig. 2. Sample hierarchy of types of kitchenware equipment

As the types form a DAG hierarchy, they may be specialized or extended, depending on the point of view and the direction of graph analysis. For example, an 'oven' node may group "microwave oven", "gas oven", "induction oven" etc. Please note that all these "ovens" share some functionality, such as "set power to 1000W", and in turn have common functionality with some other electrical devices, such as "run timer". In particular, the root (a device of type "any") could possibly be linked with some generic functionality of any electric device, such as "on", "off", and the above "timer" actions.

According to the REST principle, we specialize some types, such as the above "ovens" share the basic functionality of an oven, and some specific functions of "oven/microwave", "oven/grill", "oven/induction", "oven/induction/plate_NE", etc. Please note that the above naming schema conforms to the REST principle.

The types may also be extended to individual names. For example, a "lamp" can be specialized to a "lamp/ceiling" and a "lamp/wall", sharing exactly the same functionality but differing in, e.g., the place of installation of the physical object the mesh node is connected to. In general, detailed organization and interpretation of the hierarchy depend on the application area and the installation place. As described later in the text, the hierarchy may be reflected by nodes' data, to be processed in a distributed manner – there is no need to represent the hierarchy as a whole by any network node or somewhere in a cloud. Furthermore, last but not least, addressing the primary REST name (such as the "lamp" in the above example) would activate all the devices of the same basic type. If "on" command is sent to all the "lamps", they will be all activated. However, if such a command is sent to "lamp/wall", only this lamp is illuminated.

Note that the above-described mapping shares the basic ideas of object-oriented programming, namely encapsulation, inheritance, and abstraction. Each mesh node may be treated as a singleton, with a well-defined set of "private" variables and functions and a "public" interface. This interface accepts only the "known" (from the node's point of view) incoming messages. As the nodes share the persistent inheritance hierarchy (c.f., Fig. 2), each node may check if the incoming message is of any of the "known" types. In such a way, the node must know both the type (mapped from the REST resource name) and the query (mapped from the REST parameters) to react; however, such a reaction is individual for each node.

As the mesh network structure is dynamic, and we do not provide such functionality as a centralized directory, one cannot apply direct identification of nodes. Thus, unicast transmissions are not possible. Instead, we propose to broadcast all the messages in the "non-responding" way. Each message comes to any node within the radio range. There, a preliminary checking occurs if such a message addresses a type that is linked with the node. If any node's type matches, the node interprets the message according to the query parameters. Appropriate action is undertaken (such as switching the real device linked with the node "on" or "off"). Thus, the nodes may react differently to the same messages. For example, an "alarm" message will close the windows, open the doors, switch on the lights, activate a siren, etc.

As it may be seen, a single broadcasted message may provoke quite complicated behavior of the mesh network, depending on the individual functionality of each network node. On the contrary, some messages are possibly not served once there is no node with certain functionality. For example, if an "oven" device asks for some ventilation, and no node with "a ventilator" type is provided, no air flow is initiated. If, however, at any time, an owner of the network decides to buy a new ventilator, this device will be activated with no changes in the existing network structure and nodes' functionality/hardware/software.

Broadcast messaging limits the way of possible response from the nodes. However, direct responding may be replaced by two succeeding broadcast transmissions. For example, if a device (e.g., a smartphone) is interested in temperature measurement, its node may broadcast a message "get temperature". Thus, each thermometer node reacts by broadcasting the "current temperature" parameterized in the query by the temperature value (Fig. 3). If no "response" broadcast message is observed for a certain period, this means there is no thermometer in the network. On the contrary, if several thermometers exist, they will all send a broadcast transmission with their values. It is up to the caller to get only the first (the highest, the lowest) one or to fetch them all and compute an average value.

IV. System architecture and data flow

We assume that a mesh network is equipped with at least one gateway to any public network (LAN or Internet). The node with the gateway has two network connections (Fig. 4): a public one (such as WiFi or LTE), and a mesh-related one (such as BLE Mesh). The gateway node is responsible for the mapping of incoming REST resources to internal mesh messages representing REST calls and eventually collecting the responses (i.e., reverse broadcast messages coming back within a certain timeout window) from mesh nodes to form an HTTP public response for the external call (c.f., "fridge" node from Fig. 3).

The mapping aims to process the URL address of a REST resource in several steps:

- cutting off the address part and port number,
- mapping resource name to type name,
- passing the incoming query with no changes,
- adding some specific query parameters, such as gateway identifier, the node number of the message sender, etc.,
- encrypting the message using BLE Mesh keys.

The mesh message is sent to any other node in the network (Fig. 5). If the network is big enough, so the relaying is needed, then some nodes may re-send the message to some other nodes. Standard BLE Mesh mechanism is used to relay the messages, and to eliminate incidental message copies in a reasonable time window (usually a second).

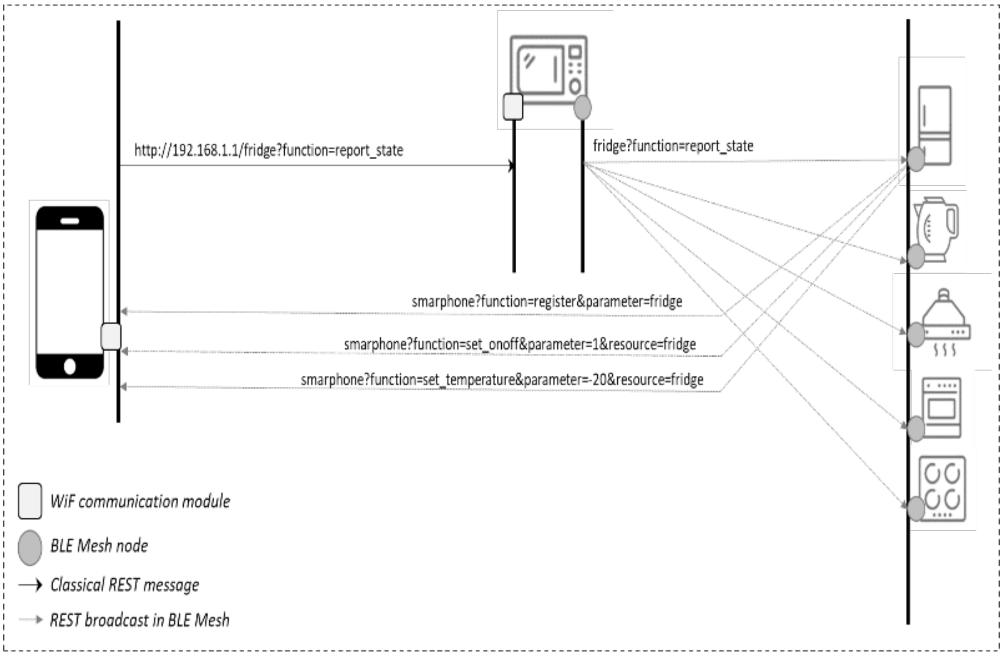

http://192.168.1.1/fridge?function=report_state
fridge?function=report_state
smarphone?function=register&parameter=fridge
smarphone?function=set_onoff&parameter=1&resource=fridge
smarphone?function=set_temperature&parameter=-20&resource=fridge
WiF communication module
BLE Mesh node
Classical REST message
REST broadcast in BLE Mesh

Fig. 3. Typical way of interaction via broadcast messaging

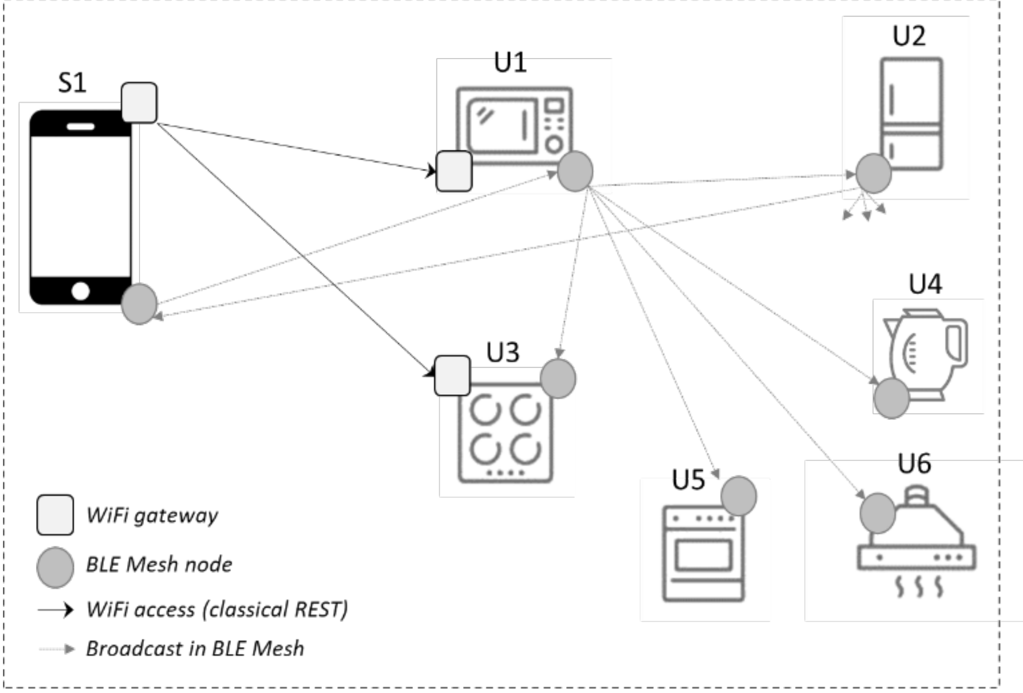


Fig. 4. Mesh network with entry gateways

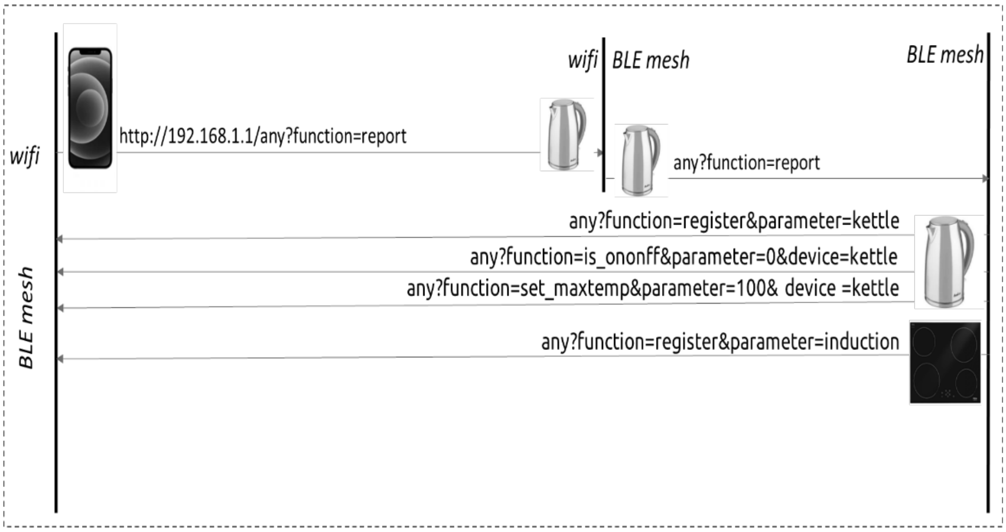

wifi
wifi
BLE mesh
BLE mesh
BLE mesh
http://192.168.1.1/any?function=report
any?function=report
any?function=register&parameter=kettle
any?function=is_ononff&parameter=0&device=kettle
any?function=set_maxtemp&parameter=100& device =kettle
any?function=register&parameter=induction

Fig. 5. Mapping messages in the gateway node

Each node, after receiving the mesh message, is responsible for (1) unfolding the non-standard characters using the standard *URLdecode()* function, (2) checking the accordance of the type, and (3) if the message type conforms to any type declared for the node, consume the message and start appropriate action(s) addressed to the real-world entity the node is connected with.

Due to the strategy used by BLE Mesh, the messages must be exchanged in the scope of so-called models. Each model is responsible for managing the messages of a certain structure and semantics. A model defines a set of states, state transitions, state bindings, messages, and other associated behavior. Each node must support one or more models, and these model or models define the node's functionality. There are two basic types of BLE Mesh models. Special Interest Group proposes SIG models [21] as a set of well-defined models of fixed functionality (starting from such simple models as Generic ON/OFF, to quite more complicated such as battery-level monitoring). These models are usually implemented as a part of the BLE Mesh library. The second group of models is called vendor models. Vendor models use a slightly different mode of identification of messages (longer identifiers). Thus they are sometimes less efficient. However, these models are not standardized, so in theory, they may represent an arbitrary set of additional services (functions) of the node, prepared by the network designer.

In the next section, we describe our reasonable for proposing the implementation of a certain set of BLE Mesh models, in turn, to implement REST messaging in the mesh network.

V. Implementation issues

As described above, one has to propose a model to exchange the information in the scope of the BLE Mesh network. The model comprises both message syntax (i.e., data length) and semantics (i.e., data type and location in the buffer). We decided to provide two implementations: a new SIG model and a vendor model. The implementation was based on Nordic Semiconductor processors (namely nrf52840) and NS BLE Mesh library, and Espressif processors (ESP-32 WROOM-E, ESP-32 S3) programmed within the IDF framework. In addition, we adapted the Nordic library for BLE Mesh for Android smartphones to administrate/test the network. The adaptation aimed in implementing the same models we designed for BLE Mesh as an extension to Nordic library. Please note that the main reason for choosing ESP-32 and nrf52840 processors was based on the fact they have a built-in BLE unit. Obviously, the solution would work for non-BLE processors such as most of the STM family with external BLE units.

It was soon detected that the choice of implementing the model as a new member of the SIG-model group was not justified. Even if, in theory, such implementation is more efficient due to shorter identifiers, it was soon detected that the library lacks documentation about some programming tricks applied by SIG programmers. For example, fixed tables of semaphores and fixed maps of model identifiers were used and indexed by model identifiers. As a result, the library itself needed to be rewritten and re-compiled. Such an adaptation should be performed every few months as a new library version comes unless SIG agrees to incorporate the model into the set of their models (which has not been planned so far).

Thus, the work concentrated on implementing a vendor model, which was less efficient, however, with no reason to update basic libraries. In contrast to any of the SIG models, the model is based on strings of characters of variable length (textual messaging). Thus, the model is quite generic, which is not the case with the existing models, and the semantic analysis of the message is to be performed at the application level. To this goal, a dedicated interpreter has been designed. Fetching a textual REST request at the input, the interpreter produces a union at the output containing the type identification (a node of the graph of the hierarchy of all types), subtype, and a list of query parameters in the form <name, value>, where "name" is a string of characters, and "value" is a number or a string of characters. "Subtype" needs more explanations. This parameter is a difference between the most-specialized type name within the hierarchy of types and the type identifier used to formulate the REST request. For example, the identifier "kettle/silver" identifies "kettle" as the basic type (as this is the most specialized node in the hierarchy of types addressing "kettle" identification), and "silver" as a subtype. Using subtypes is similar to the usage of individual names of real-world artifacts connected to the mesh nodes. It is up to each node to interpret not only the type name but also the subtype, for example, to distinguish the artifacts of the same type but different meanings or locations (such as "tv/kitchen" and "tv/mainroom"). Please note that type and subtype names conform to the REST approach to resource naming.

Queries are encoded according to the REST standard (so-called URL-encoding schema). The query addresses the detailed functionality of a node and is also interpreted at-the-place, similar to the traditional REST approach. The query elements (names and domains/formats) are not standardized. Instead, it is up to the caller to formulate the query so the mesh nodes can understand it.

Note that our textual BLE Mesh model somehow replaces any previous models. Formulating the types and queries is relatively straightforward in such a way that they are equivalent to any other call. For example, instead of the time-synchronization model, one may apply "any?setTime=12:34:56" message, asking for setting the time at each node ("any" is the root of the hierarchy of all types, cf. Fig. 2).

We also found that sometimes knowledge is needed about the existence of the nodes of a certain type in the network. Returning to one of the previous examples, a "ventilator" should inform all the "ovens" that it is possible to force the airflow, in case it becomes too hot, or a smell is irritating. To this goal, it is enough to periodically broadcast a message stating each node's type and current state. For example, the ventilator could broadcast "any?airFlow=1" informing any other node about its current state. Again, the interpretation of such information depends on each node's needs and capabilities, and the semantics of the query elements needs to be known for all the nodes interested in such information. For the rest of the nodes, such messages will be ignored as pointing to an unknown action to be done, similar to the traditional interpretation of REST requests.

VI. Sample implementation – a kitchenware application

As already mentioned, our implementation targeted "an intelligent kitchen" idea. We took several assumptions to characterize the needs of such specific networking better:

- in general, within a single network, we use one device of a given type, such as a "kettle", a "microwave_oven" etc. It is possible to use subtypes to differentiate several instances (entities of the real world) of the same type, as described in the previous section, if needed. If not parameterized, all devices of the same type react to the incoming message (such as the message "lamp?on=1" will switch on all the lamps in the location);
- we use classical RESTful usage of the mesh network as a whole, via a WiFi gateway implemented in one selected mesh node (in our case – a kettle) and a dedicated Android application connected to the same WiFi network; the Android application is equipped with some extensions to facilitate user interfacing, such as voice analysis/synthesis;
- we apply mapped REST calls (broadcasts) inside the network to control the devices in a predefined manner (any node knew its type and meaning of the query elements);
- there is no centralized directory of network services; however, some devices (especially a smartphone) may specialize in grouping/providing some information about some other devices, if needed;
- we rely on the security and relaying mechanisms from BLE Mesh (in particular, the provisioning of each new node);
- we apply periodic broadcasts of each device's state (message content depending on the device's functionality and type) to enable Android application planning and controlling the kitchen activities as a whole.

As already mentioned, we use a smartphone (Android-based) as a provisioning center and several network nodes based on processors with BLE support, mainly Espressif ESP 32 WROOM-32D/E, ESP-32 S3 (BLE 4.x), and Nordic Semiconductor nrf52840 (BLE 5.x). Selected Espressif nodes also serve as WiFi gateways. These nodes broadcast their IP addresses to all the other nodes. Note that the overall WiFi security is not broken here, as the broadcast messaging is encoded with BLE Mesh keys, and thus it is readable only by the members of the BLE network.

During the tests, we found that the smartphone application based on Nordic Semiconductor's library is insufficient for contacting the device. The library itself is huge, and the application consumes a lot of energy for transmission (both BLE and WiFi connections are active all the time). Thus we apply a mixed mode – the application listens to BLE broadcasts and sends all the requests to the network to any of the available WiFi gateways. Even if strange at the very first view, such a mixed mode was found as quite efficient and easy to implement. The communication mode depends on end-user strategy: WiFi calls are less efficient but more straightforward to implement (standard HTTP calls and REST messaging), BLE messaging needs a separate implementation of the new mesh model, but then the IP address is not necessarily known. The choice should be left to the users.

The above-mentioned application was also used as the main provisioning center for the network. As already noticed, we soon found such a way of provisioning as non-efficient, and we are working now on a new approach to provision the new nodes by any existing node from the network. Once completed, such distributed provisioning will eliminate the need for a smartphone as a network node. This work is not finished yet; it also needs some hardware extensions, such as WPS buttons known from WiFi access points. Once the work is finished, it is to be described in a separate paper. To our best knowledge, no proposal exists for dynamic provisioning within a single mesh network, where any node may act as a provisioner. Please note that we cannot rely on a single predefined provisioner, as this node may be temporarily out-of-network or even gone, thus preventing any new node to join. The above problem is also linked to the so-called "newcomer" problem, i.e., how to find an entry point to access the network for the first time. This problem also needs particular attention; we plan to work on it using intelligent BLE beacons.

Our implementation aimed to design an "intelligent" kitchen. Thus, besides the smartphone as a provisioning center, we linked the mesh nodes with the following kitchenware: a kettle (this device was equipped with a touch screen to serve as a main network node), an induction plate, a ventilator, a lamp, radio with an MP3 player.

The touch screen of the kettle could be converted to act as an interface of any network node to (1) facilitate the user interface (the interface was unified for all the devices regardless of their type, but taking into account their specificity), and (2) limit the costs of the "intelligence" of the devices to a reasonable minimum. We implemented the following automatic messaging:

- the radio automatically increases its volume while the kettle is finishing boiling the water,
- the ventilation is started after 10 minutes of using an induction field with a level greater than 50% of the maximum,

- the kettle sends an alert to the MP3 player when hot water is ready,
- the ventilation is stopped if none of the devices was used (i.e., activated) for the past 30 minutes.

The above functionality may be programmed by the producer of the devices, not necessarily the end-user. This is only a sample set of actions to be performed automatically by the network. Any new device may be incorporated at any time with no changes in the code/variables of any other device. There is no centralized directory of services, global coordinator, "main" node (maybe except the kettle – but only for economic reasons), etc. There is also no need for the "installation" of any new device (except for the provisioning process we plan to improve, as already mentioned).

Please note that our network is neither a classical sensor network nor a MANET one. This is also not quite an ad-hoc network. Even if the nodes may be switched on/off at the temporary base and coincidentally, apart from such switching, they are relatively stable – always at the same place, and usually with the same (fixed) functionality. Some nodes are used frequently, some all the time, some at request, and some rare, but the set is not evolving very often (unless a new device is bought or an old device is broken). Thus, our approach is well-suited to the above circumstances.

VII. Comparison with previous work

While we started the research for similar work to be compared with our proposal, we found that most of the existing proposals concentrated on the "smart home" idea, with an engineer's design of ready-to-market products. Some of these proposals were related to patent applications, demonstrating a growing need to provide such solutions on the market. Although very popular in the scope of client-server architecture and classical applications, we discovered that the REST idea was hardly used for controlling the behavior of home appliances and systems except the proposals based on WiFi traffic (classical REST applications). To our best knowledge, no single proposal exists for using REST like addressing in Bluetooth-based mesh networking. The existing proposals aimed in using specialized, centralized directories of services capable of mapping WiFi calls to Bluetooth direct (paired) communication. Thus, we compared our proposal with similar proposals for efficiently managing "smart" systems and devices used at home, especially in the kitchen.

Known solutions for communication and control of home devices use a centralized home network in which the entire network transmission is supervised by a center, which is most often a specialized router, sometimes a smartphone, or the most advanced, always-on home device, for example, a refrigerator. In a network with such a topology, all data sent over the network must be sent by the central node, and each device must be installed and registered in this node before its first use.

Chinese patent specification CN109218098A [22] discloses a smart home control method, a radio transmission network gateway, and a smart home control system. The gateway is the primary authorization center and the primary point that limits the functionality and capacity of the entire network. It is a typical centralized solution that requires installation in the network and registration of each new device and uses the traditional method of addressing the devices.

Another patent specification CN109088994A [23] discloses a solution in which a smartphone is used to establish a connection with one device equipped with a Bluetooth communication module at a time. The application can transcribe the command given via the graphical or voice interface to the commands sent via Bluetooth to the currently connected home device. The application also acts as a central directory of available devices that must meet very strict requirements as to the type and method of data transmission, which in practice limits the number of such devices only to the list prepared for the purposes of this invention. A severe functional limitation of this system is also that home devices cannot directly exchange any information with each other. The data must be transferred only to and from the smartphone.

The invention described in document CN109981776A [24] concerns a system that solves the problem of the limitations of the classic Bluetooth communication channel, i.e., "exclusive" operation over an established link in this type of transmission. Each device has several Bluetooth transmission modules, the first of which is used to synchronize access to the others. In this way, by establishing connections in the other modules for a while, one can transmit data between any pair of network devices through them. The biggest disadvantage of this solution is the necessity to install many communication modules in each device. A similar solution is depicted in [25].

Document CN111585855A [26] discloses a system that uses a smart wireless router integrated with a WiFi module, an infrared module, a ZigBee module, and a Bluetooth module, which enables address communication with any device available through supported forms of communication. The system does not provide for direct communication between these devices without the use of the above-mentioned router.

Contrary to all the above-presented (and similar) proposals, this paper describes a system and method of communication with intelligent home devices via a network without a central point, in which each device is equally privileged. Automatic communication among the devices is possible without human intervention. In addition, external requests are formatted in a way to replace node addresses with semantic names of functions performed by devices.

The purpose has been achieved with the use of a Bluetooth mesh topology network, in which communication among network components is possible without the need to involve the central unit. In such a network, each network device is equally privileged and can communicate with any other device directly or via any other network component. Selected devices, equipped with alternative communication modules of a different type, in particular, connected to the home WiFi network, can be used as input/output gateways for communication outside the mesh network. Transmission in the network occurs in a broadcast mode, in which one node sends information that is then received by all other nodes.

No device is individually addressed in the proposed solution, so no central directory is needed. Installing each new or replaced device or monitoring the device status is unnecessary. The transmitted signals contain digital information formatted in accordance with the REST software architecture style with respect to the fact that the addresses of devices are to be replaced with semantic names of functions performed by devices. For each device, one can define any set of functions for which that device will be responsible. The naming of the functions corresponds to the REST resources naming.

The proposal responds to the idea “different locations, similar usage”. There is no installation needed, such as when users buy new equipment, these devices are, from the very start, ready to use. Moreover, we detected un unexpected add-on while working with the smart kitchen: a possibility of targeted, personal marketing for non-existing devices. Once a requested functionality is not achieved, any other device (in our case it was the most complicated and advanced one – a kettle) may detect this fact, contact the cloud for a possible solution and broadcast some advices in the local network. As a consequence, the users are informed in JIP (just-in-place) and JIT (just-in-time) manner, thus increasing the probability of taking a decision and buying the missing equipment.

Our semi-persistent services are a response to fixed needs of the users changing places (such as several locations of a single family, or family helper), or changing organization of a mesh at a single place (such as a home or a kitchen) without a need for registration and tracking the current status of the services. Moreover, the same application may be used to control a smart kitchen, and near-by – a home audio-video system.

VIII. CONCLUSIONS

In the paper, we proposed an adaptation of the popular REST approach to BLE Mesh networking and address-free traffic. The idea enables a system and method of communication with home appliances and components of home infrastructure, such as household appliances, audio/video, lighting, heating, and air-conditioning equipment, in order to control these devices in a decentralized manner.

We use double-mode communication with home devices connected via microcontrollers to the BLE Mesh network, and WiFi. Connecting a smartphone or a computer device (e.g., a laptop or a tablet) is also possible. Gateway nodes automatically map REST messaging from WiFi networking to internal BLE messaging, conforming to REST strategy and using a hierarchy of types instead of resource names. The types enable semantic interpretation of the REST messages to be used dynamically for the such variable environment as a mesh network.

Messaging of REST-compliant communication is implemented as a broadcast transmission in the scope of a dedicated vendor model of a BLE Mesh network, implemented for the popular BLE microcontrollers as well as Android smartphones. The model is equipped with an interpreter of the incoming messages based on the semantics of the type (basic part of REST resource name), subtype (additional elements of REST resource name), and REST query complemented with some network-specific parameters. The query is provided as a list of parameters of the "name-value" type, where "name" means a command to activate a function, and "value" is a parameter of such a command. The requested functions specified in the information sent are activated for the real kitchen device associated with a given resource type, and parameterized by the query. Each device is responsible for (1) filtering all the incoming messages by their types and the accordance with the type(s) declared for the device, and (2) interpreting the query parameters. The filtering and the interpretation are programmed in the control code of the devices, and preferably switched on/off by end-users.

The devices associated with the network nodes do not have any information about the other devices on the network, including whether the device is active on the network or not. A message with a command sent by a device to an inactive device on the network does not affect the operation of the device sending the command. In the embodiment of the invention described above, the induction cooktop will not stop cooking if the cooker hood does not turn on. Still, if the cooker hood is active on the network, it will start automatically if needed.

The devices on the network are identified only by the name of the currently assigned resource and the set of functions assigned to that resource. The address or location of the device is not required for communication between devices on the network. The device cooperation can be programmed by placing appropriate REST messages in the device microcontroller code. Replacement of one device model with another will not require reprogramming these microcontrollers, and the network as a whole will work the same despite the changes.

The system is fully implemented and tested for the devices produced by Polish biggest kitchenware manufacturer Amica. It was also a base for a European patent application [27].

As for future work, we plan to extend the proposal to address the problem of the “newcomer”. If the users visit an ad-hoc location for the first time, they are not informed about the possible services to be accessed there. Thus, some experiments are needed to determine which services are available. To minimize the time spent for these experiments, we plan to include a so-called "advertisement channel" and

BLE beaconing [28] to broadcast some information for early detection of the services and their types.

We also plan to apply one of the well-known ontologies of IoT devices to provide some generic services in public places. An obvious candidate for such a service is a thermometer, but also UV-meter and PM-* detectors, to be used primarily to protect people with asthma and similar diseases. So far we concentrated on smart kitchen, but the number of ontologies (as well as the level of their complexity) for this application area is limited. If, however, we plan to extend our approach to some public places and common devices, using such external ontology is a must. Selecting given ontology depends on the application area, but our approach makes it possible to address as many different ontologies as it is needed, by a selection of an "optimal" type hierarchy.

We also plan to optimize the process of validating access to the BLE Mesh network, so-called provisioning [29], aimed at exchanging encryption keys. So far, a dedicated mesh node called a provisioner has been used for this goal. However, this node may be temporarily inaccessible for many reasons, thus preventing new users from entering the network. Thus, we plan to apply the provisioning function to any node and dissipate the necessary information among the other nodes, dynamically voting for the best "candidate" for at-the-moment provisioning.

REFERENCES

[1] R. Fielding, Representational State Transfer (REST), Ph.D. dissertation, [Online] available: https://www.ics.uci.edu/ ~fielding/pubs/dissertation/ rest_arch_style.htm, 2000

[2] L. Gupta, What is REST - REST API Tutorial, [Online] available: https://restfulapi.net/, 2022

[3] What is a URL?, Mozilla documentation, [Online] available: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/ Web_mechanics/What_is_a_URL, 2023

[4] R. Ramanathan, J. Redi, Overview of ad-hoc networks: challenges and directions, IEEE Comm, Volume 40, Issue 5, DOI 10.1109/MCOM.2002.1006968, 2002

[5] S. Al-Sultan,M. M. Al-Doori, A. H. Al-Bayatti, H. Zedan, A comprehensive survey on vehicular Ad Hoc network, Journal of Network and Computer Applications, Volume 37, Pages 380-392, 2014

[6] M. S. BenSaleh, R. Saida, Y. Hadj Kacem, M. Abid, "Wireless Sensor Network Design Methodologies: A Survey", Journal of Sensors, vol. 2020, Article ID 9592836, [Online] available: https://doi.org/10.1155/2020/9592836, 2020

[7] A. O. Bang, P. L. Ramteke, MANET: History, Challenges, and Applications, International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 9, ISSN 2319 – 4847, 2013

[8] N. Temene, C. Sergiou, V. Vassiliou, C. Georgiou, A Survey on Mobility in Wireless Sensor Networks, Ad Hoc Networks, Volume 125, 1(February), [Online] available: https://doi.org/10.1016/ j.adhoc.2021.102726, 2022

[9] R. Zagrouba, A. Kardi, Comparative Study of Energy Efficient Routing Techniques in Wireless Sensor Networks, Special Issue Wireless IoT Network Protocols, Information, 12(1), 42; [Online] available: https://doi.org/10.3390/info12010042, 2021

[10] M. Wang, Y. Chen, X. Sun, F. Xiao and X. Xu, "Node Energy Consumption Balanced Multi-Hop Transmission for Underwater Acoustic Sensor Networks Based on Clustering Algorithm," in IEEE Access, vol. 8, pp. 191231-191241, doi: 10.1109/ACCESS.2020.3032019, 2020

[11] Z. Lv, D. Chen, H. Feng, W. Wei, H. Lv, Artificial Intelligence in Underwater Digital Twins Sensor Networks, ACM Transactions on Sensor Networks, Volume 18, Issue 3, Article No. 39, pp 1–27, https://doi.org/10.1145/3519301, 2022

[12] A.M. Rahmani, P. Liljeberg, J.-S. Preden, A. Jantsch, Fog Computing in the Internet of Things - Intelligence at the Edge, Springer International Publishing AG, [Online] available: https://doi.org/10.1007/978-3-319-57639-8, 2018

[13] J. Elson, D. Estrin, An Address Free Architecture for Dynamic Sensor Networks, Research Gate, [Online] available: https://www.researchgate.n et/publication/2618136_An_Address-Free_Architecture_for_Dynamic_Sensor_Networks/citation/downlsoad , 2000

[14] A. Cilfone, L. Davoli, L. Belli, G. Ferrari, "Wireless Mesh Networking: An IoT-Oriented Perspective Survey on Relevant Technologies". Future Internet. 11 (4): 99, doi:10.3390/fi11040099, 2019

[15] Y. Junjie, Y. Zheng, C. Hao , L. Tongtong , Z. Zimu , W. Chenshu, A Survey on Bluetooth 5.0 and Mesh: New Milestones of IoT, ACM Transactions on Sensor Networks, Volume 15, Issue 3, Article No. 28, pp. 1–29, https://doi.org/10.1145/3317687, 2019

[16] M. Baert, J. Rossey, A. Shahid, J. Hoebeke. The Bluetooth mesh standard: An overview and experimental evaluation. Sensors (Basel, Switzerland) 18, 8(July), p. 2409, 2018

[17] M. Collotta, G. Pau, T. Talty, and O. K. Tonguz. 2018. Bluetooth 5: A concrete step forward toward the IoT. IEEE Communications Magazine 56, 7 (July), pp. 125-131, 2018

[18] M.R. Ghori, T.-C. Wan, G.C. Sodhy, Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols, Sensors, 20, 3590, [Online] available: https://doi.org/10.3390/s20123590, 2020

[19] A. Lacava, V. Zottola, A. Bonaldo, F. Cuomo, S. Basagni, Securing Bluetooth Low Energy networking: An overview of security procedures and threats, Computer Networks, Volume 211, 5, 2022

[20] M. Noura, M. Atiquzzaman & M. Gaedke, Interoperability in Internet of Things: Taxonomies and Open Challenges, Mobile Netw. Appl. 24, 796–809, [Online] available: https://doi.org/10.1007/s11036-018-1089-9, 2019

[21] Bluetooth Mesh Models - A Technical Overview, official Bluetooth documentation, [Online] available: https://www.bluetooth.com/ bluetooth-resources/bluetooth-mesh-models/, 2023

[22] Shenzhen Zhizhen Science And Technology Co Ltd, A kind of connection and configuration method of home gateway, China patent application CN109218098A, [Online] available: https://patents.google.com/ patent/CN109218098A/en?oq=CN109218098A, 2019

[23] Lanzhou University of Technology, Based on smart phone and single-chip microcontroller intelligent miniature household method, China patent application CN109088994A, [Online] available: https://patents.google.com/patent/CN109088994A/en?oq=CN109088994A+, 2018

[24] Foshan Shunde Midea Washing Appliances Manufacturing Co Ltd, Intelligent control equipment, the networking control method of household appliance and system, China patent application CN109981776A, [Online] available: https://patents.google.com/patent/ CN109981776A/en?oq=CN109981776A+, 2019

[25] K. Takada et al., Communication apparatus, communication system, notification method, and program product, United States Patent US 9,497,629 B2, [Online] available: https://patents.google.com/patent/ US9497629, 2014

[26] Bowei Technology Co ltd, Intelligent wireless router and intelligent home system, China patent application CN111585855A, [Online] available: https://patents.google.com/patent/CN111585855A/en?oq= CN111585855A+, 2020

[27] J. Rykowski, T. Jenek, W. Switala, System and method of communication with home devices, European patent application EP 4 132 034 A1, [Online] available: https://data.epo.org/publication-server/rest/v1.0/publication-dates/20230208/patents/ EP4132034NWA1/ document.pdf, 2022

[28] Estimote Inc., How do beacons work?, Estimote dcoumentation, [Online] available: https://community.estimote.com/hc/en-us/articles/360002656512-How-do-beacons-work

[29] K. Ren, Provisioning a Bluetooth Mesh Network Part 1, Bluetooth documentation, [Online] available: https://www.bluetooth.com/blog/ provisioning-a-bluetooth-mesh-network-part-1/

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 721–730
DOI: 10.15439/2023F5651
ISSN 2300-5963 ACSIS, Vol. 35

Exception Handling in Programmable Controllers with Denotational Model

Jan Sadolewski
0000-0001-7370-9027
Department of Computer and Control Engineering
Rzeszow University of Technology,
al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland
Email: js@kia.prz.edu.pl

Bartosz Trybus
0000-0002-4588-3973
Department of Computer and Control Engineering
Rzeszow University of Technology,
al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland
Email: btrybus@kia.prz.edu.pl

***Abstract*—The paper introduces a customized approach to handle failures in IEC 61131-3 programmable controllers. The solution assumes the utilization of a virtual machine as a runtime environment to execute control code in an isolated manner. A formal model of the runtime is presented, employing denotational semantics. Subsequently, the model is expanded by incorporating new procedures that enable the handling of runtime exceptions using ST code constructs. This formal model serves as the foundation for implementing the exception infrastructure in the CPDev development environment. The research presented in the paper, driven by industry demands, aims to facilitate the development of more reliable and resilient control systems, capable of effectively dealing with failures.**

I. Introduction

PROGRAMMABLE Logic Controllers (PLC) and Programmable Automation Controllers (PAC) have established their position in the modern world. Their applications are wide-ranging, including industrial production, energy, transportation, and Internet of Things solutions. They control processes, often using advanced algorithms, and are expected to be reliable and operate in real-time mode.

One characteristic that allows for the flexibility of these devices is the ability to program them, where an engineer creates their own control algorithm and places it in the controller. From this perspective, such a controller is versatile as it can be programmatically tailored to specific applications and its functionality can be extended or modified when changes are required in the controlled object.

In some controllers, programming is done using typical languages, most commonly C/C++. However, there are standardized mechanisms and programming solutions specifically designed for control devices. The most significant of these are the international standards IEC 61131-3 [1] and IEC 61499 [2]. Particularly, the former has become a recognized standard adopted by many manufacturers. It allows, among other things, the transfer of program code between devices from different vendors and introduces specialized programming languages such as structured text (ST), instruction list (IL), graphical block diagram (FBD), ladder diagram (LD), and sequential function chart (SFC). The language structures align well with control system programming paradigms. Hence, this article considers a system that complies with the IEC 61131-3 standard.

The stable and predictable operation of PLC and PAC controllers is a feature resulting from their applications, where errors and unexpected reactions can have serious consequences. To minimize the risk of such situations, developers employ various solutions. The mere use of the standard's languages can reduce potential problems, particularly by avoiding programming mechanisms related to manipulating pointers and dynamic memory allocation. Another solution involves constructing an isolated runtime environment for user programs. In such cases, the effects of programmer errors will not propagate beyond that runtime, allowing the device to remain operational and enabling controlled handling of exceptional situations.

One concept for creating such an isolated environment is a virtual machine [3]. In general terms, a virtual machine (referred to as VM) is understood here as a type of processor with its own instruction set and data types, implemented through software on specific hardware platforms. This means that when processing code designed for a VM, appropriate software mechanisms execute it using the native resources of the target platform, such as a specific CPU and memory. The VM processes code, typically referred to as intermediate code, which is generated by a compiler from a source program. The concept of virtual machines has gained prominence in information technology due to the widespread use of platforms such as the Java Runtime Environment [4] and the .NET Framework [5], [6].

Solutions based on virtual machines offer several important advantages. Firstly, the source program and intermediate code are independent of the target hardware platforms. This means that only one compiler for the source language is required, rather than separate cross-compilers for different platforms. Additionally, programs are executed within secure environments with memory protection, preventing potential errors from propagating beyond the designated boundaries.

However, there are also disadvantages to consider. Execution of intermediate code tends to be slower compared to executing native code on the target processor. This is because the instructions and operands of the intermediate language need to be decoded by software, whereas a standard CPU utilizes hardware decoders and pipelining. Consequently, implementing even a simple intermediate instruction requires multiple native instructions.

When designing a virtual machine as a runtime environment prepared to handle error and exceptional situations, several aspects need to be taken into account. The first aspect is the compatibility of the machine's operation with its specification. For this purpose, the authors have proposed a formal model of operations performed by the virtual machine using denotational semantics [7]. This model enables the implementation of these operations in accordance with the assumptions, for example, in languages like C/C++. In this article, the model has been expanded to include functions related to exception handling.

Another task is to supplement the languages of the IEC 61131-3 standard with additional constructs related to exception handling. This is necessary due to the lack of dedicated solutions in the standard. Hence, there is a need to introduce them. The authors take into account the extensions to the ST language available in the CODESYS package but propose their own implementation of these extensions.

II. Programming and runtime environments

The engineering environment CPDev (Control Program Developer) allows to program controllers according to the IEC standard [8]. It consists of ST/IL/FBD/LD/SFC editors, a compiler translating programs to the intermediate code [9] and a VM-based runtime system written in C/C++ [10].

The architecture of the VM is shown in Fig. 1. It includes the following components:

- code and data memories,
- code and data stacks,
- registers and pointers,
- instruction processing module.

The *Instruction processing* module fetches successive instructions from *Code memory* and executes them acquiring values of operands either from *Data* or *Code memory*. Results are stored directly in *Data memory*.

The machine does not utilize an accumulator, however it maintains other registers. The instruction pointer, also known as the program counter, is stored in the *CodeReg* register. VM increments the *CodeReg* register every time after fetching an instruction code or an operand address. The *DataReg* register is used for managing the data base addresses and is set during subprogram calls and returns, including function blocks and functions. This allows the executed code to access variables in different areas of the *Data memory* and handle multiple instances of subprograms. When entering a subprogram, the current values of *CodeReg* and *DataReg* are pushed onto the *Code stack* and *Data stack* respectively. Upon returning, the contents of these registers are popped from the stacks. This stack mechanism enables nested function blocks. Additionally, the machine includes the *Flags* register, which contains status flags that signal errors or unusual situations such as an array index outside the valid range, an unknown instruction code, or a cold start.

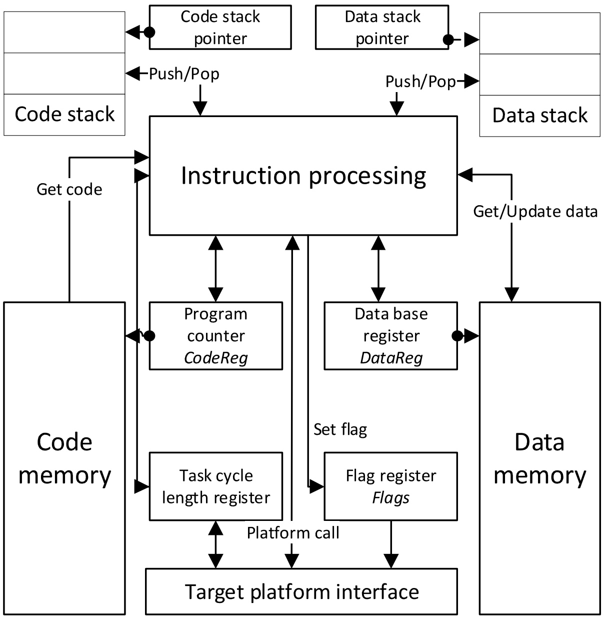


Fig. 1. Architecture of the virtual machine

The virtual machine, as designed specifically for execution of control programs, can handle all IEC 6131-3 data types. The number of bytes required to store each such type in the data memory is given in Table I.

There are two kinds of virtual machine instructions:

- functions,
- system procedures.

Examples of some functions are shown in Table II with decreasing priority. The functions return one value each to be written into the variable being the first operand (as said, an accumulator does not exist in this VM) and may have up to 15 other operands. Note that such order is different than in *Static Single Assignment* of dataflow graphs used in typical compilers [11]. Arithmetic operations are executed in limited ranges, depending on the type. In case of integers the ranges are $(-128,\ 127)$ for SINT, $(-32768,\ 32767)$ for INT, etc.

Contrary to functions, system procedures do not return values or return more than one. Table III shows typical examples. The procedures control program flow, handle memory, call

TABLE I
Data types and number of bytes

Types	Bytes
BOOL, BYTE, SINT, USINT	1
INT, UINT, WORD	2
REAL, DINT, UDINT, DWORD	4
DATE, TIME, TIME_OF_DAY	4
LREAL, LINT, ULINT, LWORD	8
DAY_AND_TIME	8
STRING, WSTRING	var

TABLE II
FUNCTIONS OF THE VIRTUAL MACHINE

Mnemonic	Meaning	Operator
EXPT	Power	**
NEG	Negation	− (unary)
MUL	Multiplication	*
DIV	Division	/
ADD	Addition	+ (arithm.)
SUB	Subtraction	− (arithm.)
CONCAT	String join	+ (text)
GT	Greater	>
GE	Greater or equal	>=
LE	Less or equal	<=
LT	Less	<
EQ	Equal	=
NE	Not equal	<>
AND	Logical and	&

subprograms, etc. From the programmers' viewpoint there is no major difference in using functions or procedures.

TABLE III
SYSTEM PROCEDURES OF THE VIRTUAL MACHINE

Mnemonic	Meaning
JMP	Unconditional jump
JNZ	Conditional jump
JR	Unconditional relative jump
JRN	Conditional relative jump
CALB	Subroutine call
RETURN	Return from subroutine
MEMCP	Copy memory block
MCD	Initialize data
FPAT	Fill memory block
GARD	Copy global memory to local area
GAWR	Copy local memory to global area

To accommodate exception handling, the architecture of the virtual machine needed to be expanded. The modifications mainly revolved around the protected code stack, which will be thoroughly explained in Section IV.

III. Runtime formal model

A formal model has been developed to specify the operation of the virtual machine using denotational semantics [12], [13]. The fundamental aspects of the model were outlined in the previous publication [14]. In this context, we now expand upon the description of the model components. The model consists of various domains that define the states, memory functions, value interpreters, limited range operators, and a universal semantic function that invokes specific functions representing individual instructions.

The domains within the model encompass abstract data types that represent the values processed by different components of the virtual machine (Fig. 1). One such domain, denoted as $BasicTypes$, comprises four sets that correspond to the memory sizes of the basic data types outlined in Table I. Another domain, named $Address$, specifies the size of addresses associated with data or code memory. The domain $Memory$, maps $Address$ to $OneByte$. Both $CodeMemory$ and $DataMemory$ are aliases for the $Memory$ domain. The domain $Stack$ represents a sequence (indicated by *) of $Address$ domains, with $CodeStack$ and $DataStack$ serving as aliases for specific stack types. The other domains are defined similarly.

$$\begin{aligned}
BasicTypes &= OneByte + TwoBytes + \\
&\quad + FourBytes + EightBytes \\
Address &= FourBytes \\
Memory &= Address \to OneByte \\
CodeMemory &= Memory \\
DataMemory &= Memory \\
Stack &= Address^* \\
CodeStack &= Stack \\
DataStack &= Stack \\
CodeReg &= Address \\
DataReg &= Address \\
Flags &= TwoBytes
\end{aligned}$$

Broadly speaking, the goal of program execution is to transition the current state of the computer into a new state. In the context of the virtual machine, the state is represented as a Cartesian product of various domains including memory, stacks, registers, and flags. More specifically, the domain denoted as $State$ can be understood as a collection of tuples $(cm, dm, cs, ds, cr, dr, flg)$, where each element corresponds to a value within its respective domain.

$$\begin{aligned}
State = {} & CodeMemory \times DataMemory \times \\
& \times CodeStack \times DataStack \times \\
& \times CodeReg \times DataReg \times Flags
\end{aligned}$$

The functions presented below model low-level operations executed on memory, stacks and flags.

- Get data from memory
  $$Get1BMem = (Address \times Memory) \to Byte$$
  $$Get2BMem = (Address \times Memory) \to TwoBytes$$
  $Get4BMem$, $Get8BMem$, etc. are defined similarly.
- Get address from memory
  $$GetAddress = (Address \times Memory) \to Address$$
  The function returns the value stored at the given $Address$ in $Memory$ which is another $Address$. Since the VM has no accumulator, it operates directly on addresses, and the function $GetAddress$ is essential for the model. $Address$ domain means $TwoBytes$ or $FourBytes$.
- Memory update
  $$\begin{aligned} Upd1BMem = {} & (Address \times Memory \times \\ & \times OneByte) \to Memory \\ Upd2BMem = {} & (Address \times Memory \times \\ & \times TwoBytes) \to Memory \end{aligned}$$
  Similarly for $Get4BMem$, $Get8BMem$, etc.

- Memory move

$$\begin{aligned} MemMove = (&Address \times Memory \times \\ &\times Address \times Memory \times \\ &\times OneByte) \to Memory \end{aligned}$$

  The source and target $Addresses$ of code or data $Memory$ should be provided. The number of bytes being moved ranges from 0 to 255 ($OneByte$).
- Stack functions

$$Push = (Stack \times Address) \to Stack$$
$$Pop = Stack \to (Address \times Stack)$$

  The functions execute stack operations needed by subprograms. Note that Pop returns a pair, viz. $Address$ and new $Stack$.
- Flag operations

$$\begin{aligned} ClearFlag &= (TwoBytes \times TwoBytes) \to \\ &\to TwoBytes \\ SetFlag &= (TwoBytes \times TwoBytes) \to \\ &\to TwoBytes \end{aligned}$$

  The $Flags$ domain is an alias to $TwoBytes$. The successive $TwoBytes$ above denote actual flags, bits to be set or reset, and new flags.
- Value conversions

$$ByteToWord = OneByte \to TwoBytes$$
$$WordToByte = TwoBytes \to OneByte$$

  For a value without a sign, $ByteToWord$ places zero bits into the more significant byte of $TwoBytes$, otherwise the byte is filled with the sign bit. $WordToByte$ reduces the value by removing the most significant bits.

The following sample functions provide numerical interpretations of $OneByte$, $TwoBytes$ and two other memory chunks.

$$\begin{aligned} BoolOf &= OneByte \to BOOL \\ FromBool &= BOOL \to OneByte \\ IntOf &= TwoBytes \to INT \\ FromInt &= INT \to TwoBytes \\ DIntOf &= FourBytes \to DINT \\ FromDInt &= DINT \to FourBytes \\ LIntOf &= EightBytes \to LINT \\ FromLInt &= LINT \to EightBytes \end{aligned}$$

Other types are interpreted analogously.

The numeric identifiers of VM instructions consist of the identifier of a group `ig` and the identifier `it` of a particular data type or procedure. In this way type-specific instructions or procedures may be selected. For some functions `it` also indicates the number of inputs.

To collectively represent the concept of decoding a group and type, followed by the execution of a specific instruction, a universal function $\mathcal{U}$ has been defined. The algorithm of the function $\mathcal{U}$ is presented in Fig. 2. It is assumed that the code register `cr` initially points to the group identifier `ig` in code memory `cm`. The algorithm starts by fetching `ig` (one byte) and incrementing `cr` to $\mathtt{cr}_1$. Then `it` is acquired and the code register incremented to $\mathtt{cr}_2$. At this moment the state of the VM is described by the tuple $(cm, dm, cs, ds, cr_2, dr, flg)$ involving memories, stacks, registers and flags. For instance, for `ig=04` and `it=02` the DIV function is called with the two operands `op1`, `op2` of type INT, whereas `it=09` means operands of type REAL. The last group `ig=1C` consists of system procedures, including JMP and CALB.

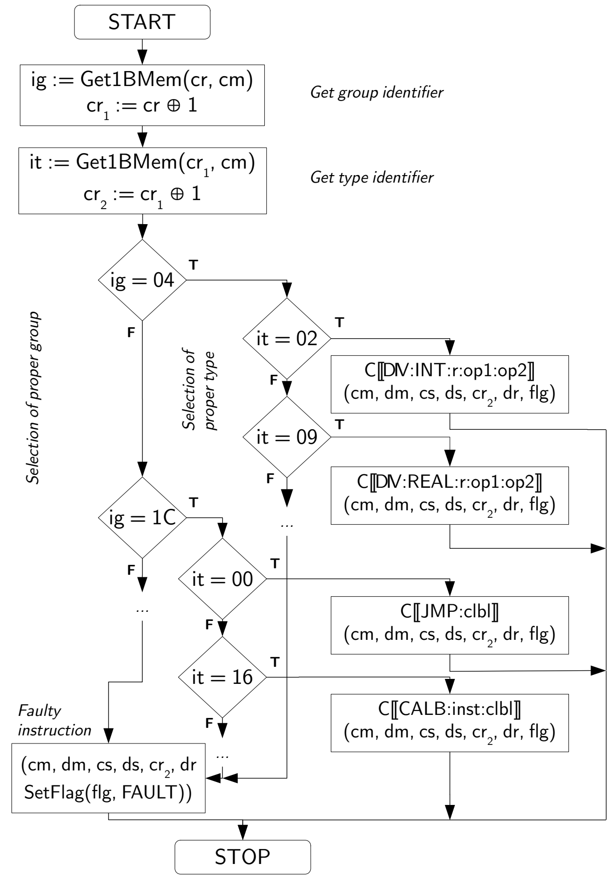


Fig. 2. Algorithm of the universal function $\mathcal{U}$

The semantics of the DIV function for INT-type operands is shown in Listing 1. The function divides two operands of type INT, sv denotes the value of the division. The result of a function execution is stored in the location labeled by the first operand, here denoted by `r`. The updated data memory is the second element of s_1 as the result of invoking $Upd2BMem$. The number stored at $raddr$ is given by $FromInt(sv)$.

The procedure GAWR presented in Listing 2 is used in algorithms involving arrays. This procedure copies elements of an array in local memory to an array in global memory. The arrays may contain elements of any type, including arrays and structures. There are four operands, source `src` and destination `dst` labels, `size` of the elements, and array index `idx`. The values $size$ and idx are addresses to data of type WORD. Since the operand `dst` refers to global memory, its

Listing 1 The semantic equation of the DIV function

$$\begin{array}{r}
\mathcal{C}[\![\texttt{DIV:INT:r:op1:op2}]\!] = \lambda s. \\
(cm, dm, cs, ds, cr, dr, flg) := s \\
r := GetAddress(cr, cm) \\
raddr := dr \oplus r \\
cr_1 := cr \oplus AddressSize \\
op1 := GetAddress(cr_1, cm) \\
op1addr := dr \oplus op1 \\
cr_2 := cr_1 \oplus AddressSize \\
op2 := GetAddress(cr_2, cm) \\
op2addr := dr \oplus op2 \\
cr_3 := cr_2 \oplus AddressSize \\
sv := IntOf(Get2BMem(op1addr, dm)) \div \\
\div\ IntOf(Get2BMem(op2addr, dm)) \\
s_1 := (cm, Upd2BMem(raddr, dm, \\
FromInt(sv)), cs, ds, cr_3, dr, flg) \\
s_1
\end{array}$$

value dst is a direct address (zero dr). The $resultaddr$ is the sum of address dst and the product of $idxval$ and $sizeval$.

Listing 2 The semantic equation of the GAWR procedure

$$\begin{array}{r}
\mathcal{C}[\![\texttt{GAWR:dst:src:size:idx}]\!] = \lambda s. \\
(cm, dm, cs, ds, cr, dr, flg) := s \\
dst := GetAddress(cr, cm) \\
cr_1 := cr \oplus AddressSize \\
src := GetAddress(cr_1, cm) \\
srcaddr := dr \oplus src \\
cr_2 := cr_1 \oplus AddressSize \\
size := GetAddress(cr_2, cm) \\
sizeaddr := dr \oplus size \\
sizeval := WordOf(Get2BMem(\\
sizeaddr, dm)) \\
cr_3 := cr_2 \oplus AddressSize \\
idx := GetAddress(cr_3, cm) \\
idxaddr := dr \oplus idx \\
idxval := WordOf(Get2BMem(\\
idxaddr, dm)) \\
cr_4 := cr_3 \oplus AddressSize \\
resultaddr := dst \oplus idxval \otimes sizeval \\
um := MemMove(dm, resultaddr, dm, \\
srcaddr, sizeval) \\
s_1 := (cm, um, cs, ds, cr_4, dr, flg) \\
s_1
\end{array}$$

It is important to note that the equations provided for the DIV and GAWR procedures do not account for erroneous operands, such as a divisor equal to zero or an array index out of bounds. Consequently, to address these failures and prevent unpredictable behavior, the model had to be extended with an exception mechanism.

IV. Adding exception handling

Exceptions have been introduced to replace a sequence of nested `if-else` instructions when performing compound operations that may fail under certain circumstances. The complex branching of algorithm paths, depicted in Figure 3, can be challenging to analyze and distinguish between the normal execution path and the path taken to handle failures. To address this issue, some programming languages have introduced a `try-catch` construct, which separates the algorithm's main path (protected code) from the failure handling path. This approach allows programmers to focus on the operations that the algorithm needs to perform, while storing the failure handling logic in a separate section of the code.

When a failure occurs, it is reported through an exception, which can take various forms, ranging from a simple value like a number or string to a specifically designed object. The presence of an exception terminates the execution of the remaining instructions within the protected code. Subsequently, the processing of the first `catch` clause begins, but only if it matches the type of the exception object. If the exception does not match the type of the first `catch` clause, the subsequent `catch` clauses will be checked for a match. If no further `catch` clauses are found, the execution switches to the surrounding `try-catch` construct. However, if such a construct is not present, the execution is terminated with an unhandled exception state, preventing further execution.

Therefore, the revised code based on Figure 3 would resemble the examples provided in the code snippet shown in Listing 3. However, it is important to note that such code may overlook certain critical tasks that must be performed even if an exception occurs. To address this concern, the `try-catch` construct has been enhanced with an additional clause called `finally`. The `finally` clause contains the code that is always executed when the control exits the protected section of code, irrespective of whether a matched exception occurs, an unhandled exception is encountered, or no exception occurs at all.

The IEC 61131-3:2013 standard does not include constructs for writing code in an exception-style manner. However, certain manufacturers offer their own extensions to the ST language to support such functionality. For instance, the CODESYS development environment provides the following keywords to indicate protected code:

- `__TRY` – beginning of the protected code,
- `__CATCH` – point where failure path of code begins,
- `__FINALLY` – beginning of mandatory code executed always,
- `__ENDTRY` – end of the protected code.

```
if(oper1())
  if(oper2())
    if(oper3())
     if(oper4())
        return success();
     else return error("oper4");
    else return error("oper3");
  else return error("oper2");
else return error("oper1");
```

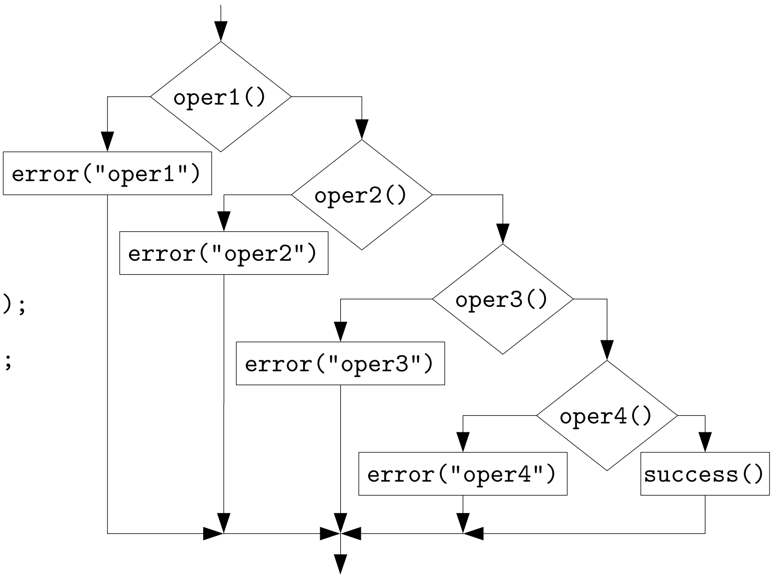


Fig. 3. An example of nested `if` commands and their block diagram

Listing 3 The `try-catch` approach

```
try {
    oper1();
    oper2();
    oper3();
    oper4();
    return success;
} catch(Exception1 e1) { ...
} catch(Exception2 e2) { ...
}
```

TABLE IV
SYSTEM PROCEDURES FOR EXCEPTIONS

Name	Operand	Type
PHPRS	excAddr	:gclabel
	finAddr	:gclabel
	contAddr	:gclabel
RAISE	excObj	:gdlabel
MEXCT	excTyp	:rdlabel
	nxtm	:gclabel
POPRS	—	—
CEXCF	—	—

In our solution, it was decided to use keywords from CODESYS for exception handling for greater compatibility. In addition to the above keywords, __THROW keyword for throwing user-defined exceptions as in the general purpose programming languages (e.g. C++, Java, C#) is added.

To accommodate these language constructs, several changes need to be made to the denotational model of the virtual machine. Firstly, the $State$ tuple needs to be expanded to include two additional components: $ProtStack$ and $ExcObj$. $ProtStack$ represents a stack (Kleene closure) of $ProtEntry$ tuples, which consist of four addresses. To modify the $ProtStack$, the following functions are introduced: $PushProt$, $PopProt$, and $PeekProt$. The $PushProt$ function adds an item to the stack, $PopProt$ removes an item from the stack, and $PeekProt$ returns a copy of the topmost item without modifying the stack. $ExcObj$ is an $Address$ that indicates the location where the exception object has been stored. Since an $Address$ always refers to a memory location, a new flag, EXCOBJ, needs to be introduced in the $Flags$ mask to indicate the presence of the exception object.

$$\begin{aligned}State = {} & CodeMemory \times DataMemory \times \\ & \times CodeStack \times DataStack \times CodeReg \times \\ & \times DataReg \times Flags \times ProtStack \times ExcObj\end{aligned}$$

$$\begin{aligned}ProtStack &= ProtEntry^* \\ ProtEntry &= Address \times Address \times Address \times Address \\ PushProt &= (ProtStack \times ProtEntry) \to ProtStack \\ PopProt &= ProtStack \to (ProtEntry \times ProtStack) \\ PeekProt &= ProtStack \to ProtEntry\end{aligned}$$

In order to handle the __TRY, __CATCH, __FINALLY, __ENDTRY keywords, as well as the additional __THROW, new system procedures are required. These procedures are listed in Table IV.

When encountering the __TRY instruction, the ST language compiler should generate the system procedure PHPRS. The purpose of this procedure is to store the current DPTR context, the address of the first __CATCH instruction, the address of the __FINALLY instruction, and the address of the __ENDTRY instruction on the $ProtStack$. The corresponding denotational model of PHPRS is presented below.

```
$\mathcal{C}[\![\texttt{PHPRS:ea:fa:ca}]\!] = \lambda s.$
  $(cm, dm, cs, ds, cr, dr, flg, ps, eo) := s$
  $dptrCtx := dr$
  $excAddr := GetAddress(cr, cm)$
  $cr_1 := cr \oplus AddressSize$
  $finAddr := GetAddress(cr_1, cm)$
  $cr_2 := cr_1 \oplus AddressSize$
  $contAddr := GetAddress(cr_2, cm)$
  $cr_3 := cr_2 \oplus AddressSize$
  $se := (dptrCtx, excAddr, finAddr, contAddr)$
  $ps_1 := PushProt(ps, se)$
  $s_1 := (cm, dm, cs, ds, cr_3, dr, flg, ps_1, eo)$
  $s_1$
```

When the ST compiler encounters the __CATCH statement, it should generate the system procedure MEXCT. The purpose of this procedure is to detect whether the exception type (excTyp operand) matches the current exception. If the exception type matches, the following instructions will be processed and the exception state will be cleared with a jump to the __FINALLY statement. If the exception type does not match, a jump to the next __CATCH statement is performed. If there are no further __CATCH statements, a jump to the __FINALLY keyword is executed. This behavior can be represented using the following denotational equation:

```
$\mathcal{C}[\![\texttt{MEXT:exct:nxtm}]\!] = \lambda s.$
  $(cm, dm, cs, ds, cr, dr, flg, ps, eo) := s$
  $exct := GetAddress(cr, cm)$
  $cr_1 := cr \oplus AddressSize$
  $nxtm := GetAddress(cr_1, cm)$
  $cr_2 := cr_1 \oplus AddressSize$
  $dext := Get4BMem(dm, dr \oplus exct \oplus typeOffset)$
  $sext := Get4BMem(dm, eo \oplus typeOffset)$
  $dcr :=$ **match** $sext = dext$ **with**
      | **true** $\rightarrow cr_2$
      | **false** $\rightarrow$ **match** $nxtm = -1$ **with**
          | **true** $\rightarrow stk := PeekProt(ps)$
                $(dct, ctch, fin, efn) := stk$
                $fin$
          | **false** $\rightarrow nxtm$
          **end**
      **end**
  $s_1 := (cm, dm, cs, ds, dcr, dr, flg, ps, eo)$
  $s_1$
```

After the last statement of __CATCH, the compiler should generate the system procedure CEXCF, which marks the end of exception handling. The denotational model of CEXCF can be defined as follows:

```
$\mathcal{C}[\![\texttt{CEXCF}]\!] = \lambda s.$
  $(cm, dm, cs, ds, cr, dr, flg, ps, eo) := s$
  $stk := PeekProt(ps)$
  $(dct, ctch, fin, efn) := stk$
  $flg_1 := ClearFlag(flg, F\_EXCPT)$
  $s_1 := (cm, dm, cs, ds, fin, dr, flg_1, ps, 0)$
  $s_1$
```

No special action is needed when the compiler encounters the __FINALLY statement. However, an action is required when the compiler encounters __ENDTRY in the ST input. In this case, the compiler should emit the POPRS system procedure, which can be represented by the following denotational model:

```
$\mathcal{C}[\![\texttt{POPRS}]\!] = \lambda s.$
  $(cm, dm, cs, ds, cr, dr, flg, ps, eo) := s$
  $tf := SetFlag(flg, F\_EXCPT)$
  $(dcr, ddr, nstk) :=$ **match** $tf = flg$ **with**
      | **true** $\rightarrow (ent, stk) := PopProt(ps)$
            $(dct, ctch, fin, efn) := ent$
            $(ctch, dct, stk)$
      | **false** $\rightarrow (ent, stk) := PopProt(ps)$
            $(cr, dr, stk)$
      **end**
  $s_1 := (cm, dm, cs, ds, dcr, ddr, flg, nstk, eo)$
  $s_1$
```

If a programmer wishes to throw their own exception in ST code, they can use the __THROW keyword. In this case, the ST compiler should emit the RAISE system procedure with the exception object. The denotational model of RAISE can be represented as follows:

```
$\mathcal{C}[\![\texttt{RAISE:excObj}]\!] = \lambda s.$
  $(cm, dm, cs, ds, cr, dr, flg, ps, eo) := s$
  $excObj := GetAddress(cr, cm)$
  $eo_1 := dr \oplus excObj$
  $flg_1 := SetFlag(flg, F\_EXCPT)$
  $stk := PeekProt(ps)$
  $(dr_1, ctch, fin, efn) := stk$
  $s_1 := (cm, dm, cs, ds, ctch, dr_1, flg_1, ps, eo_1)$
  $s_1$
```

V. Implementation in CPDev

An exception is reported in the CPDev virtual machine either automatically or manually. The call is invoked internally, when a system exception condition is met during the program execution. It may be also invoked by the programmer by calling the RAISE system procedure to manually trigger the failure path. The automatically generated system exceptions are reported when on of the runtime errors occurs. Selected system exceptions are listed in Table V.

TABLE V
SELECTED SYSTEM EXCEPTIONS

Type	Description
Division by zero	Invalid DIV instruction parameter
Modulo by zero	Invalid MOD instruction operand
Bad array index	Index array access out of bounds
Bad format	Invalid string format during parsing to numeric types (STRING_TO_INT, STRING_TO_WORD, STRING_TO_REAL, etc.)
Cycle overflow	Program execution exceeded the declared cycle time

The corresponding operations performed by th VM have been extended with exception reporting. For example, the semantic description of the DIV function from Listing 1 should be extended which checking for zero divisor as follows:

```
$divisor := IntOf(Get2BMem(op2addr, dm))$
match $divisor$ with
| 0 → $excObj := (cr_3, DIV\_BY\_ZERO\_EXC)$
    $\mathcal{C}[\![\mathtt{RAISE:op}]\!](cm, dm, cs, ds,$
        $cr_3, dr, flg, ps, excObj)$
| _ → $sv := IntOf(Get2BMem(op1addr,$
    $dm)) \div divisor$
    $s_1 := (cm, Upd2BMem(raddr, dm,$
    $FromInt(sv)), cs, ds, cr_3, dr, flg, ps, eo)$
    $s_1$
end
```

Listing 4 shows the DIV instruction utilizing a C macro for division of several numeric types. The function IG_DIV_04 implements the division for all relevant data types (group), thus avoiding repetitions of rather similar code. The function calls the parameterized macrodefinition DIV_TYPE which is common for all types. The value of an operand of a particular TYPE is determined in DIV_TYPE by the function TYPE##Of with given sizeof(TYPE). The code calls an internal function WM_RaiseException in the case when the second operand (divisor) is zero. The division result cmp updates the INT value at raddr (Upd2BMemData also increments the code register). The function IG_DIV_04 recognizes a particular type as the second nibble (half byte) of the type identifier it by masking it & 0x0F. Note that case and break are hidden for switch in the DIV_TYPE definition.

Listing 5 contains a simplified implementation of the GAWR procedure (Sec. III). The check is made if the parameter corresponding to the array index is less than zero. If so, the system exception is raised (EX_ARRAY_IDX).

In the case of an exception, the virtual machine examines the ProtStack. If the stack is empty, no __TRY...__CATCH block has been defined by the programmer. In such a case, the system may perform one of the predefined actions:

- stop execution and go into a fail-safe state (set outputs to safe values)
- perform a cold start of the controller
- perform a warm start
- restart the program cycle.

Listing 6 contains a simple ST code using the exception-related keywords. The program includes a declaration of an array BA indexed from 0 to 10. The protected code between

Listing 4 Implementation of the DIV instruction

```
#define DIV_TYPE(TYPE)  \
case IT_DIV_##TYPE & 0x000F: \
{ \
 TYPE sv = 0; \
 ADDRESS raddr = \
  dataReg + GetCodeAddress(); \
 ADDRESS op1addr =  \
  dataReg + GetCodeAddress(); \
 ADDRESS op2addr = \
  dataReg + GetCodeAddress(); \
 TYPE op1 = TYPE ## Of(GetMemData(op1addr, \
    sizeof(TYPE))); \
 TYPE op2 =   TYPE##Of(GetMemData(op2addr, \
    sizeof(TYPE))); \
 if (op2 == 0) \
           WM_RaiseException(EX_DIV0); \
 else { \
   sv = op1 / op2;\
   UpdMemData(raddr, From##TYPE(sv), \
     sizeof(sv)); } \
} \
break;


void IG_DIV_04(BYTE it)
{
        switch (it & 0x0F)
        {
                DIV_TYPE(SINT)
                DIV_TYPE(INT)
                DIV_TYPE(DINT)
                DIV_TYPE(LINT)
                DIV_TYPE(BYTE)
                DIV_TYPE(WORD)
                DIV_TYPE(DWORD)
                DIV_TYPE(LWORD)
                DIV_TYPE(REAL)
                DIV_TYPE(LREAL)
        default:  /* unknown code */
                flag |= FAULT;
        }
        return;
}
```

Listing 5 Implementation of the GAWR instruction

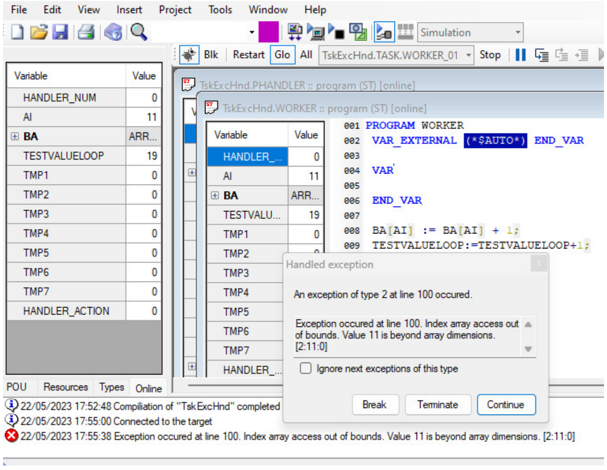
```
case VMF_GAWR & 0xFF:
{
  ADDRESS src = GetCodeAddress();
  ADDRESS dst = GetCodeAddress();
  INT idx = getINT(GetCodeAddress());
  BYTE size = getBYTE(GetCodeAddress());

  if (idx < 0)
    WM_RaiseException(EX_ARRAY_IDX);
  else
    memcpy(dst+idx*size, src, size);
}
break;
```

Listing 6 Exception handling in ST code

```
PROGRAM WORKER
  VAR
    BA : ARRAY[0..10] OF INT;
    AI : INT;
    RES, VALUE, SCALE : REAL;
    DIV_EX : DIV_BY_ZERO_EXCEPTION;
    OTHER_EX : ANY_EXCEPTION;
  END_VAR

  __TRY

    BA[AI] := BA[AI] + 1;
    RES := VALUE / SCALE;

  __CATCH(DIV_EX)

    SCALE := 1;
    DIV_EX.ACTION := RESTART_CYCLE;

  __CATCH(OTHER_EX)

    OTHER_EX.ACTION := TERMINATE;

  __ENDTRY

END_PROGRAM
```

__TRY and __CATCH increases the array element at the index pointed by the value of the variable AI. In case of a division by zero exception during the operation, the failure path from __CATCH(DIV_EX) is executed. If any other exception occurs, the next failure path from __CATCH(OTHER_EX) is taken. The exception block restores the AI value to the acceptable value of 0 and instructs the virtual machine to restart the program cycle.

An exception is usually an unexpected behavior, so CPDev IDE provides a set of tools to debug such situations. Figure 4 shows the CPDev Integrated Environment (IDE) running a program in a simulation mode. As one may observe, the exception has occured due to the fact, that the variable AI (array index) has been set to 11, so outside the array bounds. The environment allows to break the execution to examine the cause, to terminate the program completely or to continue with the default action for an exception.



Variable	Value
HANDLER_NUM	0
AI	11
⊞ **BA**	ARR...
TESTVALUELOOP	19
TMP1	0
TMP2	0
TMP3	0
TMP4	0
TMP5	0
TMP6	0
TMP7	0
HANDLER_ACTION	0

Variable	Value
HANDLER_...	0
AI	11
⊞ **BA**	ARR...
TESTVALU...	19
TMP1	0
TMP2	
TMP3	
TMP4	
TMP5	
TMP6	
TMP7	
HANDLER_...	

```
PROGRAM WORKER
 VAR_EXTERNAL (*$AUTO*) END_VAR

 VAR

 END_VAR

 BA[AI] := BA[AI] + 1;
 TESTVALUELOOP:=TESTVALUELOOP+1;
```

Handled exception

An exception of type 2 at line 100 occured.

Exception occured at line 100. Index array access out of bounds. Value 11 is beyond array dimensions. [2:11:0]

□ Ignore next exceptions of this type

Break | Terminate | Continue

```
22/05/2023 17:52:48 Compiliation of "TskExcHnd" completed
22/05/2023 17:55:00 Connected to the target
22/05/2023 17:55:38 Exception occured at line 100. Index array access out of bounds. Value 11 is beyond array dimensions. [2:11:0]
```

Fig. 4. Debugging exceptions in CPDev IDE

VI. Final remarks

The introduction of exceptions into the control environment based on a virtual machine was driven by industry demands. The presented concepts from a programmer's perspective resemble solutions available in high-level object oriented programming languages like C# or Java. However, in this case, the solution needs to be applied to embedded controllers with limited resources and performance.

To address this, the proposed mechanism for exception handling was designed to minimize the extra operations performed by the CPU. The requirement for additional memory to accommodate the $ProtStack$ is relatively easy to fulfill, even for small devices. It is anticipated that incorporating exception infrastructure supported by a formal model will facilitate the development of more robust and reliable control solutions.

Acknowledgment

This project is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

References

[1] *IEC 61131-3. Programmable Controllers. Part 3. Programming languages.* International Standard: IEC, 2013.

[2] *IEC 61499 — Function blocks.* International Standard: IEC, 2015.

[3] M. Simros, M. Wollschlaeger, and S. Theurich, "Programming embedded devices in IEC 61131-languages with industrial PLC tools using PLCopen XML," in *Proceedings of the CONTROLO'2012 Portuguese Conference on Automatic Control, Funchal, Portugal*, 2012. ISBN 9789729702532 pp. 51–56.

[4] T. Lindholm, F. Yellin, G. Bracha, and A. Buckley, *The Java® Virtual Machine Specification*. Oracle America, Inc., 2013. ISBN 9780133260441

[5] T. L. Thai and L. H., *.NET Framework Essentials*. O'Reilly Media, 2003. ISBN 9780596005054

[6] *ECMA-335, Standard. Common Language Infrastructure (CLI)*. Geneva: Ecma, 2012.

[7] A. Blikle, "An experiment with denotational semantics," *SN Computer Science*, vol. 15, no. 1, pp. 1–31, 2020. doi: 10.1007/s42979-019-0013-0

[8] D. Rzońca, J. Sadolewski, A. Stec, Z. Świder, B. Trybus, and L. Trybus, "Programming controllers in structured text language of IEC 61131-3 standard," *Journal of Applied Computer Science*, vol. 16, no. 1, pp. 49–67, 2008.

[9] D. Rzońca, J. Sadolewski, A. Stec, Z. Świder, B. Trybus, and L. Trybus, "Open environment for programming small controllers according to IEC 61131-3 standard," *Scalable Computing: Practice and Experience*, vol. Volume 10, no. 3, pp. 325–336, 2009.

[10] B. Trybus, "Development and Implementation of IEC 61131-3 Virtual Machine," *Theoretical and Applied Informatics*, vol. 23, no. 1, pp. 21–35, 2011. doi: 10.2478/v10179-011-0002-z

[11] K. Cooper and L. Torczon, *Engineering a Compiler*. San Francisco: Morgan Kaufmann, 2022. ISBN 9780128154120

[12] K. Slonneger and B. L. Kurtz, *Formal Syntax and Semantics of Programming Languages: A Laboratory-Based Approach*. Addison-Wesley Publishing Company, Inc, 1995. ISBN 9780201656978

[13] D. Schmidt, *Denotational Semantics: A Methodology for Language Development*. Kansas State University, Manhattan: Department of Computing and Information Sciences, 1997.

[14] J. Sadolewski and B. Trybus, "Denotational model and implementation of scalable virtual machine in CPDev," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems*, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30, 2022. doi: 10.15439/2022F236 pp. 587–591.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 731–738
DOI: 10.15439/2023F6502
ISSN 2300-5963 ACSIS, Vol. 35

An Innovative Drastic Metric for Ranking Similarity in Decision-Making Problems

Wojciech Sałabun
0000-0001-7076-2519
West Pomeranian University of Technology
in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: wojciech.salabun@zut.edu.pl

Andrii Shekhovtsov
0000-0002-0834-2019
National Telecommunications Institute
ul. Szachowa 1, 04-894 Warsaw, Poland
Email: a.shekhovtsov@il-pib.pl

***Abstract*—In this paper, we propose a novel approach to distance measurement for rankings, introducing a new metric that exhibits exceptional properties. Our proposed distance metric is defined within the interval of 0 to 1, ensuring a compact and standardized representation. Importantly, we demonstrate that this distance metric satisfies all the essential criteria to be classified as a true metric. By adhering to properties such as non-negativity, identity of indiscernibles, symmetry, and the crucial triangle inequality, our proposed distance metric provides a robust and reliable approach for comparing rankings in a rigorous and mathematically sound manner. Finally, we compare our new metric with distances such as Hamming distance, Canberra distance, Bray-Curtis distance, Euclidean distance, Manhattan distance, and Chebyshev distance. By conducting simple experiments, we assess the performance and advantages of our proposed metric in comparison to these established distance measures. Through these comparisons, we demonstrate the superior properties and capabilities of our new drastic weighted similarity distance for accurately capturing the dissimilarities and similarities between rankings in the decision-making domain.**

I. Introduction

DISTANCE measures are fundamental tools in many areas of data analysis, including machine learning, statistics, data mining, and many more [1], [2]. They quantify the difference or dissimilarity between pairs of objects, like vectors, sets, or more complex structures, providing a quantitative basis for their comparison [3].

A key aspect of distance measures is that they must satisfy certain properties, such as non-negativity (distances are always non-negative), identity of indiscernibles (the distance between an object and itself is zero), symmetry (the distance from A to B is the same as from B to A), and the triangle inequality (the direct distance from A to B is always shorter or equal to the distance from A to B via an intermediary point C) [4].

There are various types of distance measures, including Euclidean [5], Manhattan [6], Chebyshev [7], Hamming [6], Canberra, Bray-Curtis, and many others [8], [9], each with their own characteristics and use-cases. Some measures like Euclidean and Manhattan are primarily used for continuous variables [6], while others like Hamming are used for categorical variables [10]. Some measures are sensitive to the scale and distribution of the data, while others are more robust.

The choice of the appropriate distance measure is highly dependent on the nature of the data and the specific objectives of the analysis [11], [12]. For example, in a scenario where extreme values or outliers are important, a measure such as the Chebyshev distance could be useful as it focuses on the maximum difference in any one dimension. On the other hand, for data that represents rankings or preferences, a measure like Spearman's footrule or the Kendall tau distance might be more appropriate [13].

When comparing rankings in decision-making, distance measures play a vital role. To compare rankings, we need a way to quantify how similar or different two rankings are [14], [15]. That's where distance measures come in. They provide a numeric value representing the dissimilarity between two rankings, with lower values typically indicating greater similarity.

The choice of distance measure can have a significant impact on the comparison. Some measures are more sensitive to the exact order of the rankings, while others, like Spearman's footrule [16], are more focused on the overall similarity. Moreover, some measures are more sensitive to differences at the top of the rankings [17], [18], while others treat all positions equally. Overall, comparing rankings using distance measures can provide valuable insights in decision-making, helping decision-makers understand how different choices, evaluations, or scenarios compare to each other, and aiding in making more informed, data-driven decisions [19].

Rankings and comparisons form an integral part of decision-making processes in diverse fields such as information retrieval, sports [20], elections, and more [21]. However, a significant challenge that persists in these scenarios is quantifying the dissimilarity or distance between different rankings effectively and accurately. Traditional distance measures, while useful, can often fail to capture the nuances and subtleties inherent in the comparison of rankings. To address these limitations and introduce a more robust and versatile solution, the motivation behind this paper emerges.

In this paper, the main contribution is to propose a novel distance metric that is particularly well-suited for ranking comparisons. We aspire to create a metric that not only captures the dissimilarity between rankings accurately but also exhibits essential properties required of a true metric. A key part of our motivation is to ensure that this new measure is defined within the interval of 0 to 1, thus providing a compact and standardized representation that is easy to interpret across diverse scenarios.

The structure of the paper is as follows: In Section II, the necessary groundwork is laid by introducing and defining key distance measures. Section III is dedicated to proposing a novel distance metric, WS_{dra}, along with comprehensive proof of its properties. Section IV then provides a comparative study of this new metric against the traditional measures introduced in Section II. Finally, Section V concludes the paper by summarizing the research findings and their potential implications.

II. Preliminaries

A. Weighed similarity

The Weighted Similarity (WS) measure aims to be sensitive to significant changes in rankings while remaining robust against minor fluctuations. It also offers the advantage of being easy to interpret, with its values falling within a specified range [17].

In designing the WS measure, a key assumption is made that differences in the top rankings are more impactful than those lower down the list. This is intuitive in scenarios where top-ranked items often have more importance, such as in competitive rankings or search results.

The formula to calculate the WS measure is:

$$WS = 1 - \sum_{i=1}^{n} 2^{-x_i} \frac{|x_i - y_i|}{\max\{|1 - x_i|, |N - x_i|\}} \tag{1}$$

In this equation: WS represents the similarity coefficient's value, n is the length of the ranking, and x_i and y_i represent the place in the ranking for the i^{th} element in the respective rankings x and y.

This formula implies that WS calculates the absolute differences in the ranks of each element in two rankings, normalizes them by the maximum possible difference for that element, and then sums the results. This total is subtracted from 1 to convert it into a similarity measure. Thus, a larger WS value indicates a higher similarity between the two rankings, making WS an effective tool for comparing and analyzing rankings [17].

B. Hamming distance

Hamming distance is a metric that measures the difference between two strings of equal length. It counts the number of positions at which the corresponding symbols in the strings differ [6]. The formula for calculating Hamming distance is as follows:

$$d(x, y) = \frac{\sum_{i=1}^{n} \delta(x_i, y_i)}{n} \tag{2}$$

where x_i and y_i represent the symbols at position i in the two strings, and $\delta(x_i, y_i)$ is an indicator function that equals 0 if x_i and y_i are equal, and 1 otherwise. The Hamming distance provides a way to quantify the dissimilarity between two vectors by measuring the number of symbol mismatches [22].

C. Canberra distance

The Canberra distance is a metric used to quantify the dissimilarity between two vectors or points in a multidimensional space. It takes into account both the magnitude and direction of differences between corresponding components of the vectors [23]. The formula for calculating the Canberra distance is as follows:

$$d(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{3}$$

where x_i and y_i represent the components at position i in the two vectors. The Canberra distance considers the absolute difference between the components, normalized by the sum of their magnitudes. This normalization accounts for differences in scale and ensures that each component contributes proportionally to the overall distance calculation.

D. Bray-Curtis distance

The Bray-Curtis distance is a metric used to measure the dissimilarity between two vectors or points in a multidimensional space. It considers both the magnitude and direction of differences between corresponding components of the vectors, taking into account their relative proportions [24]. The formula for calculating the Bray-Curtis distance is as follows:

$$d(x, y) = \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} |x_i + y_i|} \tag{4}$$

where x_i and y_i represent the components at position i in the two vectors. The Bray-Curtis distance calculates the absolute difference between the components and normalizes it by the sum of their absolute values. This normalization accounts for differences in scale and ensures that each component contributes proportionally to the overall distance calculation.

E. Euclidean distance

The Euclidean distance is a metric used to measure the straight-line distance between two points in a multidimensional space. It calculates the length of the line connecting the two points, taking into account the differences between their corresponding components [6]. The formula for calculating the Euclidean distance is as follows:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

where x_i and y_i represent the components at position i in the two points. The Euclidean distance computes the squared differences between the components, sums them up, and takes the square root of the result. This computation ensures that each component's contribution to the distance calculation is positive and reflects the actual geometric distance between the points.

F. Manhattan distance

The Manhattan distance, also known as the city block distance or L1 distance, is a metric used to measure the distance between two points in a multidimensional space. It calculates the sum of the absolute differences between the corresponding components of the two points [25]. The formula for calculating the Manhattan distance is as follows:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| \tag{6}$$

where x_i and y_i represent the components at position i in the two points. The Manhattan distance measures the distance traveled along the grid-like streets of a city, where movement can only occur in vertical and horizontal directions. It sums up the absolute differences between the components, disregarding their sign.

G. Chebyshev distance

The Chebyshev distance, also known as the maximum value or L_∞ distance, is a metric that measures the dissimilarity between two vectors or points in a multidimensional space. It calculates the maximum difference between the corresponding components of the two vectors [7]. The formula for calculating the Chebyshev distance is as follows:

$$d(x, y) = \max_{i=1}^{n} |x_i - y_i| \tag{7}$$

where x_i and y_i represent the components at position i in the two vectors. The Chebyshev distance provides a measure of the largest difference between any pair of corresponding components in the vectors, which corresponds to the maximum distance in any dimension.

III. A New Proposed Drastic Metric

In the realm of data analytics and decision-making, the concept of distance plays a pivotal role, enabling us to evaluate similarities, disparities, and rank variables effectively. However, traditional distance metrics have their inherent strengths and limitations. To address these shortcomings and propel the field forward, we introduce a novel distance measure based on the WS coefficient.

Our proposed distance metric revolutionizes the notion of distance by adopting a drastic approach. Instead of penalizing discrepancies in ranking, we treat each comparison in the ranking position as a binary attribute, representing a significant or non-significant relationship. This novel perspective eliminates the conventional notion of assigning varying degrees of penalty based on the magnitude of ranking differences. In essence, our approach treats all errors equally, as an error is an error regardless of what is given in analysed position. This drastic approach fosters a fairer assessment of rankings.

Moreover, our new distance measure recognizes the inherent significance disparity across different ranking positions. It assigns greater consequence to the head of the ranking, acknowledging the top positions as more crucial than the lower ones. This acknowledgment aligns with the understanding that errors at the top of the ranking can have more significant implications than errors further down the list. By considering this significance disparity, our distance measure offers a more nuanced and accurate evaluation of rankings.

Crucially, our proposed measure is normalized within the interval from 0 to 1, enabling straightforward interpretation and comparison across diverse contexts. This normalization facilitates intuitive understanding and ensures that the distance measure remains consistent and interpretable regardless of the specific data or application domain.

By embodying these innovative characteristics, our proposed distance measure qualifies as a true metric in the rigorous mathematical sense. Its drastic approach, significance-awareness, and normalized range combine to offer a comprehensive and reliable framework for comparing rankings in various decision-making scenarios. Through empirical evaluations and theoretical analyses, we demonstrate the superiority and practical utility of our proposed distance measure, paving the way for enhanced ranking analysis and informed decision-making in diverse domains.

A. Definition

The new metric, denoted as $WS_{dra}(x, y)$, is defined as follows:

$$WS_{dra}(x, y) = \frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1 - 2^{-N}} \tag{8}$$

The metric operates on two rankings, denoted as x and y, with each ranking consisting of N elements. The key element of this metric is the function $f(x_i, y_i)$, which compares the elements at corresponding positions in the two rankings.

The function $f(x_i, y_i)$ is defined as follows:

$$f(x_i, y_i) = \begin{cases} 0, & \text{if } x_i = y_i \\ 1, & \text{if } x_i \neq y_i \end{cases} \tag{9}$$

In other words, if the elements at position i in the rankings x and y are the same, $f(x_i, y_i)$ is assigned a value of 0. Conversely, if the elements are different, $f(x_i, y_i)$ takes the value of 1.

The $WS_{dra}(x, y)$ metric computes the weighted sum of $f(x_i, y_i)$ values for each position i, using the weights given by the geometric series 2^{-i}. The weights decrease exponentially as i increases, reflecting a decreasing level of importance for elements further down the rankings. The summation of the weighted $f(x_i, y_i)$ values is then divided by the factor $1 - 2^{-N}$ to ensure normalization within the range of 0 to 1.

Overall, this new metric captures the dissimilarities between two rankings by assigning a weight to each pairwise comparison based on the function $f(x_i, y_i)$. It combines these weighted comparisons to provide a comprehensive measure of dissimilarity between the rankings x and y, where a higher value indicates greater dissimilarity. The normalization factor ensures that the metric remains consistent and interpretable across different ranking sizes.

TABLE I
A SIMPLE EXAMPLE OF TWO RANKINGS, I.E., x_i AND y_i.

i	x_i	y_i	$f(x_i, y_i)$	2^{-i}
1	1	3	1	$\frac{1}{2}$
2	2	2	0	$\frac{1}{4}$
3	3	1	1	$\frac{1}{8}$

We demonstrate a short computational example of the newly proposed metric. Consider the example shown in Table I, which illustrates two rankings, denoted as x_i and y_i. Each row in the table corresponds to a position i in the rankings, and we calculate the associated values of $f(x_i, y_i)$ and 2^{-i}. Thus, ranking x_i means the order of alternatives in the form $A_1 > A_2 > A_3$, and ranking y_i in the form $A_3 > A_2 > A_1$. To calculate the $WS_{dra}(x, y)$ value for these rankings, we use the formula (8) and we get the following result:

$$WS_{dra}(x, y) = \frac{\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 0 + \frac{1}{8} \cdot 1}{1 - \frac{1}{8}} = \frac{\frac{5}{8}}{\frac{7}{8}} = \frac{5}{7}$$

A true metric, also known as a metric space or distance metric, is a mathematical concept used to quantify the distance or similarity between objects within a set. It defines a set of rules or properties that a distance function must satisfy to be considered a true metric [2].

In a true metric, the following properties should hold:

1) Non-negativity: The distance between any two objects is non-negative. It is always equal to or greater than zero.
2) Identity of indiscernibles: The distance between two objects is zero if and only if the objects are identical.
3) Symmetry: The distance between object A and object B is the same as the distance between object B and object A.
4) Triangle inequality: The distance from object A to object B, added to the distance from object B to object C, is always greater than or equal to the distance from object A to object C.

In the following subsections, we will explore each of the presented properties to demonstrate the validity of the proposed measure as a true metric. Our objective is to carefully analyze and evaluate these properties, providing a solid foundation for the metric's credibility. Through a systematic examination, we will investigate the non-negativity, identity of indiscernibles, symmetry, and triangle inequality properties. By establishing the fulfillment of these properties, we aim to establish the proposed metric as a reliable tool for comparing rankings. The goal is to offer a well-founded framework that promotes accurate assessments and meaningful insights for decision-making.

B. Non-negativity

To prove the inequality $\frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1-2^{-N}} \geq 0$ when $f(x_i, y_i)$ can take the values 0, 1, or a combination of 0 and 1, we will consider three different cases.

Case 1: $f(x_i, y_i) = 0$ for all i: When all terms in the summation are multiplied by 0, the numerator becomes zero. The denominator, $1 - 2^{-N}$, is positive since $2^{-N} < 1$ for all positive N. Thus, the inequality holds trivially: $0 \geq 0$.

Case 2: $f(x_i, y_i) = 1$ for all i: In this case, each term in the summation will be equal to 2^{-i} since $f(x_i, y_i)$ is always 1. The numerator then becomes:

$$\sum_{i=1}^{N} 2^{-i} = 2^{-1} + 2^{-2} + \ldots + 2^{-N}$$

This sum is a finite geometric series, and its sum can be calculated as follows:

$$\sum_{i=1}^{N} 2^{-i} = \frac{2^{-1}(1 - 2^{-N})}{1 - 2^{-1}} = \frac{1 - 2^{-N}}{2 - 1} = 1 - 2^{-N}$$

Since $1 - 2^{-N}$ is positive, the numerator is non-negative. The denominator, $1 - 2^{-N}$, is also positive and is equal to the nominative. Therefore, when $f(x_i, y_i) = 1$ for all i, the inequality holds:

$$\frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1 - 2^{-N}} = 1 \geq 0$$

Case 3: $f(x_i, y_i)$ is equal 0 or 1: In this case, the numerator of the expression is a sum of terms, each multiplied by $2^{-i} f(x_i, y_i)$. Since $f(x_i, y_i)$ can be 0 or 1, the product $2^{-i} f(x_i, y_i)$ will be either 0 or 2^{-i}. It means that nominative will be limited to interval:

$$0 \leq \sum_{i=1}^{N} 2^{-i} f(x_i, y_i) < 1$$

Since both 0 and 2^{-i} are non-negative, the numerator is a non-negative number. The denominator, $1 - 2^{-N}$, is positive and it is the biggest possible value of nominative, therefore WS_{dra} will be limited to:

$$0 \leq \frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1 - 2^{-N}} \leq 1$$

Hence, when $f(x_i, y_i)$ takes 0 or 1, the inequality holds:

$$\frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1 - 2^{-N}} \geq 0$$

In all cases, we have shown that the inequality holds. Therefore, we can conclude that $\frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1-2^{-N}} \geq 0$ when $f(x_i, y_i)$ can be equal to 0, 1, or a combination of 0 and 1.

C. Identity of indiscernibles

To prove that only $WS_{dra}(x, x) = 0$ for the given expression $WS_{dra}(x, y) = \frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1-2^{-N}}$, we can substitute x for y in the expression:

$$WS_{dra}(x, x) = \frac{\sum_{i=1}^{N} 2^{-i} f(x_i, x_i)}{1 - 2^{-N}}$$

Now, let's focus on the numerator of the expression. Since $f(x_i, x_i)$ represents the function f evaluated at the same element x_i for both arguments, it will always yield the same result. Therefore, $f(x_i, x_i)$ is a constant for all i. Let's denote this constant as c, such that $f(x_i, x_i) = c$ for all i. Substituting c into the numerator, we have:

$$\sum_{i=1}^{N} 2^{-i} f(x_i, x_i) = \sum_{i=1}^{N} 2^{-i} c = c \sum_{i=1}^{N} 2^{-i}$$

The sum $\sum_{i=1}^{N} 2^{-i}$ is a finite geometric series and can be computed as:

$$\sum_{i=1}^{N} 2^{-i} = \frac{2^{-1}(1-2^{-N})}{1-2^{-1}} = \frac{1-2^{-N}}{2-1} = 1-2^{-N}$$

Now, substituting this value back into the expression, we get:

$$WS(x,x) = \frac{c\sum_{i=1}^{N} 2^{-i}}{1-2^{-N}} = \frac{c(1-2^{-N})}{1-2^{-N}} = c$$

Since c is a constant, it does not depend on the choice of x, and therefore, c is equal to $f(x_i, x_i)$ for any x_i. Since $f(x_i, x_i)$ can take the values of 0 or 1 (according to the given property), we can notice that in this case c is equal 0. Therefore, based on the given expression, we can universally prove that only $WS(x,x) = 0$ and it depends on the specific value of c (i.e., the constant $f(x_i, x_i)$).

D. Symmetry

To prove that $WS_{dra}(x,y) = WS_{dra}(y,x)$ for the expression $WS_{dra}(x,y) = \frac{\sum_{i=1}^{N} 2^{-i} f(x_i,y_i)}{1-2^{-N}}$, we need to show that the weighted sum is symmetric with respect to its arguments. Let's consider the left-hand side $WS_{dra}(x,y)$ and the right-hand side $WS(y,x)$ of the equation separately and compare them.

$$WS_{dra}(x,y) = \frac{\sum_{i=1}^{N} 2^{-i} f(x_i, y_i)}{1-2^{-N}}$$

$$WS_{dra}(y,x) = \frac{\sum_{i=1}^{N} 2^{-i} f(y_i, x_i)}{1-2^{-N}}$$

To show that $WS_{dra}(x,y) = WS_{dra}(y,x)$, we need to demonstrate that the numerator and denominator of both expressions are equal. For each term in the numerator, we have: $f(x_i, y_i)$ in the expression for $WS_{dra}(x,y)$ and $f(y_i, x_i)$ in the expression for $WS_{dra}(y,x)$. Since the order of the arguments is switched between the two expressions, we can see that $f(x_i, y_i) = f(y_i, x_i)$ for each i. Therefore, the numerator of both expressions is identical. The denominator of both expressions is the same: $1-2^{-N}$. Since the numerator and denominator of both $WS_{dra}(x,y)$ and $WS_{dra}(y,x)$ are equal, we can conclude that $WS_{dra}(x,y) = WS_{dra}(y,x)$. Hence, we have proven that the weighted sum expression $WS_{dra}(x,y)$ is symmetric with respect to its arguments, satisfying the property $WS_{dra}(x,y) = WS_{dra}(y,x)$.

TABLE II
ALL POSSIBLE BINARY COMBINATIONS FOR $f(a_i, b_i)$, $f(b_i, c_i)$, $f(a_i, c_i)$ AND THE VALUE X_i OBTAINED AS $f(a_i, b_i) + f(b_i, c_i) - f(a_i, c_i)$.

$f(a_i, b_i)$	$f(b_i, c_i)$	$f(a_i, c_i)$	X_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1
1	1	0	2

E. Triangle inequality

We'll work with the formula (8) to prove the triangle inequality property:

$$WS_{dra}(a,b) + WS_{dra}(b,c) \geq WS_{dra}(a,c)$$

Let's denote $A = WS_{dra}(a,b)$, $B = WS_{dra}(b,c)$, and $C = WS_{dra}(a,c)$. Substituting these values into the inequality, we have $A + B \geq C$. Now, let's consider the individual terms in the numerator of each expression.

$$WS_{dra}(a,b) : \sum_{i=1}^{N} 2^{-i} f(a_i, b_i)$$

$$WS_{dra}(b,c) : \sum_{i=1}^{N} 2^{-i} f(b_i, c_i)$$

$$WS_{dra}(a,c) : \sum_{i=1}^{N} 2^{-i} f(a_i, c_i)$$

Now, let's examine the numerator term-wise for the three expressions: for each i, 2^{-i} is a non-negative constant; for each term $f(a_i, b_i)$ in $WS_{dra}(a,b)$, $f(b_i, c_i)$ in $WS_{dra}(b,c)$, and $f(a_i, c_i)$ in $WS_{dra}(a,c)$, they can take the values of 0 or 1 according to the formula (9). Now, we can compare the terms between the expressions, where for each term i, we have:

$$2^{-i} f(a_i, b_i) + 2^{-i} f(b_i, c_i) \geq 2^{-i} f(a_i, c_i)$$

This inequality holds because for any given term, either $f(a_i, b_i) = f(a_i, c_i) = 1$ or $f(b_i, c_i) = 0$ (which makes the left-hand side greater than or equal to the right-hand side) or $f(a_i, b_i) = f(b_i, c_i) = 0$ (which makes the left-hand side equal to the right-hand side). The all possible cases are presents in Table II, where $X_i = f(a_i, b_i) + f(b_i, c_i) - f(a_i, c_i)$. Now, summing up these inequalities over all i from 1 to N, we have:

$$^{dra}WS(a,b) + {}^{dra}WS(b,c) \geq {}^{dra}WS(a,c)$$

$$\frac{\sum_{i=1}^{N} 2^{-i} f(a_i, b_i)}{1-2^{-N}} + \frac{\sum_{i=1}^{N} 2^{-i} f(b_i, c_i)}{1-2^{-N}} \geq \frac{\sum_{i=1}^{N} 2^{-i} f(a_i, c_i)}{1-2^{-N}}$$

$$\frac{\sum_{i=1}^{N}(2^{-i} f(a_i, b_i) + 2^{-i} f(b_i, c_i) - 2^{-i} f(a_i, c_i))}{1-2^{-N}} \geq 0$$

TABLE III
THE COMPARED RANKINGS, I.E., x_i AND $y_i^{(j)}$ FOR $j = 1, 2, ..., 7$, WHERE RED COLOR INDICATES THE DIFFERENCES WITH THE ORIGINAL RANKING.

i	x_i	$y_i^{(1)}$	$y_i^{(2)}$	$y_i^{(3)}$	$y_i^{(4)}$	$y_i^{(5)}$	$y_i^{(6)}$	$y_i^{(7)}$
1	1	2	1	1	1	3	4	5
2	2	1	3	2	2	2	2	2
3	3	3	2	4	3	1	3	3
4	4	4	4	3	5	4	1	4
5	5	5	5	5	4	5	5	1

$$\sum_{i=1}^{N} 2^{-i} \left(f(a_i, b_i) + f(b_i, c_i) - f(a_i, c_i) \right) \geq 0$$

$$\sum_{i=1}^{N} 2^{-i} X_i \geq 0$$

Therefore, X_i is always non-negativity, we can conclude that we have proven triangle inequality for WS_{dra} distance.

IV. Comparison and discussion

Table III provides a visual representation of the initial ranking x_i and sample rankings $y_i^{(j)}$ for $j = 1, 2, ..., 7$. The table is designed to compare the considered distance measures with the proposed new measure. In this table, the red color highlights the differences between each example ranking and the initial ranking.

The table consists of nine columns. The first column, labeled i, denotes the position in the rankings. The second column represents the initial ranking, denoted as x_i. The remaining columns, labeled $y_i^{(j)}$, correspond to the examplary rankings for $j = 1, 2, ..., 7$.

Each cell in the table represents the element at position i in the corresponding ranking. The red color is used to indicate any differences between the element in the examplary ranking and the initial ranking. By visually highlighting these differences, the table facilitates a clear comparison between the rankings and serves as a reference for evaluating the performance of different distance measures. Table III provides a useful reference point for understanding the subsequent analyses and discussions related to the comparisons between the considered distance measures and the proposed new measure.

Table IV provides a comprehensive summary of the similarity and distance measures for the previously presented rankings, offering valuable insights for analyzing and comparing the relationships between x_i and each $y_i^{(j)}$. The measures included in the table enable a thorough assessment of the similarities and differences among the rankings.

The table introduces a new proposed distance metric denoted as WS_{dra}, represented in blue font. This novel metric returns distances ranging from 0.5484 to 0.7742. It was developed as an enhancement of the WS coefficient. To ensure consistent information direction, the $1 - WS$ coefficient was introduced, yielding values ranging from 0.2083 to 0.5313. This modification was incorporated into the analysis, as the new distance metric was built upon this coefficient.

In addition to the newly proposed metric, Table IV includes well-known distance measures commonly used for comparison purposes. These measures, namely Hamming, Canberra, Bray-Curtis, Euclidean, Manhattan, and Chebyshev, offer additional perspectives on the dissimilarity between x_i and each $y_i^{(j)}$ ranking.

The Hamming measure, typically employed for comparing categorical data, consistently yields a distance value of 0.4000 for all comparisons. This suggests that all rankings $y_i^{(j)}$ exhibit the same level of distance and similarity with respect to x_i. However, from a decision-making standpoint, it becomes evident that this statement does not hold true, as, for example, ranking $y_i^{(1)}$ is closer to x_i than ranking $y_i^{(2)}$.

The Canberra measure calculates distances ranging from 0.6667 to 1.3333, providing insights into the relative dissimilarity between x_i and the different $y_i^{(j)}$ rankings. This measure considers both the magnitude and direction of differences between the rankings, offering a comprehensive assessment of their dissimilarity.

The Bray-Curtis measure, which evaluates dissimilarity based on the proportions of shared and unique elements, yields distances ranging from 0.0667 to 0.2667. This measure takes into account the presence and absence of specific elements, providing valuable information regarding the relative dissimilarity between the rankings.

Let's delve deeper into the two distinct sets of rankings: $y_i^{(1)}$ to $y_i^{(4)}$, and $y_i^{(1)}$ alongside $y_i^{(5)}$ to $y_i^{(7)}$. What sets these rankings apart is the presence of a singular swap between alternatives, occurring either in adjacent positions or non-adjacent ones.

In the first set of rankings, we begin by examining modifications at the top (or head) of the ranking and gradually proceed towards the bottom (or tail). Ranking tasks inherently pose a significant challenge, as they tend to assign more weight or significance to changes at the beginning of the ranking sequence rather than towards the end. For instance, let's consider a scenario where a company not placed first in the ranking wins a tender—such an event is, of course, wrong, as that company would either won or be removed from the ranking (e.g., due to withdrawal).

By comparing these rankings with the x_i ranking and employing five different distance measurement methods—Hamming, Bray-Curtis, Euclidean, Manhattan, and Chebyshev—we observe that the comparison values for all paired rankings remain constant. The respective constant values assigned to these methods are 0.4, 0.0667, 1.4142, 2.0000, and 1.0000. This indicates that these five measurements may not adequately capture the variability required for decision-making processes, as they remain insensitive to changes in ranking positions, regardless of where those changes occur. Furthermore, when considering the $1 - WS$ ratio and the Canberra distance, both measurements consistently exhibit a decreasing trend in values with each subsequent ranking, $y_i^{(j)}$,

TABLE IV
SUMMARY OF SIMILARITY AND DISTANCE MEASURES FOR x_i AND $y_i^{(j)}$ FOR $j = 1, 2, ..., 7$ RANKINGS.

Measures	$d(x_i, x_i)$	$d(x_i, y_i^{(1)})$	$d(x_i, y_i^{(2)})$	$d(x_i, y_i^{(3)})$	$d(x_i, y_i^{(4)})$	$d(x_i, y_i^{(5)})$	$d(x_i, y_i^{(6)})$	$d(x_i, y_i^{(7)})$
WS_{dra}	0.0000	0.7742	0.3871	0.1935	0.0968	0.6452	0.5806	0.5484
$1 - WS$	0.0000	0.2083	0.1458	0.0833	0.0286	0.3750	0.4375	0.5313
Hamming	0.0000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000	0.4000
Canberra	0.0000	0.6667	0.4000	0.2857	0.2222	1.0000	1.2000	1.3333
Bray-Curtis	0.0000	0.0667	0.0667	0.0667	0.0667	0.1333	0.2000	0.2667
Euclidean	0.0000	1.4142	1.4142	1.4142	1.4142	2.8284	4.2426	5.6569
Manhattan	0.0000	2.0000	2.0000	2.0000	2.0000	4.0000	6.0000	8.0000
Chebyshev	0.0000	1.0000	1.0000	1.0000	1.0000	2.0000	3.0000	4.0000

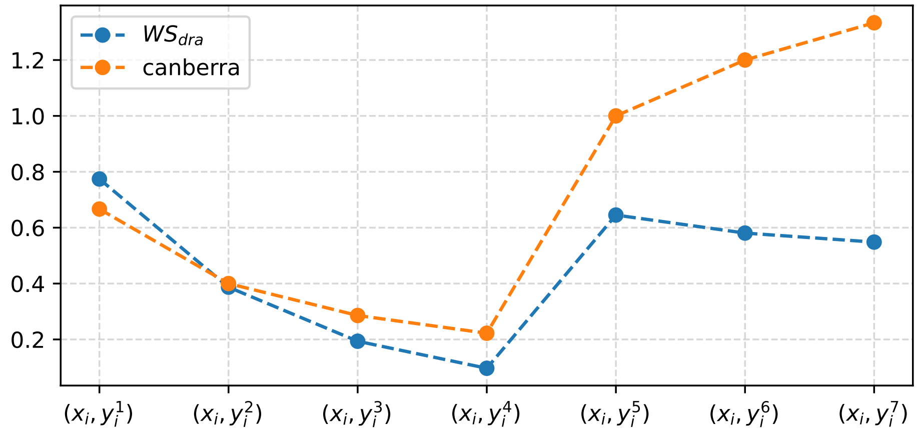


Fig. 1. Comparison of the proposed distance WS_{dra} with the Canberra distance.

where j = 1, 2, 3, 4.

Shifting focus to the second set of rankings, we encounter a scenario where the order of alternatives is swapped, albeit not in adjacent positions. This gives rise to a peculiar situation where the comparison of x_i with $y_i^{(1)}$ results in erroneous rankings at the first and second positions. Similarly, in the comparison of x_i with $y_i^{(5)}$, erroneous rankings occur at the first and third positions, and so on. Intuitively, we would expect a larger distance in the first case, with decreasing values in subsequent comparisons, as errors are initially observed at the first position and then propagate to the further subsequent positions.

However, both the Canberra distance and the $1 - WS$ ratio exhibit counter-intuitive behavior, as their values increase rather than decrease. This contradicts the initial assumption. Moreover, the $1-WS$ ratio cannot be considered a true metric due to its lack of symmetry. The comparison between the results obtained using the WS_{dra} method and the Canberra method is depicted in Fig. 1. The analysis reveals a significant correlation for the first set of alternatives, with a Pearson correlation coefficient of 0.9995. This high correlation indicates a strong agreement between the values obtained from the WS_{dra} method and the Canberra method for this set of rankings.

In the case of the second set of rankings, a similarly strong correlation is observed; however, it possesses a negative nature with a correlation coefficient of -0.9965. This negative correlation suggests an inverse relationship between the values obtained by the WS_{dra} method and the Canberra method for this particular set of rankings.

When considering both sets of rankings collectively, a moderate positive correlation of 0.6995 is observed. This indicates that, overall, there is a consistent relationship between the rankings obtained from the WS_{dra} method and the Canberra method, albeit with a moderate strength of association.

Fig. 1 visually illustrates these correlations, providing a clear understanding of the magnitude and characteristics of the relationship between the WS_{dra} method and the Canberra method for the various sets of rankings. It is important to note that the Canberra distance exceeded the value of 1, which represents the upper limit in the WS_{dra} distance metric.

The WS_{dra} distance metric is based on certain assumptions that contribute to its enhanced reliability and applicability across diverse contexts. Firstly, this metric takes into account the weighted differences between rankings, recognizing that not all changes in rankings hold equal significance. By assigning appropriate weights to these differences, the WS_{dra} metric captures the varying impact of alterations in ranking positions, providing a more accurate assessment of dissimilarity.

Additionally, the WS_{dra} metric adheres to the fundamental principle that "an error is an error." It acknowledges that any deviation or discrepancy between rankings, regardless of its location or magnitude, should be considered as an error. By treating all errors equally, the WS_{dra} metric ensures a fair and unbiased evaluation of dissimilarity, promoting a more reliable comparison between rankings.

Consequently, these examples demonstrate that the distance metric WS_{dra} provides greater reliability, as it considers the weighted differences between rankings, adhering to the principle that "an error is an error." It is important to note that the value of this metric is normalized within the range of 0 to 1, ensuring its applicability across diverse contexts.

V. Conclusion

In this paper, we introduced a new distance metric for the comparison of rankings, demonstrating its effectiveness and advantages over well-established distance measures. This novel measure, which we term the drastic WS distance, conforms to all necessary properties of a true metric and exhibits a unique capability to capture nuances in the ranking structure.

The drastic distance metric provides a compact, standardized representation within the 0 to 1 interval, making it easy to interpret across a broad spectrum of applications. Importantly, it holds the crucial properties of non-negativity, identity of indiscernibles, symmetry, and the triangle inequality. This compliance ensures that our proposed distance metric offers a mathematically sound and reliable framework for comparing rankings, further enhancing its credibility.

By conducting comparative experiments, we illustrated the superior performance of our new drastic distance metric against established measures such as Hamming, Canberra, Bray-Curtis, Euclidean, Manhattan, and Chebyshev distances. The results showcased the new metric's enhanced sensitivity and ability to accurately quantify dissimilarities between rankings, making it a potent tool in the decision-making domain.

In conclusion, the drastic distance metric proposed in this work represents a significant advancement in the area of distance measurement for rankings. With its proven mathematical robustness and practical effectiveness, it has the potential to contribute significantly to decision-making processes across various fields. Future research directions could explore more extensive applications of this metric and further refine its potential through diverse real-world use cases.

Acknowledgment

The work was supported by the National Science Centre 2021/41/B/HS4/01296 (W.S. and A.S.).

References

[1] S.-S. Choi, S.-H. Cha, C. C. Tappert, *et al.*, "A survey of binary similarity and distance measures," *Journal of systemics, cybernetics and informatics*, vol. 8, no. 1, pp. 43–48, 2010.

[2] E. Deza, M. M. Deza, M. M. Deza, and E. Deza, *Encyclopedia of distances*. Springer, 2009.

[3] S. Chen, B. Ma, and K. Zhang, "On the similarity metric and the distance metric," *Theoretical Computer Science*, vol. 410, no. 24-25, pp. 2365–2376, 2009.

[4] M. Zhu, V. Lakshmanan, P. Zhang, Y. Hong, K. Cheng, and S. Chen, "Spatial verification using a true metric," *Atmospheric research*, vol. 102, no. 4, pp. 408–419, 2011.

[5] L. Liberti, C. Lavor, N. Maculan, and A. Mucherino, "Euclidean distance geometry and applications," *SIAM review*, vol. 56, no. 1, pp. 3–69, 2014.

[6] S. Pandit, S. Gupta, *et al.*, "A comparative study on distance measuring approaches for clustering," *International journal of research in computer science*, vol. 2, no. 1, pp. 29–31, 2011.

[7] R. Coghetto, "Chebyshev distance," *Formalized Mathematics*, vol. 24, no. 2, pp. 121–141, 2016.

[8] M. M. Deza, E. Deza, M. M. Deza, and E. Deza, "Distances and similarities in data analysis," *Encyclopedia of distances*, pp. 291–305, 2013.

[9] A. Bączkiewicz, J. Wątróbski, and W. Sałabun, "Distance metrics library for mcda methods," in *R. A. Buchmann, G. C. Silaghi, D. Bufnea, V. Niculescu, G. Czibula, C. Barry, M. Lang, H. Linger, C. Schneider (Eds.), Information Systems Development: Artificial Intelligence for Information Systems Development and Operations (ISD2022 Proceedings). Cluj-Napoca, Romania: Babeș-Bolyai University.*, pp. 1–8, 2022.

[10] P. Zhang, X. Wang, and P. X.-K. Song, "Clustering categorical data based on distance vectors," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 355–367, 2006.

[11] H. A. Abu Alfeilat, A. B. Hassanat, O. Lasassmeh, A. S. Tarawneh, M. B. Alhasanat, H. S. Eyal Salman, and V. S. Prasath, "Effects of distance measure choice on k-nearest neighbor classifier performance: a review," *Big data*, vol. 7, no. 4, pp. 221–248, 2019.

[12] G. Glazko, A. Gordon, and A. Mushegian, "The choice of optimal distance measure in genome-wide datasets," *Bioinformatics*, vol. 21, no. Suppl_3, pp. iii3–iii11, 2005.

[13] R. Kumar and S. Vassilvitskii, "Generalized distances between rankings," in *Proceedings of the 19th international conference on World wide web*, pp. 571–580, 2010.

[14] W. Sałabun, J. Wątróbski, and A. Shekhovtsov, "Are mcda methods benchmarkable? a comparative study of topsis, vikor, copras, and promethee ii methods," *Symmetry*, vol. 12, no. 9, p. 1549, 2020.

[15] A. Karczmarczyk, J. Wątróbski, G. Ladorucki, and J. Jankowski, "Mcda-based approach to sustainable supplier selection," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 769–778, IEEE, 2018.

[16] D. K. Bukovšek and B. Mojškerc, "On the exact region determined by spearman's footrule and gini's gamma," *Journal of Computational and Applied Mathematics*, vol. 410, p. 114212, 2022.

[17] W. Sałabun and K. Urbaniak, "A new coefficient of rankings similarity in decision-making problems," in *Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, June 3–5, 2020, Proceedings, Part II 20*, pp. 632–645, Springer, 2020.

[18] C. Genest and J.-F. Plante, "On blest's measure of rank correlation," *Canadian Journal of Statistics*, vol. 31, no. 1, pp. 35–52, 2003.

[19] A. Shekhovtsov, V. Kozlov, V. Nosov, and W. Sałabun, "Efficiency of methods for determining the relevance of criteria in sustainable transport problems: A comparative case study," *Sustainability*, vol. 12, no. 19, p. 7915, 2020.

[20] W. Sałabun, A. Shekhovtsov, D. Pamučar, J. Wątróbski, B. Kizielewicz, J. Więckowski, D. Bozanić, K. Urbaniak, and B. Nyczaj, "A fuzzy inference system for players evaluation in multi-player sports: The football study case," *Symmetry*, vol. 12, no. 12, p. 2029, 2020.

[21] B. Bera, P. K. Shit, N. Sengupta, S. Saha, and S. Bhattacharjee, "Susceptibility of deforestation hotspots in terai-dooars belt of himalayan foothills: A comparative analysis of vikor and topsis models," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 8794–8806, 2022.

[22] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," *Advances in neural information processing systems*, vol. 25, 2012.

[23] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello, "Canberra distance on ranked lists," in *Proceedings of advances in ranking NIPS 09 workshop*, pp. 22–27, Citeseer, 2009.

[24] E. W. Beals, "Bray-curtis ordination: an effective strategy for analysis of multivariate ecological data," in *Advances in ecological research*, vol. 14, pp. 1–55, Elsevier, 1984.

[25] M. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pp. 503–507, IET, 2013.

DOI: 10.15439/2023F3485
ISSN 2300-5963 ACSIS, Vol. 35

Classifying Speech Acts in Political Communication: A Transformer-based Approach with Weak Supervision and Active Learning

Klaus Schmidt*, Andreas Niekler*, Cathleen Kantner†, Manuel Burghardt*
*Leipzig University, Email: [name].[surname]@uni-leipzig.de
†University of Stuttgart, [name].[surname]@sowi.uni-stuttgart.de

***Abstract*—We present a study on the automatic classification of speech acts in the domain of political communication, based on J. R. Searle's classification of illocutionary acts. Our research involves creating a dataset using the US State of the Union corpus and the UN General Debate corpus (UNGD) as data sources. To overcome limited labelled data, we employ a combination of weak supervision and active learning techniques for dataset creation and model training. Through various experiments, we investigate the influence of external and internal factors on speech act classification. In addition, we discuss the potential for further analysis of speech act usage, using the trained model on the UNGD corpus. The findings demonstrate the effectiveness of Transformer-based models for automatic speech act classification, highlight the benefits of weak supervision and active learning for dataset creation and model training, and underscore the potential for large-scale statistical analysis of speech act usage in the domain of political communication.**

I. Introduction

When we use language, we often want to go beyond the mere conveying of information, but rather want to accomplish a communicative goal and perform a social function, such as making a promise or issuing a threat. In his seminal work "How to do things with words", J. L. Austin has called these kinds of utterances illocutionary acts, which is not an act of just saying something, but an act of doing something by saying something [1]. According to Austin, there are two types of utterances: constatives, which make statements about the world, and performatives, which are acts performed by the speaker with the intention of fulfilling a social function. This concept was later developed further by Searle and the general term *speech act* has been developed since for such performative utterances [2]. Unlike interrogative and imperative sentences that are marked with either a "?" or "!" at the end of a sentence, performatives are not orthographically highlighted as such. Indeed, a central insight of Austin is the fact that there is apparently no trivial indicator differentiating constatives from performatives. As humans, we thus have to rely on linguistic and contextual information to properly recognize performatives and to act upon them accordingly.

While speech acts have an important socio-linguistic function in everyday communication, they are also playing a crucial role in political communication, as they are used by politicians to perform specific actions and influence their audience [3, 4]. Examples include promises for the purpose of getting elected, or requesting legislative action. By analysing the use of speech acts in political discourse, researchers can gain a better understanding of the intentions and motivations of political actors and how they seek to influence public opinion.

In this study, we present a robust computational framework for detecting and correctly identifying speech acts in political communication based on large transformer-based language models for automated and scalable analysis of speech acts in large datasets. The contributions of this paper are as follows:

- Our approach is based on Transformer-based language model classifiers [5], and we employ a combination of weak supervision and active learning techniques for dataset creation and model training.
- Through various experiments, we investigate the influence of external and internal factors on speech act classification.
- We show a concrete use case that demonstrates the use and potential of automatic speech act annotation in large text corpora containing political communication.

The use of speech acts in political communication has up to now received only little attention despite their pervasiveness and utility in this mode of communication (see [6, 7]).

Underlying this concept is the idea that the interconnected network of deontic moral forces within society are established and sustained through specific types of speech acts. J.R. Searle postulates that declarative speech acts, in particular, play a pivotal role in creating institutions and institutional facts. Nevertheless, it has not yet resulted in significant changes, possibly due to the substantial challenges involved in operationalizing his ideas for empirical research."

The current literature on the use of speech acts in political communication has mainly focused on individual sentences or small corpora, lacking a larger, macroscopic perspective. However, with the advent of digital resources that are freely accessible and machine-readable, there is an opportunity to analyse larger datasets of political speeches, for instance parliamentary debates [8] or speeches of singular politicians [9]. This availability of material opens up new perspectives for rather empirical research questions along the lines of political sciences, international relations, computational social science and corpus linguistics.

An issue surrounding this topic is the vast amounts of data that need to be processed in order to gain valuable insights into macroscopic patterns of their uses among political actors. Especially since certain types of utterances are quite infrequent and sporadic, the criterion applies that manual sampling would certainly not be able to produce representative distributions for quantitative evaluations in any reasonable amount of time. With the advances of computational approaches for accessing different language phenomena from Natural language Processing (NLP) and Artificial Intelligence (AI) made in recent years, it is now possible to find this language phenomenon in an automated way and with good quality in large amounts of data.

Future applications will be the quantitative analysis of political communication in international institutions [10]. This will provide a computational method for studying how to do international politics with words [11]. With the methodology shown here, the following research questions, for example, become conceivable: How can speech act theory be applied to understand declarations of war, inaugurations, pardons, government statements, etc.? How do international obligations and the self-binding of sovereign states to international norms and rules emerge under conditions of anarchy?

II. Related Work

Automatic speech act classification has been a subject of research for some time, focusing on dialogue acts [12, 13, 14]. Earlier studies on Korean employed various methods, including Hidden Markov Models [15], maximum entropy models [16], and supervised machine learning algorithms [17]. Unlike Austin and Searle's speech acts, dialogue acts specifically target synchronous language used in direct communication and many of the classification schemes used in research do not align with Austin's or Searle's speech act classification.

In recent years, researchers have increasingly utilized Deep Neural Networks (DNNs) in Natural Language Processing (NLP) due to advancements in computing capabilities. This has led to the adoption of more sophisticated approaches in addressing the issue at hand. Notably, recurrent neural networks (RNNs) [18] and convolutional neural networks (CNNs) [19] have been explored as viable options for tackling the challenge of speech act detection. Existing research on speech and dialogue act detection primarily focuses on synchronous language in online communication. However, little attention has been given to speech act detection in political communication, except for notable exceptions such as [20]. The emergence of the Transformer architecture [5] and pre-trained Transformer-based language models like BERT [21] has significantly changed the way NLP is practised, and automated processing of even difficult language problems is becoming increasingly possible. To the best of our knowledge, no previous work exists on classifying speech acts according to Searle's classification of illocutionary acts.

III. Experiment Design

In our experiments, we developed a Transformer-based model specifically designed to classify sentences according to Searle's classification of illocutionary acts, referred to as "speech acts" in this study. We constructed the dataset using weak supervision and further refined it through active learning techniques. Additionally, we explored possible avenues for future research, which encompass improving the model's performance and assessing its applicability and relevance in computational political science studies.

A. Corpora

To construct a suitable dataset for model training and initial analysis, we have selected two corpora of similar yet distinct sub-domains within political communication: the US State of the Union (SOTU) addresses and the speeches delivered at the UN General Debate. The SOTU corpus comprises the State of the Union presidential addresses given annually from 1790 to 2017.[1] In our experiments, the SOTU corpus' temporal range was limited to between 1990 and 2017. The UN General Debate corpus (UNGD) comprises speeches delivered by national representatives, including presidents and foreign ministers, from each UN member country during the annual UN General Debate [8]. Spanning the years 1970 to 2020, this corpus provides a rich collection of political speeches from a diverse range of international stakeholders. Aside from the transcriptions of speeches in English, the UNGD corpus also includes a variety of metadata for each speech such as the year and session of the UN General Debate that the speech was held in, the name of the speaker, their position, the original language of delivery as well as the represented country. We have adopted a standardized format for referencing individual speeches in the UNGD corpus, using the year of the speech and a three-letter country code based on ISO 3166-1 alpha-3 to denote the speaker's country he or she represents, such as ”1979_IRQ” for the 1979 speech of the representative of Iraq.

We formulated the task of identifying speech acts as a multi-class classification problem. To achieve this, we have employed the theoretical framework of illocutionary acts introduced by Searle [2] and identified five classes of speech acts and one open class for sentences where none of the labels can be assigned:

- *ASSERTIVE*, the speaker commits to the truth of a stated expression.
- *EXPRESSIVE*, the speaker expresses their personal thoughts and feelings
- *COMMISSIVE*, the speaker commits themselves to a future action.
- *DIRECTIVE*, the speaker issues orders or instructions to the recipients.
- *DECLARATIVE*, the speaker, using granted institutional powers, alters or defines (social) realities.
- *NONES*, an open class for sentences that contain none of the above speech act types.

[1] https://www.kaggle.com/datasets/rtatman/state-of-the-union-corpus-1989-2017

B. Heuristic Labelling using Weak Supervision

When creating a dataset to train our model, we need to define the assignment of labels by existing linguistic properties of each class. To assign speech act annotations to linguistic expressions containing the linguistic properties just mentioned, we use the weak supervision library *skweak* [22]. This library enables the generation of so-called weak labels through annotator functions. Although these labelling functions rely on simple heuristics and may not achieve particularly high precision, they effectively enable us to efficiently search for speech acts employed in political communication during the dataset creation process.

A crucial aspect of Austin's seminal work on speech acts pertains to the notion that certain verbs are closely linked to a specific class of speech act, which he coined *performative verbs*. We relate closely to this basic concept in our definition of the relevant linguistic features for speech acts, in order to define heuristics for the labelling functions. We chose the following verb relationships to supervise the labelling process:

- **Assertive**: *think, know, believe, convince, presume, assume, admit*
- **Expressives:** *apologize, condole, lament, deplore, forgive, welcome, thank, forgive, boast*
- **Commissives:** *promise, vow, guarantee, offer, will, refuse, volunteer*
- **Directives::** *ask, order, command, request, beg, plead, pray, invite, permit, advise, must, should*
- **Declaratives::** *announce, declare, dismiss, nominate, pronounce, pass, adopt, support, oppose, advocate, condemn*

To capture general speech act features, we developed several labelling functions that assign a general "speech act" label [*SA*] which is added to the matching sentences. This approach allows us to assign weak labels to encompass general features that are not confined to a specific speech act. Sentences containing performative verbs are assigned the speech act label considered characteristic for them, as shown in the list above. We derived the labelling rules from sample sentences in Austin's work *"How to do things with words"* [1]. Note, that multiple labelling functions can be applied to a single sentence if the criteria are met. In a final step, these weak labels, which represent a supervision signal, are subsequently aggregated to form final labels. This aggregation model takes into account the varying degrees of confidence associated with each weak label. By leveraging the sequential dependencies of the weak labels between sentences, the aggregation ensures a more accurate labelling outcome. The labelling functions can be briefly described as follows:

- **Subject is 1st person**: Assigns the label [*SA*] if the subject of a sentence is either "I" or "we".
- **Main verb is present tense**: Assigns the label [*SA*] if the main verb of a sentence is in present tense.
- **Object is 2nd person**: Assigns the label [SA] if the object of the sentence is the pronoun "you".
- **Sentence contains imperative**: Assigns the label [*DIRECTIVE*] if the sentence is an imperative without an overt subject.
- **Sentence contains interrogative**: Assigns the label [*DIRECTIVE*] if the main verb precedes the subject and the sentence ends with a question mark "?".
- **Sentence contains performative verb**: Assigns the label associated with the performative verb [*ASSERTIVE, EXPRESSIVE, COMMISSIVE, DIRECTIVE, DECLARATIVE*] if the lemma of a performative verb is present in the sentence.

Processing the SOTU as well as the UNGD corpus using this weak supervision approach yields a list of sentences that likely contain speech acts.

C. Active Learning

Active learning is an approach that utilizes query strategies to select the most informative samples from an unlabelled pool of data, guided by a classifier trained on a set of existing labelled data. The goal is to intelligently choose data points for labelling, in order to improve the performance and efficiency of the learning process. Over the years, many query strategies have been proposed by various researchers, most of them using prediction based query strategies [23]. Active learning follows a process that involves two pools: a labelled pool and an unlabelled pool. The goal is to transfer data from the unlabelled pool to the labelled pool. In prediction-based query strategies, a primary model is trained on the labelled pool, which is then used to identify the most informative samples to query from the unlabelled pool. These queried samples are then presented to a domain expert (also known as oracle in active learning jargon), who manually confirms or rejects the labels of the primary model. The labelled samples are subsequently added to the labelled pool. This process continues iteratively until a predefined stopping criterion is met. With this approach, active learning allows the progressive enhancement of the model's performance by actively selecting the most informative samples for annotation.

We used the active learning library Small-Text [24] to facilitate experiments with active learning. This library provides a user-friendly and consistent interface which makes it very simple to set up the experiments. After some initial testing, we selected the prediction-based query strategy *Prediction Entropy* as our choice and set a fixed iteration size of 10 iterations as the stopping criterion. In each iteration, we queried 20 samples.

To effectively utilize the library, it is essential to choose a transformer-based language model as the underlying framework for the classification task. Among the various models we tested, we ultimately opted for the light-weight DistilBERT [25] with the default configuration provided through HuggingFace.[2] This decision was primarily driven by DistilBERT's significantly reduced computational overhead, with minimal performance drawbacks compared to the resource-intensive BERT model. Since the iterative approach involves many training processes, the model accelerates them significantly,

[2] https://huggingface.co/distilbert-base-uncased

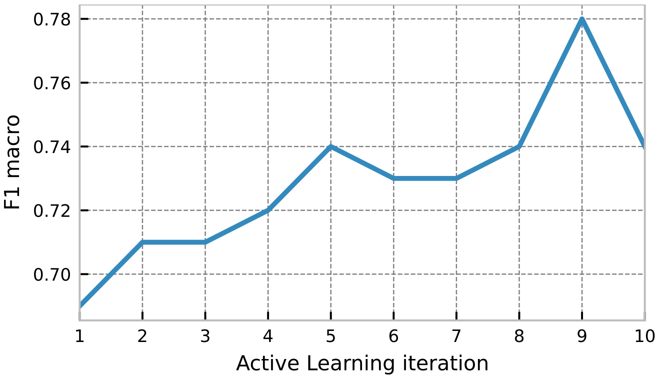


Figure 1. Learning curve of f1 macro performance on test set over all classes

making them more manageable in an interactive setting with domain experts.

The results from the weak supervision process were subsequently filtered by a domain expert, resulting in a valid semi-manual training dataset for initializing the active learning process. Despite the linguistic features serving as a basic codebook for annotating queried samples, a considerable number of samples proved to be non-trivial. Given the fuzzy nature of speech act use and their interpretation, disagreements are unavoidable. For training, we stratified the dataset by class and performed a split, allocating 60% of the data for training and 40% for evaluation. This choice of a 60:40 split was made to ensure that both sets capture a representative sample of each class. By evaluating each iteration's performance using F1-scores for individual classes as well as the macro average, we could effectively track the model's progress over iterations and visualize it through a learning curve.

IV. Experiments and Results

A. Active Learning and Classifier Performance

After applying the weak supervision approach and correcting its output, our initial dataset consisted of 175 samples in the labelled pool and 118 samples in the evaluation set. Once the initial model was trained using the samples found in the labelled pool, we further refined it through the active learning process. In each iteration, we queried, annotated, and then added 20 newly labelled samples to the labelled pool. After each addition, the model was retrained using the augmented labelled pool. This iterative process continued until the stopping criterion of 10 iterations was met.

Looking at Fig. 1, after performing ten active learning iterations, we observed a rise in F1-macro performance from 0.69 of the initial model to 0.78 of the best model in the ninth iteration before the performance cratered somewhat again in the tenth iteration. In Table I we show detailed information about the classification performance of the model trained in the ninth iteration.

Examining the individual F1-scores, we noticed large discrepancies between the highest score of the commissive class and lowest score of the expressive class. In terms of precision, assertives do have a medium performance, indicating that the training examples provide very ambiguous features to the classifier. In terms of recall, expressives and declaratives show comparatively medium performance. In the subsequent experiments presented in sections IV-B, IV-D as well as the showcase demonstrated in section IV-E, we utilized the model trained during the 9th iteration, which incorporated an additional 180 training samples obtained through the active learning process.

Class	Precision	Recall	F1	Support
Assertive	0.71	0.89	0.79	19
Expressive	0.82	0.60	0.69	15
Commissives	0.83	1.0	0.91	24
Directives	0.78	0.96	0.86	26
Declaratives	1.0	0.67	0.80	18
Nones	0.80	0.50	0.62	16
macro average	**0.82**	**0.77**	**0.78**	**118**

Table I
Table presenting the performance of the classifier from the 9th active learning iteration over all speech act classes

B. Adding context sentences in the training data

The current model employed in our research utilizes single sentences as the units of classification. While this approach may simplify the classification task, it is not necessarily evident that speech acts occur independently of each other. Therefore, it is important to acknowledge the potential limitations of this local single sentence approach.

To better understand the relationships between single sentences, we conducted an investigation by visualizing the occurrence of all speech acts from five randomly selected speeches. Figure 2 reveals a strong clustering behaviour in the use of speech acts, even from this small sample.

We conducted an additional analysis by calculating the pointwise mutual information (PMI) between each speech act and its neighbouring co-occurring speech acts in a sample of 100 randomly selected speeches. The PMI measures the statistical association between two events, in this case, speech acts and their neighbouring counterparts. Positive PMI scores indicate a higher than random likelihood of co-occurrence. On the other hand, negative scores indicate a lower than expected likelihood of co-occurrence. Our analysis revealed a significantly higher co-occurrence between speech acts of the same class, as would be expected by random chance. Furthermore, we observed moderately positive associations between directives and commissives, as well as between expressives and commissives. These findings suggest that considering the context of neighbouring sentences might be crucial for accurate classification of speech acts.

To test this hypothesis, we investigated the effectiveness of extending the classification unit by incorporating a window of neighbouring context sentences around a sentence from the trainings set. This approach enables the classifier to capture the contextual information surrounding a sentence. However, as depicted in Figure 3, our findings indicate that this approach

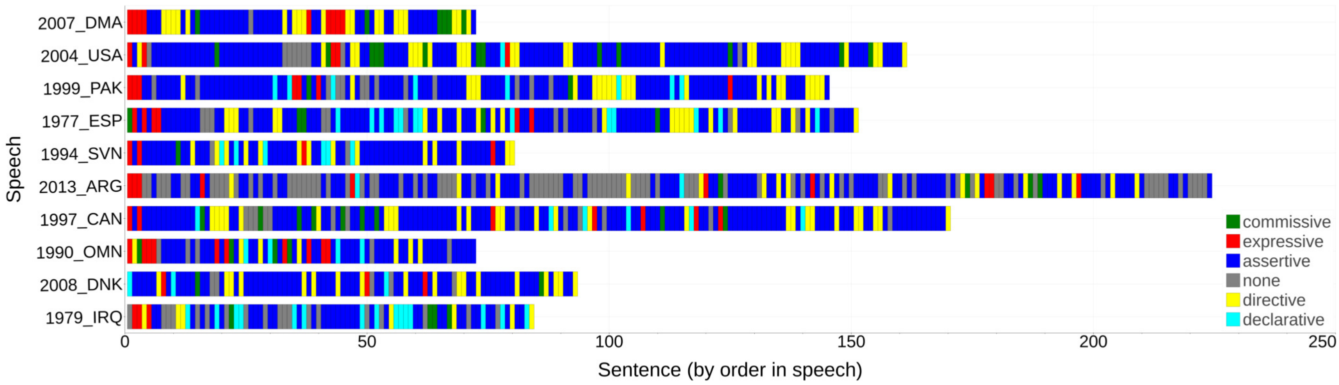


Figure 2. Speech act usage pattern over the course of a speech. The bars represent a speech in full length and the colours mark a respective speech act. It is recognizable that the same speech acts often occur consecutively in the sequence.

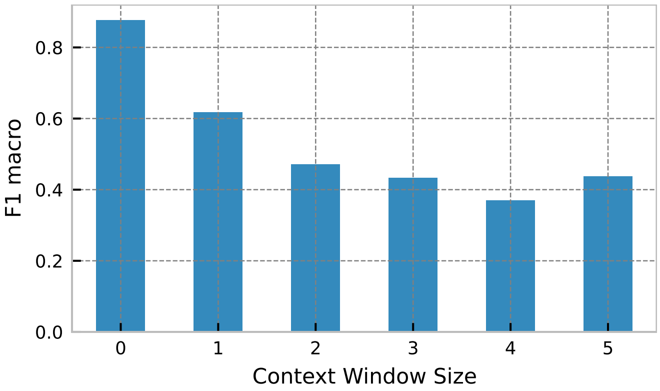


Figure 3. Model performance on the entire dataset (train+test) for various context sizes. The Number of context sentences is equivalent to 2 × Context Window Size.

did not yield a substantial improvement over using a single sentence window.

C. Feature Importance of Metadata

In the preceding section, we demonstrated the significance of linguistic features in the classification of speech acts, as evidenced by the rather promising results obtained from the transformer model. In addition to the raw speeches, the UNGD Corpus includes various metadata, such as the name and position of the speaker, the country represented, and the original language of the speech. To explore the potential significance of metadata in relation to the occurrence of speech acts within the speeches, we trained a separate model in the form of a Random Forest (RF) regressor. This model is aimed to predict the distribution of each speech act class based on the available metadata alone. The objective of this approach is to examine the potential influence of extra-linguistic features, such as metadata, on the distribution of speech acts by examining the *importance* of each feature. The hyper-parameters for the RF model were left at their default values provided by scitkit-learn, which consisted of using 100 trees and the Gini impurity criterion.

We compared the performance of the trained model to a baseline dummy classifier and observed a substantial improvement in goodness of fit for commissives (reducing the mean squared error from 10.75 to 2.4) and directives (reducing the mean squared error from 42.8 to 9.1). These results indicate a significant correlation between the metadata and the distribution of commissives and directives.

To identify the most influential features, we conducted an analysis of feature importances in the model. In RFs, feature importance quantifies the average decrease in impurity across all trees for each feature [26]. Figure 4 showcases the results of this analysis for commissives and directives, providing interesting insights. The feature importance analysis indicates that certain variables have a notable impact on the distribution of speech acts. For commissives, the role of the speaker is a significant variable, with the category of *president* standing out prominently. Additionally, the country represented by the speaker also plays a crucial role, with *USA* and *Japan* being particularly influential factors.

Similarly, for both commissives and directives, the original language of the speech exhibits a strong effect on the distribution of speech acts. Notably, speeches delivered in English demonstrate a substantial influence on the observed distributions.

D. Linguistic Features for Qualitative Assessment

A long-lasting shortcoming of using deep neural networks architectures such as Transformers is the black box nature of these models. Different approaches under the banner of *Explainable AI* such as CAPTUM [27] and shapley values [28] have been investigated in order to elucidate which features are important in a model's prediction. In our investigations, we used CAPTUM which is implemented with the *transformers_interpret* library.[3] This approach assigns *attribution scores* to each of the (sub-)token features which quantify the importance of each feature for the classification of a particular class. These scores are scaled between 1, indicating maximum attribution for classifying this as a particular class and -1,

[3] https://github.com/cdpierse/transformers-interpret

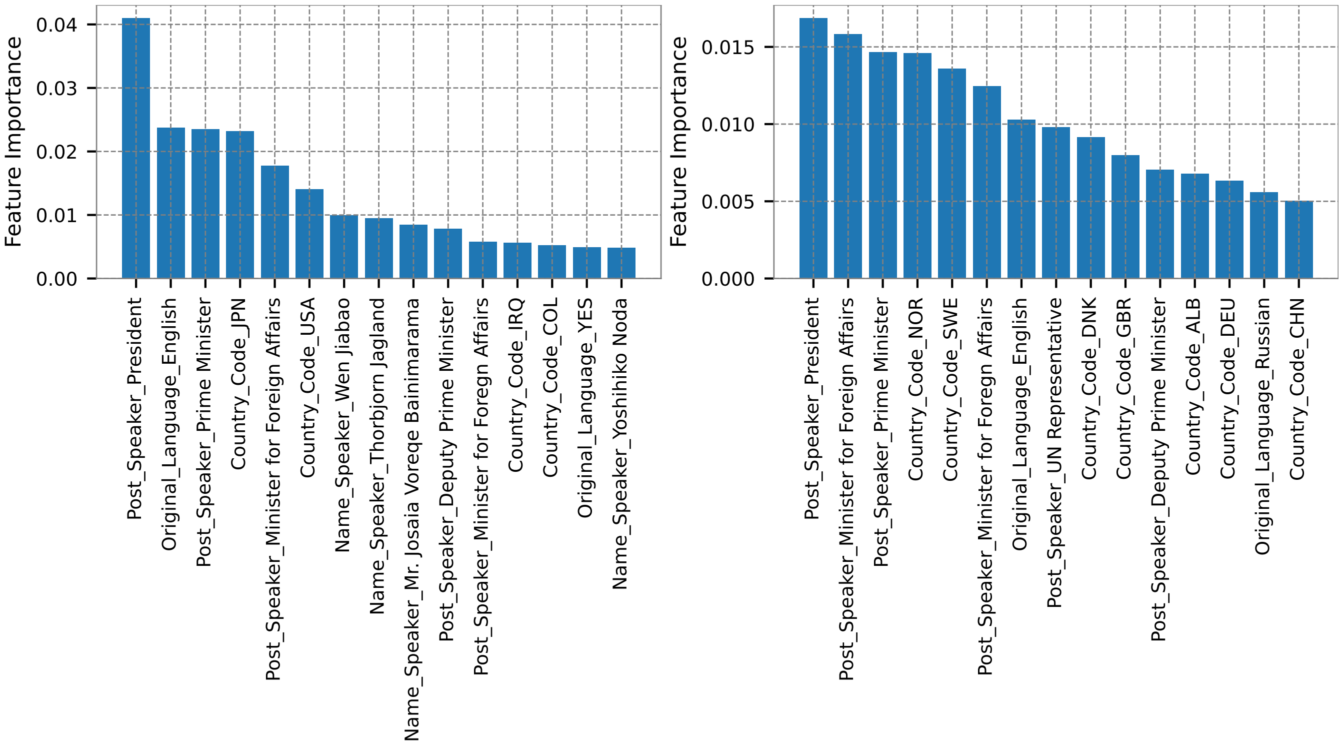


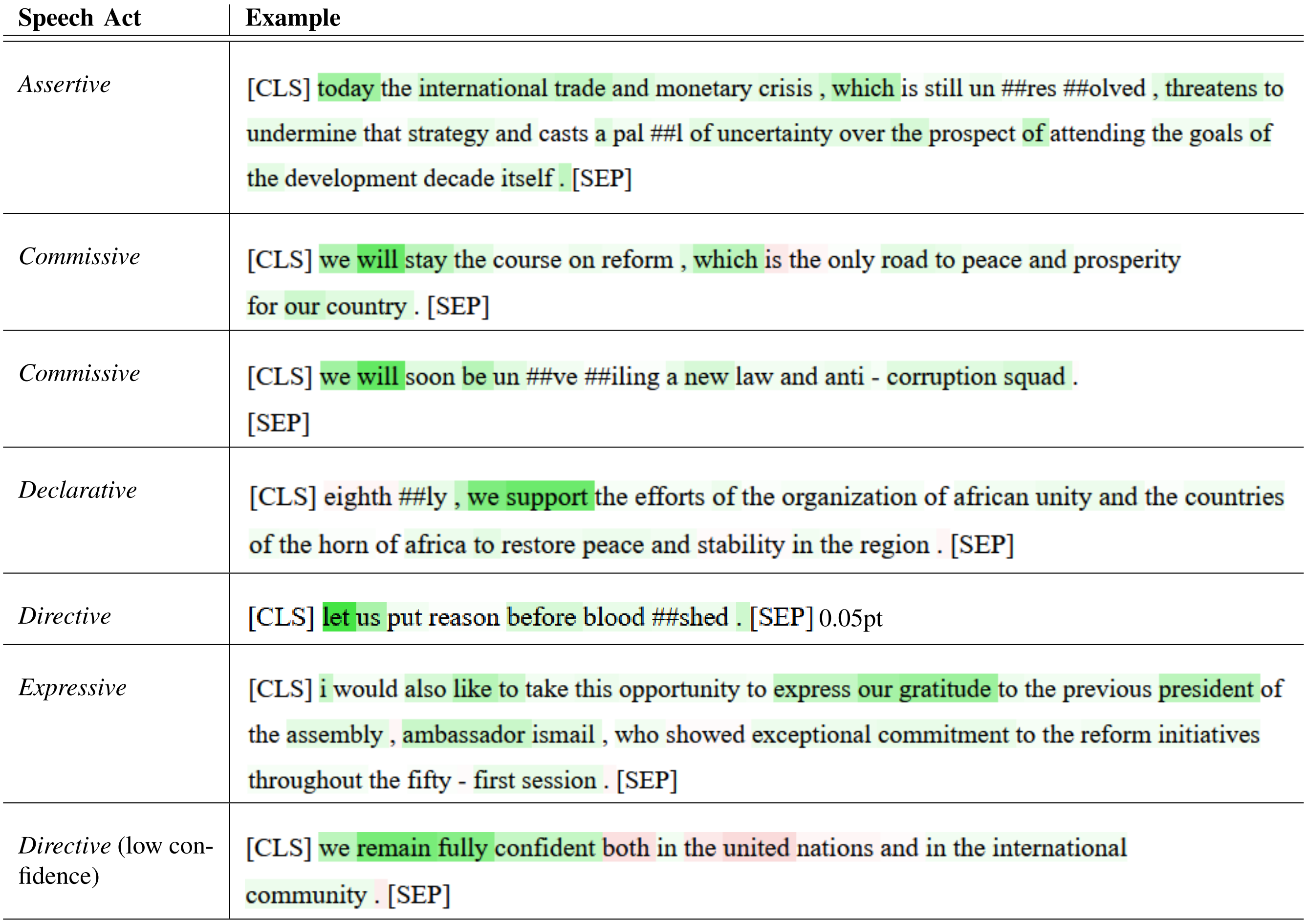
Figure 4. Importance of metadata in commissives and directives

indicating that this feature is strongly associated with class other than the predicted one.

Firstly, we look at the examples with most confidence so that we can understand whether the words shown in Section III-B, each typical of the respective speech act types, are also important features for classification. Second, we investigated which kind of samples expose the lowest confidence in the model in terms of prediction probability in the hope of identifying features that cause uncertainty within the model's prediction confidence. We see a strong connection between Austins's performative verbs and the words that are identified as important features through the training process. As the examples shown in Table 5 demonstrate, the most important features are performative verbs that fit almost prototypically with the examples shown earlier in Section III-B. For example, the phrases *we will* for commissives, *we support* for declaratives or *let us* for directives contain typical performative verbs that are also found in our dataset and are also important features in the classification (see Figure 5). Of interest, however, are the words that do not belong to the group of performative verbs but are nevertheless learned as positive features in each speech act type, as it is shown in Figure 5. Here, assertiva have strong thematic connections to international relations (e.g. *international trade*). Other examples for assertives from the dataset also contain words like *economy* or *cooperation*. Commissiva, on the other hand, have a strong connection to a vocabulary combining promises regarding problems to be solved (e.g. *corruption squad, road to peace*). Declaratives often include country names, as some utterances emphasize support for different states (*Horn of Africa, region*). Directives and expressives function somewhat differently here. Used as rhetorical or linguistic means of communication, expressives typically include addressing specific persons or expressions of appreciation (*ambassador, president, gratitude*). In the case of directives, we often find statements to allow oneself a comment or a motivating request (*reason before bloodshed*).

For the example with a low confidence shown in Figure 5, it turns out that other rather unspecific words are also negatively evaluated and appear in greater accumulation. This greater accumulation of negative words is very characteristic for low confidence classifications in the dataset, although no specific thematic or communicative connections or patterns are recognizable.

This observation shows us perspectives for further use of speech acts. On the one hand, certain speech acts can be expected in certain concrete thematic or regional contexts and one can match this expectation with real political communication (*assertive, declarative, commissive*). Discrepancies or patterns can then be described for different discourses and thus allow for a qualitative assessment. Second, rhetorical devices can be derived as a component of strategic communication (expressives, directives). This is especially interesting, since analyses are conceivable here that refer to power relations between the involved actors.



Speech Act	Example
Assertive	[CLS] today the international trade and monetary crisis , which is still un ##res ##olved , threatens to undermine that strategy and casts a pal ##l of uncertainty over the prospect of attending the goals of the development decade itself . [SEP]
Commissive	[CLS] we will stay the course on reform , which is the only road to peace and prosperity for our country . [SEP]
Commissive	[CLS] we will soon be un ##ve ##iling a new law and anti - corruption squad . [SEP]
Declarative	[CLS] eighth ##ly , we support the efforts of the organization of african unity and the countries of the horn of africa to restore peace and stability in the region . [SEP]
Directive	[CLS] let us put reason before blood ##shed . [SEP] 0.05pt
Expressive	[CLS] i would also like to take this opportunity to express our gratitude to the previous president of the assembly , ambassador ismail , who showed exceptional commitment to the reform initiatives throughout the fifty - first session . [SEP]
Directive (low confidence)	[CLS] we remain fully confident both in the united nations and in the international community . [SEP]

Figure 5. Example sentences visualized with the library CAPTUM [27] where the positive contributing features are shown in green and the negative contributing features are shown in red. The brightness indicates the importance of the features. The tokenization is taken from the transformer model.

E. Showcase: speech acts as time series data

This section will emphasize the practical application of the model through a demonstrative analysis in the UNGD corpus. We will use sentence-based annotations of speech acts to quantitatively measure the shift in communicative patterns within the UN General Debate. Specifically, we want to demonstrate a method for exploring potential influences of significant political events on speech act usage patterns. It is important to note that this demonstration serves as an illustrative example rather than a rigorous scientific investigation of speech act usage. By highlighting the viability of our approach, we aim to emphasize its potential as a valuable research tool in the fields of political and social sciences. For this purpose, we decided to focus on the political events that unfolded in Ukraine from 2013 to 2014. Later being known as the Maidan Uprising, a series of protests and civil unrest unravelled on November 21, 2013, primarily centred around Kyiv's Maidan Nezalezhnosti (Independence Square) [29]. These protests ultimately led to the Revolution of Dignity in February 2014, albeit with a heavy toll of 108 protesters and 13 police officers losing their lives in the clashes [30]. Consequently, the Ukrainian parliament, with a significant majority of 328 out of 450 votes (approximately 73% of the votes), voted to remove President Viktor Yanukovych from power. Yanukovych disputed the legality of the vote and sought assistance from Russia. Russia criticized the events, labelling them a ”coup”. Subsequently, pro-Russian protests erupted in southern and eastern Ukraine. Russia occupied and eventually gained control of Crimea, while armed pro-Russian groups seized government buildings and declared Donetsk and Luhansk as independent states. This resulted in the onset of the Donbas war, prompting international efforts to diplomatically address and find resolution to the unfolding events within the international community.

To investigate the change in the strategic political communication following those events, we analysed the speech act usage of *directives*, which primarily attempt to motivate the addressees to take an action, of Ukrainian representatives in the UN General Debate. By choosing this scenario, we aim to clearly demonstrate the ways in which events can affect the language used in international political discourse by identifying and quantifying speech acts as an indicator variable.

To identify a statistically significant change in usage pat-

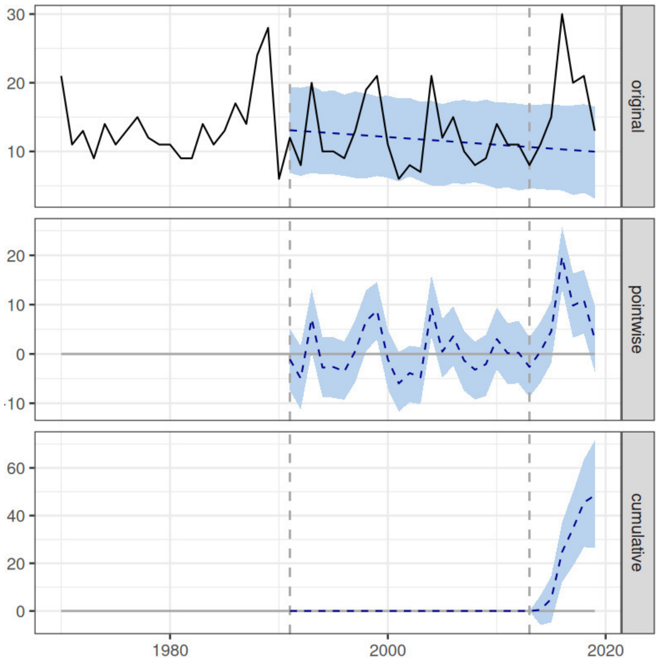


Figure 6. CausalImpact plot comparing directive speech act usage in absolute numbers between the years 1991-2013 and 2014-2019

terns of *directives* between the period starting in 2014 and the most recent available speech in 2019, we utilized the R library *CausalImpact* [31]. Specifically, we compared the *directive* usage frequency of Ukrainian representatives after the Maidan events (2014-2019) with those during the period before the events, which spans from the dissolution of the Soviet Union in 1991 up until 2013. The software creates a Bayesian structural time-series model which is used to predict how the response would have evolved after an event, called intervention, if the event had never occurred. The result is the cumulative effect of the intervention on the time series.

In the model, we see a positive effect on the use of *directive* speech acts in the period between 2014-2019. The first panel in Figure 6 showcases the observed data and a calculated prediction derived from the data of the pre-intervention period. The pre-intervention period is enclosed by two dotted vertical lines and serves as the baseline for comparison with the post-intervention period. The second panel shows the pointwise causal effect of the model, which is the calculated difference between the observed data and the prediction. The third panel presents the cumulative effect of the model, which is the summation of the pointwise causal effect over time. The positive effect on the use of *directive* speech acts observed during the period after the Maidan protests in Ukraine in 2014 is statistically significant and unlikely to be due to random fluctuations. It is important to note that during this period, not only the frequency of directive speech acts increased significantly, but the speeches also showed a general increase in length. This suggests that the involvement and activation of the international community by Ukrainian representatives has increased and become a larger part of Ukraine's political communication.

V. Discussion

In this paper, we investigate for the first time how the linguistic phenomenon of speech acts described by Searle can be automatically identified in text corpora using a transformer-based approach and be made accessible for quantitative and qualitative analyses. In doing so, the study focuses on the identification of speech acts within very large datasets of political communication, which makes the results interesting for political science research. Furthermore, the study presents very transparently which linguistic properties are included for the definition of speech acts. Our experiments demonstrate the effectiveness of using a combination of weak supervision and active learning for dataset creation in the classification of speech acts. Weak supervision enables the identification of samples that satisfy defined linguistic features, while active learning can provide new examples from an unlabelled dataset. The results show a 0.09 increase in F1 score within 9 iterations of the model, indicating the usefulness of these techniques in expanding small datasets and improving model performance. The problems we encountered in identifying *expressives* and *declaratives* in terms of recall indicate that there may be variance between the examples in the training data and the test data, or the linguistic characteristics of these speech acts may be ambiguous, making it difficult for the classifier to properly delineate between these and other classes. However, evaluating the entire process still poses challenges, especially with the limitations of our evaluation set, which has a rather small size. The properties of this set might introduce a high degree of variance and possible biases in the evaluation. To address these challenges, future research will focus on developing a robust evaluation framework and larger evaluation sets for the methods employed.

Aside from linguistic features, our observations indicate that extra-linguistic factors play a role in influencing the distribution of speech acts. Specifically, variables such as the speaker's role, the country represented, and the original language of the speech have shown significant influence on the occurrence and distribution of speech acts. These findings emphasize the importance of considering contextual factors when studying speech acts in political discourse. Understanding the impact of these variables can contribute to the development of more accurate speech act classification frameworks and provide valuable insights into the factors shaping the use of speech acts.

Additionally, our findings suggest tentative evidence of relationships between speech acts, as utterances of the same class tend to occur in proximity to each other. This observation not only contributes to our understanding of how speech acts are used in actual communication, but also provides valuable insights for the classification task. While our initial attempts at utilizing neighbouring utterances did not yield satisfactory results, further exploration in this area holds promise for enhancing the accuracy of speech act classification. One possible approach is to employ techniques such as linear chain Conditional Random Fields (CRFs) modelling [32] or similar sequential models. Such modelling can capture the sequential dependencies among speech acts and exploit the contextual information provided by neighbouring utterances. By incorporating such techniques, we can potentially improve the classification performance by considering the sequential nature of speech acts.

The application of our model concerning the use of *directives* by the representatives of Ukraine demonstrate the potential of utilizing the trained model for quantitative research. We presented a statistical analysis technique to explore the impact of political events on speech act usage patterns. By examining specific periods and comparing them to relevant baselines, we can gain insights into how political events shape language use in international discourse. The approach presented here can be extended to investigate other geopolitical events and their influence on speech act patterns, opening the potential for providing a deeper understanding of the dynamics of political communication, which is crucial for researchers and practitioners in fields such as political science, international relations, and diplomacy. In addition, our analysis of the relevant features with CAPTUM demonstrated that the qualitative evaluation of speech acts has the potential to enhance our understanding of the strategic use of speech acts. Automatic speech act detection can thus contribute to various fields by facilitating empirical studies on persuasive strategies employed by political actors. It not only enables the monitoring of shifts in rhetoric and discourse, but also provides insights into the motivations and intentions behind political speeches. By employing and further developing this technology, researchers can delve deeper into understanding the intricate dynamics of political communication and gain a comprehensive understanding of the strategies and objectives employed by politicians.

References

[1] J. L. Austin, *How to do things with words*. Cambridge, Mass., Harvard University Press, 2003., 1962.

[2] J. R. Searle, "A classification of illocutionary acts," *Language in society*, vol. 5, no. 1, pp. 1–23, 1976. doi: 10.1017/s0047404500006837

[3] P. L. Berger and T. Luckmann, *Die gesellschaftliche Konstruktion der Wirklichkeit: Eine Theorie der Wissenssoziologie*. Frankfurt am Main: Fischer, 1966.

[4] J. Habermas, *Theorie des kommunikativen Handelns: Handlungsrationalität und gesellschaftliche Rationalisierung*. Frankfurt: Suhrkamp, 1995, vol. 1.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. doi: 10.5555/3295222.3295349. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[6] S. S. M. Hashim and S. Safwat, "Speech acts in political speeches," *Journal of Modern Education Review*, vol. 5, no. 7, pp. 699–706, 2015. doi: 10.15341/jmer(2155-7993)/07.05.2015/008

[7] M. Ulum, D. Sutopo, and W. Warsono, "A comparison between trump's and clinton's commissive speech act in america's presidential campaign speech," *English Education Journal*, vol. 8, no. 2, pp. 221–228, 2018.

[8] A. Baturo, N. Dasandi, and S. J. Mikhaylov, "Understanding state preferences with text as data: Introducing the un general debate corpus," *Research & Politics*, 2017. doi: 10.1177/2053168017712821

[9] G. Peters and J. T. Woolley, "The state of the union, background and reference table," The American Presidency Project, Santa Barbara, CA, 1999–2021. [Online]. Available: https://www.presidency.ucsb.edu/node/324107/

[10] J. Duffield, "What are international institutions?" *International Studies Review*, vol. 9, pp. 1–22, 2007. doi: 10.1111/j.1468-2486.2007.00643.x

[11] C. Daase, S. Engert, M.-A. Horelt, J. Renner, and R. Strassner, *Apology and Reconciliation in International Relations: The Importance of Being Sorry*. London: Routledge, 2015.

[12] C. Moldovan, V. Rus, and A. C. Graesser, "Automated speech act classification for online chat." *MAICS*, vol. 710, pp. 23–29, 2011.

[13] B. Bayat, C. Krauss, A. Merceron, and S. Arbanowski, "Supervised speech act classification of messages in german online discussions," in *The Twenty-Ninth International Flairs Conference*, 2016.

[14] Y. Liu, K. Han, Z. Tan, and Y. Lei, "Using context information for dialog act classification in DNN framework," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017. doi: 10.18653/v1/D17-1231 pp. 2170–2178. [Online]. Available: https://aclanthology.org/D17-1231

[15] S. Lee and J. Seo, "Korean speech act analysis system using hidden markov model with decision trees," *International Journal of Computer Processing of Oriental Languages*, vol. 15, no. 03, pp. 231–243, 2002. doi: 10.1142/s0219427902000625

[16] W. S. Choi, H. Kim, and J. Seo, "An integrated dialogue analysis model for determining speech acts and discourse structures," *IEICE TRANSACTIONS on Information and Systems*, vol. 88, no. 1, pp. 150–157, 2005. doi: 10.1093/ietisy/e88-d.1.150

[17] N. Song, K. Bae, and Y. Ko, "Effective korean speech-act classification using the classification priority application and a post-correction rules," *Journal of KIISE*, vol. 43, no. 1, pp. 80–86, 2016. doi: 10.5626/jok.2016.43.1.80

[18] R. Li, C. Lin, M. Collinson, X. Li, and G. Chen,

"A dual-attention hierarchical recurrent neural network for dialogue act classification," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019. doi: 10.18653/v1/K19-1036 pp. 383–392. [Online]. Available: https://aclanthology.org/K19-1036

[19] D. Yoo, Y. Ko, and J. Seo, "Speech-act classification using a convolutional neural network based on pos tag and dependency-relation bigram embedding," *IEICE Transactions on Information and Systems*, vol. 100, no. 12, pp. 3081–3084, 2017. doi: 10.1587/transinf.2017edl8083

[20] S. Subramanian, T. Cohn, and T. Baldwin, "Target based speech act classification in political campaign text," in *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019. doi: 10.18653/v1/S19-1030 pp. 273–282. [Online]. Available: https://aclanthology.org/S19-1030

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019. doi: 10.18653/v1/N19-1423 pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[22] P. Lison, J. Barnes, and A. Hubin, "skweak: Weak supervision made easy for NLP," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Aug. 2021. doi: 10.18653/v1/2021.acl-demo.40 pp. 337–346. [Online]. Available: https://aclanthology.org/2021.acl-demo.40

[23] C. Schröder and A. Niekler, "A survey of active learning for text classification using deep neural networks," 2020. [Online]. Available: https://arxiv.org/abs/2008.07267

[24] C. Schröder, L. Müller, A. Niekler, and M. Potthast, "Small-text: Active learning for text classification in python," 2021. [Online]. Available: https://arxiv.org/abs/2107.10314

[25] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[26] G. James, T. Hastie, R. Tibshirani, and D. Witten, *An Introduction to Statistical Learning: With Applications in R*. Springer, 2013. ISBN 978-1-4614-7137-0

[27] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[28] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[29] "Euromaidan," page Version ID: 1153492594. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Euromaidan&oldid=1153492594

[30] "Revolution of dignity," page Version ID: 1154376283. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Revolution_of_Dignity&oldid=1154376283

[31] K. H. Brodersen, F. Gallusser, J. Koehler, N. Remy, and S. L. Scott, "Inferring causal impact using Bayesian structural time-series models," *Annals of Applied Statistics*, vol. 9, pp. 247–274, 2015. doi: 10.1214/14-aoas788

[32] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. doi: 10.5555/645530.655813. ISBN 1558607781 p. 282–289.

DOI: 10.15439/2023F5070
ISSN 2300-5963 ACSIS, Vol. 35

Investigating the Effect of Partial and Real-Time Feedback in INMAP Code-To-Architecture Mapping

Zipani Tom Sinkala, Sebastian Herold
0000-0002-7288-5552
0000-0002-3180-9182
Department of Mathematics and Computer Science, Karlstad University, Karlstad, Sweden
Email: {tom.sinkala, sebastian.herold}@kau.se

***Abstract*—InMap is an interactive and iterative information retrieval-based automated mapping algorithm that produces *code-to-architecture* mapping recommendations. In its original form, InMap requires an architect to provide feedback for each *code-to-architecture* mapping recommendation in a given set produced (*complete feedback*). However, architects may delay/ defer deciding on some of the mapping recommendations provided. This leads us to ask, how would InMap perform if only a subset of the recommendations provided *(partial feedback)* or only a single recommendation *(real-time feedback)* is reviewed by the architect? Through carefully designed mapping experiments, we show that an architect giving *partial* or *real-time feedback* does not harm the recall and precision of the recommendations produced by InMap. On the contrary, we observed from the results of the systems tested a net increase of 2-5% (depending on the approach). This shows that in addition to InMap's original *complete feedback* approach, the two new approaches of collecting feedback presented in this paper, i.e. *partial* and *real-time,* create flexibility in how software architecture consistency checking tool developers may choose to collect mapping feedback and how architects may opt-to provide feedback, with no harm to the recall and precision of the results.**

***Index Terms*—software architecture conformance, automated source code mapping, software architecture consistency, software maintenance.**

I. Introduction

SOFTWARE engineering industry practitioners use *Software Architecture Consistency Checking (SACC)* to check the consistency of a software's implementation with its architectural design [1, 8, 9, 17]. Its importance is derived from the fact that if a software system's implementation no longer conforms to its originally designed architecture, this may lead to failure in fulling its architectural design goals or in meeting intended software quality attributes such as reliability, availability, performance or security [1, 7, 15, 17].

Reflexion Modelling is an effective SACC technique as far as industry acceptance and tool support are concerned [6, 10, 12, 14]. It represents software architecture as a decomposition of a software system into *sub-components/architectural modules* and the *relationships/dependencies* allowed among them [10]. It maps the existing codebase onto the defined architectural modules revealing any irregular dependencies amongst the modules. If no irregular dependencies exist, the system's codebase is said to conform to its architecture. However, if dependencies not prescribed in the architecture exist, then the software's codebase is said to have diverged from its intended architecture – a phenomenon known as *architecture drift* or *architecture degradation* [10, 14, 16].

Mapping code to architecture in Reflexion Modelling is a tedious manual process, especially for large systems [1]. Nevertheless, techniques exist that attempt to 'automate' *code-to-architecture* mapping in SACC methods [3, 5, 12, 18]. Accomplishing this well is no trivial task, as ideally, mapping is done by a system expert knowledgeable about the system, its architecture and its codebase [10]. However, these techniques attempt to correctly predict which architectural module a *portion of the code* would be mapped to. They use various approaches to try and tackle this problem – from code dependency and clustering [5, 11] to machine learning, information retrieval and natural language processing techniques [3, 12, 19, 21, 22].

Furthermore, mapping techniques also differ in how an architect is involved in *verifying* the correctness of the resulting mapping. Some techniques only involve the architect after a complete mapping of the codebase, meaning the architect is given a complete mapping and then can decide if it is correct [3, 11, 12]. Other mapping techniques do it progressively or *interactively* in a *human-in-the-loop* approach. This means that as the mapping takes place, the architect is asked to review the mapping suggestions provided to improve them [19, 20] or resolve the cases that are difficult to map automatically [5].

Interactive mapping techniques have a similar approach to obtaining feedback from an architect as mapping progresses. They require an architect to decide on the correctness of every member of a set of mapping recommendations given. However, it is common to have cases where an architect may need more knowledge of a system [3, 5, 19, 21, 22] and may therefore prefer to *delay/defer* deciding on some mapping recommendations provided. In other words, given a set of 10 mapping recommendations, it is expected to have the architect decide about all 10 before progressing with the mapping. However, the limitation of this approach is that it does not accommodate a situation where an architect may feel that they are only able to make a decision about some of the recommendations produced (not all) or a case where the architect would want to defer deciding on a given recommendation to a later point in the mapping exercise. We thus investigate the effect, in terms of the impact on the recall and precision of the results, of providing *partial feedback* in a code-to-architecture mapping technique we developed in prior studies called InMap [18, 19, 20, 21, 22]. Additionally, we are interested in investigating InMap's behaviour if we update the list of recommendations provided in *real-time*; that is, the cases where a new set of mapping recommendations are provided each time an architect gives feedback for a singular recommendation.

Our contribution through this paper is evidence that an architect giving *partial* or *incomplete feedback* for the mapping recommendations produced by InMap does not harm the recall and precision of the technique's recommendations. InMap, in this modified form, retained recall values similar to its original *complete-feedback* form for all six systems under test. Additionally, we found that the precision of the *partial-feedback* approach is, on average, +/-2% compared to the *complete-feedback* approach. We also show that updating InMap's mapping recommendations in *real-time*, as an architect provides feedback, gives a net increase of 5% for the systems tested. This entails that in addition to InMap's original complete feedback approach, the two new approaches of collecting feedback introduced in this paper, i.e. partial batch and real-time, provide flexibility in how SACC tool developers may choose to collect mapping feedback from an architect when using *human-in-the-loop* mapping techniques. We also show that our InMap mapping technique offers flexibility in how architects may opt to provide feedback with no harm to the recall and precision of the results.

Section II highlights related works. Section III gives an overview of InMap. In Sec IV, we discuss our new approaches to collecting feedback in-depth. Section V explains our experimental setup. The results of the experiments are presented in Section VI and discussed in Section VII. The paper is concluded in Section VIII.

II. Related Work

A few techniques exist that attempt to (auto)-map source code to software architecture in SACC methods like Reflexion Modelling. We broadly classify these into two categories with regard to how they collect feedback from an architect. We classify, as *collaborative mapping* or *just-in-time feedback*, techniques involving the architect as the mapping occurs. These techniques are incremental and involve a *human-in-the-loop*. They get feedback from the architect as the mapping progresses on the premise that the architect's knowledge of the software system can help 'steer' the mapping in the right direction [19, 20]. We classify, as *non-collaborative mapping* or *feedback after-the-fact*, those techniques that attempt to entirely automate the mapping process without involving the architect during the mapping process. An architect only reviews the final results of these techniques after they complete their mapping process, implying the architect is not directly involved in the mapping [7].

A. Collaborative Mapping / Just-In-Time Feedback

Christl et al. propose **HuGME,** a mapping recommendation technique that analyses source code elements' dependencies. HuGME clusters source code elements using an architect's knowledge about its intended architecture [4, 5]. A *dependency-based attraction function*, which minimises coupling and maximises cohesion, is used, which yields a matrix of attraction scores for unmapped entities [23]. All unmapped entities that result in only one candidate having a score higher than the mean of all scores result in a sole recommendation. All unmapped entities with two or more mapping candidates are presented to the architect in descending order as recommendations. HuGME presents the recommendations to the architect to let cluster decisions be made entirely by the architect. HuGME does not attempt to map all source code entities in one complete step; instead, it maps a subset at a time, getting feedback from the architect until no more mapping is possible. This classifies it as a *collaborative mapping technique*. HuGME needs about 20% of a system's codebase to be manually *pre-mapped* by an architect before proceeding with automated mapping and thus suffers from pre-mapping drawbacks [18, 19].

Bittencourt et al. have a mapping recommendation technique that uses *information retrieval (IR).* It has a similar automated mapping approach to HuGME, except that they use *dependency-based attraction functions* with an *IR-based similarity function* [3]. They compute the similarity of an unmapped entity to an architectural module by searching for specified terms within the source of an unmapped entity. They search for an architectural module's name and the names of its mapped entities/classes, class methods and fields. Their technique requires manual pre-mapping before it can automate mapping; hence it suffers from pre-mapping drawbacks similar to HuGME [18, 19]. The results of Bittencourt et al.'s technique show that when there was a smaller pre-mapped code base, there was a decrease in the *f_1-score* of their technique [3].

Naim et al. propose a technique that uses *dependency analysis* and *information retrieval* methods, called ***Coordinated Clustering of Heterogeneous Datasets (CCHD),*** to compute a *similarity score* for source entities [11]. CCHD profits from an architect's feedback on a recovered architecture to iteratively adjust the results until there are no more recommendations for change. These modified results train a classifier that automatically places new code in the "right" architectural module. However, the technique is not necessarily meant for automated mapping in *software architecture consistency checking,* but rather, it was designed for *software architecture recovery* tasks.

None of the above-discussed collaborative mapping approaches directly addresses the different ways an architect can provide feedback as the mapping progresses, namely *complete, partial or real-time.* They all make use of a complete-batch feedback approach for the recommendations they provide.

B. Non-Collaborative Mapping / Feedback After-the-Fact

Olsson et al. use *information retrieval* and *dependency analysis* in their automated mapping technique called **Naive Bayes Classification (NBC)** [12]. They use Bayes' theorem to build a probabilistic model of classifications using words taken from the source entities of a software system. The model provides the probability of words (or tokens) being part of a source entity. This is then enriched with syntactical information on a source entity's incoming and outgoing dependencies, a method called *Concrete Dependency Abstraction* [12]. NBC needs a pre-mapped set to return fruitful results and inadvertently suffers from the downsides that come with the need to pre-map. In recent studies, Olsson et al. refined their technique to create a pre-mapped set using an approach similar to InMap [13]. However, the technique uses a *feedback after-the-fact* approach, implying the architect checks whether the mapping is correct at the end of the process.

C. Manual Methods Supported by Tools

Naming patterns (or *regular expressions*) are commonly used in SACC industry tools. Expressions such as ***/cli/*** or **.cli.* or net.commons.cli.** can be used to map source entities (whether classes or packages) to an architecture module named *CLI.* This approach is used in popular SACC tools such as **Sonargraph Architect** and **Structure101 Studio.** These tools also provide *drag & drop* functionality. However, the limitation of using naming patterns or drag & drop functionality is that they do not solve the problem of reducing the monotony of the mapping process because they are both manual tasks. In large systems with complex mapping configurations, this is a demanding task.

In summary, there exist techniques that attempt to get feedback as the mapping progresses, like HuGME [4, 5] and InMap [19, 21, 22] and others that get feedback from the architect once the mapping task is complete, like NBC [12, 13]. Of those that collect feedback as the mapping progresses, none directly address the alternative ways architects may provide feedback, for example, as a *complete batch,* as a *partial batch* or *as a single recommendation* at a time.

III. InMap Interactive Mapping

In [18, 19], we propose a collaborative mapping technique known as InMap. Using natural language descriptions of a system's architecture modules, InMap can automate mapping a completely unmapped system with no loss in the recall and precision of the recommendations produced [19]. It iteratively and interactively provides mapping recommendations that an architect can review, a batch/set at a time. It uses the architect's feedback in one iteration to guide the recommendations provided in a subsequent iteration. This process continues until the entire system is mapped or no more recommendations can be produced. In our prior studies, InMap automated the mapping of completely unmapped systems with an average recall and precision of 0.97 and 0.82, respectively, for the systems tested. However, InMap does not cater for the fact that an architect may not have full knowledge of the set of recommendations provided and may instead opt to provide partial feedback on the recommendations given. Instead, it assumes or requires that the architect provides feedback for all recommendations produced in a given iteration/batch before a new set of recommendations can be provided in the following iteration.

A. The Algorithm

The InMap mapping technique is comprised of seven steps [19, 21] summarised as follows:

1. A software system's source files are filtered to omit third-party package libraries or system packages/classes that the architect does not want to be part of the mapping exercise.

2. The contents of the filtered source files are stripped of any special characters and programming language-specific keywords.

3. The pre-processed source files are indexed as an inverted index.

4. InMap constructs a query for each architectural module.

5. InMap uses the queries from *Step 4* to search the indexed source files for the similarity of every unmapped class to each architectural module. The query for each architectural module is a combination of (i) its name; (ii) its natural language architectural description; (iii) the names of classes mapped to the module; and (iv) the names of methods contained within classes mapped to the module. In InMap's first iteration, when there are no mapped classes, it uses only information from items (i) and (ii) to construct the query. However, once the first set of classes is mapped, InMap adds items (iii) and (iv) to the query. These last two items 'enrich' the query, as it were, to search for the similarity of any unmapped class to the architectural module in question. Consequently, after each iteration of newly mapped classes, the query to produce the next set of recommendations is different. The queries are used to search the index, resulting in a set of scores for every *class-module pair.* The scores are based on the similarity information retrieval function, *tf-idf.* The *tf-idf* scores

are called *class-to-module similarity scores (SS_{cm})*, where *c* and *m* are a class-module pair in the system under review. *Step 5* results in a matrix of *class-to-module similarity scores (SS_{cm})* for every class against every module. We extended InMap to include hierarchical information contained in a system's codebase, i.e., packages [20, 21], to condense the number of recommendations made to complete the mapping process in SACC techniques. This version of the technique in step five produces a matrix of *package-to-module similarity scores SS_{pm}* derived from *class-to-module similarity scores (SS_{cm})*.

6. InMap uses the matrix of *entity-to-module* scores to produce an ordered list of the best-scoring entities for the given architectural modules in terms of similarity. In this case, an entity is either a class or a package. InMap also uses page size to trim the ordered list to the most likely correct recommendations.

7. The architect reviews the recommendations produced, giving feedback by accepting or rejecting them. After this step, InMap returns to *Step 4* and iterates *Step 4* through *7* until no more recommendations can be produced.

At the time of our study, InMap existed in two versions, a class-based version and a package-based (or hierarchical) version. More detailed descriptions of both mapping algorithms and how their similarity score calculations are derived can be found in [19] and [21], respectively.

B. Research Questions

For our study, we hypothesise that there is more than one way a software architect may choose to provide feedback about code-to-architecture mapping recommendations produced by interactive techniques like InMap. They may do it as a *complete batch* which is how InMap works in its original form [19, 21, 22], but they may also do it as a *partial batch.* For example, if presented with a page of recommendations, the architect might be unsure about a few of them and would opt to postpone making a decision about them. This leads to asking,

> ***RQ1****: What is the effect, in terms of recall and precision, of an architect giving partial batch-feedback in InMap compared to complete batch-feedback?*

Another interesting scenario to investigate is the implication of InMap collecting and updating its list of recommendations in real-time. Rather than waiting for an architect to give feedback for all recommendations provided on a page before presenting a new set of recommendations (what we would describe as an *interactive batch update process*), what would be the behaviour of InMap, in terms of recall and precision if it updated its list of recommendations immediately an architect gives feedback on an individual recommendation (what we would describe as an *interactive real-time update process*)? In other words,

> ***RQ2****: What is the effect, in InMap in terms of recall and precision, to consider feedback from the architect as soon as we receive it (real-time feedback), and how does it compare with batch feedback?*

IV. Method

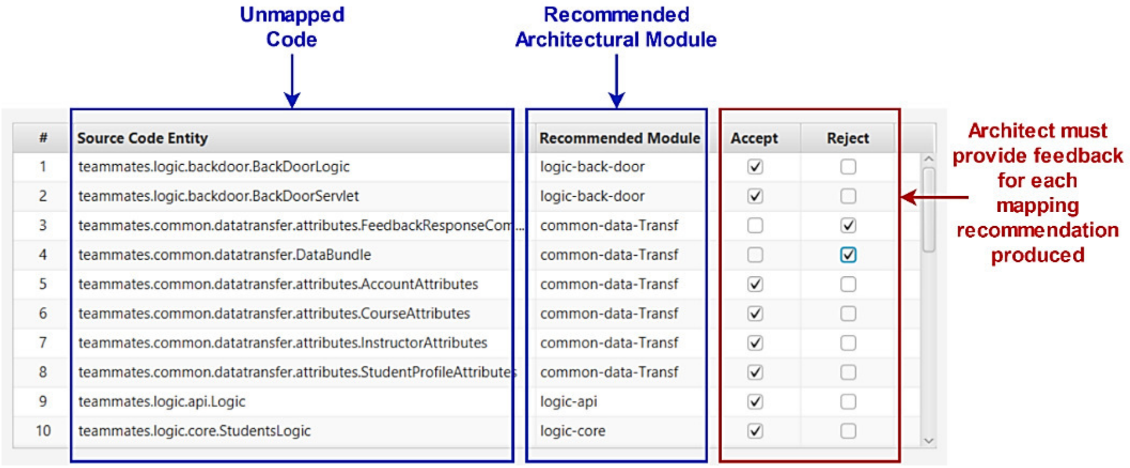
To properly investigate our research questions, we describe and formally define all three highlighted approaches to collecting feedback from an architect, the prior existing *complete-batch feedback* approach, as well as the two new approaches introduced in this study, namely *partial-batch feedback* and *real-time feedback*. We also illustrate how InMap was modified to accommodate the two new feedback approaches.

A. Complete-Batch Feedback

Complete-batch feedback is used by most interactive mapping techniques [3, 5, 19, 20]. The algorithm gets complete feedback from the architect for all mapping recommendations on a page – see *Fig. 1* for an illustration. The mapping algorithm only generates a new set/page of mapping recommendations in a subsequent iteration once feedback is given for all entities in the given set of the current iteration. This implies:

# of **required recommendations in feedback** = # of recommendations on page

How Complete-Batch Mapping Recommendations in InMap Works: Recall that in step *six*, InMap derives the highest scoring class-to-module pairs, from the class-to-module similarity scores (SS_{cm}) matrix, or package-to-module pairs, from the package-to-module similarity scores (SS_{pm}) and gives them as class/package-to-module mapping recommendations. InMap presents, as recommendations, either the class/package-module pairs above the arithmetic mean of the highest similarity scores obtained for a pair; or the best 30 recommendations (if those above the mean are greater than 30). Thirty gave the most optimal results based on the systems tested. In step *seven*, this final filtered list is presented to the architect to review the recommendations given. An architect gives feedback on the page (batch of 30) recommendations produced. In its original form, the architect's feedback provided to InMap must be complete. The architect is expected to provide feedback (an accept/reject) for every recommendation listed on the page.



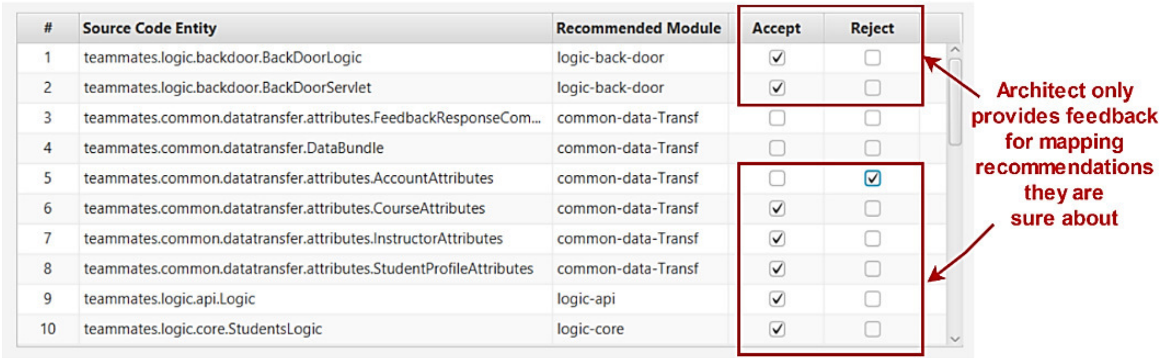
Unmapped Code

Recommended Architectural Module

#	Source Code Entity	Recommended Module	Accept	Reject
1	teammates.logic.backdoor.BackDoorLogic	logic-back-door	☑	□
2	teammates.logic.backdoor.BackDoorServlet	logic-back-door	☑	□
3	teammates.common.datatransfer.attributes.FeedbackResponseCom...	common-data-Transf	□	☑
4	teammates.common.datatransfer.DataBundle	common-data-Transf	□	☑
5	teammates.common.datatransfer.attributes.AccountAttributes	common-data-Transf	☑	□
6	teammates.common.datatransfer.attributes.CourseAttributes	common-data-Transf	☑	□
7	teammates.common.datatransfer.attributes.InstructorAttributes	common-data-Transf	☑	□
8	teammates.common.datatransfer.attributes.StudentProfileAttributes	common-data-Transf	☑	□
9	teammates.logic.api.Logic	logic-api	☑	□
10	teammates.logic.core.StudentsLogic	logic-core	☑	□

Architect must provide feedback for each mapping recommendation produced

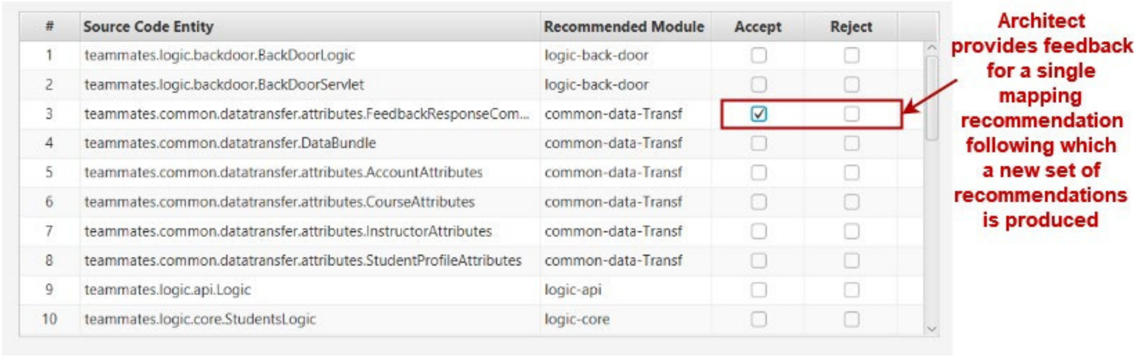
Fig 1. Illustration of *complete-batch* feedback. For this approach of collecting feedback, an architect must provide feedback for each mapping recommendation produced.



#	Source Code Entity	Recommended Module	Accept	Reject
1	teammates.logic.backdoor.BackDoorLogic	logic-back-door	☑	□
2	teammates.logic.backdoor.BackDoorServlet	logic-back-door	☑	□
3	teammates.common.datatransfer.attributes.FeedbackResponseCom...	common-data-Transf	□	□
4	teammates.common.datatransfer.DataBundle	common-data-Transf	□	□
5	teammates.common.datatransfer.attributes.AccountAttributes	common-data-Transf	□	☑
6	teammates.common.datatransfer.attributes.CourseAttributes	common-data-Transf	☑	□
7	teammates.common.datatransfer.attributes.InstructorAttributes	common-data-Transf	☑	□
8	teammates.common.datatransfer.attributes.StudentProfileAttributes	common-data-Transf	☑	□
9	teammates.logic.api.Logic	logic-api	☑	□
10	teammates.logic.core.StudentsLogic	logic-core	☑	□

Architect only provides feedback for mapping recommendations they are sure about

Fig 2. Illustration of *partial-batch* feedback. For this approach of collecting feedback, an architect can choose which mapping recommendations they want to provide feedback for and can opt to delay or defer deciding on some.



#	Source Code Entity	Recommended Module	Accept	Reject
1	teammates.logic.backdoor.BackDoorLogic	logic-back-door	□	□
2	teammates.logic.backdoor.BackDoorServlet	logic-back-door	□	□
3	teammates.common.datatransfer.attributes.FeedbackResponseCom...	common-data-Transf	☑	□
4	teammates.common.datatransfer.DataBundle	common-data-Transf	□	□
5	teammates.common.datatransfer.attributes.AccountAttributes	common-data-Transf	□	□
6	teammates.common.datatransfer.attributes.CourseAttributes	common-data-Transf	□	□
7	teammates.common.datatransfer.attributes.InstructorAttributes	common-data-Transf	□	□
8	teammates.common.datatransfer.attributes.StudentProfileAttributes	common-data-Transf	□	□
9	teammates.logic.api.Logic	logic-api	□	□
10	teammates.logic.core.StudentsLogic	logic-core	□	□

Architect provides feedback for a single mapping recommendation following which a new set of recommendations is produced

Fig 3. Illustration of *real-time* feedback. For this approach of collecting feedback, an architect provides feedback for a single mapping recommendation, following which a new set of recommendations is instantly produced and presented.

B. Partial-Batch Feedback

Partial-batch feedback is an alternative approach to collecting feedback about code-to-architecture mapping recommendations. In this form, an architect provides feedback for a subset of the recommendations provided on a given page. This could be because the architect needs to gain sufficient knowledge about some of the code entities (classes/packages) listed in the recommendations provided and would prefer to delay/defer decisions about those entities. The mapping algorithm would use the partial feedback to provide a new set/page of mapping recommendations that may or may not include the entities the architect did not decide on in prior iterations. *Fig. 2* illustrates this. This implies:

1 < # of **required recommendations in feedback** < # of recommendations on page

How Partial-Batch Mapping Recommendations in InMap Works: In step *seven* of InMap's algorithm, an architect is presented with 30 mapping recommendations to review. The architect does not have to give feedback for all 30 recommendations produced and may opt to skip some, essentially delaying or deferring a decision about them. We say delay or defer as these recommendations could reappear, given that they were not outrightly rejected. However, it is also possible that they may not reappear, given that InMap uses the architect's feedback provided to decide the next set of mapping recommendations to produce.

C. Real-Time Feedback

Real-time feedback is another alternative approach to collecting feedback from the software architect about code-to-architecture mapping recommendations. In this form, the mapping algorithm would produce a new set/page of recommendations immediately after an architect provides feedback on any of the recommendations on the page. *Fig. 3* illustrates this. This implies:

# of **required recommendations in feedback** = 1

How Real-Time Mapping Recommendations in InMap Works: In this case, despite InMap providing 30 mapping recommendations, the list of recommendations provided gets updated immediately after the architect provides feedback on a single code entity, i.e. in real-time. Note that it would be ideal to implement this in a way that the position in the ordered set did not matter, implying an architect can give feedback on any individual code entity from anywhere in the list, and the mapping algorithm refreshes the recommendations instantaneously. However, it is important to acknowledge that recommendation results are always presented with the best candidate at the top or beginning of the list and the worst at the bottom or end of the list. Therefore, in both the *partial-batch* and *real-time* feedback approaches, we consider the rank of the recommendation.

V. Evaluation

A. Experimentation

To test the effect of our two proposed alternatives to collecting feedback, we ran experiments on InMap in its original form, i.e. *complete-batch feedback* as our control. We then ran the same set of tests on our *partial-batch feedback* approach and our *real-time feedback* approach. We extended the InMap evaluator tool developed in prior studies of InMap [18, 19, 20, 21, 22] to accommodate the evaluation of our two proposed feedback approaches. The evaluation tool simulates a "human architect" accepting and rejecting the recommendations produced. It uses the oracle class/package mappings provided by knowledge experts of each system, as reported in prior studies of InMap [19, 21, 22]. We used the optimal parameter settings for both InMap's class-based version [18, 19] and InMap's hierarchical package-based version [20, 21], observing what effect both *partial-batch* and *real-time feedback* have on both the class-based and package-based versions of InMap. A batch size of 30 was used for the control experiment, i.e. the *complete-batch feedback*. However, for the *partial-batch feedback,* we tested a range of batch sizes, in addition to a batch size of 30, to observe if different batch sizes affect the results.

For every experiment, we collected the *recall*, *precision* and *f_1-scores* (as a harmonic mean between recall and precision) of the recommendations produced. InMap produces mapping recommendations in descending order of the most likely correct recommendation based on similarity scores. We take it as a norm that as an architect reviews a list of recommendations provided, they start reviewing the list and making decisions in sequential order from the beginning/top to the end/bottom instead of reviewing it randomly. This approach is similar to how most other recommendation or information retrieval-based systems are designed. They surmise that the best candidate in a list of results is found at the beginning/top of the list and the worst candidate at the end/bottom of the list. Therefore, in reviewing both *partial-batch* and *real-time feedback,* we consider the rank of a recommendation.

B. Systems Under Test

We evaluated our modified versions of InMap (*partial-batch* and *real-time feedback*) against InMap in its original form (*complete-batch feedback*) using six Java-based open-source systems used in prior InMap studies. These are *Ant*, a command line and API tool for automating processes; *ArgoUML*, a desktop application for modelling in UML; *JabRef,* a desktop application for managing bibliographic references; *Jittac,* an Eclipse IDE plugin for applying reflexion modelling; *ProM,* a desktop application for mining processes; and *TeamMates* a web application for peer reviews and feedback. These systems all have varying characteristics in terms of the number of lines of code, number of architectural modules, length of architectural descriptions, number of source files, number of classes and number of packages, to name a few. Our prior studies documented their characteristics [18, 19, 20, 21, 22].

C. Replication Package

A replication package of the evaluation tool; the six case systems tested along with their independently produced ground-truth mappings provided by experts knowledgeable about the respective systems; and the complete, partial and real-time feedback mapping approaches and results are all available in the online open-repository at the following link https://doi.org/10.6084/m9.figshare.13714150.

VI. Results

Tables 1, 2 and *3* show the results obtained for both InMap's class-based and package-based algorithms when run using the six test-case systems. They show the recall, precision and f_1-score for each version of the algorithm checked against each feedback approach. Green denotes an increase, whereas red denotes a decrease.

Table 1 gives the results obtained for the control experiment, i.e. the *complete-batch feedback* approach, which is how InMap was initially designed to collect feedback from an architect. The average recall for the six systems tested on the class-based version of InMap was 0.972; the average precision was 0.788, and the average f_1-score of 0.870. The package-based version had an average recall of 0.865, average precision of 0.745 and an average f_1-score of 0.800.

Tables 2 and *3* show the results of the two new approaches, i.e*., partial-batch* and *real-time feedback.* The results show that as far as the recall is concerned, these two ways of collecting feedback seem not to affect the recall – both approaches maintained an average of 0.972 for the class-based version and 0.865 for the package-based version of InMap.

With regard to the precision, we see some variances, albeit minor. For the class-based version of InMap, ProM's precision improved in both approaches, with the most significant result coming from the *real-time feedback* approach (3% increase). However, whereas *partial-based feedback* was the same for Ant compared to *complete-batch,* it surprisingly reduced by 1% for the real-time-based approach. Jittac improved in both approaches, with *real-time feedback* recording a higher increase in precision (5%) between the two approaches. ProM also increased precision for the package-based version using the *real-time feedback* approach. Ant again recorded a slight reduction in this case. It was interesting to observe that InMap's package-based version had more movement in precision compared to the class-based version.

The average f_1-scores of all three techniques show that *partial feedback* did not negatively affect the results compared to *complete feedback.* Furthermore, the f_1-scores show that *real-time feedback* gave the best results for all three approaches to collecting feedback.

VII. Discussion

A. Findings

The results of our investigation show that if we update the list of recommendations provided in real-time, the precision of the recommendations improves on average. This is likely because InMap uses the feedback given by an architect in deciding the next set of mapping recommendations to give. In other words, it benefits from getting information early in the mapping process. When we have a batch size of, say, 30 for both the batch processes, that is, *complete* and *partial*, the architect gives feedback for a minimum of 2 and a maximum equal to the batch/page size, in this case, 30. Now, if we consider that each class/package contains a unit of information that InMap could use to make a more accurate mapping recommendation, then in the batch process, we are delaying the feedback loop. The larger our batch/page size, the more significant the delay in relaying what could otherwise be helpful information in predicting the unmapped source entities. In other words, the results show that even though InMap uses information from the architect's feedback to work out the most suitable recommendations, it does not necessarily benefit from having lots of information given to it at once, say feedback on 100 classes in one go. Instead, having smaller units of information fed into the mapping loop early on is more beneficial. Thus, SACC tool developers and architects are more likely to benefit from using a *real-time feedback* approach. However, although *real-time feedback* offers the best results, the difference between both batch processes was shown to be minor. Therefore, if a tool developer or the users of the tools prefer not to work in real-time but give feedback a batch at a time, that works reasonably well too.

The results also show that for the batch feedback mapping approaches (i.e. *complete feedback* and *partial feedback*), on average, if an architect opts to delay decisions about some of the recommendations provided in a list, the recall and precision of the recommendations InMap provides are not negatively affected. Meaning given a batch size of 30, whether an architect chooses to provide feedback on all 30, i.e. *complete feedback*, or whether an architect decides to provide feedback for at minimum 2 or out of the 30 while delaying the rest, maybe because the architect is unsure and would like to see more/other recommendations first, this would not affect the results negatively. On the contrary, the results showed a slight improvement. This could be attributed to the same reasons that *real-time feedback* showed the best results. When an architect gives *partial feedback*, this is a smaller chunk of information than the batch size. Moreover, since InMap has shown that it benefits from receiving feedback as early as possible, giving feedback, for example, for 8 out of 30 recommendations, provides a smaller chunk of information earlier in the feedback loop than providing recommendations for all 30 at once. However, again, in this case, it does not imply that if a tool developer or architect opts for a *complete batch feedback* approach, then the accuracy of the recommendations will reduce drastically. On the contrary, the results showed *partial-batch feedback* recorded a slight increase over *complete-batch feedback*, which already had some reasonably good results.

We must note that there is a limit to the complete-batch size that can be or should be used. Firstly, if we set the batch size to 50 or 100, it is not practical to make an architect provide complete feedback for such a large quantity before providing

TABLE I.
RESULTS OF *COMPLETE-BATCH* FEEDBACK

System	Class Recall	Class Precision	Class F_1 Score	Package Recall	Package Precision	Package F_1 Score
AT	1.00	0.70	0.82	0.77	0.89	0.82
AU	0.99	0.75	0.85	0.78	0.56	0.65
JR	0.99	0.95	0.97	0.95	0.87	0.91
JT	0.99	0.80	0.88	0.82	0.58	0.68
PM	0.98	0.58	0.73	0.87	0.65	0.74
TM	0.88	0.95	0.91	1.00	0.92	0.96
Avg	**0.972**	**0.788**	**0.870**	**0.865**	**0.745**	**0.800**

TABLE II.
RESULTS OF *PARTIAL-BATCH* FEEDBACK

System	Class Recall	Class Precision	Class F_1 Score	Package Recall	Package Precision	Package F_1 Score
AT	1.00	0.70	0.82	0.77	0.89	0.82
AU	0.99	0.75	0.85	0.78	0.56	0.65
JR	0.99	0.95	0.97	0.95	0.87	0.91
JT	0.99	0.80	0.88	0.82	**0.61**	**0.70**
PM	0.98	**0.59**	**0.74**	0.87	0.65	0.74
TM	0.88	0.95	0.91	1.00	0.92	0.96
Avg	***0.972***	***0.790***	***0.872***	***0.865***	***0.750***	***0.803***

TABLE III.
RESULTS OF *REAL-TIME* FEEDBACK

System	Class Recall	Class Precision	Class F_1 Score	Package Recall	Package Precision	Package F_1 Score
AT	1.00	**0.69**	0.82	0.77	**0.87**	0.82
AU	0.99	0.75	0.85	0.78	0.56	0.65
JR	0.99	0.95	0.97	0.95	**0.86**	0.90
JT	0.99	0.80	0.88	0.82	**0.63**	0.71
PM	0.98	**0.61**	**0.75**	0.87	**0.68**	0.72
TM	0.88	0.95	0.91	1.00	0.92	0.96
Avg	***0.972***	***0.792***	***0.873***	***0.865***	***0.753***	***0.805***

a new set of recommendations of similar size. It is tedious to do so for such a large number at a time and does not help reduce the effort required by an architect, which is a core motivation for automating the mapping processes. Secondly, this study and prior studies [18, 19] have shown that smaller units of information fed back into the algorithm at a time improve the results. So whereas the results might be similar for a batch size of 20, 30 or 40, the same cannot be said for sizes of 50, 100 or 150. Generally speaking, a batch size of 30 was shown to produce the best average results in our previous studies, and the larger the batch size is beyond 30, the less our precision becomes, on average.

Both findings on *real-time* and *partial-batch feedback* show that InMap allows flexibility in how a SACC tool collects mapping recommendation feedback because all three approaches have, on average, f_1-scores within 0.005 of each other. Furthermore, if the developer of a SACC tool decides to implement all three approaches, then this flexibility extends to the architect. They can choose their preferred way of providing feedback, i.e. *complete-batch*, *partial-batch* or *real-time.* Furthermore, they can alternate among these approaches throughout the mapping process without committing to a singular approach.

Interestingly, the package-based version of InMap showed better improvement in the average of the f_1-score for both *partial* and *real-time feedback* over the class-based version, 0.005 versus 0.003, respectively. However, the class-based version of the InMap achieved higher f_1-scores for all approaches compared to the package-based version, 0.87 vs 0.80; therefore, there was less opportunity for improvement for the class-based version compared to the package-based version because it already had reasonably high-scoring results. That said, an approach combining both the class-based and package-based versions of InMap, such as the one introduced in [22], would likely derive improvements from both versions.

B. Limitations & Validity

The two new feedback approaches introduced build on top of InMap's *class* and *package similarity* functions; therefore, the same factors that affect the external validity of InMap's results are inherited by the alternative feedback approaches presented in this paper. That is to say, aspects such as the code commenting quality and style; the number of classes, packages and modules; and the length and quality of the architecture description could likely affect the external validity of our results. Therefore, more case systems with variable attributes would add to the soundness of the results. Nonetheless, the results of the six systems tested and their varying characteristics provide a fair case for the two feedback approaches investigated.

VIII. Conclusion & Future Work

This paper presents two alternative approaches to how an architect can provide feedback on mapping recommendations provided, either as *partial-batch* or *real-time feedback*. They are an alternative to InMap's *complete-batch feedback* approach. Our *partial-batch feedback* approach showed a net increase of 2% in precision for the systems tested, and our *real-time feedback* approach showed a net increase of 5% in precision for the systems tested. This shows that providing *partial–batch feedback* does not harm the precision of the recommendation produced compared to when *complete-batch feedback* is provided. Furthermore, it shows that providing feedback in *real-time* improves the recommendations produced. Moreover, because the results for *complete-batch feedback* were already reasonably good, the 2-5% increase that these two new approaches provide allows for flexibility in how SACC tools that use *human-in-the-loop* approaches can collect feedback; or flexibility in how architects themselves opt to provide feedback. This implies that tools that use InMap's automated interactive mapping algorithm have some flexibility in how they choose to gather feedback from an architect as mapping progresses (*complete* vs *partial* and *batch* vs *real-time*) without suffering a loss in recall and with an insignificant difference in precision. It also offers flexibility for an architect, assuming a SACC tool implements the different ways mapping recommendation feedback can be collected.

In future work, we would like to do more detailed studies with more systems to further test the soundness of the conclusion of this study. We would also like to see how the two new feedback approaches introduced in this work would fair on the version of InMap that integrates both the class-based and package-based versions of the algorithm into one [22]. Lastly, we would like to carry out an exploratory study that examines the cases that are difficult for InMap to map with either mapping version of InMap or either approach to collecting feedback.

References

[1] N. Ali et al, "Architecture Consistency: State of the Practice, Challenges and Requirements," in *Empirical Software Engineering*, 23(1), 2018, pp. 224–258, https://doi.org/10.1007/s10664-017-9542-0

[2] M. Bauer, M. Trifu, "Architecture-Aware Adaptive Clustering of OO Systems," Proceedings – 8th European Conference on Software Maintenance and Reengineering, 2004, pp. 3–14, https://doi.org/10.1109/CSMR.2004.1281401

[3] R.A. Bittencourt et al, "Improving Automated Mapping in Reflexion Models Using Information Retrieval Techniques," *Proceedings – Working Conference on Reverse Engineering, WCRE*, 2010, pp. 63–172, http://dx.doi.org/10.1109/WCRE.2010.26

[4] A. Christl et al, "Automated Clustering to Support the Reflexion Method," in *Information and Software Technology*, 49(3), 2007, pp. 255–274, https://doi.org/10.1016/j.infsof.2006.10.015

[5] A. Christl et al, "Equipping the Reflexion Method with Automated Clustering," *12th Working Conference on Reverse Engineering*, 2005, https://doi.org/10.1109/WCRE.2005.17

[6] F.A. Fontana et al, "Tool Support for Evaluating Architectural Debt of an Existing System: An Experience Report," *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, 2016, pp. 1347–1349, http://dx.doi.org/10.1145/2851613.2851963

[7] N. Medvidovic, R.N. Taylor, "Software Architecture: Foundations, Theory, and Practice", *ACM/IEEE 32nd International Conference on Software Engineering*, 2010, pp. 471–472, https://doi.org/10.1145/1810295.1810435

[8] J. Knodel, "Sustainable Structures in Software Implementations by Live Compliance Checking," *Fraunhofer-Verl*, Stuttgart, 2011.

[9] J. Knodel, D. Popescu, "A Comparison of Static Architecture Compliance Checking Approaches," *Proceedings of the 6th Working IEEE/IFIP Conference on Software Architecture*, 2007, https://doi.org/10.1109/WICSA.2007.1

[10] G.C. Murphy et al, "Software Reflexion Models: Bridging the Gap between Source and High-Level Models," *IEEE Transactions on Software Engineering*, 27(4), 2001, pp. 364–380, https://doi.org/10.1109/32.917525

[11] S.M. Naim et al, "Reconstructing and Evolving Software Architectures Using a Coordinated Clustering Framework", in *Automated Software Engineering*, 24(3), 2017, pp. 543–572, https://doi.org/10.1007/s10515-017-0211-8

[12] T. Olsson et al, "Semi-Automatic Mapping of Source Code using Naive Bayes," *Proceedings of the 13th European Conference on Software*

Architecture ECSA, *Lecture Notes in Computer Science*, 13365, 2022, pp. 65-85, https://doi.org/10.1007/978-3-031-15116-3_4

[13] L. Passos et al, “Static Architecture-Conformance Checking: An Illustrative Overview,” in *IEEE Software*, 2010, 27(5), pp. 82–89, https://doi.org/10.1109/MS.2009.117

[14] D.E. Perry, A.L. Wolf, “Foundations for the Study of Software Architecture,” in *SIGSOFT Softw. Eng. Notes*. 17, 4, 1992, pp. 40–5, https://doi.org/10.1145/141874.141884

[15] J. Rosik et al, “Assessing Architectural Drift in Commercial Software Development: A Case Study,” in *Software Practice and Experience*, 41, 2011, pp. 63–86, https://doi.org/10.1002/spe.999

[16] L. de Silva, D. Balasubramaniam, “Controlling Software Architecture Erosion: A Survey,” in *Journal of Systems and Software*, 85(1), 2012, pp. 132–151, https://doi.org/10.1016/j.jss.2011.07.036

[17] Z.T. Sinkala, S. Herold, “InMap: Automated Interactive Code-to-Architecture Mapping,” *Proceedings of the ACM Symposium on Applied Computing*, 2021, pp. 1439–1442, https://doi.org/10.1145/3412841. 3442124

[18] Z.T. Sinkala, S. Herold, “InMap: Automated Interactive Code-to-Architecture Mapping Recommendations,” *Proceedings – IEEE 18th International Conference on Software Architecture*, 2021, pp. 173–183, https://doi.org/10.1109/ICSA51549.2021.00024

[19] Z.T. Sinkala, S. Herold, “Towards Hierarchical Code-to-Architecture Mapping Using Information Retrieval,” *Companion Proceedings – IEEE 15th European Conference on Software Architecture,* 2021.

[20] Z.T. Sinkala, S. Herold, “Hierarchical Code-to-Architecture Mapping,” in *ECSA 2021 Tracks and Workshops – Revised Selected Papers*, 2022, https://doi.org/10.1007/978-3-031-15116-3_5

[21] Z.T. Sinkala, S. Herold, “An Integrated Approach to Package and Class Code-to-Architecture Mapping Using InMap,” *Proceedings – IEEE 20th International Conference on Software Architecture*, 2023, https://doi.org/10.1109/ICSA56044.2023.00023

[22] T.A. Wiggerts, “Using Clustering Algorithms in Legacy Systems Remodularization,” *Proceedings of the 4th Working Conference on Reverse Engineering*, 1997, pp. 33–43, https://doi.org/10.1109/WCRE.1997.624574

DOI: 10.15439/2023F2398
ISSN 2300-5963 ACSIS, Vol. 35

Hashtag Discernability – Competitiveness Study of Graph Spectral and Other Clustering Methods

Bartłomiej Starosta, Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, Dariusz Czerski
Institute of Computer Science, Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
Email: b.starosta,klopotek,stw,d.czerski@ipipan.waw.pl

***Abstract*—Spectral clustering methods are claimed to possess ability to represent clusters of diverse shapes, densities etc. They constitute an approximation to graph cuts of various types (plain cuts, normalized cuts, ratio cuts). They are applicable to unweighted and weighted similarity graphs. We perform an evaluation of these capabilities for clustering tasks of increasing complexity.**

I. Introduction

DOCUMENT clustering (or text clustering) has a multitude of applications, including topic extraction, fast information retrieval, filtering, authorship discovery, topic drift detection in news streams and social media, automatic document organization etc. ([1], [2], [3], [4])

Two clustering methods are of particular interest in this area, the Graph Spectral Clustering (GSC) and spherical k-means.

Graph Spectral Clustering methods [1] are generally praised for possessing ability to represent clusters of diverse shapes, densities etc. They constitute an approximation to graph cuts of various types (plain cuts, normalized cuts, ratio cuts). They are applicable to unweighted and weighted similarity graphs.

Spherical k-means algorithm [5] is a variant of k-means algorithm that measures similarity of documents based on their cosine similarity, that is quite popular in the domain of text analysis (e.g. for search engines).

In this paper we pose the question: If the grouping method correctly groups certain datasets, can we expect that a combination of these datasets will also be correctly clustered? We will examine the following problem in more detail. Assume that a clustering method can cluster correctly documents from categories $[A, B]$, $[B, C]$, and $[C, A]$. Can we expect the algorithm to cluster correctly data from the mixed set $[A, B, C]$? Let us illustrate this with three datasets, tweets, marked with (single) tags 'lolinginlove', 'tejran', 'anjisalvacion'.

We used standard Python implementation of spectral clustering from scikit-learn library.[1] The affinity matrix was constructed from a k-nearest neighbors connectivity matrix, with the default value of $k = 10$.

In one of the experiments the clustering illustrated in Fig. 1 was obtained for the hashtags 'lolinginlove', 'tejran'. For the hashtags 'tejran', 'anjisalvacion' the nearest neighbor spectral clustering achieves the best clustering agreement visible in Fig. 2. For the hashtags 'lolinginlove', 'anjisalvacion', the nearest neighbor spectral clustering achieves the clustering agreement visible in Fig. 3.

	T\P	0	1	
lolinginlove:	0	1258	0	1258
tejran:	1	8	337	345
		1266	337	1603

F-score: 0.990046

Fig. 1. Spectral clustering with affinity "nearest neighbors" example 1; row labels - "true" clusters, column labels - clustering result

	T\P	0	1	
tejran:	0	324	21	345
anjisalvacion:	1	5	727	732
		329	748	1077

F-score: 0.968385

Fig. 2. Spectral clustering with affinity "nearest neighbors" example 2; row labels - "true" clusters, column labels - clustering result

So, for each pair of the three hashtags we see a very good agreement of clusterings with the target (hashtags). If we look at the hashtags ['lolinginlove', 'tejran', 'anjisalvacion'], we get clustering agreement visible in Fig. 4. We see that more errors are committed here than for each pair of hashtags presented in Figs. 1, 2 and 3, though the increase does not seem to be large in absolute numbers. We will return to this issue in the next section.

Here and in further sections, F-score is computed as follows. We assume that the clustering is to predict the hashtag. The "true" hashtag is identified as the majority hashtag in the cluster. For a given hashtag H we proceed as follows. True positives (TP) are those cases when cluster membership agrees

	T\P	0	1	
lolinginlove:	0	1258	0	1258
anjisalvacion:	1	0	732	732
	1258	732	1990	

F-score: 1.000000

Fig. 3. Spectral clustering with affinity "nearest neighbors" example 3; row labels - "true" clusters, column labels - clustering result

[1] Consult https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html for details.

	T\P	0	1	2	
lolinginlove:	0	1258	0	0	1258
tejran:	1	7	314	24	345
anjisalvacion:	2	0	5	727	732
		1265	319	751	2335

F-score: 0.970334

Fig. 4. Spectral clustering with affinity ”nearest neighbors” example 4; row labels - ”true” clusters, column labels - clustering result

with this hashtag. False positives (FP) are the cases which belong to the cluster for which the hashtag H is the true hashtag, but the hashtag for the given document is different from H. True negatives (TN) are the cases which belong to the cluster for which the hashtag H is not the true hashtag, and the hashtag for the given document is different from H. False negatives (FN) are the cases which belong to the cluster for which the hashtag H is not the true hashtag, but the hashtag for the given document is the hashtag H. Computation of precision and recall follows the standard pattern and the F-score is computed for each hashtag separately, and then the average is taken as the F-score for the clustering.

In this paper we study the extent to which this behaviour extends to larger number of clusters. This study is a starting point for a future revision of the studied clustering algorithms.

II. Conceptual Considerations

Despite the example shown above, it is not entirely obvious that given a grouping method that allows to correctly group documents from the categories $[A, B]$, $[B, C]$, $[C, A]$, we can expect that the algorithm will correctly group data from the mixed set $[A, B, C]$.

If the sets $A \cup B$, $B \cup C$ and $C \cup A$ have block diagonal document similarity matrices (after proper reordering the documents), and the blocks are actually within A, B, C then in fact the $[A, B, C]$ similarity matrix will be block diagonal too so that GSC algorithm will cluster A, B, C correctly. This can be seen immediately by inspection of block matrix structure, i.e.

$$S_{A,B} = \begin{bmatrix} S_{A,A} & 0 \\ 0 & S_{B,B} \end{bmatrix} S_{B,C} = \begin{bmatrix} S_{B,B} & 0 \\ 0 & S_{C,C} \end{bmatrix}$$

$$S_{A,C} = \begin{bmatrix} S_{A,A} & 0 \\ 0 & S_{C,C} \end{bmatrix}$$

implies

$$S_{A,B,C} = \begin{bmatrix} S_{A,A} & 0 & 0 \\ 0 & S_{B,B} & 0 \\ 0 & 0 & S_{C,C} \end{bmatrix}$$

Recall that combinatorial Laplacian is computed as $L = D - S$, where S is the similarity matrix and D is the diagonal matrix with elements being sums of corresponding rows of S. Hence

$$L_{A,B} = \begin{bmatrix} L_{A,A} & 0 \\ 0 & L_{B,B} \end{bmatrix} \text{, etc.}$$

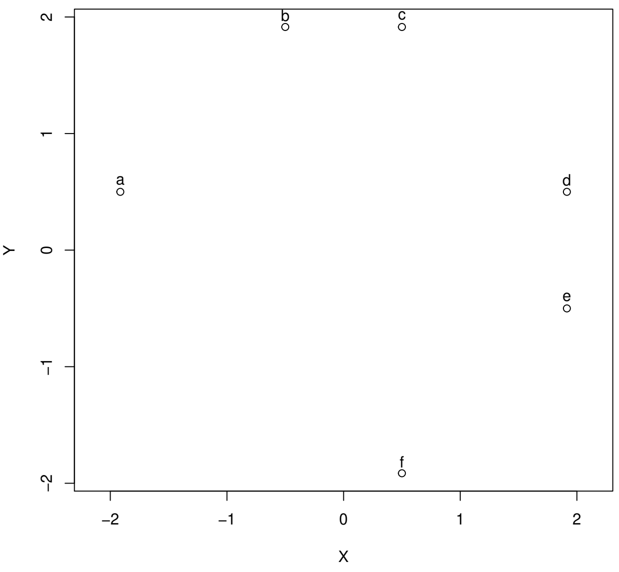


Fig. 5. Visualization of datapoints used to illustrate the increasing clustering problem for k-means

and

$$L_{A,B,C} = \begin{bmatrix} L_{A,A} & 0 & 0 \\ 0 & L_{B,B} & 0 \\ 0 & 0 & L_{C,C} \end{bmatrix}$$

Eigenvalues of $L_{A,B}, L_{B,C}, L_{A,C}$ will become eigenvalues of $L_{A,B,C}$ with corresponding eigenvectors being only extended with zeros appropriately. So theoretically it should be easy to separate the sets A, B, C based on eigenvectors of $L_{A,B,C}$. However, this enthusiasm needs to be mitigated because such a pure block structure rarely occurs, see our example Fig. 1, Fig. 2, Fig. 3, so the ”noise” is inherited in sets with more hashtags as visible in Fig. 4. But there are also further concerns. Spectral clustering is based on lowest eigenvalue eigenvectors of respective Laplacians. But as shown in [6], the two lowest eigenvectors of $L_{A,B}, L_{B,C}, L_{A,C}$ do not need to be lowest three eigenvectors of $L_{A,B,C}$. For higher number of clusters, the situation may be more complex.

If the dataset $A \cup B \cup C$ is well separated in the sense of k-means algorithm, so that a clustering with k-means will yield A, B, C as clusters, then its application to $A \cup B$, $B \cup C$ or $C \cup A$ will also return correct pairs of clusters. But this is not necessarily true for k-means in the reverse direction. Well-separatedness of $A \cup B$, $B \cup C$ and $C \cup A$ does not imply well-separatedness of $A \cup B \cup C$. Let us illustrate this point with a bit artificial example. Consider the datapoints $\mathbf{a} = (-(0.5+\sqrt{2}), 0.5)$, $\mathbf{b} = (-0.5, 0.5+\sqrt{2})$, $\mathbf{c} = (0.5, 0.5+\sqrt{2})$, $\mathbf{d} = (0.5+\sqrt{2}, 0.5)$, $\mathbf{e} = (0.5+\sqrt{2}, -0.5)$, $\mathbf{f} = (0.5, -(0.5+\sqrt{2}))$, see Fig. 5 for visualization. Consider ”hashtags” with their ”documents”: $A = \{\mathbf{a}, \mathbf{b}\}$, $B = \{\mathbf{c}, \mathbf{d}\}$, $C = \{\mathbf{e}, \mathbf{f}\}$. Clustering with k-means of $A \cup C$ into two clusters

TABLE I
TWT.10 DATA SET - HASHTAGS AND CARDINALITIES OF THE SET OF RELATED TWEETS USED IN THE EXPERIMENTS

No.	hashtag	count
0	90dayfiance	316
1	tejran	345
2	ukraine	352
3	tejasswiprakash	372
4	nowplaying	439
5	anjisalvacion	732
6	puredoctrinesofchrist	831
7	1	1105
8	lolinginlove	1258
9	bbnaija	1405

will yield A, C, similarly any two hashtag combinations. But clustering with k-means of $A \cup B \cup C$ will yield three clusters $\{\mathbf{a}\}, \{\mathbf{b}, \mathbf{c}\}, \{\mathbf{d}, \mathbf{e}, \mathbf{f}\}$, not A, C, E.

In all these cases, if some noise is added to fuzzify the well-separatedness, the noise can be more destructive for the set A, B, C than for any of the three mentioned subsets – this affects GSC as well as k-means clustering. This is easily imagined by considering k-means algorithm. The cluster center of A when clustering fuzzified A and B may lie in a different position than when clustering fuzzified A and C.

This behavior will be subsequently illustrated by a series of experiments.

III. Data

We used tweets retrieved from the stream endpoint of Twitter API (a random sample of about 1% of English tweets), collected by one of the Authors for the time period from mid September 2019 till end of November 2022. From this set we extracted the subset TWT.10 used in experiments. It is a collection of top thread tweets related to hashtags listed in Table I. While selecting the data, we imposed the restriction that the tweets had to have one single hashtag (which we treated as an indication of being devoted to a single theme).

IV. Methods

We study two standard versions of Graph Spectral Clustering, available from scikit-learn, and the 6 versions of spherical k-means and 6 versions of our proprietary so-called K-embedding based clustering algorithm.

More precisely the clustering experiments were performed with popular Python libraries: numpy [7], scipy [8], scikit-learn [9] and soyclustering [10] which is an implementation of spherical k-means [11]. In particular, we used

1) `SpectralClustering` class from scikit-learn with two distinct settings of the `affinity` parameter: `precomputed` (affinity from similarity matrix) and `nearest_neighbors` (affinity from graph of nearest neighbors) - as a representative of the spectral clustering, and
2) `SphericalKMeans` class from soyclustering with the following combinations of (`init`, `sparsity`) parameter pairs (the mentioned 6 versions, short names given for reference): ”sc.n”: (’similar_cut’, None), ”sc.sc”: (’similar_cut’, ’sculley’), ”sc.md”: (’similar_cut’, ’minimum_df’), ”k++.n”: (’k-means++’, None), ”k++.sc”: (’k-means++’, ’sculley’), ”k++.md”: (’k-means++’, ’minimum_df’), and
3) K-embedding clustering (our implementation, exploiting spherical k-means – see subsection IV-C). Same combinations of parameter pairs (versions) were used as for `SphericalKMeans` above. The following numbers of eigenvectors were tried: $r = 12$ and higher.

The advantages and disadvantages of these methods are briefly discussed below.

A. Spectral analysis

In fact spectral clustering algorithms constitute a large family, see e.g. [12], [13], [14], which have numerous desirable properties (like detection of clusters with various shapes, applicability to high dimensional datasets, capability to handle categorical variables), yet they suffer from various shortcomings, common to other sets of algorithms, including multiple possibilities of representation of the same dataset, producing results in a space different from the space of original problem, curse of dimensionality, etc. These shortcomings are particularly grieving under large and sparse data set scenario, like in Twitter data.

Let us briefly recall the typical spectral clustering algorithm in order to make it understandable, how distant the clustering may be from the applier's comprehension [12]. The first step consists in creating a similarity matrix of objects (in case of documents based on tf, tfidf, in unigram or n-gram versions, or some transformer based embeddings are the options – consult e.g. [15] for details), then mixing them in case of multiple views available. The second step is to calculate a Laplacian matrix. There are at least three variants to use: combinatorial, normalized, and random-walk Laplacian, [12]. But other options are also possible, like: some kernel-based versions, non-backtracking matrix [16], degree-corrected versions of the modularity matrix [17] or the Bethe-Hessian matrix [18]. Then computing eigenvectors and eigenvalues, eigenvector smoothing (to remove noise and/or achieve robustness against outliers) choice of eigenvectors, and finally clustering in the space of selected eigenvectors (via e.g. k-means). The procedure may be more complex, e.g. one may add loops back to preceding steps based on feedback from quality analysis, like degree of deviation from block-structure of the Laplacian.

From this diversified set we chose the two mentioned implementations available from scikit-learn.

B. Spherical k-means

Spherical k-means was developed in [5] by observing that the squared Euclidean distance between two vectors, $\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \|\mathbf{x}_i\|^2 - 2\mathbf{x}_i^T \mathbf{x}_j + \|\mathbf{x}_j\|^2$, in case of normalized vectors reduces to

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = 2(1 - \mathbf{x}_i^T \mathbf{x}_j) \ , \tag{1}$$

and $\mathbf{x}_i^T\mathbf{x}_j = \cos\angle(\mathbf{x}_i, \mathbf{x}_j)$. This makes it very efficient in case of sparse vectors, a typical representation of text documents. Such a variant of k-means suffers dependence on initialization, thus further improvements are proposed, e.g. [19], [20], [21] and [22].

C. K-embedding

K-embedding has the following underlying idea. Let us think for a moment about a particular embedding of the nodes of the graph, based on [23]. Let A be a matrix of the form:

$$A = \mathbf{1}\mathbf{1}^T - I - S \ , \tag{2}$$

where S stands for an affinity matrix, I is the identity matrix, and $\mathbf{1}$ is the (column) vector consisting of ones, both of appropriate dimensions. (Note that here we have to assume that the diagonal of S consists of zeros). Let K be the matrix of the (double centered) form [24]:

$$K = -\frac{1}{2}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)A(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) \ , \tag{3}$$

with $n \times n$ being the dimension of S. $\mathbf{1}$ is an eigenvector of K, with the corresponding eigenvalue equal to 0. All the other eigenvectors must be orthogonal to it as K is real and symmetric, so for any other eigenvector $\mathbf{v}$ of K we have: $\mathbf{1}^T\mathbf{v} = 0$.

Let Λ be the diagonal matrix of eigenvalues of K, and V the matrix where columns are corresponding (unit length) eigenvectors of K. Then $K = V\Lambda V^T$. Let $\mathbf{z}_i = \Lambda^{1/2}V_i^T$, where V_i stands for i-th row of V. Let $\mathbf{z}_i, \mathbf{z}_\ell$ be the embeddings of the nodes i, ℓ, resp. This embedding shall be called *K-embedding*. Then

$$\|\mathbf{z}_i - \mathbf{z}_\ell\|^2 = 1 - S_{i\ell} \tag{4}$$

for $i \neq \ell$. Hence upon performing k-means clustering in this space we *de facto* try to maximize the sum of similarities within a cluster. Note that $K = V\Lambda V^T$ may be quite well approximated if we drop from Λ low eigenvalues and from V their corresponding eigenvectors (which we do in our experiments).

V. Evaluation

For each of the algorithms we perform the following tests. For each pair of datasets associated with two hashtags from Table I (45 pairs in all) the clustering will be performed by each of the mentioned algorithms 10 times (due to stochastic nature of these algorithms) and the average F-score will be computed. Ten pairs with the highest average F-scores will be taken for the next phase. Now datasets associated with 3 hashtags will be created out of these selected pairs plus each of the hashtags not present in the selected pairs. This process is continued till all 10 hashtags are exhausted. In figures, the average value of F over all computations with the given hashtag cardinality is presented plus the average of the top 10 groups of hashtags. The results are summarized in Figs. 6–19.

The next experiment was to compare the F-score obtained by a given set of hashtags, considered in the preceding experiment, and its subsets obtained by removing one of the hashtags. For example, we considered the F-score of ['lolinginlove', 'tejran', 'anjisalvacion'] and the average of F-scores for the subsets ['lolinginlove', 'tejran'], ['lolinginlove', 'anjisalvacion'] and ['tejran', 'anjisalvacion']. We computed the Spearman and Pearson correlations for such pairs (F-score of a set of hashtags and average of its subsets) and presented the results in Table II for each of the analysed clustering algorithms. We have created also a more detailed view for one of the algorithms: spectral clustering with affinity nearest neighbors. Fig. 20 presents the histogram of differences between the average F-score of subgroups and the F-score of the group. Fig. 21 presents the relation between the average F-score of subgroups and the F-score of the group as a scatterplot.

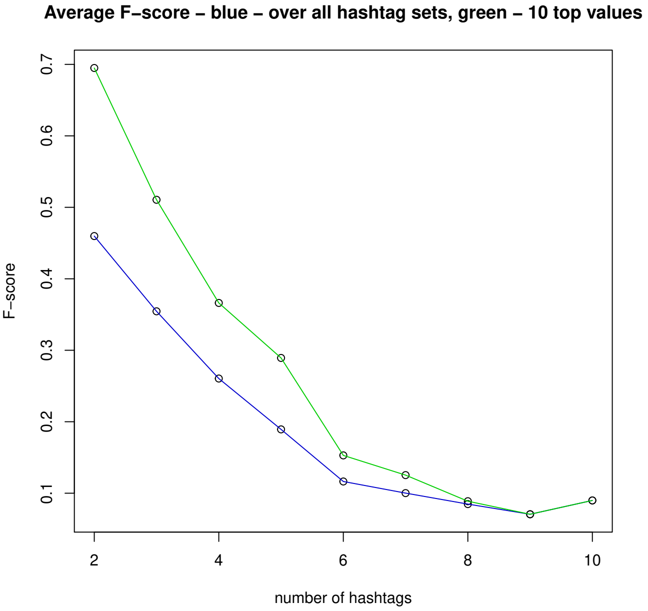


Fig. 6. F-scores for various numbers of hashtags; spectral clustering with affinity nearest_neighbors

TABLE II
Correlation between the F-score of a given group of hashtags and their subgroups of cardinality lower by one.

algprithm	pearson	p.val	spearman	p.val
spectral nearest_neighbors	0.7745	0	0.8358	0
spectral precomputed	0.7374	0	0.7437	0
spherical sc.md	0.7036	0	0.7711	0
spherical sc.sc	**0.8306**	0	**0.8538**	0
spherical k++.n	0.7647	0	0.8189	0
spherical sc.n	0.7778	0	0.8167	0
spherical k++.md	0.7796	0	0.8129	0
spherical k++.sc	0.8099	0	0.8502	0
K-embedding.12plus sc.md	0.6057	0	0.6041	0
K-embedding.12plus sc.sc	0.6948	0	0.6678	0
K-embedding.12plus k++.n	0.7975	0	0.8294	0
K-embedding.12plus sc.n	0.6901	0	0.7113	0
K-embedding.12plus k++.md	0.7976	0	0.8483	0
K-embedding.12plus k++.sc	0.7924	0	0.8460	0

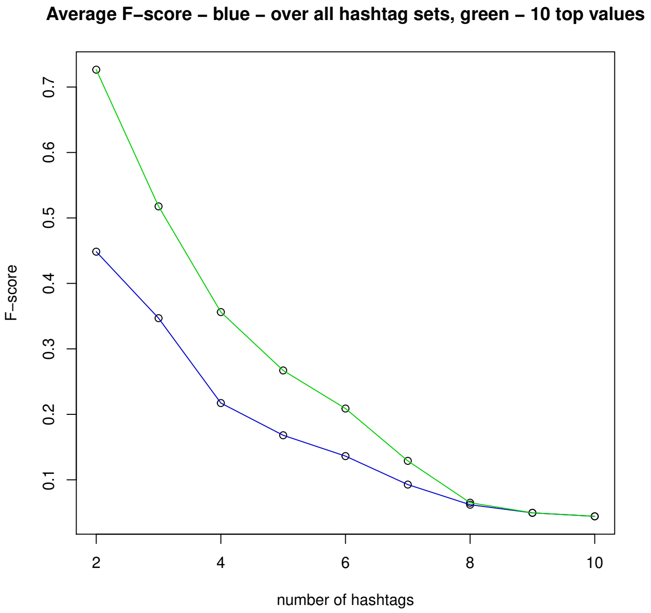


Fig. 7. F-scores for various numbers of hashtags; spectral clustering with affinity precomputed

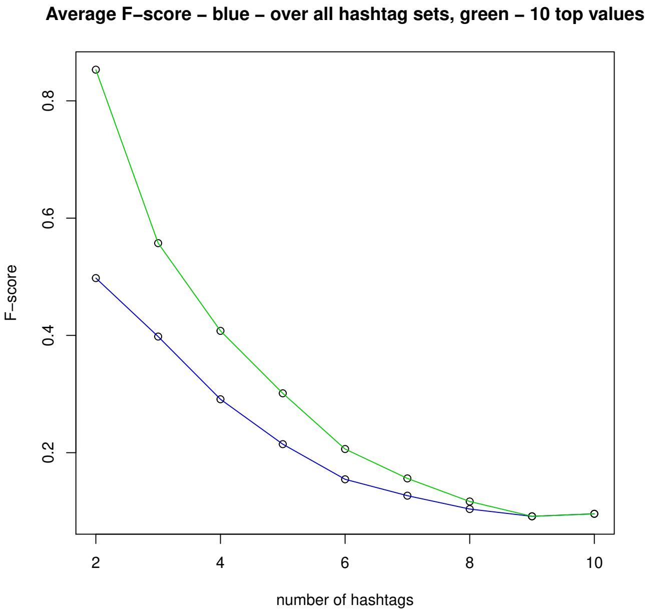


Fig. 9. F-scores for various numbers of hashtags; spherical k-means clustering with sc.sc configuration

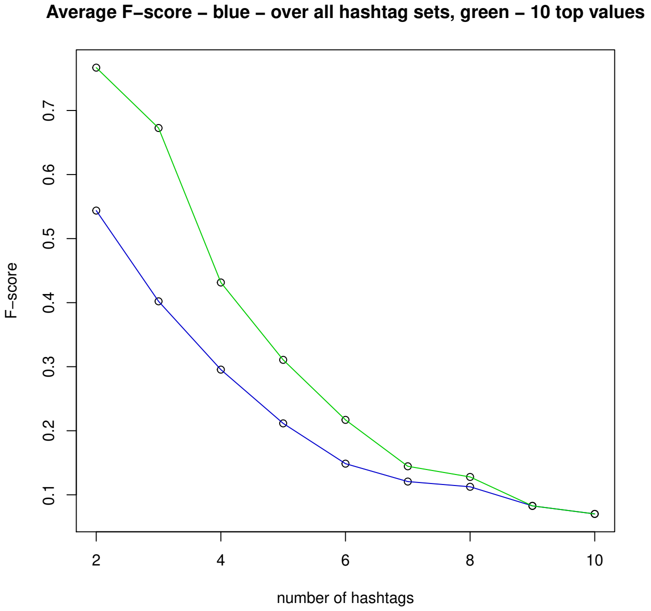


Fig. 8. F-scores for various numbers of hashtags; spherical k-means clustering with sc.md configuration

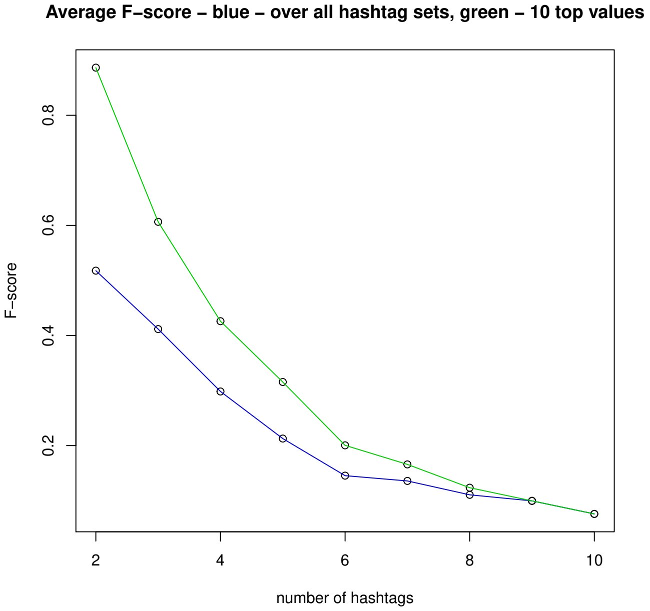


Fig. 10. F-scores for various numbers of hashtags; spherical k-means clustering with sc.n configuration

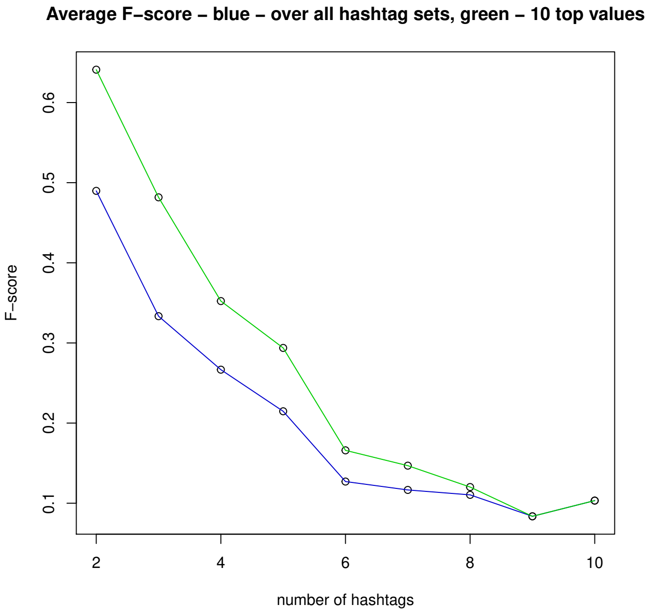


Fig. 11. F-scores for various numbers of hashtags; spherical *k*-means clustering with k++.md configuration

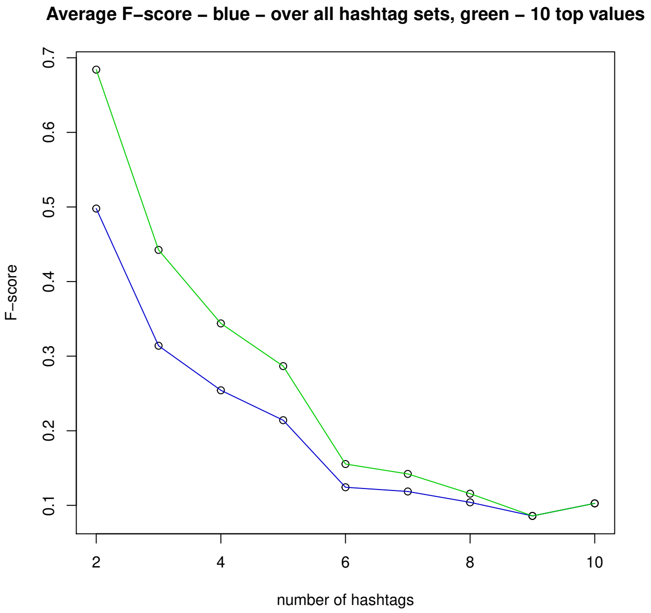


Fig. 12. F-scores for various numbers of hashtags; spherical *k*-means clustering with k++.n configuration

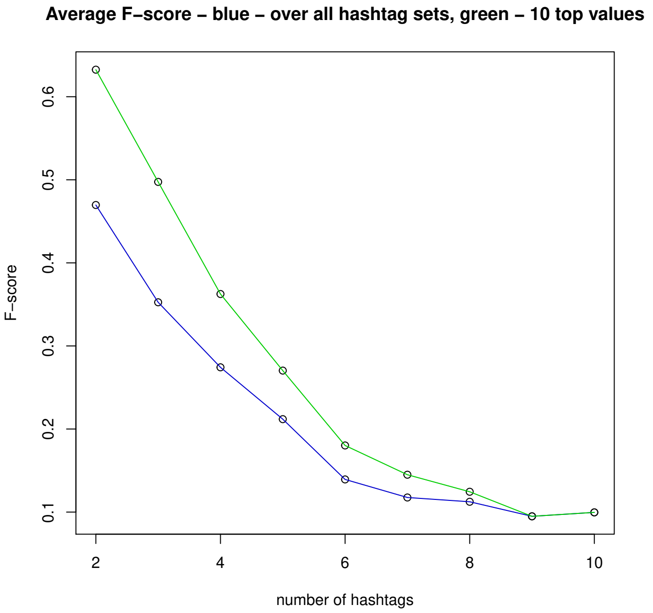


Fig. 13. F-scores for various numbers of hashtags; spherical *k*-means clustering with k++.sc configuration

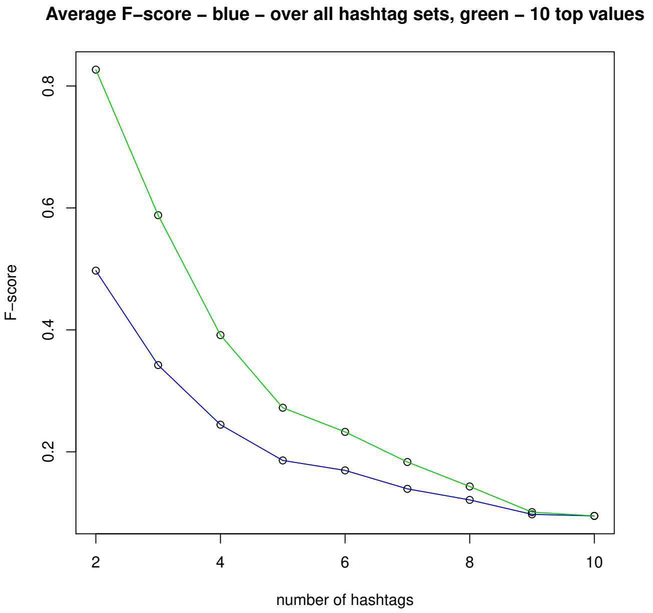


Fig. 14. F-scores for various numbers of hashtags; K-embedding based clustering with sc.md configuration

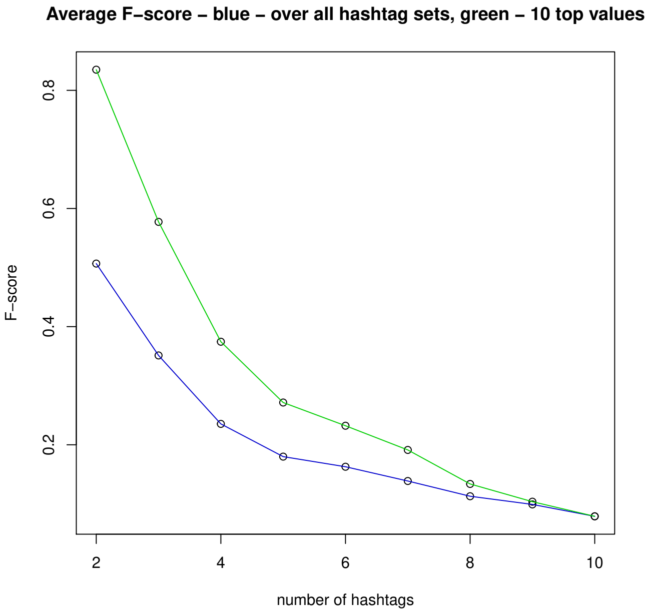


Fig. 15. F-scores for various numbers of hashtags; K-embedding based clustering with sc.sc configuration

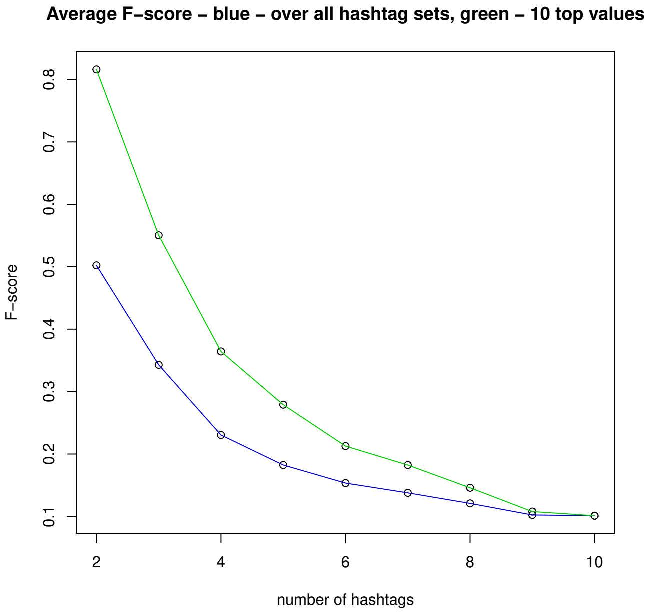


Fig. 16. F-scores for various numbers of hashtags; K-embedding based clustering with sc.n configuration

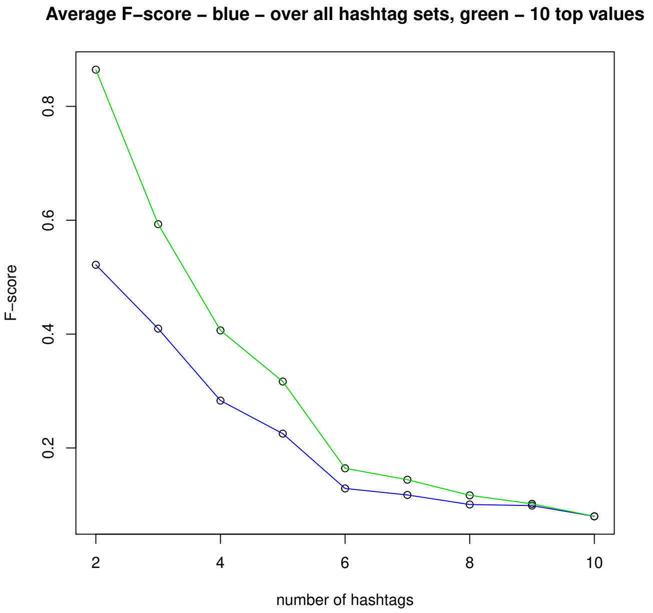


Fig. 17. F-scores for various numbers of hashtags; K-embedding based clustering with k++.md configuration

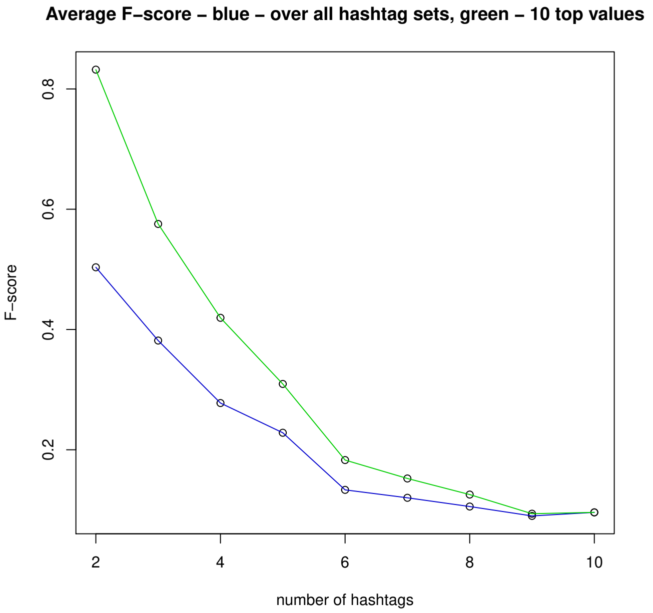


Fig. 18. F-scores for various numbers of hashtags; K-embedding based clustering with k++.n configuration

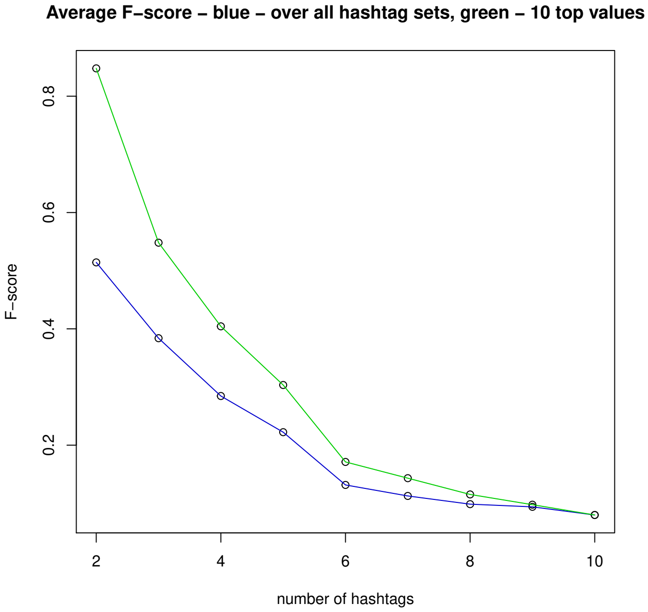


Fig. 19. F-scores for various numbers of hashtags; K-embedding based clustering with k++.sc configuration

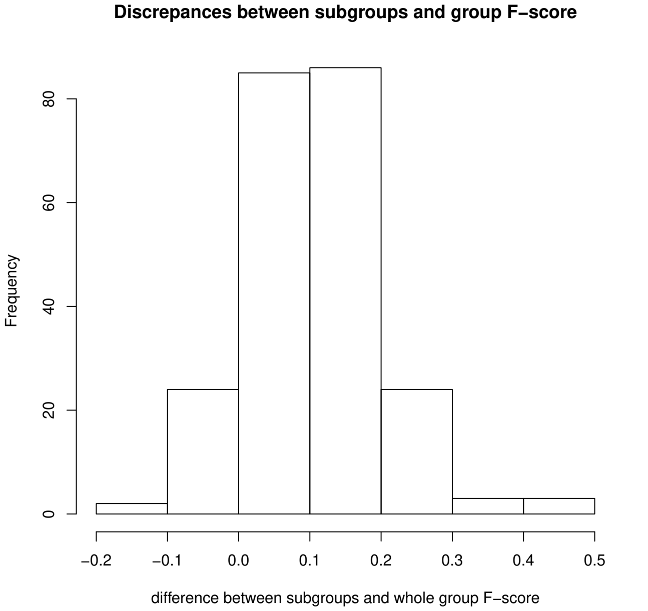


Fig. 20. Difference (negated) between F-score of the given group that was clustered and the average F-score of its subgroups (with one less hashtag); spectral clustering with affinity nearest_neighbors

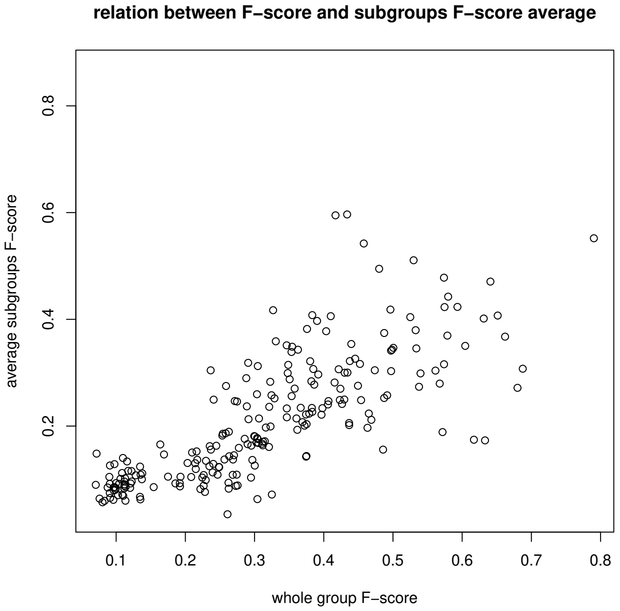


Fig. 21. Relationship between F-score of the given group that was clusters and the average F-score of its subgroups (with one less hashtag); spectral clustering with affinity nearest_neighbors

VI. Results

As visible from Figs. 6–19, the increase of the number of intended clusters to be discovered constitutes a problem for the clustering algorithms, with even 9-fold decrease of F-score when going from 2 to 10 clusters. This behaviour is consistent throughout all the investigated methods though minor variations of the shape of the curves may be observed.

Spherical k-means clustering with sc.n configuration appears to perform best for the 10 top pairs of hashtags (Fig. 10) and with sc.sc configuration (Fig. 9), followed by K-embedding based clustering with most configurations (Figs. 14–19, except 16).

In most cases the top average of the F-score for next higher number of cluster is usually higher than the average score for the entire previous number of clusters, which indicates that better separation of subgroups gives some advantage for the capability to separate the entire group.

Table II shows Spearman and Pearson correlations between the F-score achieved by grouping a dataset related to a given set of hashtags and by grouping datasets obtained by removing data of one of the hashtags, split by the clustering algorithm. The correlations are generally high and are statistically very significant. This means that clustering capability of subsets of hashtags can be a good indicator of clustering capability for the set of hashtags. The algorithm spherical sc.sc seems to perform best for such a criterion, followed by spherical k++.sc and in the column on Spearman correlation – K-embedding.12plus k++.md.

A more detailed insight into this relationship for one of the algorithms is presented in Figs 20 and 21. Fig. 20 convinces us, however, that generally this clustering capability decreases (the F-score of a group is usually lower than that of the average of the subgroups). Fig. 21 shows additionally, that the high correlations between group and subgroups of hashtags are to be expected rather for low values of F-score. Higher F-score values are responsible for higher variation in supergroup F-score.

VII. Conclusions

The performed experiments demonstrate that, in spite of the generally praised properties, graph spectral clustering methods have still a large space for improvements with respect to increasing number of clusters to be detected. Even if all the subsets of intended clusters may be well separated by the algorithms, their mixture does not so. Same observation can be made about the spherical k-means algorithm.

References

[1] S. T. Wierzchoń and M. A. Kłopotek, *Modern Clustering Algorithms*, ser. Studies in Big Data. Springer Verlag, 2018, vol. 34. ISBN 978-3-319-69307-1. doi: https://doi.org/10.1007/978-3-319-69308-8

[2] P. Łoziński, D. Czerski, and M. A. Kłopotek, "Grammatical case based IS-A relation extraction with boosting for polish," in *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, FedCSIS 2016, Gdańsk, Poland, September 11-14, 2016*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 8. IEEE, 2016. doi: 10.15439/2016F391 pp. 533–540. [Online]. Available: https://doi.org/10.15439/2016F391

[3] J. Dörpinghaus, S. Schaaf, J. Fluck, and M. Jacobs, "Document clustering using a graph covering with pseudostable sets," in *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3-6, 2017*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 11, 2017. doi: 10.15439/2017F84 pp. 329–338. [Online]. Available: https://doi.org/10.15439/2017F84

[4] P. Borkowski, M. A. Kłopotek, B. Starosta, S. T. Wierzchoń, and M. Sydow, "Eigenvalue based spectral classification," *PLoS ONE*, vol. 18, no. 4, p. e0283413, 2023. doi: https://doi.org/10.1371/journal.pone.0283413

[5] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143–175, Jan 2001. doi: https://doi.org/10.1023/A:1007612920971

[6] S. T. Wierzchoń and M. A. Kłopotek, "Spectral cluster maps versus spectral clustering," in *Computer Information Systems and Industrial Management*, ser. LNCS, vol. 12133. Springer, 2020. doi: 10.1007/978-3-030-47679-3_40 pp. 472–484.

[7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, p. 357–362, 2020. doi: 10.1038/s41586-020-2649-2 https://numpy.org.

[8] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. doi: 10.1038/s41592-019-0686-2 https://scipy.org.

[9] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013. doi: 10.48550/arXiv.1309.023 pp. 108–122, https://scikit-learn.org.

[10] H. Kim and H. K. Kim, "clustering4docs github repository," 2020, https://pypi.org/project/soyclustering/. [Online]. Available: https://github.com/lovit/clustering4docs

[11] H. Kim, H. K. Kim, and S. Cho, "Improving spherical k-means for document clustering: Fast initialization, sparse centroid projection, and efficient cluster labeling," *Expert Systems with Applications*, vol. 150, p. 113288, 2020. doi: https://doi.org/10.1016/j.eswa.2020.113288. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420301135

[12] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007. doi: https://doi.org/10.48550/arXiv.0711.0189

[13] P. Macgregor and H. Sun, "A tighter analysis of spectral clustering, and beyond," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022. doi: https://doi.org/10.48550/arXiv.2208.01724 pp. 14 717–14 742. [Online]. Available: https://proceedings.mlr.press/v162/macgregor22a.html

[14] Y. Xu, A. Srinivasan, and L. Xue, *A Selective Overview of Recent Advances in Spectral Clustering and Their Applications*. Cham: Springer International Publishing, 2021, pp. 247–277. ISBN 978-3-030-72437-5. doi: 10.1007/978-3-030-72437-5_12

[15] C. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. doi: https://doi.org/10.1017/CBO9780511809071

[16] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborov, and P. Zhang, "Spectral redemption in clustering sparse networks," in *Proc. the National Academy of Sciences*, vol. 110[50], 2013. doi: 10.48550/arXiv.1306.5550 pp. 20 935–20 940.

[17] H. T. Ali and R. Couillet, "Improved spectral community detection in large heterogeneous networks," *Journal of Machine Learning Research*, vol. 18, no. 225, pp. 1–49, 2018. [Online]. Available: http://jmlr.org/papers/v18/17-247.html

[18] A. Saade, F. Krzakala, and L. Zdeborová, "Spectral clustering of graphs with the bethe hessian," 2014. [Online]. Available: https://arxiv.org/abs/1406.1880. doi: 10.48550/ARXIV.1406.1880

[19] Y. Endo and S. Miyamoto, "Spherical k-means++ clustering," in *Modeling Decisions for Artificial Intelligence*, V. Torra and T. Narukawa, Eds. Cham: Springer International Publishing, 2015. doi: https://doi.org/10.1007/978-3-319-23240-9_9. ISBN 978-3-319-23240-9 pp. 103–114.

[20] S. Ji, D. Xu, L. Guo, M. Li, and D. Zhang, "The seeding algorithm for spherical k-means clustering with penalties," *J. Comb. Optim.*, vol. 44, no. 3, p. 1977–1994, oct 2022. doi: 10.1007/s10878-020-00569-1. [Online]. Available: https://doi.org/10.1007/s10878-020-00569-1

[21] J. Knittel, S. Koch, and T. Ertl, "Efficient sparse spherical k-means for document clustering," in *Proceedings of the 21st ACM Symposium on Document Engineering, DocEng '21*. ACM, New York, NY, United States, 2021. doi: https://doi.org/10.1145/3469096.3474937 pp. 1–4.

[22] R. Pratap, A. Deshmukh, P. Nair, and T. Dutt, "A faster sampling algorithm for spherical k-means," in *Proceedings of The 10th Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Zhu and I. Takeuchi, Eds., vol. 95. PMLR, 14–16 Nov 2018, pp. 343–358. [Online]. Available: https://proceedings.mlr.press/v95/pratap18a.html

[23] R. A. Kłopotek, M. A. Kłopotek, and S. T. Wierzchoń, "A feasible k-means kernel trick under non-euclidean feature space," *International Journal of Applied Mathematics and Computer Science*, vol. 30, no. 4, pp. 703–715, 2020. doi: https://doi.org/10.34768/amcs-2020-0052 Online publication date: 1-Dec-2020.

[24] J. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, vol. 53(3-4), pp. 325—-338, 1966. doi: https://doi.org/10.1093/biomet/53.3-4.325

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 769–780
DOI: 10.15439/2023F3882
ISSN 2300-5963 ACSIS, Vol. 35

An Experimental Framework for Secure and Reliable Data Streams Distribution in Federated IoT Environments

Jakub Sychowiec
Cybernetics Faculty
Military University of Technology
Warsaw, Poland
jakub.sychowiec@wat.edu.pl

Zbigniew Zieliński
Cybernetics Faculty
Military University of Technology
Warsaw, Poland
zbigniew.zielinski@wat.edu.pl

***Abstract*—An increasing number of Internet of Things (IoT) applications are based on a federated environment. Examples include the creation of federations of NATO countries and non-NATO entities participating in missions (Federated Mission Networking) or the interaction of civilian services and the military when providing Humanitarian Assistance And Disaster Relief. Federations are often formed on an ad hoc basis, with the primary goal of combining forces in a federated mission environment at any time, on short notice, and with optimization of the resources involved. One of the leading security challenges in a federated environment of separate IoT administrative domains is effective identity and access management, which is the basis for establishing a relationship of trust and secure communication between IoT devices belonging to different partners. When carrying out missions involving the military and ensuring security, meeting requirements for immediate interoperability is important. In the paper, an attempt has been made to develop a system architecture framework for secure and reliable data streams distribution in a multi-organizational federation environment, where data authentication is based on IoT device identity (fingerprint). Moreover, a hardware-software IoT gateway has been proposed for the verification process and the integration of Hyperledger Fabric's distributed ledger technology, the Apache Kafka message broker, and data-processing microservices implemented using the Kafka Streams API library. The performance tests conducted confirm the suitability of the developed system framework for processing and distributing audio-video data in a federation IoT environment. Also, a high-level security and reliability assessment was conducted in the paper.**

***Index Terms*—Internet of Things, Blockchain, Distributed Ledgers, Device Authentication**

I. Introduction

WITH technological advances in mobile radio networks, particularly concerning the implementation of 5G technology, and the development and general availability of many electronic devices equipped with a radio interface, we are seeing an increase in industrial-scale applications of the Internet of Things (IoT), occurring in both the civilian and military spheres. Examples of such applications include smart transportation, smart power grids, smart cities, or the Internet of Battle Things (IoBT) [1]. The consequence is the generation of enormous amounts of data by various IoT devices. One of the main problems attracting the attention of many researchers today is the acquisition, analysis, and fusion of these data, their secure and reliable distribution, and context-dependent information sharing. The problem is compounded in these applications when some institutions or organizations using IoT form a federation to enable different parties to cooperate. A prerequisite for effective cooperation between partners in a federation is sharing certain resources belonging to different participants and exchanging information.

One example is the creation of a federation formed of NATO countries and non-NATO mission actors (Federated Mission Networking) [2], where each actor retains control over its capabilities and operations while accepting and meeting the requirements outlined in pre-negotiated and agreed-upon arrangements, such as security policy. The main idea is to join forces in a federated mission environment at any time (zero-day interoperability), on short notice, and with optimization of the resources involved. The expected result is better command and full control of operations and decision-making through improved information sharing. Another example is the interaction of civilian services and the military, which form a federation when providing humanitarian assistance in eliminating natural disasters (HADR - Humanitarian Assistance And Disaster Relief). There are many situations in which federation partners need to exchange information, for example, about the location of each other's troops, detected threats, etc. IoT in a federation environment enhances the ability to get an accurate real-time [3] picture of the situation during an operation, e.g., by deploying mobile IoT devices such as unmanned aerial vehicles (UAVs). Hence, IoT devices operated by different federation partners must securely communicate with each other. To ensure the timely transmission of situational awareness data, the UAV may need to use a partner's resources within the communication range. In this case and many others, it is necessary to establish a trust relationship through mutual authentication of devices belonging to different federation partners, such as between a specific UAV and the partner's data distribution system.

In the case of information exchange in federated environments, where troops belonging to different NATO countries participate, or the military participates jointly with civilian services in HADR operation in an urbanized area, it is most often assumed that 5G mobile radio networks will be used as the main communication medium. To obtain a precise picture of the situation of the so-called situational awareness, there is often a need to transmit image data to other partners forming a federation. In such an environment, audiovisual data streams generated by IoT devices or city surveillance cameras (CCTV) must be considered trustworthy before they can be properly processed and transmitted to selected command posts. The primary way to confirm the reliability of such data is to authenticate the IoT devices that generate it. The presented needs for ensuring the reliability and security of data exchange bring challenges, the solution of which determines the implementation of IoT. The basic problem remains: *how to carry out the acquisition and fusion of data from various sources with different levels of reliability, and operate in computing environments with varying degrees of trust securely and reliably?* To solve it, it is necessary to know the answer to the subquestions in the first place:

- *Identity management gap* - How to manage the identity of devices? How to identify devices?
- *Security gap* - How to securely distribute data among participants in a federated environment? (Taking into account the priorities assigned to devices).
- *Network integration and interoperability gap* - How to organize interconnections, especially between unclassified systems (civilian systems) and military systems?
- *Resilience and centralization gap* - How to ensure data availability in constrained (partially isolated) environments?

Taking into consideration the aforementioned requirements for federated IoT environments and presented sub-questions, it is necessary to use and integrate multiple technologies, e.g., a data authentication mechanism where a unique identity image (fingerprint) is used, distributed ledger, 5G technology, and data distribution and processing systems. A point worth noting is that data distribution systems are often based on the MQTT protocol [4], [5]. The main disadvantage of this type of solution is the need for additional components to acquire and distribute streaming data, such as video data from daylight or infrared cameras.

This paper proposes a framework architecture for secure and reliable data streams distribution in a multi-organizational federation environment, where data authentication is based on IoT device identity. Moreover, a hardware-software IoT gateway has been proposed for the verification process and the integration of Hyperledger Fabric's distributed ledger technology [6], the Apache Kafka message broker, and data-processing microservices implemented using the Kafka Streams API library [7].

The remainder of the article is structured as follows: Section II provides an overview of the related research work that formed the basis for our solution. Section III describes the proposed framework architecture and its main elements, along with the security mechanisms used to enhance confidentiality, integrity, availability, and accountability for data in-transit and at-rest. The main operations for our experimental framework were described in Section IV. The test environment, workloads scenarios, and preliminary benchmarks with results were presented in Section V. The section also includes a high-level security risk assessment considering several security and reliability threats. Section VI presents conclusions and planned future work.

II. Background and Motivation (Related Work)

In this section, we will present related works that have had the greatest impact on the proposed framework architecture for secure and reliable data streams distribution in Federated IoT Environments. These works address the basic problems related to:

- securing data processed by IoT devices with the usage of blockchain technology;
- unique IoT device identification based on the distinctive features (fingerprints);
- the integration of heterogeneous military and civilian systems based on IoT devices, where the requirement for zero-day interoperability must be ensured.

Additionally, at the end of this section, we have briefly discussed our solution against the analyzed works.

A. Blockchain integration with the Internet of Things

The literature presents numerous attempts to integrate the Internet of Things and blockchain (distributed ledger) technology. The work [8] describes the challenges and benefits of integrating blockchain with the Internet of Things and its impact on the security of processed data. Similarly, in the works [9], [10], where a proposal for a 4-tier structural model of Blockchain and the Internet of Things (BIoT) is presented. Guo et al. [11] proposed a mechanism for authenticating IoT devices in different domains, where cooperating distributed ledgers operating in the master-slave mode were used for data exchange. Xu et al. [12] presented the DIoTA framework based on a private Hyperledger Fabric blockchain, which was used to protect the authenticity of data processed by IoT devices. The work [13] proposed an access control mechanism for devices, which used the Ethereum public blockchain placed in the Fog Layer and public key infrastructure based on elliptic curves.

B. Unique IoT device identification using fingerprint methods

Apart from classification methods for identifying a group or type of similar IoT devices [14], an interesting area of research is fingerprint techniques [15], [16], which aim to identify a unique image of a device identity through the appropriate selection of its distinctive features. The fundamental premise of fingerprint methods is the occurrence of manufacturing errors and configuration distinctions, which implies the nonexistence of two identical devices. Subsequently, the main challenge associated with fingerprinting techniques is the selection of non-ephemeral parameters that make it possible to distinguish devices uniquely. Generally, three main fingerprint methods can be identified for IoT devices as a result of distinction:

1) hardware and software features of the device;
2) characteristics of generated network traffic;
3) characteristics of generated radio signals.

The authors of the LAAFFI framework [17] presented a protocol designed to authenticate devices in federated environments based on unique hardware and software parameters extracted from a given IoT device. Concerning distinctive radio features, Sanogo et al. [18] evaluated the Power Spectral Density parameter. The work [19] indicates a proposal to use neural networks to identify devices based on the Physical Unclonable Function in combination with radio features: frequency offset, in-phase (I) and quadrature (Q) imbalance, and channel distortion. Charyyev et al. [20] proposed the LSIF fingerprint technique, where the Nilsimsa hash function was used to determine a unique IoT device network flow. In contrast, the work of [21] demonstrated the Inter-Arrival Time (IAT) differences between successively received data packets as a unique identification parameter.

C. Zero-day interoperability ensuring for heterogeneous military and civilian systems

Meeting the requirement for zero-day interoperability is a significant challenge for NATO coalition countries. Consequently, attempts are being made to integrate data exchange systems belonging to various partners to create an environment called Federated Mission Networking (FMN). For mentioned environment, NATO countries have regularly defined and revised requirements for years [2] and established research groups to identify the optimal solution for coalition data processing systems. Jansen et al. [4] presented an experimental environment consisting of four organizations between which data is distributed in two configurations. The first configuration uses two MQTT broker types (Mosquitto, VerneMQ), while the second configuration is broker-less and disseminates MQTT messages via broadcast and UDP protocol. Suri et al. [5] made an analysis and performance evaluation for eight data exchange systems used within mobile tactical networks. For the research conducted, the superiority of the DisService protocol over solutions such as Redis and RabbitMQ was demonstrated. Additionally, the work [22] proposes a data exchange system for IoT devices based on the MQTT protocol, where data is encrypted using elliptic curves. Moreover, Yang et. al [23] presented a system architecture designed for anonymized data exchange between participants using the Federation-as-a-Service (FaaS) cloud service model. The proposed system architecture was based on the Hyperledger Fabric ledger.

D. Discussion

For most of the reviewed publications, a trusted third-party infrastructure and a private distributed ledger were used to enhance the security of processed data. Compared to the work of Guo et al. [11] (master-slave chain) and Xu et al. [12] (DIoTA framework), our proposed solution is based on a single global instance of the distributed ledger. At the same time, we can freely transfer devices between organizations that are part of the federation, where these devices can use elements of another organization's infrastructure for secure data exchange. The works of [4], and [5] only address the issue of efficient data exchange and do not consider how to secure data streams. Additionally, these works did not consider the evaluation of Kafka, which also enables the handling of MQTT protocol messages. In our work, we proposed using the Kafka broker and stream processing microservices for data distribution.

Moreover, within the proposed framework, we took into account interoperability between military and civilian systems and the limited nature of such environments. Hence, for the proposed system, we have considered the recommendations made by the NATO IST-150 working group [4], which studied disconnected, intermittent, and limited (DIL) tactical networks. Our system uses a publish-subscribe model and Commercial Off-The-Shelf elements that are generally available. Subsequently, we have minimized operational costs, which is essential for ensuring immediate interoperability.

In addition, in our work, we have separated the key used to secure the communication channel for IoT devices from the key used for the data authenticity protection mechanism. Unlike the DIoTA framework, where an HMAC-based commitment scheme and randomly generated keys are used to authenticate messages, we proposed to use the device's unique distinctive features. Consequently, we proposed using a hybrid identity image defined by a combination of several fingerprint methods, primarily based on the parameters of the generated radio signals.

So far, in the publications that we analyzed, we have not noted a solution that, for the problem of secure data exchange, integrates elements of the Hyperledger Fabric blockchain, Kafka broker, and stream-processing microservices.

III. Proposed framework

This section proposes an experimental framework architecture for secure and reliable data (message) stream distribution in a multi-organizational federation environment. Figure 1 shows an example of the system structure for a federation formed from two organizations (Org1, Org2). The Apache Kafka message brokers acquire, merge, and replicate data generated by producers (publishers) and make data available to consumers (subscribers). At the same time, the proposed system enables verification of message streams based on device identity, which is stored redundantly in a distributed Hyperledger Fabric blockchain. Moreover, a hardware-software IoT gateway has been proposed for the verification process. Through which microservices using the stream processing library Kafka Streams API can communicate with the distributed ledger. A crucial aspect of the system architecture is the ability to freely transfer devices between organizations and utilize their

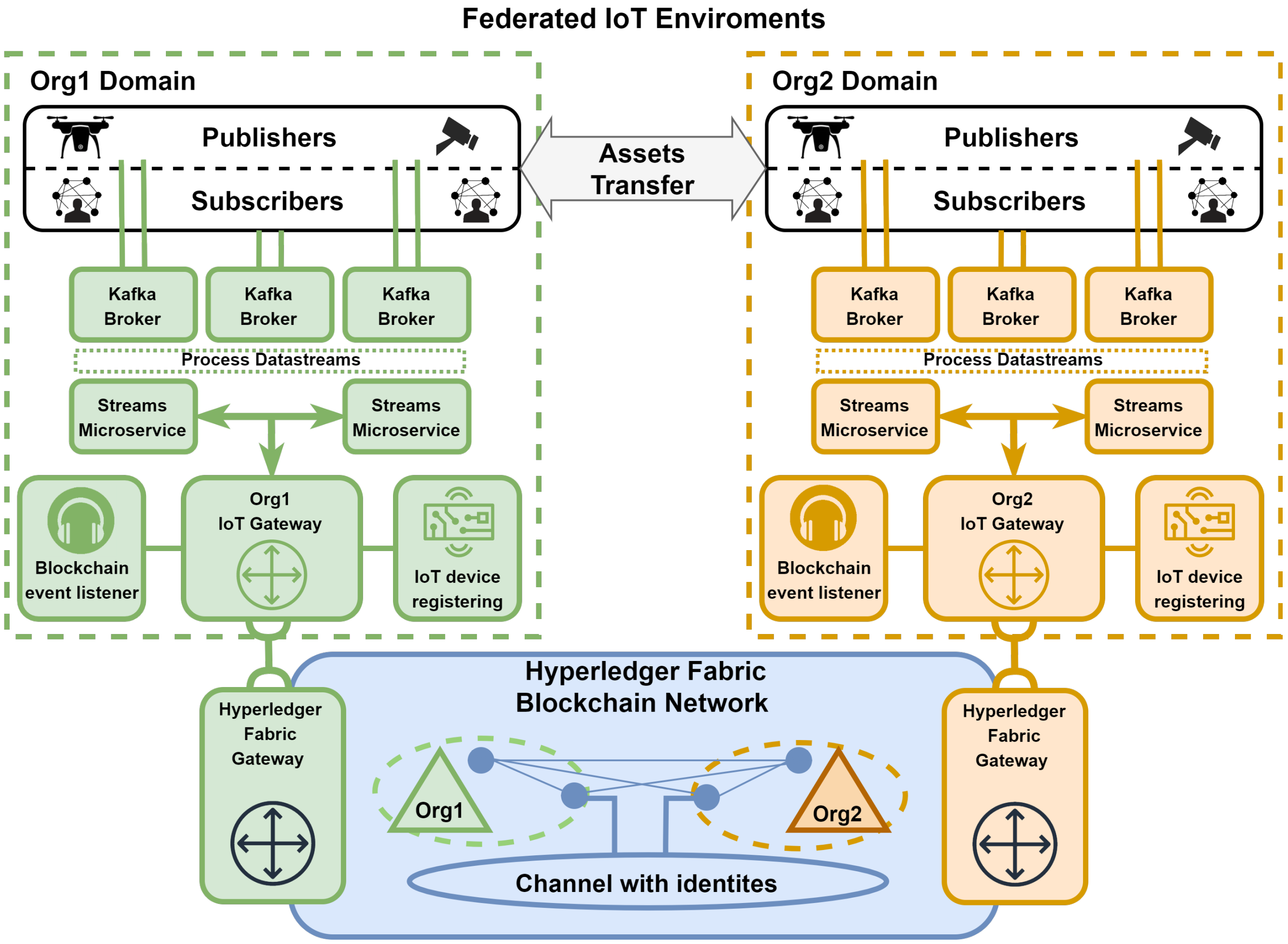


Fig. 1. Proposed framework general overview

Kafka brokers for secure data exchange. In addition, data can be exchanged in any scheme in accordance with the predefined policy, e.g., one-to-many. Furthermore, through the IoT gateway, it is possible to listen for events related to transactions of registering the identity of new devices.

Figure 2 illustrates in detail the proposed solution, where the messages generated by the producers are tagged (sealed) with their identity and then sent using the available communication medium to the broker on a specific topic (e.g., cctv-1-in). Messages sealed in this way are read from the broker by microservices and undergo a verification process. The microservice queries the Hyperledger Fabric blockchain for an image of the device identity for comparison with the identity extracted from the message. The IoT gateway handles all communication with the distributed ledger via an interface to Hyperledger Fabric Gateway services running on the ledger nodes. Successfully verified messages are saved on a dedicated topic (e.g., cctv-1-out) and provided to consumers. Messages which fail verification are discarded by the microservice or written to a previously designated topic to identify faulty (malicious) devices. As pointed out, a device identity image is used to verify the message. The identity is determined in the registration phase by using hybrid fingerprint techniques with a focus on the specificity of the generated radio signals. For a new identity image to be registered in the distributed ledger, the Hyperledger Fabric chaincode is called, which handles the transaction of adding a new identity. Successful registration of the device in the ledger is achieved by obtaining consensus among the organizations belonging to the federation. At the same time, the addition of a new identity implies the generation of an event by the blockchain, which can be handled by dedicated event listening applications (Blockchain Event Listener). In the context of the described solution, these applications were used in terms of reducing the delays associated with the processing of sealed messages. For this purpose, the event listening mechanism was integrated with the Kafka broker and microservices that use local data stores. Also, a dedicated topic (e.g., device-identity) is used for this operation. As a result of the proposed operation, the device identity image can be read from two stores: the on-chain store, called *world state* for Hyperledger Fabric, or the local off-chain store.

In the following headings, the main components of the experimental system and hardware-software IoT gateway are described, along with the reasons for the selection of the proposed elements. Additionally, the description of the various components includes their built-in security mechanisms that enhance confidentiality, integrity, availability, as well as accountability for data in-transit and at-rest.

A. Hyperledger Fabric blockchain

As part of the proposed system architecture, the Hyperledger Fabric solution was chosen. The work [8] provides a performance comparison of various distributed ledger technologies and consensus protocols. In the context of integration with the Internet of Things, mainly Hyperledger Fabric and Ethereum solutions were pointed out as legitimate due to the overall results obtained in experimental studies. Hyperledger Fabric achieved a 10 000 tps transaction throughput [8], for Ethereum throughput was lower. However, only the Proof of Work consensus protocol was benchmarked, and Proof of Stake that is currently used for Ethereum was not included in tests.

Hyperledger Fabric technology is a permissioned blockchain that uses the Practical Byzantine Fault Tolerance (PBFT) consensus protocol. For protocols of this type, all parties must know each other. As a consequence, the Fabric ledger uses public key infrastructure (certificates). The execution of complex business logic (e.g., device registry) is possible by calling multilingual chaincode (Go, Java, Node.js). Chaincode implements a group of smart contracts (transaction steps) and defines an endorsement policy, i.e. which organizations must authorize the transaction.

B. Hyperledger Fabric Gateway

The IoT gateway handles communication with the distributed ledger through an interface to the Hyperledger Fabric Gateway services [6], which allows for:

- performing queries to the world state store and reading the identity from it;
- registering, updating, and revoking IoT device identities from the ledger by calling chaincode;
- handling events generated as a result of approved transactions and blocks.

Moreover, a dynamic mode is proposed to use for the connection profile. This profile uses the ledger nodes' built-in mechanism to identify changes in the network topology on an ongoing basis. As a result, microservices will be able to operate reliably despite the failure of some nodes. Also valuable is the checkpointing mechanism, which makes it possible to resume event listening without losing events due to connection losses.

C. Device Fingerprint

As a part of the registration phase, the image of the IoT device identity will be defined, which will be stored within the device, the Hyperledger Fabric blockchain, and optionally in the local off-chain data store. The identity image will be used as a signing key for messages sent to the Kafka broker. The exact procedures of key management are out of the scope of this article. Consequently, only a general procedure for the mentioned key is presented.

In the registration phase, the device administrator places the device in RF Shielded chamber, which suppresses possible interference affecting the radio waves emitted by the device. Then, using dedicated software and measurement equipment, the device's distinctive features are subjected to a series of tests to define a unique identity image. In this study, a hybrid approach combining several fingerprinting methods is proposed, which is mainly based on the parameters of the generated radio signals. The rationale for this choice is:

- limitations arising from the heterogeneity of the environment and the need to maintain the mobility of IoT devices;
- devices' vulnerability to extreme environmental factors (e.g., temperature, humidity);
- autonomy from the protocols used in the network.

After defining the identity image, the next step is to add it to the distributed ledger. To do this, the chaincode is called, which handles the transaction of adding a new device. Then, an identity image is uploaded to the device. The whole procedure is performed through a secure communication channel with the distributed ledger.

Referring to the format of messages sent to the Kafka broker, Figure 3 shows the general structure for a single message that the broker supports. *Msg_key* and *Msg_value* are a sequence of bytes (binary stream) and represent the payload of a message. The broker or producer can specify the timestamp. Also, the producer can apply message compression or add metadata. The broker assigns the partition number and offset.

The described structure of Kafka messages makes it possible for the broker to accept and handle any format of data. The producer using a data serialization mechanism is the one who determines how to convert the data format of a given protocol (e.g., MQTT) into a bytes representation. Whereas the recipient, through deserialization, defines how to structure the byte string from the broker.

Due to the described Kafka message format and serialization mechanism, it is proposed to use dedicated software running on an IoT device to protect the message depending on its purpose and the required level of security (e.g., classified information). Using this software, it will be possible to:

- authenticating messages;
- signing messages with the identity image and its distinctive characteristics;
- encrypting the message;

Furthermore, this software will be implemented taking into account the parameters of the minimum classes of resource-limited devices, which are defined by RFC 7228 [24].

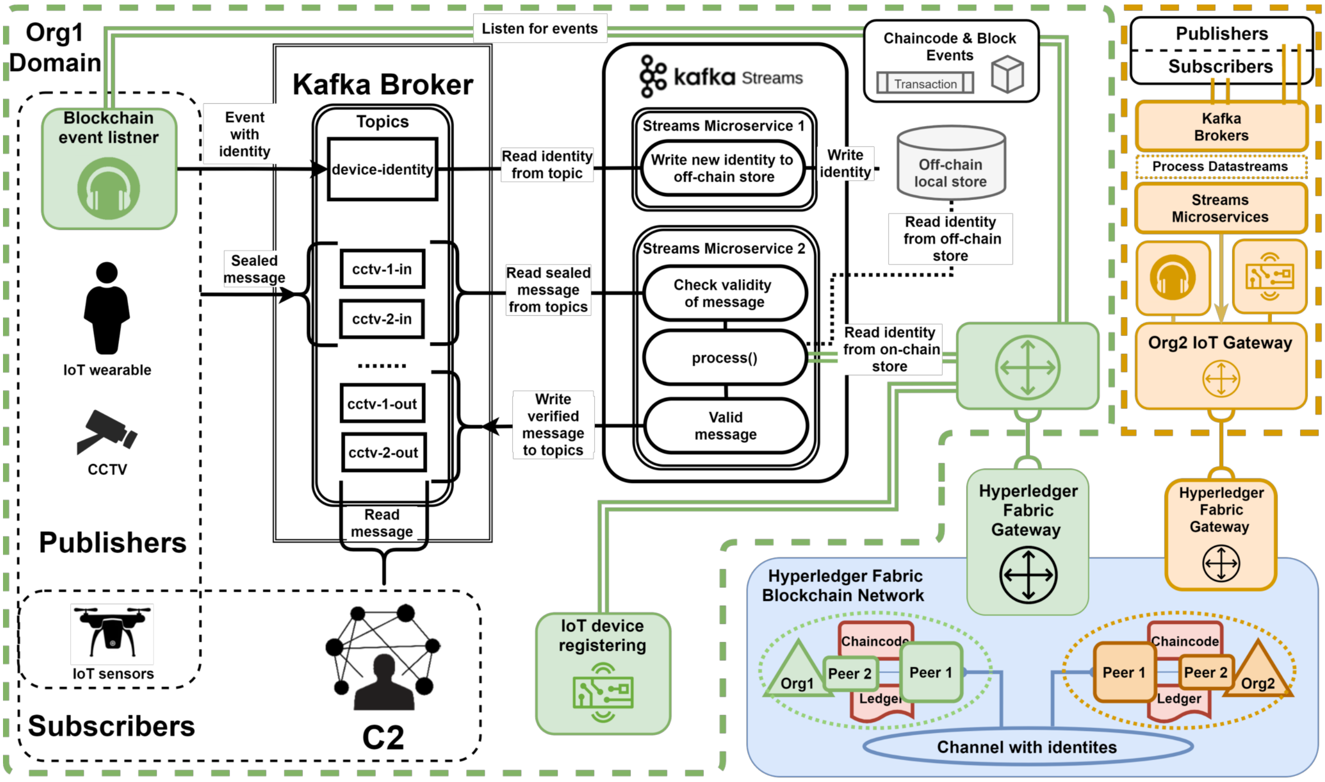


Fig. 2. Proposed framework detailed overview

Msg_key binary data	Msg_value binary data
Timestamp	
Compression Type	
Metadata / Headers	
Key	Value
...	
Key	Value
Partition + offset	

Fig. 3. Kafka message structure

D. Apache Kafka

Considering the multiplicity of data sources (devices) and the need to process messages generated by real-time systems, an Apache Kafka solution has been proposed to enable the streaming processing of data records (messages). For example, Kul et al. [25] presented a framework that uses Kafka and neural networks for tracking (tracking) vehicles, where the dataset was represented as data streams from city surveillance cameras.

Apache Kafka is based on a producer-broker-consumer (publish-subscribe) model and the classification of messages based on their topics. Due to the built-in synchronization mechanism and distributed data (registry) replication between brokers, it is possible to maintain the availability and reliability of data records. In addition, the mechanism of serialization and compression (e.g., lz4, gzip) of data records makes the proposed solution independent of the data format and the protocols used in the network (which is important for heterogeneous environments).

E. Kafka Streams API library

Performing complex operations on data records individually (stream processing) or groups of records (batch processing) requires the selection of an appropriate framework/library. Karimov et al. [26] and Poel et al. [27] evaluated solutions for processing data records. In both works, the Apache Flink framework exceeded in the overall grade other solutions: Kafka Streams, Spark Streaming, and Structured Streaming.

However, in the context of the proposed system architecture, the Kafka Streams API library was chosen, which uses the built-in primitives of Apache Kafka technology like failover and fault-tolerance. Moreover, the library uses a semantic guarantee pattern in which each record (message) is processed exactly once end-to-end. As a result, despite the failure of one of the stream processors (microservice), records will not be lost or double-processed. Spark and Structured Streaming were rejected because these technologies use a micro-batching processing technique, where aggregated data records are processed within defined time windows (threshold). Also, the Apache Flink framework was rejected since it requires a separate processing cluster, which influences operational costs for maintaining the entire infrastructure.

F. On-chain and Off-chain database

It is essential to define and distinguish two categories of data stores within the proposed framework architecture: *on-chain* and *off-chain* stores. World state and transaction log belongs to the on-chain category and refers to the Hyperledger Fabric solution. The world state is a database that determines the current state of the ledger. The transaction log is an exclusively incremental store that acts as a change data capture mechanism where approved and rejected transactions are stored. In contrast, the off-chain category refers to local data stores for applications and microservices that use the Streams API library. For the proposed framework within the on-chain category, the registered identities of IoT devices will be stored. And off-chain stores will serve as additional identity storage to reduce possible delays in the message verification process.

IV. Framework Basic Operations

In this part of the article, the main operations for our experimental framework were presented, taking into account relationships between system elements and message flow.

A. Verification of message streams

In order to verify messages using the distributed ledger, a custom stream processing logic was proposed. Also, a hardware-software IoT gateway was used through which microservices communicate with the Hyperledger Fabric solution. As an optional element, the usage of local off-chain data stores was included. Figure 4 shows a high-level sequence diagram where the steps and message flow are marked for data stream verification operations:

- Step 1: producer (publisher) generates a message and seals it with its own identity image that was beforehand uploaded to the device and the Hyperledger Fabric blockchain during the registration phase;
- Step 2: sealed messages are sent to the Apache Kafka broker to the specific topic using the available and secure communication channel;
- Step 3: microservices sequentially reads the messages from the topic to verify them;
- Step 4: streams microservice *process()* method carries out verification of the message, where the identity image is extracted from the message;
- Step 5: a query to the local off-chain store is made to retrieve the device identity;
- Step 6: the local data store returns the appropriate identity or an error related to its absence;
- Step 7: if identity is retrieved, step 10 is executed. Otherwise, the identity not found error results in a query for the device identity to the distributed ledger, which is executed via an interface to Fabric Gateway services;
- Step 8: the distributed ledger returns the appropriate identity or an error related to its absence;
- Step 9: identity obtained from the ledger is added to the local data store;

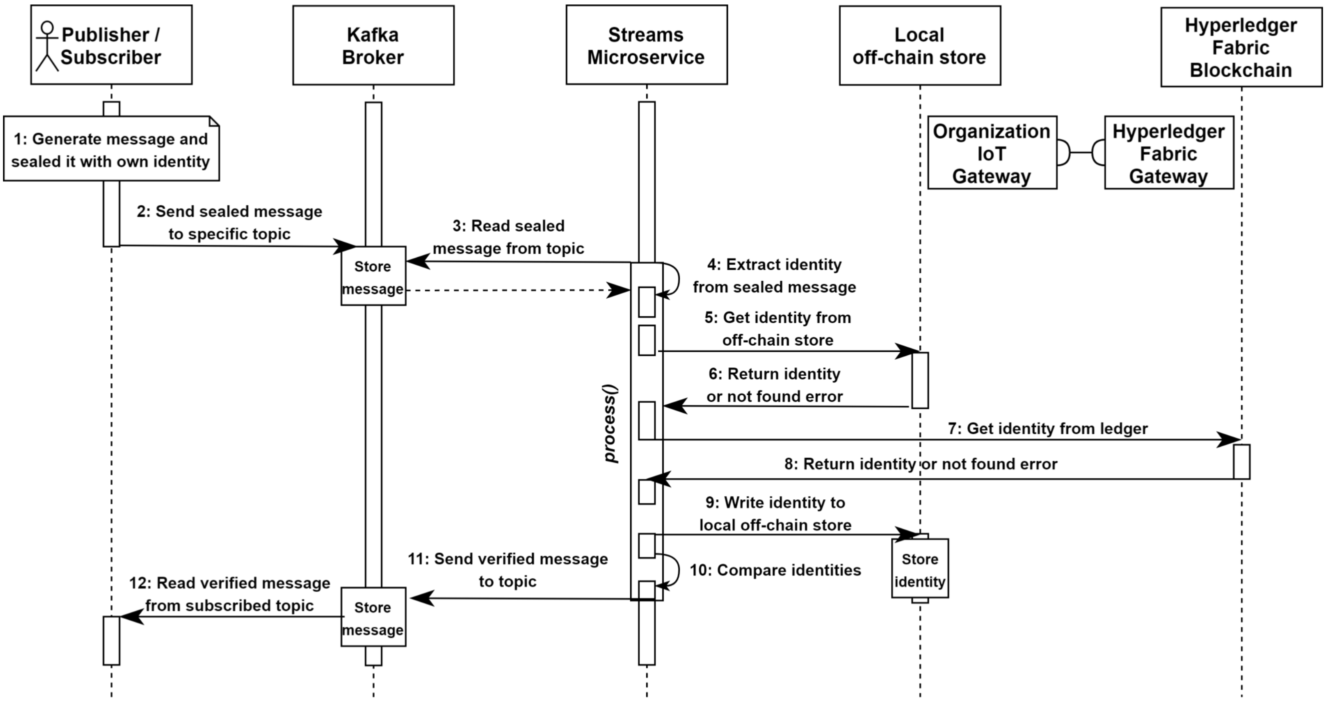

Publisher / Subscriber
Kafka Broker
Streams Microservice
Local off-chain store
Hyperledger Fabric Blockchain
Organization IoT Gateway
Hyperledger Fabric Gateway
1: Generate message and sealed it with own identity
2: Send sealed message to specific topic
3: Read sealed message from topic
Store message
4: Extract identity from sealed message
5: Get identity from off-chain store
6: Return identity or not found error
process()
7: Get identity from ledger
8: Return identity or not found error
9: Write identity to local off-chain store
10: Compare identities
Store identity
11: Send verified message to topic
12: Read verified message from subscribed topic
Store message

Fig. 4. Sequence diagram for verification of message streams

- Step 10: identities are compared with each other;
- Step 11: as a result of a successful identity comparison, the message is saved to the Kafka topic. The message that does not pass verification is discarded or saved to a previously designated topic to identify faulty (malicious) devices;
- Step 12: depending on the subscribed topics, the shared messages can be read sequentially by the consumer.

B. Adding identities through an event listener

The operation of adding (updating) identity to the local off-chain data store is optional and was proposed because of the possibility of reducing time delays for message verification. Figure 5 shows a high-level sequence diagram for the described operation, where:

- Step 1: the identity of the IoT device is defined;
- Step 2: chaincode, which handles the transaction of adding the new identity to the distributed ledger, is invoked;
- Step 3: the transaction is executed after obtaining approvals of organizations specified by the endorsement policy;
- Step 4: blockchain event listener application listens for events emitted by the distributed ledger;
- Step 5: for an approved and executed transaction, an event related to the registration of a new identity image is emitted;
- Step 6: when a specific event is received by the application (Blockchain Event Listener), the identity image is extracted from the event payload;
- Step 7: the identity is uploaded to a dedicated Kafka topic;
- Step 8: the streaming processing microservice sequentially reads the identities from the dedicated topic;
- Step 9: identity is added to the local off-chain data store, which can be used by the other streaming processor microservices.

Optionally, the application or microservice can invoke a synchronization query to the Hyperledger Fabric blockchain to compare the integrity of the master identity image from the ledger with the one extracted from the event payload or written by the microservice to the local data store.

V. Framework Evaluation

Performance benchmarking of streaming data processing systems is an extensive challenge that arises from the problem of the global notion of time. This section describes preliminary benchmarks for our framework, where we evaluated the processing-time latency for microservices that were implemented using the Kafka Streams API. The main purpose of the tests was to confirm whether our framework is feasible to process message streams, especially audiovisual streams. Even if our processing logic is dependent on performing identity read operations from the distributed ledger.

A. Setup

The various components of our experimental framework were deployed using the Amazon Web Services (AWS) cloud environment. Figure 6 shows the test environment, which includes:

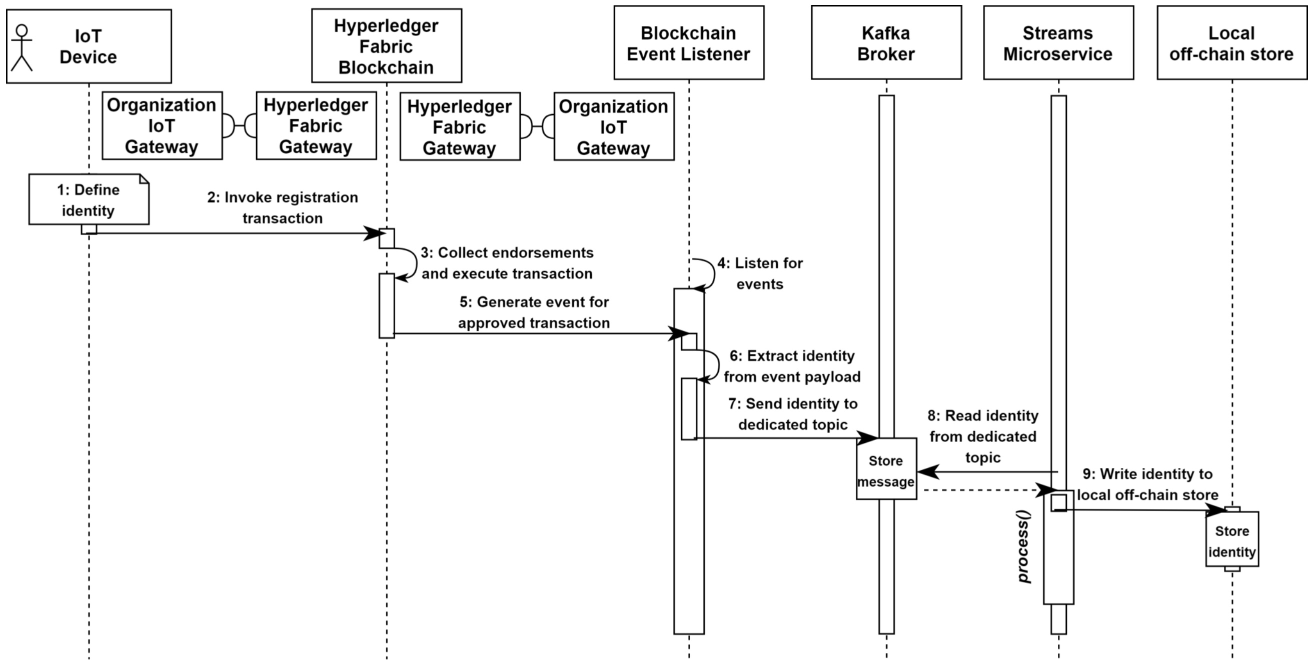

IoT Device
Hyperledger Fabric Blockchain
Blockchain Event Listener
Kafka Broker
Streams Microservice
Local off-chain store
Organization IoT Gateway
Hyperledger Fabric Gateway
Hyperledger Fabric Gateway
Organization IoT Gateway
1: Define identity
2: Invoke registration transaction
3: Collect endorsements and execute transaction
4: Listen for events
5: Generate event for approved transaction
6: Extract identity from event payload
7: Send identity to dedicated topic
8: Read identity from dedicated topic
Store message
9: Write identity to local off-chain store
process()
Store identity

Fig. 5. Sequence diagram for adding identities through an event listener

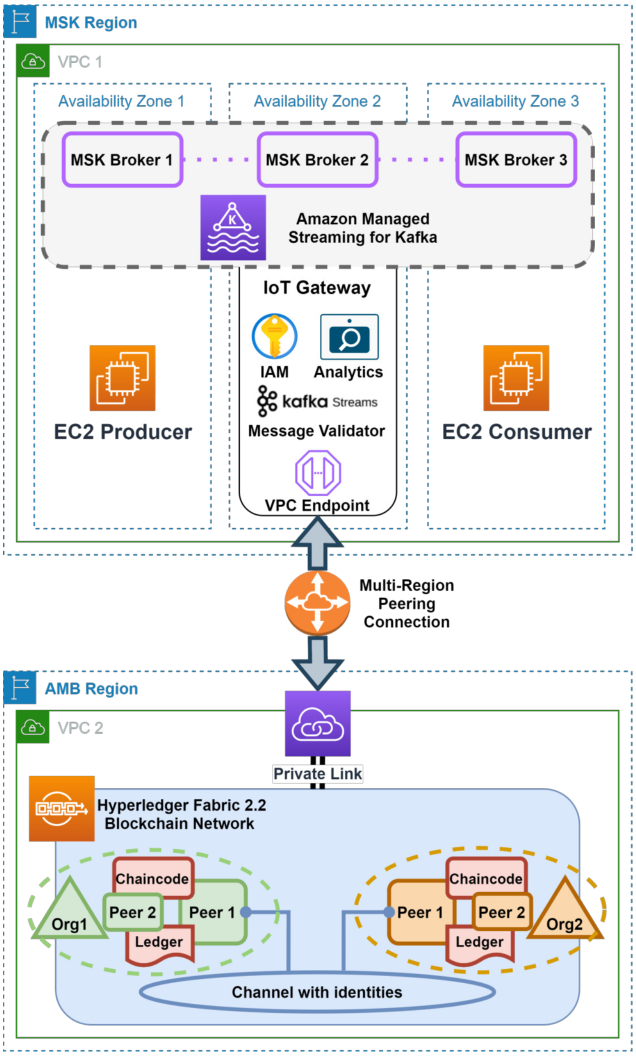


Fig. 6. Test environment overview

- two AWS regions to simulate geographical distances: MSK Region that belongs to Org1, and AMB Region for Org1 and Org2. The Multi-Region link was set using VPC Peering Connection;
- a single Amazon Managed Streaming for Apache Kafka version 2.8.1, deployed in the MSK Region isolated (availability) zones, where three kafka.t3.small brokers (vCPU: 2, Memory: 2GiB, Network Bandwidth: 5Gbps) were set. Each broker has a default configuration with a single partition and a replication factor of 3;
- a single Amazon Managed Blockchain Starter Edition for Hyperledger Fabric version 2.2, deployed in the AMB Region, where a single channel for identities was created within the blockchain network. Also, each member (Org1, Org2) of the channel has two nodes (peers) running of type bc.t3.small (vCPU: 2, Memory: 2GiB, Network Bandwidth: 5Gbps);
- two Amazon Elastic Compute Cloud (EC2) virtual machines of type t2.micro (vCPU: 1, Memory: 1GiB), deployed in the MSK Region in two isolated zones to simulate message producer and consumer;
- a single EC2 instance (t2.micro) deployed in the MSK Region, running as a microservice that verifies messages;
- a single AWS PrivateLink interface (VPC Endpoint) that enables the communication between the microservice and elements of the Hyperledger Fabric.

The AWS cloud due to its pay-as-you-go model and pluggable architecture for Commercial Off-The-Shelf services: Apache Kafka (Amazon Managed Streaming) and Hyperledger Fabric (Amazon Managed Blockchain), enables efficient deployment of our framework. Simultaneously minimizing the operational costs associated with the provisioning, configuration, and maintenance of its various components. As a consequence, our framework is suitable for federated environments for which it is required to ensure zero-day interoperability.

B. *Processing scenarios*

In conducting performance studies (benchmarks) of streaming data processing systems, it is necessary to consider three main metrics [26], [27]: latency, throughput, and the usage of hardware-software resources (CPU, RAM). Furthermore, the overall performance evaluation can be affected by the input parameters (e.g., system configuration) and processing scenarios (workloads) [25]. In the context of the proposed framework, several parameters are listed below:

- parallelization of stream processors (microservices);
- the kind (e.g., join, windowed aggregation) and type of operations (e.g., stateless, stateful);
- configuration for Kafka brokers: number of brokers, partitions, and replication factor [25];
- number of organizations that joined a federated environment;
- the number of nodes of the distributed ledger, and registered devices (identity count);
- the selected programming language for microservices and chaincodes (e.g., Java, Go, Node.js).

Generally, latency defines as the interval of time it takes for a system under test (SUT) to process a message, calculated from the moment the input message is read from the source until the output message is written by SUT. Hence, it is important to distinguish the latency metric [26] into its two types: *event-time latency* and *processing-time latency*. The first mentioned refers to the interval between a timestamp assigned to the input message by the source (e.g., broker) and the time the SUT generates an output message. The second one refers to the interval calculated between the time when an input message is ingested (read) by the SUT, and the time the SUT generates an output message. In this paper, we only prepared and conducted two workload scenarios to test the performance of the proposed processing logic for microservices, where the processing-time latency was measured:

1) Scenario I: involved verifying the input (sealed) message by performing a comparison operation between

TABLE I
PROCESSING-TIME LATENCY METRICS (IN MILLISECONDS)

Identity Count	Avg	Avg Dev	Min	Max	Pop Std Dev	Percentiles [p=0.9; p=0.95; p=0.99]
10000	38.3 (0.76)	3.0 (0.00)	32.7 (0.56)	133.8 (7.80)	5.5 (0.70)	[44.3 (0.76); 47.7 (1.24); 56.7 (2.16)]
20000	39.5 (1.20)	4.0 (0.00)	32.2 (1.28)	143.7 (9.90)	6.6 (0.60)	[46.7 (1.90); 50.7 (1.70); 61.4 (2.00)]
35000	39.9 (1.12)	3.7 (0.42)	32.6 (1.08)	131.2 (5.64)	5.9 (0.54)	[46.5 (1.10); 50.5 (1.30); 60.2 (2.16)]
50000	38.2 (1.28)	3.3 (0.42)	31.7 (1.3)	130.8 (7.76)	5.5 (0.60)	[44.6 (1.72); 47.8 (2.00); 55.7 (2.70)]
100000	38.6 (0.72)	3.3 (0.42)	32.0 (0.40)	139.3 (5.16)	5.5 (0.50)	[44.7 (0.56); 47.8 (0.64); 55.8 (1.64)]

TABLE II
TIME OF DATASTREAM (1000 MSG) RETRIEVAL BY CONSUMER (IN SECONDS)

	Identity Count				
	10000	20000	35000	50000	100000
Scenario I	38.86 (0.64)	39.95 (1.39)	40.19 (0.98)	38.59 (1.22)	39.15 (0.61)
Scenario II	20.42 (0.77)	21.03 (1.20)	21.15 (1.06)	20.39 (1.12)	20.06 (0.69)

the extracted device identity, with the identity stored in the distributed ledger.

2) Scenario II: involved verifying the sealed message by performing a comparison operation between the extracted device identity, with the identity stored in the off-chain data store. For this scenario, all device identities from the distributed ledger were also stored in the off-chain data store.

For all scenarios, the burst at startup technique was applied [27], where each input message was beforehand generated and sealed with a pseudo-randomly device identity. Once a certain number of input messages were generated, a single instance of the microservice responsible for verifying them was invoked. At the same time, within the scenarios, every second message pointed to an identity that was not registered in the distributed ledger. This approach was designed to minimize the impact of optimization (caching) mechanisms.

C. Discussion

Table 1 presents results for workload scenario I, where we determined the processing-time metric:

- average latency (Avg) and average absolute deviations of data points from their mean value (Avg Dev);
- minimum (Min) and maximum (Max) latency;
- standard deviation based on the entire population (Pop Std Dev);
- quantiles of order: p90, p95, p99.

Table 2 presents the average times for consumers to read message streams at the same time when SUT was processing it.

Both tables present the averaged results along with the average absolute deviation shown in brackets calculated for 10 repetitions of each processing scenario. Also, the input parameters for both scenarios were: a number of registered identities (identity count) in the distributed ledger and a fixed number of 1 000 messages. The optimal number of beforehand generated messages was determined through empirical tests for mentioned scenarios. During this, we noted slight increases in the accuracy of the measurements in comparison to 10 000 or 100 000 messages.

Regarding the results (Tab. 1), changing the number of registered identities did not affect the processing-time latency associated with the verification of a single message. An average delay of ~39ms was measured. The minimum delay was 31ms. In contrast, the average deviations for quantiles of the order p90 and p99 do not consecutively exceed ~2ms and ~3ms. For 100 000 registered identities, quantiles of p90 latencies were below ~45ms, and for p99 below ~56ms.

The results shown in Table 2 for workload scenario I are promising as the average time for consumers to read message streams was ~39 seconds. In the context of audiovisual streams, the measured time (1 000/39) represents ~25 frames per second (~25fps). The work of [28] demonstrated that CCTV cameras with a minimum 8fps frame rate are required to correctly identify objects on video. In addition, the results for workload scenario II confirmed the rightness of local off-chain data store usage as a mirco-caching mechanism for message verifying. The usage of mentioned data store almost doubly reduced the reading time of the message stream. An important point for scenario II is that for every second message, the identity did not exist in both data stores.

The collected results were also compared with the results in the works of [25], [26], and [27]. Hence, the rationale of the proposed processing logic for microservices was confirmed. Additionally, the obtained low average deviations indicate the stability of the proposed framework deployed with the AWS Cloud. This characteristic is important in the context of further performance studies (e.g., maximum, sustainable throughput).

D. Security and reliability risk assessment

We have conducted a high-level security risk assessment considering several security and reliability threats across the Application, Network, and Perception layers of the IoT system.

Application Layer:

1) *Storage attack* - the attack consists of changing device identity features. To prevent this attack, access to the device should be properly secured to prevent the change of data stored in it. In our framework, if the device identity is changed, the device will not be able to authenticate itself. It is almost impossible to change data in a distributed ledger without the knowledge and consent of the organization that owns the IoT device.
2) *Malicious insider attack* - the attack consists of the use of credentials by an authorized person. In the framework, access to data stored in the Hyperledger Fabric is possible only by an authenticated and authorized entity that uses an appropriate private key and a valid X.509 certificate. Each access attempt is logged. Resistance to this attack can be enhanced by using Security Information and Event Management (SIEM).
3) *Distributed ledger node failures* - in our framework, we propose each organization has a minimum of two nodes. Since the data in the blockchain is replicated, the failure of a single node does not affect the operation of the entire network.
4) *Kafka cluster (brokers) failure* - in our framework, it is possible to maintain the availability of data generated by IoT devices by using built-in synchronization mechanism and setting an appropriate replication factor.
5) *Denial of Service* – in our framework, the number of ledger nodes, Kafka brokers, microservice, and IoT gateways could be increased to handle more requests. Moreover, using SIEM we can identify specific properties of requests involved in DoS to detect a source of overload and reject all malicious requests at the gateway level.

Network Layer:

1) *Eavesdropping* - the attack involves eavesdropping on transmissions and obtaining messages (credentials). In our framework, we have separated the key used to secure the communication channel for IoT devices from the key used for the data authenticity protection mechanism. Only the registration phase is critical and must be carried out in a protected, trusted environment.
2) *Man-in-the-Middle* - the attack consists in changing the messages sent between the IoT device and the verifying microservices. Any change to the message will prevent it from being verified due to the data authenticity protection mechanism. Moreover, invalid messages can be logged to identify faulty (malicious) devices.

Perception Layer:

1) *Device capture* - the attacker can access the IoT device and generate messages sealed with its identity. In this situation, it is assumed that for such a device, its behavioral pattern (distinctive features) will change. As a consequence, it will be possible to use analytics tools (SIEM) to detect these changes, mainly related to network fingerprints. Moreover, when a compromised device is detected, it can be immediately marked and revoked from the distributed ledger. Also, hardware modules such as TPM can be used to increase device resilience against capture and manipulation.
2) *Malicious device* - the attack involves adding a fake IoT device to the network. In our structure, the process of registering a device takes place once in a protected environment. Therefore, we assume that the process will be coordinated by an authorized person. Therefore, it is not possible to register a fake device. If a device is not registered, it will not be authenticated and messages from such a device will be rejected, and consequently the device will be detected and blocked.
3) *Device tampering* - the attack consists in changing software or hardware components of the IoT device. In our framework, any changes to a unique device fingerprint would generate numerous failed verification attempts.
4) *Sybil attack* - The attack consists in having a multi-identity device by the IoT device. In our framework, this situation is prevented via a secure registration process.
5) *Side-channel (timing) attacks* - attacks consist in obtaining the key by analyzing the implementation of the protocol (e.g., current power consumption, time dependencies). The framework could be susceptible to a timing attack when the device will use unique data to seal messages. In this situation, it is possible to predict from where these data are read, but not the values of these data, therefore we believe that this attack is rather difficult to perform in practice.

VI. Conclusion and Future Work

One of the still unresolved problems is the acquisition, analysis, and fusion of enormous amounts of data generated by various IoT devices, and their secure and reliable distribution. In order to fill the gap, we proposed an experimental framework architecture for secure and reliable message stream distribution in a multi-organizational federation environment.

Deploying our framework within AWS Cloud infrastructure showed that it is suitable for environments where immediate interoperability is required. Moreover, preliminary performance benchmarking and obtained results (~25fps) confirmed the rationale for the usage of our solution to process audiovisual streams. Also, obtained low processing-time latency average deviations indicate the stability of the proposed system. This characteristic is important in the context of further performance studies, like event-time latency and maximum throughput.

Our framework has the potential to serve as the backbone for multiple applications. For instance, it could be incorporated into UAV detection and neutralization systems. This would allow both civilian and military organizations to have full control over air-defense activities, where IoT devices that are part of the smart city and military infrastructure can securely disseminate data about UAV location via our solution. Another possible scenario could involve setting up an ad-hoc system to coordinate international operations aimed at providing humanitarian assistance in eliminating natural disasters (known as HADR - Humanitarian Assistance And Disaster Relief). Our system can be of great help in this case, as it allows for the reliable exchange of data from CCTV cameras and health devices such as SOS wristbands. This would reduce response time for those in need of assistance and lead to better decision-making through improved information sharing.

Future work will focus on the development of a detailed design of a protocol for secure communication of IoT devices with a distributed registry, along with a protocol to enable message sealing that utilizes the identity of the IoT device. Additionally, we intend to conduct a comprehensive evaluation of a security and reliability risk, and we will implement dedicated software to seal messages with device identity.

REFERENCES

[1] M. Manso et al., "Connecting the Battlespace: C2 and IoT Technical Interoperability in Tactical Federated Environments". MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM), 2022, pp. 1045-1052, DOI: 10.1109/MILCOM55135.2022.10017950.

[2] F. Johnsen, M. Hauge, "Interoperable, adaptable, information exchange in NATO coalition operations", Journal of Military Studies. 11, 2022, pp.49-62, DOI: 10.2478/jms-2022-0005.

[3] H. Kopetz, W. Steiner, "Real-Time Systems: Design Principles for Distributed Embedded Applications", Springer, 2022, DOI: 10.1007/978-3-031-11992-7_13.

[4] N. Jansen et al., "NATO Core Services profiling for Hybrid Tactical Networks — Results and Recommendations," 2021 International Conference on Military Communication and Information Systems (ICMCIS), The Hague, Netherlands, 2021, pp. 1-8, DOI: 10.1109/ICMCIS52405.2021.9486415.

[5] N. Suri et al., "Experimental Evaluation of Group Communications Protocols for Tactical Data Dissemination", MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 2018, pp. 133-139, DOI: 10.1109/MILCOM.2018.8599749.

[6] Hyperledger Fabric documentation. Accessed: May. 22, 2023. [Online]. Available: https://hyperledger-fabric.readthedocs.io/

[7] Apache Kafka documentation. Accessed: May. 22, 2023. [Online]. Available: https://kafka.apache.org/

[8] Xu Wang et al., "Survey on blockchain for Internet of Things", Computer Communications 136, 2019, pp. 10-29, DOI: 10.1016/j.comcom.2019.01.006

[9] L. Ramasamy et al., "A Survey on blockchain for industrial Internet of Things", Alexandria Engineering Journal. 61., 2021, pp. 6001-6022, DOI: 10.1016/j.aej.2021.11.023.

[10] O. Alfandi et al., "A survey on boosting IoT security and privacy through blockchain", Cluster Computing. 24, 2021, pp. 37-55, DOI: 10.1007/s10586-020-03137-8.

[11] S. Guo et al., "Master-slave chain based trusted cross-domain authentication mechanism in IoT", Journal of Network and Computer Applications. 172, 2020, DOI: 10.1016/j.jnca.2020.102812.

[12] L. Xu et al., "DIoTA: Decentralized-Ledger-Based Framework for Data Authenticity Protection in IoT Systems", IEEE Network. 34, 2020, pp. 38-46, DOI: 10.1109/MNET.001.1900136.

[13] U. Khalid et al., "A decentralized lightweight blockchain-based authentication mechanism for IoT systems", Cluster Computing 23, 2020, pp. 2067–2087, DOI: 10.1007/s10586-020-03058-6

[14] A. Sivanathan et al., "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics", IEEE Transactions on Mobile Computing. 18, 2019, pp. 1745-1759, DOI: 10.1109/TMC.2018.2866249.

[15] Q. Xu et al., "Device Fingerprinting in Wireless Networks: Challenges and Opportunities", IEEE Communications Surveys & Tutorials. 18, 2016, pp. 94-104, DOI: 10.1109/COMST.2015.2476338.

[16] A. Jagannath et al., "A Comprehensive Survey on Radio Frequency (RF) Fingerprinting: Traditional Approaches, Deep Learning, and Open Challenges", 2022, DOI: 10.36227/techrxiv.17711444.

[17] M. Jarosz et al., "Formal verification of security properties of the Lightweight Authentication and Key Exchange Protocol for Federated IoT devices," 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 2022, pp. 617-625, DOI: 10.15439/2022F169.

[18] L. Sanogo et al., "Intrusion Detection System for IoT: Analysis of PSD Robustness", Sensors. 23., 2023, pp. 2353, DOI: 10.3390/s23042353.

[19] B. Chatterjee et al., "RF-PUF: Enhancing IoT Security Through Authentication of Wireless Nodes Using In-Situ Machine Learning", IEEE Internet of Things Journal. 6, 2019, pp. 388-398, DOI: 10.1109/JIOT.2018.2849324.

[20] B. Charyyev and M. H. Gunes, "IoT Traffic Flow Identification using Locality Sensitive Hashes", ICC 2020 - 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 2020, pp. 1-6, DOI: 10.1109/ICC40277.2020.9148743.

[21] C. Neumann et al., "An Empirical Study of Passive 802.11 Device Fingerprinting", 32nd International Conference on Distributed Computing Systems Workshops, Macau, China, 2012, pp. 593-602, DOI: 10.1109/ICDCSW.2012.8.

[22] F. De Rango et al., "Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks", Pervasive and Mobile Computing. 61, 2019, pp. 101105, DOI: 10.1016/j.pmcj.2019.101105.

[23] M. Yang et al., "Differentially Private Data Sharing in a Cloud Federation with Blockchain", IEEE Cloud Computing. 5, 2018, pp. 69-79, DOI: 10.1109/MCC.2018.064181122.

[24] RFC 7228: Terminology for Constrained-Node Networks Accessed: May. 22, 2023. [Online]. Available: https://www.rfc-editor.org/rfc/rfc7228.

[25] S. Kul et al., "Event-Based Microservices With Apache Kafka Streams: A Real-Time Vehicle Detection System Based on Type, Color, and Speed Attributes", IEEE Access. 9, 2021, pp. 83137-83148, DOI: 10.1109/ACCESS.2021.3085736.

[26] J. Karimov et al., "Benchmarking Distributed Stream Data Processing Systems" IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 2018, pp. 1507-1518, DOI: 10.1109/ICDE.2018.00169.

[27] G. van Dongen, D. Van den Poel, "Evaluation of Stream Processing Frameworks", IEEE Transactions on Parallel and Distributed Systems. 31, 2020, pp. 1845-1858, DOI: 10.1109/TPDS.2020.2978480.

[28] H. Keval, A. Sasse, "To catch a thief - You need at least 8 frames per second", Proceedings of the 16th ACM international conference on Multimedia, 2008, pp. 941-944, DOI: 10.1145/1459359.1459527.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 781–789
DOI: 10.15439/2023F5221
ISSN 2300-5963 ACSIS, Vol. 35

On the Applicability of the Pareto Principle to Source-Code Growth in Open Source Projects

Korneliusz Szymański
Email: korneliusz.szymanski@gmail.com

Mirosław Ochodek
0000-0002-9103-717X
Poznan University of Technology,
ul. Piotrowo 2, 60-695 Poznan, Poland
Email: miroslaw.ochodek@put.poznan.pl

***Abstract*—Context: research on understanding the laws related to software- project evolution can indirectly impact the way we design software development processes, e.g., knowing the nature of the code-repository content growth could help us improve the ways we monitor the progress of OSS software development projects and predict their future development Goal: our aim is to empirically verify a hypothesis that the OSS code repositories grow in size according to the Pareto principle. Method: we collected and curated a sample of 31,343 OSS code repositories hosted on GitHub and analyzed their content growth over time to verify whether it follows the Pareto principle. Results: we observed that, on average, monotonically growing OSS repositories reach 75% of their final content size within the first 25% revisions. Conclusions: the content size of monotonically growing OSS repositories seems to grow in size according to the Pareto principle with the 75/25 ratio.**

I. Introduction

THE 80/20 rule is often referred to as a means of quantitatively modeling cause-effect relationships between real-world variables. A generalization of this rule is a well-known Pareto principle. The principle states that *roughly 80% of outcomes come from 20% of causes*. This phenomenon has been also studied in the context of Open Source Software (OSS) development. Most of the studies investigated the principle by studying the patterns in the OSS community ways of working—i.e., commits [1], communication [2], [3], issue trackers, while others focused on the modeling the distribution of code smells, architecture, data, and software defects [4], [5].

This research focuses on studying the applicability of the Pareto principle to code repository content growth over the lifespan of OSS projects. In particular, our goal is to empirically verify a hypothesis stating that *monotonically growing OSS code repositories increase their content size over time according to the Pareto principle*, which means that, on average, an OSS code repository reaches 80% of its final content size in the first quantile of the project's lifespan. We narrow our study to projects that have a natural monotonic tendency to grow in size over time (e.g., they are not subjected to significant code removal activities).

Research on understanding the laws related to software-project evolution can indirectly impact the way we design software development processes. For instance, knowing the nature of the OSS code-repository contents growth could help us improve the ways we monitor the progress of OSS software development projects and predict their future development.

This paper is organized as follows. Section II provides more details on the Pareto principle and discussed selected studies on the Pareto principle in Software Engineering. Section III presents the design of our research method to study the applicability of the Pareto principle to the code-repository growth in OSS. Section IV presents and discusses the results and main threats to the validity of our study. The main findings are summarized in Section V.

II. Background and Related Work

A. The Pareto principle

The Pareto principle, also known as the 80/20 or 80 by 20 principle, was formulated in the early 1950s by Joseph Juran [6], but it was based on a relationship that Vilfredo Pareto had noted before—namely that 80% of the world's wealth is owned by 20% of humanity. Since then, numerous studies have shown that this principle holds also for other variables. However, this is not always an exact 80/20 ratio, the values are rather illustrative and will not apply to every situation, but in many cases, the Pareto principle works perfectly. The Pareto principle not only works in many fields, such as resource management, people management, and time management, but also in scientific fields, such as economics, accounting, medical sciences, or computer science.

The Pareto principle is also a generalization for the Pareto distribution presented in Figure 1. This principle is also characterized by the Pareto index, denoted as the alpha coefficient in the figure, which is an indicator of the Pareto principle strength.

B. The Pareto principle in Software Engineering

A large number of OSS projects available on platforms such as GitHub or SourceForge created an opportunity for researchers to study a massive corpus of software projects. Free access to code repositories based on version control systems (VCS) makes it possible to study the evolution of projects over time. This includes numerous studies on the applicability of the Pareto principle to different areas covered by Software Engineering. Most of them focused on studying the applicability of the Pareto principle to model the distribution

This work was supported by the Poznan University of Technology within the project 0311/SBAD/0738.

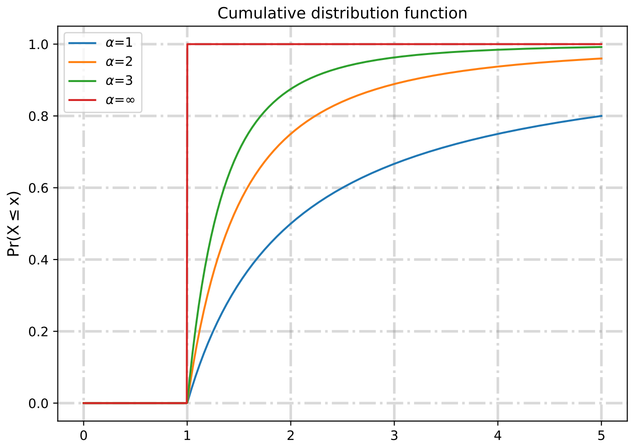


Fig. 1: Examples of cumulative density distribution functions for the Pareto distribution.

of defects, activity, or collaborators in software projects. Here, we briefly summarize the most relevant papers in this area:

- *Architecture smells and Pareto principle: A preliminary empirical exploration* [7] investigated to what extent architecture smell occurrences adhere to the Pareto principle. The authors analyzed 750 Java and 361 C# repositories and detected seven types of architecture smells. They reported that ca. 45% of the Java repositories and 66% of C# repositories followed the Pareto principle in this aspect. This study investigates the Pareto principle in open source, however, its focus is on architectural smells, while our study focuses on the size growth of OSS projects.
- *Long-term evaluation of technical debt in open-source software* [8] studied the evolution and characteristics of technical debt in OSS. In particular, the authors investigated the evolution of three large OSS Java applications (110 releases) with the use of the SonarQube technical debt detector. They reported that the Pareto principle was satisfied for the studied applications, as 20% of issue types generated around 80% of total technical debt. The focus of the study is different that ours, however, the main similarity is that both studies investigate release histories.
- *Towards a theoretical model for software growth* [9] studied 700,000 C source-code files of the FreeBSD operating system to compare different complexity measures in the context of measuring software growth. One of their observations was that all the measures followed a double Pareto distribution. This observation is convergent with the outcomes of our study, however, the original study did not consider the time-related aspect of code evolution.
- *Evidence for the Pareto principle in open source software activity* [3] focuses on analyzing the activity of users in mailing lists on a sample of three OSS software projects. This study is loosely related to our work since the studied object differs from ours.
- *On the central role of mailing lists in open source projects: an exploratory study* [2] regards communication in OSS projects and focuses on communication with the OSS projects' mailing lists. In particular, they empirically verified a hypothesis stating that a few key discussion participants are responsible for most of the messages posted on the mailing list. However, the study did not provide strong evidence confirming the applicability of the Pareto principle to this case.
- *Evaluation and application of bounded generalized Pareto analysis to fault distributions in open source software* [5] aimed at investigating distributions of faults in OSS software projects to see if it follows the Pareto distribution. Therefore, since the object of the study differs from ours, we consider it to be loosely related to our study.
- *Revisiting the applicability of the Pareto principle to core development teams in open source software projects* [1] studies the ratio of produced code size to developers' activity on a sample of 2,496 GitHub projects. This work is not directly related to our research since it focuses on the Pareto principle of code size in ratio to developers' activity and not code size growth over the project lifespan.
- *Empirical study of software quality evolution in open source projects using agile practices* [4] studies the evolution of software quality in two open source projects (Eclipse and Netbeans). They investigated the relationship between Object-Oriented metrics and defect proneness. They observed that a small percentage of compilation units hold most of the reported issues—i.e., it follows the Pareto principle.

As it follows from our literature analysis, most of the papers focus on studying the existence of the Pareto principle in different areas of software development, however, not in the context of repository content increments. Also, most of the studies focus on small samples of OSS projects and often have a form of case studies. On the contrary, our study is based on a massive dataset of OSS projects.

III. Research Methodology

A. Research goal, questions, and metrics

The goal of our study is to investigate a hypothesis that *monotonically growing OSS code repositories increase their content size over time according to the Pareto principle.* Since we aim to verify the hypothesis based on empirical evidence, we frame the problem using the Goal-Question-Metric framework [10]. Our goal (G) is to examine the applicability of the Pareto principle to OSS repository content size growth over the project lifespan. From the goal follows a research question (Q): **What is the average repository content size growth over time in OSS projects?**

Finally, we use the following metrics (M) as the basis for answering the research question:

- **M1: Repository contents size at a given point in time** — this metric presents repository contents size which is expressed by the total lines count of all files in the repository in the given revision. There is no distinction between the types of data that the files contain. Source code files, resources files, comments files, etc. are treated as contents files. We use the **wc** tool [11] to count the number of lines in the files. A point in time is determined based on the resolution defined as the M2 metric.

- **M2: Lifespan unit** defines a resolution unit of the project's lifespan. It is logically expressed as a % value but technically converted to a float in the range between 0 to 1. The lower the metric value, the higher the resolution. The base form of the Pareto principle is the 80/20 ratio. Therefore, it requires a maximum lifespan unit to be at the quintile level (20%). Based on the selected Lifespan unit, the project is sampled at unit-sized points in time, and the number of lines of code is calculated (M1).
- **M3: Repository final size** is the final size of the repository expressed in bytes. A too-small repository may not have visible differences in content growth. Conversely, a repository that is too large may indicate that it is a fixed-size dataset that does not change over time.
- **M4: Repository lifespan** is expressed as the number of commits. Repository lifespan could have a visible impact on the content growth analysis. A repository that has a too short commit history may bias the results since we analyze the distribution of the M1 measure in time. Unfortunately, in practice, some repositories may contain only a single commit. Also, this metric is related to the M2 metric. For instance, when M2 is set to 20%, the minimal repository lifespan (M4) is required to have minimum of 5 commits ($1/0.2 = 5$).

B. Study design

Answering the research question requires a multi-stage procedure, starting from collecting a large corpus of OSS project repositories, analyzing how the code changes over time in these repositories, and drawing conclusions about the applicability of the Pareto principle to the distribution of code increments.

1) Data acquisition and filtering: In the first step, we collect a dataset of projects hosted on GitHub.[1] In particular, we use GitHub API to fetch the data from the projects. The service allows the database to be filtered for repositories that match the expected criteria, but the response output is limited to 1,000 rows. The sample of repositories could be too small for our use case. Also, the quality of projects hosted on GitHub could vary visibly, starting from what we would call "engineered" projects to some toy examples, code snippets, or even repositories that do not contain any code.

Our workaround to these issues is to use a curated list of 1,857,423 GitHub repositories created by Munaiah et al. [12] (RepoReapers[2]) and further curated by Pickerill et al. [13]. The filtered list contains GitHub repositories belonging to so-called "engineered" projects. Munaiah et al. define an engineered project as *"a software project that leverages sound software engineering practices in one or more of its dimensions such as documentation, testing, and project management."* Selecting such projects allow us to narrow the search and increase the relevancy of the obtained sample of OSS projects.

In the following step, we further filter the list of repositories based on the M2, M3, and M4 metrics to select only relevant code repositories. We do not pre-set any filtering thresholds for these measures, but instead, we determine them empirically by observing how making these criteria stricter influences the dataset properties.

Finally, we take into account projects whose content growth is incremental. Therefore, we use linear regression as a tool to identify projects with anomalies related to how their size grows over time. The linear regression is expressed as:

$$y = ax + b$$

where a is a slope and b is an intercept. The slope can be used to evaluate the repository-growth tendency. We are going to determine the a threshold empirically by observing how it affects the presence/absence of outliers in our dataset. We use the PostgreSQL function regr_slope[3] to calculate the slope.

2) Data analysis: The analysis procedure is based on the Git version control tools. Git tools allow comparing changed, added, or deleted lines of code. An atomic set of changes recorded by Git is called a commit or revision. Git repositories consist of a tree of commits. Using the git-ls-tree command, the structure of the commit tree and its metadata can be obtained. The metadata contains information about changes to the file's lines. Using the wc [11] tool, it is possible to count the total amount of code at a specific point in time at which a commit was created.

The data is further normalized and aggregated to obtain the distribution of repository content growth. Each repository has a different project beginning and ending date. The duration of projects has to be expressed as normalized time. Let's define project duration normalized time as NT. NT is a discrete variable with a distribution of every threshold s (M2) which is expressed as a value from 0 which is the beginning date of a project and 1 which is the end date of a project.

$$NT = 0, s, 2s, 3s, ..., 1$$

Each project has a different content size at the end. This means a repository's content size also has to be expressed as a normalized value to a repository's final content size (M1). Let $L(p, NT)$ represents repository content size as a count of lines, where p is the repository. The ending time then is equal to 1 ($NT = 1$).

$$L_{final}(p) = L(p, 1)$$

Having the final repository content size, the normalized repository content size at a given normalized point in time can be calculated:

$$NL(p, NT) = \frac{L(p, NT)}{L_{final}(p)}$$

[1] GitHub https://github.com

[2] https://reporeapers.github.io/

[3] https://www.postgresql.org/docs/9.0/functions-aggregate.html

Having projects lifespan as normalized points in time and normalized repositories content size makes it possible to calculate the distribution of repository content growth:

$$\overline{NL}(P, NT) = \frac{1}{n}\sum_{i=1}^{n} NL(P_i, NT))$$

where P is set of repositories.

In order to ascertain the completeness of the Pareto principle in our study, the distribution of code increments should be close to the theoretical Pareto distribution. The relevant point of the distribution is the point of the searched 80/20 ratio. As a result of the study, the resulting graph may intersect the vicinity of the point, but the whole distribution may not converge to the Pareto distribution. The expected graph should be incremental with a significant increase in the first 20-30% and then a gentle increase up to 100%.

IV. Results and Discussion

A. Data fetching

We fetched code repositories from GitHub using the curated list of links to repositories belonging to so-called “engineered projects” [12], [13]. For each repository, we first downloaded the repository metadata from GitHub API. Next, each repository was downloaded and analyzed. We used the GitStats tool [14] to analyze the tree structure of Git commits. As a result, we collected data from **35,890** OSS software repositories.

The distribution of programming languages of projects is presented in Table I. This classification of projects is based on the GitHub metadata. The table presents the data for the programming languages used in over 100 projects. The distribution of the repository size for the code repositories in our sample is summarized in Table II. The average size of code repository is around 3,343 KB with a standard deviation of ca. 213 KB. The average size is influenced by very large repositories (see Figure 2), thus, the median size is visibly smaller and equal to ca. 180 KB.

TABLE I: The distribution of programming languages of projects

Language	**Projects count**
Java	8886
Python	5756
Ruby	4188
PHP	3313
C++	3134
C#	2196
C	2153
JavaScript	633
HTML	380
CSS	188
Objective-C	124

TABLE II: A summary of code repository sizes (measured in KB).

Average	Median	Standard deviation
3,343.31KB	180KB	212.95KB

B. Calculating the measures

We used the GitStats and wc tools to analyze the repositories and calculate measures M1, M2, M3, and M4. An example of the output generated by the toolset for the Spring Boot repository[4] is as follows:

```
Project name
    spring-boot
Generated
    2022-08-13 00:02:29 (in 344 seconds)
Generator
    GitStats (version 55c5c28),
    git version 2.25.1, gnuplot 5.2
    patchlevel 8
Report Period
    2012-10-21 19:53:52 to 2022-08-12 17:47:38
Age
    3583 days, 3032 active days (84.62%)
Total Files
    8748
Total Lines of Code
    727599 (1726809 added, 999210 removed)
Total Commits
    39150 (average 12.9 commits per active day,
    10.9 per all days)
Authors
    1113 (average 35.2 commits per author)
```

In addition to the report, the toolset allowed us to collect information about the growth of the code repository in time. Figure 3 presents the growth of the Spring framework repository (M1) over time. As we can see, the size (M1) of this repository increases monotonically over time, therefore, it belongs to the population of the projects in the scope of our analysis.

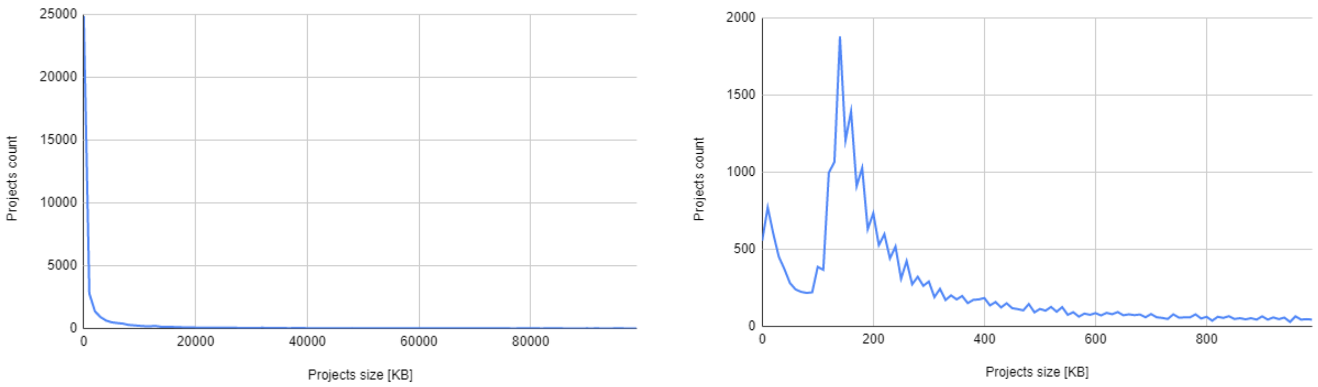
C. Applying filtering criteria

We decided to set the project lifespan unit (M2) to $s = 0.05$, which allowed us to analyze the appearance of the Pareto principle with a minimum resolution of 5%. As it was stated in Section III-A, the maximum lifespan unit for studying the Pareto principle shall not exceed 20% ($s = 0.2$). We set the threshold to 0.05 to increase the resolution since it is rather unlikely to observe the exact 80/20 ratio in a sample of real-life data, but rather some minor variation of that ratio, e.g., 75/25, 70/30, or similar ratios.

We set the following thresholds values of the M3 and M4 measures to initially filter the code repositories:

1) M3: Repository total size of up to 100 MB — we limited the size of the repository to exclude outliers (see Figure 2) and reduce the processing time of a single repository.

[4] Spring Boot – https://spring.io/projects/spring-boot



(a) All projects in the sample

(b) Projects with code repositories of up to 1,000KB

Fig. 2: The distribution of the sizes of the code repositories (KB).

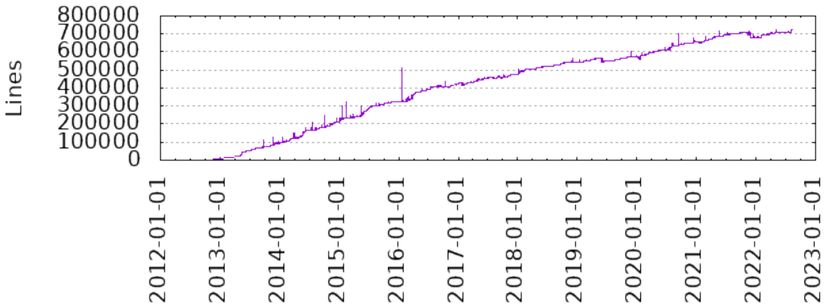


Fig. 3: An example of the repository growth plot for the Spring framework.

2) M4: Repository lifespan — we constraint the minimum repository lifespan to at least 20 commits, which follows from the selected lifespan unit (M2). A lifespan unit of 5% determines the minimum number of commits to 20.

In the following step, we used our toolset to calculate the count of lines (M1) over time (L). The results were stored in a PostgreSQL database. Next, we calculated the normalized repository content size (NL) and normalized project lifespan (NT). Finally, we used both measures to calculate the distribution of repository content growth ($\overline{NL}$).

Figure 4 shows $\overline{NL}$ for the sample of projects filtered based on the M3 and M4 measure thresholds. As the content growth is measured with respect to the final content size of the repository, the 100% content size is achieved at the last point in time. However, we can see that the average normalized size reaches up to 722% of the final size. The reason for that is the presence of outlier projects with visibly non-monotonic, non-incremental growth over time. An example of such a project could be Unity3d-Async-Task[5]:

```
Project name
    Unity3d-Async-Task
Generated
    2022-08-23 19:22:00 (in 0 seconds)
Generator
    GitStats (version 55c5c28),
    git version 2.25.1, gnuplot 5.2
    patchlevel 8
Report Period
    2015-03-07 19:24:45 to 2015-10-01 17:45:15
Age
    209 days, 21 active days (10.05%)
Total Files
    1
Total Lines of Code
    3 (48396 added, 48393 removed)
Total Commits
    58 (average 2.8 commits per active day,
    0.3 per all days)
Authors
    4 (average 14.5 commits per author)
```

As it is visible in Figure 5, the content of this repository was drastically reduced at a single point in time. As it stands from the README.md file, the source code from this repository was moved to another location. Such projects bias the general view of how the content of OSS code repositories grows over time and thus should be removed from the analysis. Unfortunately, manual filtering of such repositories was not feasible due to the size of the sample. Therefore, we decided to use linear regression (as we stated in Section III-B) to identify and remove outlying projects with respect to their

[5] Unity3d-Async – https://github.com/NVentimiglia/Unity3d-Async-Task

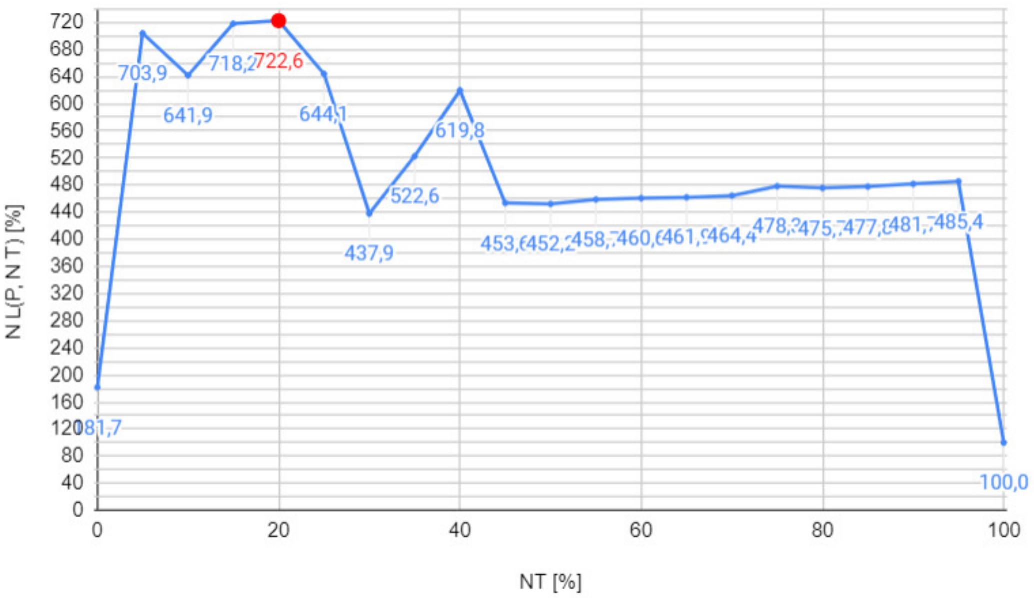


Fig. 4: Aggregated normalized lines count per normalized time expressed in percentage before filtering out the repositories with non-monotonic size growth over time.

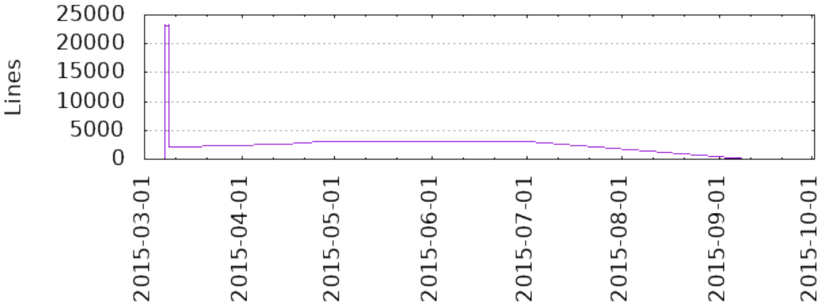


Fig. 5: Content growth in lines of code for Unity3d-Async-Task.

content growth. After applying linear regression and analyzing the slope of the regression functions, we accepted projects for which the slope was between 0 and 5. This allowed us to filter out projects with both suspiciously massive size reductions and suspicious bulk code addition operations in single commits.

The results of applying the filtering criteria are presented in Figure 4. The intensity of line-count growth became growing monotonically. Interestingly, as we can see in the figure (see $NT = 0\%$), projects, on average, are uploaded to GitHub when they contain 30% of their target size. After that, their content growth increases significantly, up to 20% of their lifespans when they reach 90% of their final size, and later, the growth rate flattens out. Unfortunately, we can also see a suspicious drop in size within the last 5% of the projects' lifespan, with a drop of 15% of their content (the pick point for a project content size is ca. 120% of its final size instead of the expected 100%). After investigation, it turned out to be, once again, the effect of outliers in the sample (e.g., moving repositories) that affected the overall picture. Therefore, we decided to introduce one more filtering criterion to identify projects with extensive code removal operations at the end of their lifespans and remove them from the sample. We applied linear regression to the last 10% of the projects' lifespans. Figure 7 shows the results of filtering the sample based on the slope of the regression function (RE). We decided to set the filtering threshold for the slope to $RE = -0.5$, however, as follows from the figure, the results were similar for all considered RE values. The final sample included **31,343** out of the 35,890 initially fetched repositories.

D. The Pareto principle

The analysis of the repository content growth for the curated sample of projects (see Figure 7) shows that, on average, monotonically growing OSS projects are uploaded to GitHub with 28.8% of their final content size. Later, their content

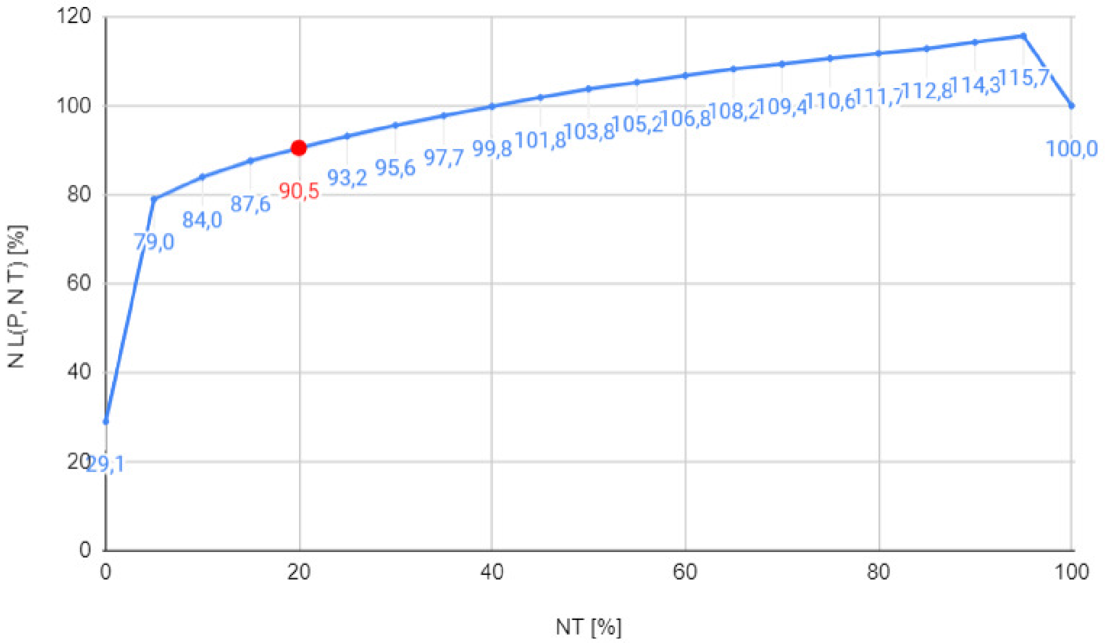


Fig. 6: Aggregated normalized lines count per normalized time expressed in percentage after filtering out the repositories with non-monotonic size growth over time.

size increases dynamically until it reaches ca. 60% of the final content size and over the remaining project lifespan, the repository content increases steadily.

The goal of our study was to investigate a hypothesis that repository content growth for monotonically growing OSS projects is subjected to the Pareto principle. If it is true, we would expect to observe that at the first quintile of the normalized project lifespan, the normalized content size of repositories should be around 80%. The ratio observed in the analyzed sample of projects is 75/25 (on average, 75% of the final repository content size was delivered within the first 25% of normalized project lifespan). Could we accept this observation as a manifestation of the Pareto principle?

As the literature on the Pareto principle suggests, the principle is not strictly bound to the 80/20 proportion. For instance, Dunford, Su, and Tamang [15] state that *"the Pareto Principle is a simplified version of the mathematics behind the Pareto distribution"* and *"the numbers 20 and 80 are not mathematically fixed, but are used as a rule of thumb"*; Boboia and Polinicencu [16] state that *"The 80/20 Rule is not a strict formula. Sometimes the cause-effect relation is closer to the 70/30 ratio than to the 80/20 ratio, but very rarely 50% of the causes lead to 50% of the results."* Therefore, taking into account the results of our study and literature regarding the Pareto principle, we conclude that **the observed 75/25 ratio fits the definition of the Pareto principle and allow us to accept the hypothesis that the content growth over time in monotonically growing OSS repositories is subjected to that principle.**

E. Threats to validity

We identified several threats to validity of our study. The internal validity threats relate to the contents of repositories in the collected sample and their potential impact on the calculated measures, in particular:

- *Extreme sizes of repositories* — our sample may contain repositories that are too small or too large. The size of the repository may indicate inappropriate project content and purpose for our research. To reject such repositories, we used the filtering criteria M3 and M4.
- *Irrelevant repository content* — many of the repositories available on GitHub are toy examples, code snippets, or do not contain any source code. To mitigate the influence of such repositories, we used the curated list of GitHub repositories that are likely to contain "engineered" projects.
- *Non-monotonically growing repositories* — the content of repositories may not be incremental. We used linear regression models to automatically filter out projects with visible deviations in content growth from what we understand to be a monotonically growing project. However, since we were not able to manually inspect each and every project, we should accept the fact that some such projects could be still present in the curated sample.

The external validity regards the possibility of generalizing the results of our study to the whole considered population. First of all, we narrowed the population to monotonically growing OSS projects which gave us better control over the

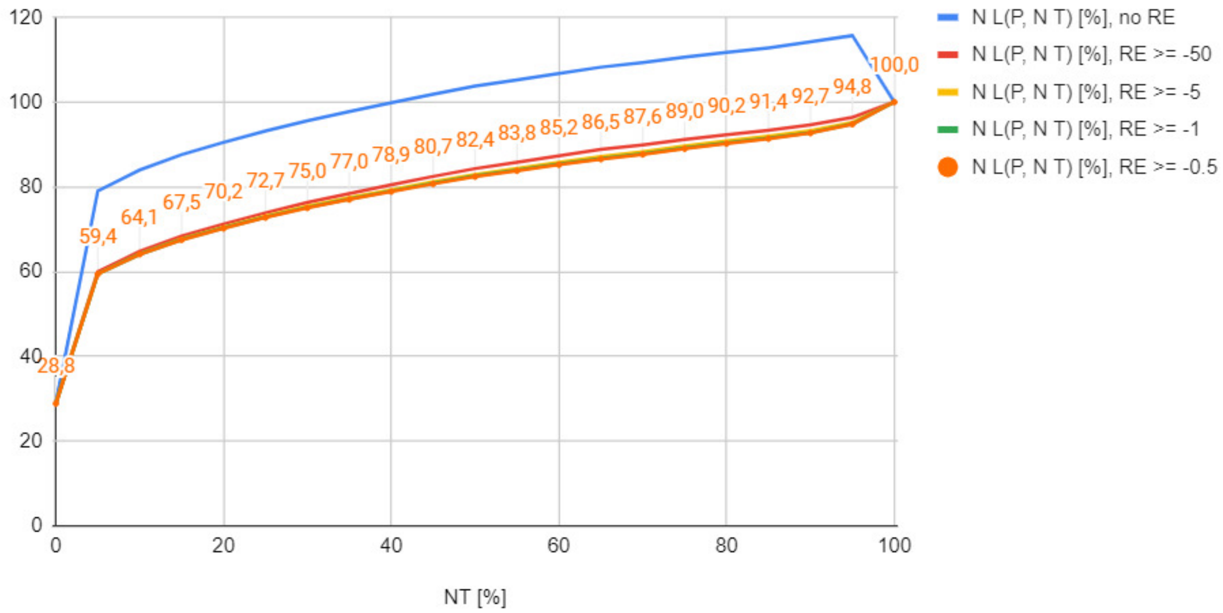


Fig. 7: Aggregated normalized lines count per normalized time expressed in percentage after filtering out the repositories with extensive code-removal actions at the end of their lifespans.

generalizability of our findings (but also narrows it). Secondly, we collected and curated a very large sample of OSS code repositories, which we believe to be a representative sample of the population under study.

V. Conclusions

We analyzed the intensity of OSS repositories' content growth over time to verify a hypothesis that the number of lines in the monotonically growing repositories increases over the project lifespan according to the Pareto principle.

We studied a sample of 31,343 OSS repositories hosted on GitHub and observed that, on average, 75% of the final content size of the repositories is produced within the first 25% revisions in such repositories. Therefore, we claim that this phenomenon is a manifestation of the Pareto principle.

We also observed that monotonically growing OSS projects are, on average, become hosted on GitHub when they already contain 30% of their final code size[6]. Therefore, in many cases, it might be impossible to retrieve the history of code changes for the first ca. 30% of the project's final size. Although studying the causes of this phenomenon was outside of the scope of this study, we suspect that this might be caused by either beginning work on an OSS project as a private project/project hosted outside of the GitHub version control system, or due to the fact that many projects are based on framework skeletons or auto-generated stubs.

The observations made in this study could help to monitor the growth of OSS projects and help to evaluate the current state of such projects.

For future research, we recommend studying the applicability of the Pareto principle to content growth depending on the various sub-criteria such as programming language, community size, etc.

References

[1] K. Yamashita, S. McIntosh, Y. Kamei, A. E. Hassan, and N. Ubayashi, "Revisiting the applicability of the pareto principle to core development teams in open source software projects," in *Proceedings of the 14th international workshop on principles of software evolution*, 2015, pp. 46–55.

[2] E. Shihab, N. Bettenburg, B. Adams, and A. E. Hassan, "On the central role of mailing lists in open source projects: An exploratory study," in *New Frontiers in Artificial Intelligence: JSAI-isAI 2009 Workshops, LENLS, JURISIN, KCSD, LLLL, Tokyo, Japan, November 19-20, 2009, Revised Selected Papers 1*. Springer, 2009, pp. 91–103.

[3] M. Goeminne and T. Mens, "Evidence for the pareto principle in open source software activity," in *First International Workshop on Model-Driven Software Migration (MDSM 2011)*, 2011, p. 74.

[4] A. Murgia, G. Concas, S. Pinna, R. Tonelli, I. Turnu *et al.*, "Empirical study of software quality evolution in open source projects using agile practices," in *Proc. of the 1st International Symposium on Emerging Trends in Software Metrics*, vol. 11, 2009.

[5] C.-Y. Huang, C.-S. Kuo, and S.-P. Luan, "Evaluation and application of bounded generalized pareto analysis to fault distributions in open source software," *IEEE Transactions on Reliability*, vol. 63, no. 1, pp. 309–319, 2013.

[6] J. M. Juran, "Pareto, Lorenz, Cournot, Bernoulli, Juran and others," in *Critical evaluations in business management*, J. C. Wood and W. M. C., Eds. Routledge, 2004, ch. 1, pp. 47–49.

[7] A.-M. Chaniotaki and T. Sharma, "Architecture smells and pareto principle: A preliminary empirical exploration," in *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*. IEEE, 2021, pp. 190–194.

[6] Please note that we refer to the final size and not to the final content of the repository.

[8] A.-J. Molnar and S. Motogna, “Long-term evaluation of technical debt in open-source software,” in *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2020, pp. 1–9.

[9] I. Herraiz, J. M. Gonzalez-Barahona, and G. Robles, “Towards a theoretical model for software growth,” in *Fourth International Workshop on Mining Software Repositories (MSR’07: ICSE Workshops 2007)*. IEEE, 2007, pp. 21–21.

[10] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach, “Goal question metric (gqm) approach,” *Encyclopedia of software engineering*, 2002.

[11] D. M. Paul Rubin, “word cound,” https://linux.die.net/man/1/wc, lines word count tool, GNU General Public License.

[12] N. Munaiah, S. Kroh, C. Cabrey, and M. Nagappan, “Reporeapers,” https://reporeapers.github.io/results/1.html, 2017, [Online; accessed 12-June-2022]".

[13] P. Pickerill, H. J. Jungen, M. Ochodek, M. Maćkowiak, and M. Staron, “Phantom: Curating github for engineered software projects using time-series clustering,” *Empirical Software Engineering*, vol. 25, no. 4, pp. 2897–2929, 2020.

[14] S. Byeon, “Gitstats, https://pypi.org/project/gitstats/.”

[15] R. Dunford, Q. Su, and E. Tamang, “The pareto principle,” *The Plymouth Student Scientist*, vol. 7, no. 1, pp. 140–148, 2014.

[16] A. Boboia and C. Polinicencu, “Application of the pareto analysis regarding the research on the value of preparations in community pharmacies from cluj-napoca, romania,” *Farmacia*, vol. 60, no. 4, pp. 578–585, 2012.

DOI: 10.15439/2023F4164
ISSN 2300-5963 ACSIS, Vol. 35

Scheduling Jobs to Minimize a Convex Function of Resource Usage

Evelin Szögi
0009-0008-5818-374X
ELKH SZTAKI
Kende str. 13-17, Budapest, 1111, Hungary
and
Department of Operations Research
Loránd Eötvös University
Email: szogi.evelin@sztaki.hu

Tamás Kis
0000-0002-2759-1264
ELKH SZTAKI
Kende str. 13-17, Budapest, 1111, Hungary
Email: kis.tamas@sztaki.hu

***Abstract*—In this paper we describe polynomial time algorithms for minimizing a separable convex function of the resource usage over time of a set of jobs with individual release dates and deadlines, and admitting a common processing time.**

I. Introduction

IN THIS paper we study variants of the following scheduling problem. There are n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$, and a common resource required by a subset of the jobs. Each job J_i has a release date r_i, a deadline d_i, and requires $\mu_i \in \{0, 1\}$ unit of the common resource. All jobs have the same processing time p. A schedule $\mathcal{S}$ specifies a starting time S_i for each job J_i, and it is *feasible*, if $r_i \le S_i \le d_i - p$ holds for each job J_i.

The goal is to find a feasible schedule $\mathcal{S}$, which minimizes a convex function $f : \mathbb{R}_{\ge 0} \to \mathbb{R}$ of the load of the resource throughout the scheduling horizon, i.e.,

$$\min_{\mathcal{S}} \int_{r_{\min}}^{d_{\max}} f(\ell^{\mathcal{S}}(t))dt,$$

where $r_{\min} = \min_i r_i$ is the earliest release date and $d_{\max} = \max_i d_i$ is the last deadline, and $\ell^{\mathcal{S}}(t)$ is the load of the resource at time point t in schedule $\mathcal{S}$, i.e., $\ell^{\mathcal{S}}(t) = |\{i \mid S_i \le t \le S_i + p, \mu_i = 1\}|$.

A variant of this problem, where $p = 1$, $\mu_i = 1$ for all $J_i \in \mathcal{J}$, and for each job J_i a subset of time slots $D_i \subset \mathbb{Z}_+$ is given, rather than an interval $[r_i, d_i]$, is known as the *load balancing problem* and it has been extensively studied by several authors, see e.g., [9], [10], [6]. As it is established in all these papers, this special case has some very nice properties: (i) there always exists a universally optimal solution $\mathcal{U}$, which is optimal for any convex function f of the load $\ell^{\mathcal{U}}$, and (ii) if a schedule $\mathcal{S}$ is *not* optimal, then there exists a pair of time slots $t_0, t_1 \in \mathbb{Z}$, such that $\ell^{\mathcal{S}}(t_0) \ge \ell^{\mathcal{S}}(t_1) + 2$ along with a subset of jobs, which can be rescheduled such that the load of t_0 decreases by one, while the load t_1 increases by one, and the load of all other time slots do not change. The convexity of f ensures that the objective function value of the new schedule is smaller than that of $\mathcal{S}$. Yet another variant is when the jobs have arbitrary integer processing times, but the preemption is allowed, i.e., the processing of any job can be interrupted and resumed later. This variant is studied in [8]. The authors have shown that the above two properties of optimal solutions are preserved. Drótos and Kis [7] study the following resource leveling problem. There is a set of m machines, a set of renewable resources, and a set of n jobs associated with release times, deadlines and resource requirements, each pre-assigned to one of the machines. The jobs have to be sequenced on the machines, while minimizing the sum of the convex functions of the loads of the resources over time. They show that if the starting times of all tasks on all but one machines are fixed, the problem is NP-hard. However, if the ordering of tasks on the remaining machine is also given, then a polynomial time algorithm exists. They also give a heuristic as well as exact branch-and-bound algorithm for solving the problem.

The above scheduling problem can be extended to parallel machine problems, where the jobs have to be assigned to machines, and the jobs assigned to the same machine have to be sequenced. The latter problem was introduced by Blazewicz [4], where all jobs have processing time $p = 1$, and require 0 or 1 unit of a common resource of capacity c. In a feasible solution the jobs are scheduled between their release dates and deadlines, and at most m jobs are processed concurrently, where m is the number of the parallel machines. Moreover, at most c jobs of resource requirement 1 are processed in parallel at any time. Blazewicz described a proprietary polynomial time algorithm for deciding whether a feasible schedule exists. This problem can be reformulated as a scheduling problem with a separable convex cost function. We define a piecewise-linear convex function f as follows: $f(x) = 0$ for $x \le c$ and $f(x) = x - c$ for $x \ge c$. It is easy to see that there is a feasible schedule in which at most c jobs with resource requirement 1 are scheduled concurrently if and only if there exists a feasible schedule of cost 0 w.r.t function f.

In this paper, we deal with two variants of the scheduling

This research has been supported by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy".

problem with non-preemptive jobs, all of processing time p:

Problem P_1. All jobs require one unit of the common resource, i.e., $\mu_i = 1$ for each job J_i, and the common processing time p is arbitrary positive integer.

We will show by way of an example that in unlike the load balancing problem with $p = 1$, for general p, there is no universally optimal solution (Section III). Furthermore, improving a non-optimal schedule may be far more complicated than in the case with unit length jobs. We will reduce the problem to a minimum cost circulation problem with convex cost functions on the arcs in an appropriately defined network, which permits the application of efficient combinatorial methods for finding optimal solutions (Section IV).

Problem P_2. Only a subset of the jobs require one unit of the common resource, and $p = 1$. In addition, there are m machines (resources of unit capacity), and each job has to be assigned to one of the machines. The jobs assigned to the same machine must be processed in non-overlapping time slots.

We describe a network-flow based method with convex cost functions on the arcs in Section V. As a by-product, our method can also answer the decision problem of [4].

II. Related work

We have already summarized the most relevant results on load balancing, resource leveling, and deadline scheduling of jobs on parallel machines using a bounded capacity resource in the introduction. In the following we focus on parallel machine scheduling problems with equal job processing times.

Brucker and Kravchenko [5] investigate the problem where equal-length jobs have to be scheduled on m identical parallel machines. For each job, a release time and a deadline is given. They present a polynomial time algorithm that finds a feasible schedule and minimizes the weighted sum of the completion times. Their method is based on an integer programming formulation of the problem, solving the linear relaxation and rounding the solution appropriately. A similar problem is considered by Kravchenko and Werner [12], but the time interval between the earliest release date and the latest deadline is divided into several smaller intervals, and for each of them, the number of available machines is given. They present a linear programming approach to find a feasible schedule that minimizes the maximum number of machines used by the jobs. Further results can be found in [2], [3], and for a survey, see [13].

III. Properties of optimal solutions

In the introduction we emphasized that the load balancing problem with common job processing time $p = 1$ admits a universally optimal solution, i.e., one which is optimal for any convex cost function. The following example shows that for $p > 1$ this is not the case by providing two different convex functions with two different unique optimal solutions.

Example III.1. *There are 9 jobs, where J_1 and J_2 have release dates $r_1 = r_2 = 0$ and deadlines $d_1 = d_2 = 5$, and J_3 through J_8, have release dates $r_3 = \cdots = r_8 = 5$ and deadlines $d_3 = \cdots = d_8 = 10$. Furthermore, job J_9 has release date $r_9 = 0$ and deadline $d_9 = 14$. The processing time of all the jobs is $p = 5$. Notice that in a feasible schedule, the place of jobs $J_1, \ldots, J_8$ is fixed: J_1 and J_2 are processed from 0 to 5, and $J_3, \ldots, J_8$ are processed from 5 to 10. The feasible schedules differ only in the starting time of J_9. Firstly, we want to minimize the function $f_1(x) = x^2$ of the load. It is not difficult to check that in the optimal solution, J_9 is processed in the interval $[9, 14]$. Let S_1 denote this solution, and the cost of S_1 is $5 \cdot 2^2 + 4 \cdot 6^2 + 7^2 + 4 \cdot 1^2 = 217$. We get a feasible, but not optimal solution by processing J_9 in $[0, 5]$. Let S_2 denote the solution we get in this way. The cost of S_2 with respect to f_1 is $5 \cdot 3^2 + 5 \cdot 6^2 = 225 > 217$, and indeed, S_2 is not an optimal solution w.r.t. f_1. Now consider the convex function $f_2(x) = 0$ for $x \leq 6$, and $f_2(x) = x - 6$ for $x \geq 6$. It is easy to show that S_2 is the only optimal solution w.r.t. f_2.*

The example above suggests that the approaches for $p = 1$ may not be straightforwardly generalized for $p > 1$.

IV. A combinatorial approach for problem P_1

We first give a linear programming formulation of problem P_1 in Section IV-A. Then, we show that the problem can be equivalently described as a minimum-cost circulation problem in a network with piecewise-linear convex cost functions on the arcs (Sections IV-B), and how to determine an optimal solution for our scheduling problem from an optimal circulation in Section IV-C. Finally, based on these results, we propose a new combinatorial algorithm for a parallel machine scheduling problem (Section IV-D).

A. Initial problem formulation and solution method

Firstly, we introduce additional notation. Let $\mathcal{I} = \{I_1, I_2, \ldots, I_L\}$ be the set of all different time slots of length p in

$$\bigcup_{i=1}^{n} \left(\{[r_i + kp, r_i + kp + p] \mid k \in \mathbb{Z},\ r_{\min} \leq r_i + kp \leq d_{\max} - p\}\right.$$
$$\left.\cup \{[d_i + kp, d_i + kp + p] \mid k \in \mathbb{Z},\ r_{\min} \leq d_i + kp \leq d_{\max} - p\}\right).$$

The following lemma states an important property about the structure of an optimal schedule, and it generalizes Lemma 3 of [2].

Lemma 1. *There is an optimal schedule, where each job is processed in one of the intervals in $\mathcal{I}$.*

Proof. Suppose the statement of the lemma does not hold for a problem instance with jobs $\mathcal{J}$ and processing time p. Let $\mathcal{S}^*$ be an optimal schedule such that the number of jobs which are not scheduled in some time slot in $\mathcal{I}$ is minimal. Let H be the subset of all those jobs that are not scheduled in some time slot in $\mathcal{I}$ by $\mathcal{S}^*$.

Let $\delta \in \mathbb{R}^{n+1}$ be a vector representing the load of the resource in $\mathcal{S}^*$, that is, for $\ell \in \{0, \ldots, n\}$, δ_ℓ equals the total size of time intervals in which the load of the resource is ℓ. Clearly, $\sum_{\ell=0}^{n} \ell \cdot \delta_\ell = n \cdot p$, and the cost of $\mathcal{S}^*$ is $\sum_{\ell=0}^{n} f(\ell)\delta_\ell$.

Let $\epsilon > 0$ be the smallest value such that starting all the jobs in H by ϵ time earlier, or later, at least one of the jobs in

H is scheduled in a time slot $I \in \mathcal{I}$. Let $\mathcal{S}_1$ be the resulting schedule. Such a shift induces a vector $\mu \in \mathbb{R}^{n+1}$ such that $\delta + \mu$ represents the load of the resource in schedule $\mathcal{S}_1$.

The cost of the schedules $\mathcal{S}^*$ and $\mathcal{S}_1$ are related by

$$cost(\mathcal{S}_1) = cost(\mathcal{S}^*) + \Delta,$$

where $\Delta = \sum_{\ell=0}^{n} f(\ell)\mu_\ell$.

Since $\sum_{\ell=0}^{n} \ell \cdot \mu_\ell = 0$ must hold, $\delta - \mu$ also represents the resource usage of some feasible schedule $\mathcal{S}_2$, namely, the one obtained by shifting all jobs in H in the opposite direction by ϵ. The cost of $\mathcal{S}^*$ and $\mathcal{S}_2$ are related by

$$cost(\mathcal{S}_2) = cost(\mathcal{S}^*) - \Delta.$$

Unless $\Delta = 0$, this implies that $\mathcal{S}^*$ is not optimal. Hence, $\Delta = 0$, and $\mathcal{S}_1$ is an optimal schedule in which some job in H is scheduled in a time slot in $\mathcal{I}$, which contradicts the choice of $\mathcal{S}^*$. □

Let $\mathcal{C} = \{c_0, c_1, \dots, c_H\}$ be the ordered set of all different left and right endpoints of $I_1, I_2, \dots, I_L$ such that $c_0 = r_{\min}$ and $c_H = d_{\max}$. Let K_h denote the interval $[c_{h-1}, c_h]$ for $h = 1, \dots, H$.

For any subset X of the jobs, let $N(X)$ consist of all those time slots $I_k \in \mathcal{I}$, which are feasible for at least one of the jobs in X, that is, $N(X) = \{I_k \mid I_k \subseteq [r_i, d_i]$ for some $J_i \in X\}$. We say that $N(X)$ is *connected* if the time slots in $N(X)$ are consecutive. Let $\mathcal{X}$ denote those subsets X of $\mathcal{J}$ such that $N(X)$ is connected.

In our formulation we have the following three types of variables:

- x_k: the number of jobs processed in time slot I_k;
- b_X: the number of jobs scheduled in the time slots $N(X)$;
- t_h: the number of jobs that require the resource in K_h, that is, $t_h = \sum_{I_k \supset K_h} x_k$.

Consider the following mathematical program:

$$\text{minimize} \sum_{h=1}^{H} |K_h| f(t_h)$$

$$\begin{aligned}
&\text{s.t.} && \sum_{I_k \in N(X)} x_k - b_X = 0, \quad \text{for all } X \in \mathcal{X} && (1)\\
& && b_X \geq |X|, \quad \text{for all } X \in \mathcal{X} && (2)\\
(IP): & && b_{\mathcal{J}} = n && (3)\\
& && \sum_{k: I_k \supseteq K_h} x_k - t_h = 0, \quad \text{for each interval } K_h && (4)\\
& && x_k \geq 0, \quad \text{for all } k = 1, \dots, L && (5)\\
& && x_k \in \mathbb{Z}, \quad \text{for all } k = 1, \dots, L. && (6)
\end{aligned}$$

Note that $|K_h| = c_h - c_{h-1}$, while $|X|$ denotes the cardinality of X. In order to show that (IP) is a proper formulation for problem P_1, we define the bipartite graph $G_{(x,b,t)} = (V_{\mathcal{I}} \cup V_{\mathcal{J}}, E)$ for a feasible solution (x, b, t) of (IP). For each job $J \in \mathcal{J}$, $V_{\mathcal{J}}$ contains a unique node v_J, and for each time slot $I_k \in \mathcal{I}$, $V_{\mathcal{I}}$ contains x_k nodes $v_k^{(1)}, \dots, v_k^{(x_k)}$ corresponding to I_k. For each job J_i and $I_k \subset [r_i, d_i]$, E contains the edge $(v_{J_i}, v_k^{(l)})$ for $l = 1, \dots, x_k$. For any $X \subseteq \mathcal{J}$, let V_X denote the set of nodes $\{v_J \mid J \in X\}$, and $N(V_X)$ the set of time slot nodes adjacent to any node in V_X in $G_{(x,b,t)}$. Then, constraints in (1) and (2) ensure that for any subset of jobs $X \in \mathcal{X}$, $N(V_X) \geq |V_X|$. The following result shows that this condition holds for all nonempty subsets of the nodes, not only for those in $\mathcal{X}$.

Lemma 2. *Let (x, b, t) be a feasible solution to (IP). Then, for every nonempty subset X of the jobs $\mathcal{J}$, $|N(V_X)| \geq |V_X|$.*

Proof. Let $X \subseteq \mathcal{J}$ be arbitrary subset of the jobs and let V_X denote the corresponding subset of nodes in $V_{\mathcal{J}}$. Let $N(V_X) \subseteq V_{\mathcal{I}}$ denote the nodes that are adjacent to at least one node in V_X. Then $N(V_X)$ can be partitioned into $N(V_{X_1}), N(V_{X_2}), \dots, N(V_{X_r})$ such that $X = X_1 \cup X_2 \dots \cup X_r$, the X_l are disjoint and $N(X_l)$ is connected for each $l = 1, \dots, r$. Since (x, b, t) is a feasible solution to (IP), $|N(V_{X_i})| \geq |V_{X_i}|$ holds for all $i = 1, 2, \dots r$ and $|N(V_X)| \geq |V_X|$ follows. □

Constraint (3) ensures $|V_{\mathcal{J}}| = n$, therefore, we have:

Lemma 3. *Let (x, b, t) be a feasible solution to (IP). Then $G_{(x,b,t)}$ admits a perfect matching.*

Proof. It follows from Lemma 2 and from the well-known theorem of Hall (see e.g. [1]). □

The following result shows how to map feasible solutions of (IP) to feasible schedules of the same cost and vice versa.

Lemma 4. *For any feasible solution (x, b, t) of (IP), there is a feasible schedule, where x_k jobs are processed in time slot I_k and the load of the interval K_h is t_h. Conversely, from every feasible schedule, where the jobs are scheduled in the time slots of $\mathcal{I}$, one can obtain a solution (x, b, t) of the same cost satisfying (1) - (6).*

Proof. To prove the first part of the lemma, suppose (x, b, t) satisfies (1) - (6). Then by Lemma 3, $G_{(x,b,t)}$ admits a perfect matching M. If $(v_J, v_I) \in M$, schedule job J in time slot I. Then for any $I_k \in \mathcal{I}$, the number of jobs processed in I_k is the number of nodes in $V_{\mathcal{I}}$ representing I_k which is exactly x_k. t_h represent the load of interval K_h in the solution by eq. (4), therefore, the load of K_h is t_h in the schedule.

To show the second part, let $\mathcal{S}$ denote a feasible schedule. For all time slots $I_k \in \mathcal{I}$, let x_k denote the number of jobs processed in I_k, and for each interval K_h, $h = 1, \dots, H$, let t_h denote the number of jobs that are executed during K_h. For an arbitrary $X \in \mathcal{X}$, let b_X denote the total number of jobs that are processed in the time slots of $N(X)$. Then (x, b, t) satisfies (1) - (6) and it has the same cost as $\mathcal{S}$. □

The following statement shows how to map optimal solutions (x, b, t) to perfect matchings in $G_{(x,b,t)}$, and vice versa.

Proposition 1. *If (x, b, t) is an optimal solution to (IP), then any perfect matching in $G_{(x,b,t)}$ corresponds to an optimal*

Algorithm 1 Calculation of optimal schedule

```
Require: $n \geq 0$, $p \geq 1$, $\{r_i, d_i\}$ for $i = 1, \ldots, n$, function $f$
Ensure: Optimal schedule $\mathcal{S}$;
1: Solve $(IP)$, and let $(x, b, t)$ be an optimal solution;
2: Define the graph $G_{(x,b,t)}$, and find a perfect matching $M$ in it;
3: Construct schedule $\mathcal{S}$ by assigning the jobs to the time slots as specified by $M$;
```

schedule. Conversely, any optimal schedule $\mathcal{S}$ induces an optimal solution (x, b, t) to (IP).

Proof. Follows easily from Lemma 3 and Lemma 4. □

By Proposition 1, we can solve the scheduling problem by Algorithm 1: Since finding a perfect matching in a bipartite graph can be done in polynomial time [1], it remains to solve (IP) efficiently. In the remainder of this section, we sketch our approach for solving (IP) in polynomial time by a combinatorial method based on network flows.

Firstly, we observe that the size of (IP) can be polinomially bounded in the size of the input.

Lemma 5. *The size of the linear system (1)-(6) is polynomial in the size of the input.*

Proof. To prove that the size of system (1)-(6) is polynomial in the size of the input, it is enough to show that the number of constraints in (2) and (4) is polynomial in the input size. Observe that $L = |\mathcal{I}|$ is $\mathcal{O}(n^2)$. All $X \in \mathcal{X}$ can be obtained the following way. One can construct $\mathcal{O}(L^2)$ possible connected $N(X)$ sets of time slots by determining the earliest and latest time slot. Then it remains to check whether there is an $X \subseteq \mathcal{X}$ such that $N(X)$ is exactly the set of time slots feasible for at least one jobs in X. Therefore $|\mathcal{X}|$ is polynomial in L. There are H intervals K_h and the endpoints of each of them coincide with endpoints of time slots in $\mathcal{I}$, therefore $H = \mathcal{O}(L)$ and the number of constraints in (4) is polynomial in the input size. □

Since we are only interested in the values of f at integer loads only, we can replace f with a piecewise linear convex function $\tilde{f}$ with integer break points, where $\tilde{f}(z) = f(z)$ for all $z \in \mathbb{Z}_{\geq 0}$. Furthermore, one can assume that the first break point of $\tilde{f}$ is at 0 and the last one is at most n, since there are n jobs. It is not difficult to see that for such an $\tilde{f}$, the optimal value of (IP) coincides with the optimal value of

$$(IPWL): \qquad \text{minimize} \sum_h |K_h| \tilde{f}(t_h) \quad \text{s.t. } (1) - (6).$$

In fact, the matrix of (1) - (6) is totally unimodular, see Lemma 6, which permits to get rid of the integrality condition (6) by a result of Meyer. Meyer [14] has shown that an optimization problem with a separable piecewise linear convex cost function with integer break points, and a totally unimodular constraint matrix always admits and integer optimal solution. This means that in (IPWL) we can drop constraint (6), while preserving integer optimal solutions:

$$(PWL): \qquad \text{minimize} \sum_h |K_h| \tilde{f}(t_h) \quad \text{s.t. } (1) - (5).$$

Karzanov and McCormick [11] describe efficient polynomial time algorithms for minimizing a separable convex cost function over the linear space $Mx = 0$, provided M is a totally unimodular matrix. More specifically, for every coordinate x_e of x, there is a convex function $w_e : \mathbb{R} \to \mathbb{R}$. If E is the set of coordinates of x, the problem is to find x such that $Mx = 0$, while $\sum_{e \in E} w_e(x_e)$ is minimized. It is assumed that $\{x : Mx = 0\}$ contains a non-zero point and the minimization problem has a finite optimal solution. The authors have shown how to solve the problem efficiently for different classes of convex functions, assuming there is an oracle that solves the following problems:

1) given a point $r \in \mathbb{R}$, return $c_e^{\vdash}(r)$ and $c_e^{\dashv}(r)$, where $c_e^{\vdash}(r)$ and $c_e^{\dashv}(r)$ are the right and left derivatives of w_e at r. The convexity of w_e implies the existence of the right and left derivatives;
2) given a slope $s \in \mathbb{R}$, return a point r with $c_e^{\dashv}(r) \leq s \leq c_e^{\vdash}(r)$.

In the case of piecewise linear convex functions, such an oracle is easy to implement.

Karzanov and McCormick proposed two different algorithms for solving such problems: the Minimum Mean Canceling Method (MMCM) and the Cancel and Tighten algorithm. Both methods are iterative, and in general case, these algorithms involve solving linear programs in each iteration. Although the number of iterations is polynomial in the size of the input, the running time is not as impressive due to solving linear programs. However, when $\{x : Mx = 0\}$ is the space of circulations in a graph G (that is, M is the node-edge incidence matrix of G), then there is no need to solve linear programs, and each iteration of the Cancel and Tighten algorithm takes $\mathcal{O}(|E(G)| \log |V(G)|)$ time, and the total number of iterations is $\mathcal{O}(|V(G)| \log(|V(G)|C))$, where C denotes the absolutely largest finite slope. The impressive running time motivates the question whether our problem can be reformulated as a minimum cost circulation problem in a network with piecewise-linear convex cost functions on the arcs.

B. Reformulation as a circulation problem in a network

First of all, observe that in the objective function of (PWL), we have convex functions only for variables t_h, and the system has lower or upper bound constraints for variables x_k and b_X. The function corresponding to t_h is $|K_h|\tilde{f}(\cdot)$. The breakpoints of $\tilde{f}(\cdot)$ are $0, 1, \ldots, n$, and the slopes between these breakpoints are $s_l = |K_h|(\tilde{f}(l) - \tilde{f}(l-1))$ for $l = 1, \ldots, n$, noting that $s_1 \leq s_2 \leq \cdots \leq s_n$, since $\tilde{f}$ is convex (if $s_l = s_{l+1}$ then

b_l is not a real breakpoint, but it is not a problem). Then we have

$$w_h(z) = \begin{cases} |K_h|\tilde{f}(0) + s_1 z & \text{if } 0 \le z \le 1 \\ |K_h|\tilde{f}(1) + s_2(z-1) & \text{if } 1 \le z \le 2 \\ \dots & \\ |K_h|\tilde{f}(n-1) + s_n(z-n+1) & \text{if } n-1 \le z. \end{cases}$$

It is not difficult to show that we get an equivalent problem by introducing convex cost functions w_k and w_X for the variables x_k and b_X and moving the lower and upper bound constraints to w_k and w_X the following way. Let K be a sufficiently large positive number. Since we want a variable x_k to be non-negative, w_k has a breakpoint at 0 and the slope before 0 is $-K$ and the slope after 0 is 0, that is

$$w_k(z) = \begin{cases} -Kz & \text{if } z \le 0 \\ 0 & \text{if } z \ge 0. \end{cases}$$

Similarly, if $X \neq \mathcal{J}$, w_X has a breakpoint at $|X|$ and the slope before $|X|$ is $-K$ and after $|X|$ is 0, i.e.,

$$w_X(z) = \begin{cases} -K(z-|X|) & \text{if } z \le |X| \\ 0 & \text{if } z \ge |X|. \end{cases}$$

If $X = \mathcal{J}$, w_X has a breakpoint at n and the slope before n is $-K$ and after n it is K, that is

$$w_{\mathcal{J}}(z) = \begin{cases} -K(z-n) & \text{if } z \le n \\ K(z-n) & \text{if } z \ge n. \end{cases}$$

Therefore, (PWL) can be reformulated as

$$\begin{aligned} &\text{minimize} \sum_{k=1}^{L} w_k(x_k) + \sum_{X \in \mathcal{X}} w_X(b_X) + \sum_{h=1}^{H} w_h(t_h) \\ &\quad\text{s.t.} \\ (PWL2): &\sum_{I_k \in N(X)} x_k - b_X = 0 \quad \text{for all } X \in \mathcal{X}; \\ &\sum_{k: I_k \supseteq K_h} x_k - t_h = 0 \quad \text{for all intervals } K_h. \end{aligned}$$

The following lemma plays a crucial role in our method.

Lemma 6. *Let M denote the matrix of the following system:*

$$\sum_{I_k \in N(X)} x_k - b_X = 0 \quad \textit{for all } X \in \mathcal{X};$$

$$\sum_{k: I_k \supseteq K_h} x_k - t_h = 0 \quad \textit{for all intervals } K_h.$$

Then M^T is a network matrix.

Proof. Observe that M can be written as $M = (A, -I)$, where $A \in \mathbb{R}^{a \times L}$ is an interval matrix and $-I \in \mathbb{R}^{a \times a}$ is a negative identity matrix, where $a = |\mathcal{X}| + H$ is the number of constraints in (1) and (4). Therefore if y is a column of M^T, then y has some 1 entries in consecutive positions in the first L coordinates, and in the remaining a coordinates, y has a unique -1 entry. All other coordinates of y are 0. We construct a network D with a spanning tree T and show that each non-tree edge corresponds to a column of M.

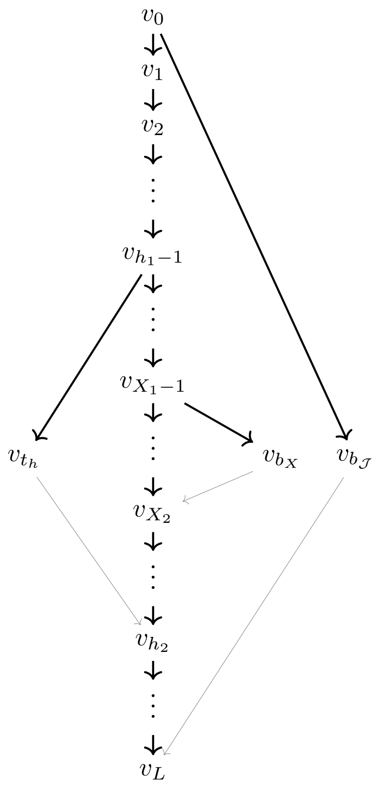
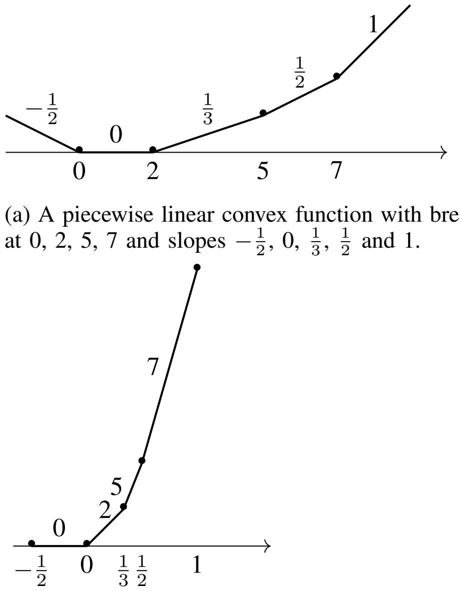


Fig. 1: Network D. Thin arcs correspond to non-tree edges, thick arcs are the edges of T.

Let $P = v_0 \to v_1 \to \cdots \to v_L$ denote a directed path of length L, where the kth edge $v_{k-1} \to v_k$ represents the kth p-length time slot I_k, and we say it corresponds to variable x_k. At the beginning, $D = P$ and $T = P$. Then we add new nodes and edges to D and T the following way: we take all constraints from (1) and (4) one by one, and for each of them, we connect nodes representing the first and the last time slot in the constraint with a new path of length 2. More precisely, consider constraints in (1) and suppose equation $\sum_{I_k \in N(X)} x_k - b_X = 0$ is one of them. We add a new node denoted by v_{b_X}. Let X_1 denote the index of the first time slot and X_2 the index of the last time slot in $N(X)$. We add edges $v_{X_1-1} \to v_{b_X}$ and $v_{b_X} \to v_{X_2}$. We extend T with the first new edge $v_{X_1-1} \to v_{b_X}$, and we say the tree edge $v_{X_1-1} \to v_{b_X}$ corresponds to variable b_X. The other new edge $v_{b_X} \to v_{X_2}$ becomes a non-tree edge. We proceed similarly with equations in (4) of the form $\sum_{k: I_k \supseteq K_h} x_k - t_h = 0$: we connect the first and the last time slot in $I_k : I_k \supseteq K_h$ with a 2-length path containing two new edges and extend T with the first new edge in the same way as before. Fig. 1 illustrates the network constructed this way. It is not difficult to check that a column in M^T corresponding to a constraint from (1) or (4) is represented by a non-tree edge in the network. □



(a) A piecewise linear convex function with breakpoints at 0, 2, 5, 7 and slopes $-\frac{1}{2}$, 0, $\frac{1}{3}$, $\frac{1}{2}$ and 1.

(b) The dual piecewise linear convex function. The breakpoints are the slopes and the slopes are the breakpoints of the function in Fig. 2a.

Fig. 2: A piecewise linear convex function and its dual.

For simplicity, tree edges in D corresponding to variables x_k, b_X and t_h are denoted by e_{x_k}, e_{b_X} and e_{t_h}. In the next part, we dualize the problem and solve the dual problem instead of the original primal formulation. To this end, we need dual variables for non-tree edges represented by the rows of M. For a non-tree edge, let z_X denote the corresponding dual variable if the edge derives from a constraint in (1), and let z_h be the dual variable if it derives from a constraint in (4). For a non-tree edge e, let C_e denote the tree edges in e's fundamental cycle.

In order to obtain the dual of $(PWL2)$, we have to determine the duals of the functions w_k, w_X and w_h, respectively. It is known (see e.g. [11]) that the dual $\tilde{f}^*(\cdot)$ of a piecewise-linear convex function $\tilde{f}(\cdot)$ is obtained by exchanging the slopes and breakpoints of $\tilde{f}$, that is, the slopes of $\tilde{f}$ will be the breakpoints of $\tilde{f}^*$ and the breakpoints of $\tilde{f}$ will be the slopes $\tilde{f}^*$, see Fig. 2 for an illustration.

The dual functions of w_X, w_k and w_h are denoted by w_X^*, w_k^* and w_h^*, respectively, and these functions have the following forms.

If $X \neq \mathcal{J}$, we have

$$w_X^*(z) = -|X|z, \qquad 0 \le z \le K,$$

and if $X = \mathcal{J}$,

$$w_X^*(z) = -nz, \qquad -K \le z \le K.$$

For w_k^*, we have

$$w_k^*(z) = 0, \qquad z \ge 0,$$

and finally w_h^* can be written as

$$w_h^*(z) = \begin{cases} 0, & \text{if } z \le s_1, \\ z - s_1, & \text{if } s_1 \le z \le s_2, \\ -s_1 + s_2 + 2(z - s_2) & \text{if } s_2 \le z \le s_3, \\ \dots & \\ -\sum_{i=1}^{r-1} s_i + (r-1)s_r + & \\ \quad r(z - s_r), & \text{if } s_r \le z \le s_{r+1}, \\ \dots & \\ -\sum_{i=1}^{n-1} s_i + (n-1)s_n + & \\ \quad n(z - s_n), & \text{if } z \ge s_n. \end{cases}$$

Using w_X^* and w_h^*, the dual of $(PWL2)$ can be concisely expressed as follows:

$$\begin{aligned} & \text{minimize} \sum_{X \in \mathcal{X}} w_X^*(z_X) + \sum_{h=1}^{H} w_h^*(-z_h) \\ & \text{s.t.} \\ & \qquad \sum_{e_{b_X}: e_{x_k} \in C_{e_{b_X}}} z_X + \sum_{e_{t_h}: e_{x_k} \in C_{e_{t_h}}} z_h + \lambda_k = 0, \\ (DP) \quad & \qquad \lambda_k \ge 0, \ \text{for all } k = 1, \dots, L \\ & \qquad 0 \le z_X \le K, \quad \text{for all } X \in \mathcal{X}, \ X \neq \mathcal{J} \\ & \qquad -K \le z_{\mathcal{J}} \le K. \end{aligned}$$

Notice that the lower bound for λ_k coincides with the left endpoint of the domain of w_k^*, and the lower and upper bounds for z_X and $z_{\mathcal{J}}$ coincide with the left and right endpoints of the domain of w_X^* and $w_{\mathcal{J}}^*$.

Since the matrix of problem $(PWL2)$ is the transpose of a network matrix, (DP) is a network circulation problem, the only problem with it is that in the objective function we have $w_h^*(-z_h)$, i.e., the negative of z_h is substituted in the convex function w_h^*. However, it is easy to overcome this issue by reversing the arcs e_{t_h}. So, our final network has the same set of nodes and arcs as D, except that the arcs e_{t_h} are directed oppositely. For an edge e_{x_k}, the lower bound for the flow value λ_k is 0 and there is no upper bound, while the cost is 0. For an edge e_{b_X}, let the cost function be the piecewise linear w_X^* and if $X \neq \mathcal{J}$, the lower bound is 0 and the upper bound is K, and when $X = \mathcal{J}$, the lower bound is $-K$ and the upper bound is K. For an edge e_{t_h}, we have no lower or upper bounds and the cost function is the piecewise linear w_h^*. There are no lower or upper bounds for the flows on non-tree edges in D, and the cost function is 0.

Proposition 2. *The minimum cost circulation problem defined above is equivalent to the problem (DP). The cost of a minimum cost circulation is equal to the optimal value of (DP).*

The minimum cost circulation problem defined above can be solved by the Cancel and Tighten method described in [11]. Remember, the number of iterations is $\mathcal{O}(|V(G)| \ \log(|V(G)|C))$, and one iteration takes

$\mathcal{O}(|E(G)| \log |V(G)|)$ time. Observe that $C = \mathcal{O}(n)$ holds in our case. It can be assumed that the length of the scheduling horizon $d_{\max} - r_{\min}$ is at most $2np$. Therefore, the number of different p-length time slots is $\mathcal{O}(n^2)$, and the number of constraints in (1) and (4) is $\mathcal{O}(n^4)$. Hence, $|V(G)| = \mathcal{O}(n^4)$ and $|E(G)| = \mathcal{O}(n^4)$ in our case.

C. Solution of the primal problem $(PWL2)$

By Proposition 2, an optimal solution to (DP) can be obtained by solving a minimum cost network circulation problem with convex cost functions on the arcs, using an algorithm of [11]. It remains to show how to determine the primal optimal solution for $(PWL2)$. Since the dual space of circulations is the space of co-circulations, the optimal primal solution is represented by an appropriate co-circulation. One can read out the following lemma from [11].

Lemma 7. *Let x_e^*, $e \in E$ denote an optimal solution to the minimum cost circulation problem. If h_e, $e \in E$ is a co-circulation satisfying $c_e^{\dashv}(x_e^*) \leq h_e \leq c_e^{\vdash}(x_e^*)$ for all $e \in E$, then h_e, $e \in E$ is an optimal solution to the dual problem, where $c_e^{\dashv}$ and $c_e^{\vdash}$ are the corresponding left and right derivatives, respectively.*

If an optimal solution x_e^*, $e \in E$ is given, then by Lemma 7, an optimal co-circulation is easy to obtain. For $e \in E$, let u_e and v_e denote the starting and ending node of e, respectively. Then finding a co-circulation satisfying $c_e^{\dashv}(x_e^*) \leq h_e \leq c_e^{\vdash}(x_e^*)$, $e \in E$ is equivalent to finding node potentials π satisfying $\pi(v_e) - \pi(u_e) \leq c_e^{\vdash}(x_e^*)$ and $\pi(u_e) - \pi(v_e) \leq -c_e^{\dashv}(x_e^*)$ for all $e \in E$. Such node potentials π can be found by a shortest path algorithm in the directed graph we get by adding edges $e \in E$ to the graph in reverse direction as well. The edge lengths are $c_e^{\vdash}(x_e^*)$ for edges with original orientation and $-c_e^{\dashv}(x_e^*)$ for edges with reverse orientation. Since $c_e^{\vdash}(x_e^*)$ and $c_e^{\dashv}(x_e^*)$ are integral values in our case, the optimal co-circulation found by a shortest path algorithm is integral as well.

D. Application to a parallel machine scheduling problem to minimize the total completion time of the jobs

In this subsection we show how to apply the previously introduced techniques to a problem investigated by Brucker and Kravchenko [5]. There are n jobs with common processing time p, each of them having a release date and deadline. In addition, there are m identical parallel machines. In a feasible schedule each job is processed between its release date and deadline on one of the machines, and at most m jobs are processed concurrently at any time. The goal is to find a feasible schedule, if one exists, that minimizes $\sum C_i$, where C_i denotes the completion time of J_i.

To begin with, we formulate the problem similarly to (IP). Recall the definitions of the set of time slots $\mathcal{I}$, and set of intervals $\{K_h\}_{h=1,\dots,H}$. Let $C(I_k)$ denote the right endpoint of $I_k \in \mathcal{I}$. Variables x_k, b_X and t_h denote the same quantities as in (IP).

$$
(IP'):\quad
\begin{aligned}
&\text{minimize} \sum_{k=1}^{L} C(I_k) x_k && (7)\\
&\text{s.t.} \sum_{I_k \in N(X)} x_k - b_X = 0, \quad \text{for all } X \in \mathcal{X} && (8)\\
&b_X \geq |X|, \quad \text{for all } X \in \mathcal{X} && (9)\\
&b_{\mathcal{J}} = n && (10)\\
&\sum_{k: I_k \supseteq K_h} x_k - t_h = 0, \quad \text{for each interval } K_h && (11)\\
&t_h \leq m, \quad \text{for all } h = 1, \dots, H && (12)\\
&x_k \geq 0, \quad \text{for all } k = 1, \dots, L && (13)\\
&x_k \in \mathbb{Z}, \quad \text{for all } k = 1, \dots, L. && (14)
\end{aligned}
$$

The objective function (7) expresses the total completion time of the jobs. The rest of the constraints are analogous to that of (IP). For a feasible solution (x, b, t), one can construct a bipartite graph $G_{(x,b,t)} = (V_{\mathcal{I}} \cup V_{\mathcal{J}}, E)$ in the same way as in Section IV-A. Analogously to Lemma 2, one can show that $G_{(x,b,t)}$ admits a perfect matching and Proposition 1 holds. Therefore, the problem can be solved efficiently if one can solve (IP') efficiently. From Lemma 5, it follows that the size of (IP') is polynomial in the size of the input, and similarly to Lemma 6, one can show that the transpose of the matrix of the system is a network matrix. Since the cost functions on the arcs are linear functions of the flows, the dual is a circulation problem with liner costs on the arcs, and a simple minimum cost flow computation finds an optimal solution.

V. A SOLUTION TO PROBLEM P_2

This section is devoted to problem P_2. We aim to schedule the jobs on m parallel machines in a way that the load is as balanced as possible. We deal only with the case, where the processing time of the all jobs is $p = 1$, but the resource requirement of the jobs can be 0 or 1. We show that this problem can be solved by a single minimum cost flow computation in a network with convex costs on the arcs in Section V-A. Then, we apply our formulation to the decision problem of [4] in Section V-B.

A. A network flow formulation to solve P_2

In order to describe a network flow representation of the scheduling problem, we define the time slots for the jobs. Let $\mathcal{I}' = \{I_1, \dots, I_{L'}\}$ be the set of all different unit-length time slots in the set

$$\bigcup_{i=1}^{n} \{[r_i + k, r_i + k + 1] \mid \forall\ k \in \mathbb{Z} \\ \text{s.t. } 0 \leq k \leq \min\{n-1, d_i - r_i - 1\}\}\,.$$

Note that for each job it suffices to consider only the first n unit-length time slots, since there are n jobs. We define the network D' as follows. For every job, there is a job node in

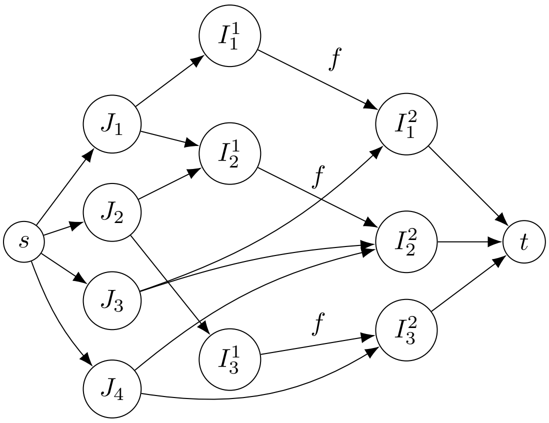


Fig. 3: Network D' for Problem P_2, where jobs J_1 and J_2 require one resource unit and jobs J_3 and J_4 require 0. The cost is measured by the convex function f on three edges, the remaining edges have zero cost.

D'. For simplicity, the job nodes are denoted by $J_1, \ldots, J_n$. For each $I_k \in \mathcal{I}'$, there are two time slot nodes in D' denoted by I_k^1 and I_k^2. Furthermore, there is a source node s and a sink node t. From s, there is an arc to every job node of capacity 1. If J_i requires 1 unit of the resource, that is, $\mu_i = 1$, then there are arcs from J_i to all the nodes I_k^1 such that I_k is feasible for J_i. When $\mu_i = 0$, there are arcs from J_i to all the nodes I_k^2, such that I_k is feasible for the job. All these arcs have infinite capacity. For all $I_k \in \mathcal{I}$, there is an arc from I_k^1 to I_k^2 of infinite capacity, and the cost function on this arc is f. The cost function on all other arcs is 0. Moreover, there is an arc from I_k^2 to t of capacity m. See Fig. 3 for an illustration.

Proposition 3. *The optimal feasible schedules are in one to one correspondence with the integral minimum cost feasible flows, where the total flow leaving s is n.*

Despite of D' having convex costs on some edges, a similar network having only linear costs can be constructed in a similar way as in [8], and the problem can be solved by any minimum cost network flow algorithm.

B. Application to a scheduling problem with a resource of bounded capacity

By choosing the convex function f properly, one can decide the feasibility problem considered by Blazewicz [4] as described in Section I. If c denotes the resource capacity, then let f denote the following piecewise linear function: $f(x) = 0$ if $x \leq c$ and $f(x) = x - c$ if $x \geq c$, where c is the capacity of the resource. Then there exists a feasible schedule using m machines, where each job is processed in a time slot between its release time and deadline if and only if there is feasible flow of zero cost in the previously constructed network D'. Notice that there is no need to solve a flow problem with arc costs. Let the network D'' be obtained from D' by removing all arc costs and setting the capacity of the arcs from I_k^1 to I_k^2 to c. Then we have the following result.

Proposition 4. *The scheduling problem of [4] with a bounded capacity resource admits a feasible solution if and only of the network D'' admits a feasible flow of value n.*

VI. Preliminary computational results

We have implemented Algorithm 1 including the Cancel and Tighten method in C++ for solving Problem P_1 on randomly generated problem instances. The goal of the test runs was to assess how sensitive is the method to two problem parameters: the number of the jobs n, and the ratio of the common processing time and the size of the time windows of the jobs, i.e, $q = p/(d_i - r_i)$. We generated three problem instances for each combination $(n, q) \in \{20, 50, 100\} \times \{0.1, 0.4\}$. The common job processing time was $p = 8$. The time horizon spanned 100 time units, and for each job J_i the release date and deadline satisfied the constraint $r_i \geq 0$, $d_i = \lceil r_i + p/q \rceil$, and $d_i \leq 100$. We used the same piecewise linear convex function f in all cases, where $f(0) = 0$, f has breakpoints at $1, 2, \ldots, n$, and the slope after breakpoint i is i.

The code was compiled with Visual Studio 2019, and the tests were run on a notebook computer with Intel Core I7 processor and Windows 11. We summarize the computational results in Table I. We provide averages (rounded to nearest integers) over 3 problem instances for each combination of the parameters n and q. In each case, we provide the average number of nodes and edges of the network D, the average number of iterations, the average CPU time, and also the average optimum values. All values are rounded to the nearest integers. As we can see, the CPU time strongly correlates with the number of graph edges. On the other hand, the number of iterations lightly increases with the number of the jobs, but for the same number of jobs, it is smaller for $q = 0.4$ than for $q = 0.1$. In fact, this is what we expected, since problem instances with a larger ratio q permit less freedom to choose the starting times of the jobs.

VII. Conclusion

In this paper we gave polynomial algorithms to two load balancing problem. A possible direction for a future research is to investigate a problem slightly more general than Problem P_2. While the common processing time in P_2 is one time unit, it is an interesting question what can be said if the common processing time is greater than one time unit. The complexity of this problem is still open. One possible next step would be to derive an approximation algorithm for the problem.

References

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows.* Prentice-Hall, Inc., New Jersey, 1993.

[2] Philippe Baptiste. Scheduling equal-length jobs on identical parallel machines. *Discrete Applied Mathematics*, 103(1):21–32, 2000.

[3] Philippe Baptiste, Peter Brucker, Sigrid Knust, and Vadim G. Timkovsky. Ten notes on equal-processing-time scheduling. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2:111–127, 2004.

[4] Jacek Błażewicz. Deadline scheduling of tasks with ready times and resource constraints. *Information Processing Letters*, 8(2):60–63, 1979.

[5] Peter Brucker and Svetlana Kravchenko. Scheduling jobs with equal processing times and time windows on identical parallel machines. *Journal of Scheduling*, 11:229–237, 08 2008.

TABLE I: Preliminary computational results for randomly generated inputs. n is the number of the jobs and q is the ratio of the job length and the size of the time windows of the jobs.

	$n = 20$, $q = 0.1$	$n = 20$, $q = 0.4$	$n = 50$, $q = 0.1$	$n = 50$, $q = 0.4$	$n = 100$, $q = 0.1$	$n = 100$, $q = 0.4$
Avg. number of graph nodes	87	86	92	89	92	92
Avg. number of graph edges	289	374	360	924	379	1785
Avg. number of iterations	1275	1102	1511	1293	1633	1493
Avg. optimal objective value	223	229	1028	1071	3685	3804
Avg. CPU time (millisec)	36	58	52	74	59	135

[6] Mihai Burcea, Wing-Kai Hon, Hsiang-Hsuan Liu, Prudence W. Wong, and David K. Yau. Scheduling for electricity cost in a smart grid. *Journal of Scheduling*, 19(6):687–699, 2016.

[7] Márton Drótos and Tamás Kis. Resource leveling in a machine environment. *European Journal of Operational Research*, 212(1):12–21, 2011.

[8] Péter Györgyi, Tamás Kis, and Evelin Szögi. A polynomial time algorithm for solving the preemptive grid-scheduling problem. *unpublished*, 2023.

[9] Bruce Hajek. Performance of global load balancing by local adjustment. *IEEE Transactions on Information Theory*, 36(6):1398–1414, 1990.

[10] Nicholas JA Harvey, Richard E Ladner, László Lovász, and Tami Tamir. Semi-matchings for bipartite graphs and load balancing. *Journal of Algorithms*, 59(1):53–78, 2006.

[11] Alexander V. Karzanov and S. Thomas McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM Journal on Computing*, 26(4):1245–1275, 1997.

[12] Svetlana Kravchenko and Frank Werner. Minimizing the number of machines for scheduling jobs with equal processing times. *European Journal of Operational Research*, 199:595–600, 12 2009.

[13] Svetlana Kravchenko and Frank Werner. Parallel machine problems with equal processing times: a survey. *Journal of Scheduling*, 14:435–444, 10 2011.

[14] R. R. Meyer. A class of nonlinear integer programs solvable by a single linear program. *SIAM J. Control Optim.*, 15(6):935–946, 1977.

DOI: 10.15439/2023F7181
ISSN 2300-5963 ACSIS, Vol. 35

Target search with an allocation of search effort to overlapping cones of observation

Hugo Vaillaud
Thales DMS
Sorbonne Université, CNRS, LIP6
Paris, France
hugo.vaillaud@fr.thalesgroup.com

Claire Hanen
Sorbonne Université, CNRS, LIP6
UPL, Université Paris Nanterre
Paris, France
claire.hanen@lip6.fr

Emmanuel Hyon
Sorbonne Université, CNRS, LIP6
UPL, Université Paris Nanterre
Paris, France
emmanuel.hyon@lip6.fr

Cyrille Enderli
Thales DMS
Elancourt, France
cyrille-jean.enderli@fr.thalesgroup.com

***Abstract*—This paper addresses the problem of an aerial moving target search with a radar on an airborne platform. An observation of the radar is modeled as a cone covering a set of regions of the search area. We assume overlapping cones of observation, and we want to find the discrete allocation plan of search effort to the cones in order to optimize target detection. For the stationary target search with overlapping cones, we present a dynamic programming algorithm that computes the optimal allocation. An approximate greedy heuristic, which is more appropriate in a real time context, is also presented and assessed. The moving target search problem is solved with the Forward And Backward (FAB) algorithm coupled with the different stationary search algorithms. In this paper, we use a radar detection model that has been shown to be more realistic than the ones usually considered. Also, several models of movement of the target are considered with different Markovian transition matrices. We compare the performance of the mentioned algorithms on several scenarios.**

***Index Terms*—Moving Target search, Dynamic Programming, FAB algorithm, Overlapping Observation Cones.**

I. Introduction

SEARCHING for a moving target with a sensor is a complex problem, especially when the search area is large. This has been the subject of much research in the past, as evidenced in literature reviews by Stone [1] and by Rapp [2].

In general, one seeks to optimize an objective related to the probability of detecting one or more moving targets. The search area is discretized into identical regions and a sensor plan is computed over a fixed horizon. All regions are associated with prior probabilities of presence of targets. The probability of detecting targets in a region depends on the amount of search effort invested in it, the visibility of the region, and the probability that targets are present in the region.

Depending on the type of the platform (drone, satellite) or sensor (radar, camera), several approaches exist (see the review in [2] and references included therein). One can try to solve a *path constrained search*, where the subset of regions that will be observed and the order of visit of these consecutive regions must be determined (see [3], [4]). One can instead try to find the quantities of effort (discrete or continuous), constrained by a budget, to allocate independently to each region, at each time step, in order to maximize the probability of detecting the targets over the time horizon considered [1], [5], [6]. The search plan can be computed using the Forward-and-Backward algorithm (FAB) [7], [5], [6], Branch and Bound methods [8], or MILP [9] among others (see [2]).

In the problem we address, the search effort cannot be allocated independently to each region. Indeed, each observation of the radar is characterized by a *cone* that covers a set of regions. Moreover we assume that the cones may overlap, so that a region can be covered by several observation cones. This model applies for 2D as well as for 3D dimensional cones. As pointed out in chapter 14 of [10], an important parameter of the search plan design is *beam search spacing*, that defines the maximum number of consecutive overlapping cones. In this paper, we investigate its role in the efficiency of search plans and algorithms. We consider here discrete search effort on each cone which means we deal with the number of observations.

In [11], the authors consider a similar context for a continuous search effort and propose a customized version of the Forward-And-Backward algorithm to solve it, without proving its optimality. Path search have been considered in [9] where several regions are observed at the same time from each position and a region can be observed from several positions. The solution proposed is a mixed integer programming method. In [12], the authors addressed the problem of a moving target search with a radar where the cones of observation are disjoint: each region is covered by a single cone of observation. They show that for continuous effort the FAB algorithm is still optimal, and adapt the FAB algorithm to the discrete effort case. None of these works address the problem of allocating a discrete search effort to overlapping cones.

Another relevant issue in target search is the detection model. Chapter 2 of [13] presents an extensive literature review on detection models and quotes that in most of the papers the visibility of a region depends only on the distance from the radar. However, one of the main characteristics of radar measurements is that the detection probability also decreases on the borders of an observation cone. Such a model was first proposed in [11]. In this paper we consider a general model where the detection probability of a single observation depends arbitrarily on the region and on the observation cone.

In this work, we address a moving target search with a radar by considering overlapping cones of observation for the allocation of discrete search effort and with a new detection model. Unfortunately, in this case the usual algorithm to optimize the probability of finding a stationary target given by [1] is not optimal. Thus, we face up a combinatorial optimisation problem to compute the optimal quantity of effort to allocate at each time step.

We present a dynamic programming algorithm that finds the optimal allocation of effort for a stationary target search case with overlapping cones of observation. Operational considerations require a short execution time of the algorithm. Therefore, we study different computational heuristics. Among them we propose a greedy algorithm which approximates efficiently the optimal solution. All stationary algorithms are then used in a FAB algorithm to solve the moving target search. We then present numerical experiments to compare the performance of these different algorithms in different scenarios. Among them we consider different models of target movement.

This paper is organized as follows. Section II details the framework of the target search. In Section III we present the dynamic programming algorithm that finds the optimal allocation for a stationary target search while Section IV is devoted to heuristics and an upper bound. We recall the moving target search problem and the FAB algorithm in part V. At last Section VI focuses on the numerical experiments.

II. Framework

The search problem that we consider is the following:

a) the target: An airborne platform faces a search area, which is partitioned into J regions. We assume a single target. The target position is unknown but we assume that we know its movement model. Similarly as in previous works, namely [1], the target moves from one region to another, at each time step $t \in \{0, \dots, T\}$, following a Markovian transition model with some transition function $\pi_t(i,j)$ defined for all pair of regions $i, j \in \{1, \dots, J\}$ and $t = 0, \dots, T$. As soon as we know the prior probability $p_0(j)$ that the target is initially located in a region j, we deduce $p_t(j)$ which is the probability that the target is in region j at time t. A trajectory is a sequence of regions $\omega = (\omega_t)_{t=0,\dots,T}$.

b) the sensor: We also assume a single sensor that observes the search area such that at each time step, it has a budget of observations that it can allocate to one or several angles. An observation made by the sensor in an angle encompasses all the regions covered by the cone defined by the angle and the position of the sensor. We define M_a as the set of regions observed in angle a (see Figure 1) and conversely by Y_j the set of angles covering region j. We say that two angles a, a' overlap if $M_a \cap M_{a'} \neq \emptyset$.

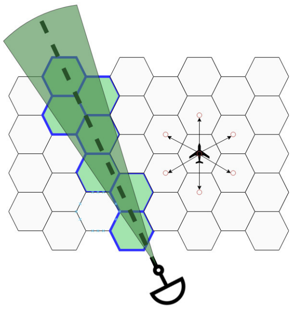


Fig. 1. A single cone of observation and the regions being observed.

We assume that the set of angles A is indexed such that each region of the search area is observable from at least one angle and from at most N consecutive angles. In Figure 2, we present three examples for $N = 1, 2, 3$. The set A is indexed from left to right. In each example, there are four cones of observation and a region indicated by a red circle. The cones that cover the region are represented in green such that an area covered by several green cones appears darker. The other cones of observation are represented in white. For $N = 1$, the search area is partitioned into disjoint cones of observation. For $N = 2$ the red circle is covered by angles 2 and 3, and for $N = 3$, the red circle is covered by angles $1, 2, 3$.

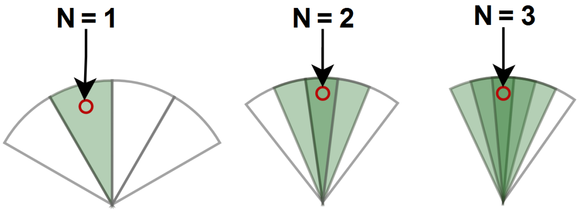


Fig. 2. Example of overlapping cones of observation.

c) the search plan: We define the search plan Z as a matrix of shape $|A| \times (T+1)$. It defines at each time step $t = 0, \dots, T$, an allocation vector $\zeta(t)$ whose components $\zeta_a(t)$ define the number of observations made in the angle a at time t for $a \in A$.

We define $\kappa(a) \in \mathbb{R}^+, \kappa(a) \geq 1$ the cost of investing an effort to the angle a. The cost of an allocation vector $\zeta(t)$ is

$$K(\zeta(t)) = \sum_{a \in A} \zeta_a(t) \cdot \kappa(a) \,.$$

At each time step, we construct an allocation vector $\zeta(t)$ such that its cost is bounded by a budget C_t.

d) The detection model: The range of a radar is not uniform in an observation cone: it is at its highest in the center of the cone and decreases on the borders. In our model, we assume that a region-angle couple (j, a) such that $a \in Y_j$ is associated with a visibility coefficient $\alpha_{j,a} \in [0,1]$ representing the conditional probability of detecting the target in the region j when the sensor makes one observation in the angle a, provided that the target is in the region j. We also assume that the observations are independent.

We call this model "*realistic*": the coefficient depends on the distance of the region to the radar and the angular offset to the center of the observation cone. We also consider in our experiments the usual model where the coefficient only depends on the distance. It is called "*distance*" model. In this case $\alpha_{j,a} = \alpha_j$ for all $a \in Y_j$. Figure 3 illustrates these two models.

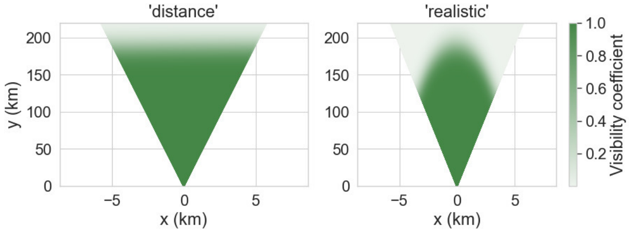


Fig. 3. Two types of detection models in a single cone of observation.

Consider a single time step t of the plan and the allocation of effort vector ζ at this time step (t is omitted in the notation). The detection function $b(j, \zeta)$ is the conditional probability of detecting the target in a region j for an effort vector ζ given that the target is located in j. It can be expressed by the following property:

Property 1 (Detection function)**.**

$$b(j, \zeta) = 1 - \prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} . \tag{1}$$

Proof. Let D_j and $\bar{D_j}$ be respectively the events associated with a detection and a non detection of the target in the region j. Let C_j be the event associated with the target located in j. By definition, The detection function in a region is defined as: $b(j, \zeta) = \mathbb{P}(D_j|C_j) = 1 - \mathbb{P}(\bar{D_j}|C_j)$.

Now, we know that for each observation of j in angle a, the probability of missing the target, given it is present, is $1-\alpha_{j,a}$.

So, if ζ_a independent observations are made in angle a, the conditional probability of missing the target in region j, given C_j is $(1 - \alpha_{j,a})^{\zeta_a}$.

Finally, as region j is covered by all angles $a \in Y_j$, and as all observations are independent, the probability $\mathbb{P}(\bar{D_j}|C_j)$ of missing the target, given it is in region j, is $\prod_{a \in Y_j}(1-\alpha_{j,a})^{\zeta_a}$ so that the property hold.

□

When we use the "distance" model, since the value of $\alpha_{j,a}$ is the same for any a in Y_j, then the formula is simplified to

$$b(j, \zeta(t)) = 1 - (1 - \alpha_j)^{\sum_{a \in Y_j} \zeta_a(t)} . \tag{2}$$

e) the objective function: Let $P_t(Z, \omega)$ be the conditional probability of detecting the target before the time horizon t given the target follows the trajectory ω and the allocation plan is Z. $P_t(Z, \omega)$ is defined as follows:

$$P_t(Z, \omega) = 1 - \prod_{s=0}^{t} \big(1 - b(\omega_s, \zeta(s))\big) . \tag{3}$$

As in [1], we consider a generic objective function composed of any linear combination of such probabilities:

$$\hat{P}_T(Z) = \mathbb{E}\left[\gamma_{T+1}(\omega) + \sum_{t=0}^{T} \gamma_t(\omega) \cdot P_t(Z, \omega)\right] . \tag{4}$$

Two remarkable objectives can be formulated by adjusting the coefficients γ_t:

- The maximization of the probability $P_T(Z)$ of detecting the target before the time horizon T by executing the allocation plan Z: ($\gamma_t = 0$ for $t = 0, \ldots, T-1$, $\gamma_T = 1$ and $\gamma_{T+1} = 0$).
- The minimization of the mean completion time $M_T(Z)$ (i.e. the expected number of time steps until the target is detected or the time horizon T is reached). By setting $\gamma_t = 1$ for $t = 0, \ldots, T$ and $\gamma_{T+1} = -(T+1)$, we get the objective $-M_T(Z)$ to be maximized.

The global objective of the planning algorithm is thus to find the feasible plan (satisfying the budget constraint) Z^* that maximizes $\hat{P}_T(Z)$.

III. Optimal stationary target search

The Forward and Backward algorithm decomposes the moving target search problem into successive resolutions of stationary target search problems at each time step. Hence, we first consider the problem of stationary target search. The aim is to minimize the probability of missing the target at a single time step t, given the probabilities of the target presence in every region, and a budget C for the observations.

For the sake of clarity, in this section we do not use the time step t in our notations and $\zeta_a(t)$ is denoted by ζ_a. We present a dynamic programming algorithm to compute the optimal allocation of effort to angles bounded by a budget C for the search of a stationary target when the cones of observation overlap.

The stationary target search problem can be formulated as a convex mathematical program:

$$\min \quad 1 - P(\zeta) = \sum_{j=1}^{J} p(j) \cdot \prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} \tag{5}$$
$$\text{s.t.} \sum_{a \in A} \kappa(a) \cdot \zeta_a \leq C \text{ and } \zeta_a \in \mathbb{N}, \forall a \in A .$$

To our knowledge, the complexity of this problem is unknown, even in the simpler case considered in [1] where the effort applies independently to regions and assuming unit costs. The complexity of the algorithm presented in [1] depends linearly on the maximal number of observations, which is constrained by the budget, and thus is pseudo-polynomial. However from a practical point of view, in the operational instances the maximum number of observations is usually far lower than the number of regions, or cones in our case which makes the algorithm polynomial.

A. Properties

The algorithm uses the fact that at most N consecutive cones of observation overlap (as illustrated in Figure 2) to decompose the problem into elementary subproblems and computes the optimal solution iteratively by considering consecutive angles.

We consider the tuple $(a_k)_{k=1,\ldots,|A|}$ of the angles of A in increasing order. For each phase $k \in \{1, \ldots, |A|\}$ of the algorithm, we define the set of regions R_k that are observable from angle a_k but not from any angle $a_{k'}$ with $k' > k$.

For example in Figure 2, when $N = 2$, there are 4 cones of observations associated to the angles $(a_k)_{k=1,\ldots,4}$. The red circle is covered by the cones a_2 and a_3 but not by a_4. The circle belongs to R_3. Note that in an operational instances the number of angles is far greater than 4.

Property 2. *The sets $(R_k)_{k=1,\ldots,|A|}$ form a partition of the set of regions.*

Note that, since the R_k are disjoint, Property 2 implies that the probability of detecting the target $P(\zeta)$ (given by Eq. (5)) can be computed by summing the probabilities of detecting the target in each R_k, for $k = 1, \ldots, |A|$.

Let us denote by $n_k = \min(N, k)$ the maximum number of angles that can observe a region in R_k. Notice that for $k = |A|$, R_k contains all regions observable by angle a_k, so there might be a region in R_k covered by N angles.

Property 3. *A cone M_a intersects R_k if and only if $a \in \{a_{k-n_k+1}, \ldots, a_k\}$.*

Property 3 implies that the probability of missing the target in R_k can be determined using the allocation of effort on $\{a_{k-n_k+1}, \ldots, a_k\}$, which can be expressed as a list of n_k values: $\zeta_{a_k}, \ldots, \zeta_{a_{k-n_k+1}}$ for $k = 1, \ldots, |A|$.

We define $W_k(\zeta_{a_k}, \ldots, \zeta_{a_{k-n_k+1}})$ the probability of missing the target in R_k as:

$$W_k(\zeta_{a_k}, \ldots, \zeta_{a_{k-n_k+1}}) = \sum_{j \in R_k} p(j) \prod_{l=k-n_k+1}^{k} (1 - \alpha_{j,a_l})^{\zeta_{a_l}} . \tag{6}$$

Let us define the partial objective of order k as:

$$Part_k(\zeta) = \sum_{l=1}^{k} W_l(\zeta_{a_l}, \ldots, \zeta_{a_{l-n_l+1}}) . \tag{7}$$

The partial objective $Part_k(\zeta)$ is thus the part of the global objective (5) that excludes the part of the area covered by the cones of the angles $a_{k+1}, \ldots, a_{|A|}$. Notice that our objective is then:

$$1 - P(\zeta) = Part_{|A|}(\zeta) \tag{8}$$

B. The algorithm

We now define the dynamic programming scheme to minimize $Part_{|A|}(\zeta)$, where each phase k corresponds to the decision concerning the effort ζ_{a_k}. The $n_k - 1$ past decisions at phase k are summarized into a state $E = (B, x)$ where:

- B is an integer between 0 and C representing the remaining budget available for the decisions of the phases k to $|A|$;
- x is a list of $n_k - 1$ elements representing the last decisions made in reverse chronological order (thus concerning the angles $a_{k-1}, \ldots, a_{k-n_k+1}$).

In this context, the initial state is $(C, null)$ (where $null$ represents an empty list). If x is a list of values we denote by $x[p]$ the p^{th} element of the list, and by $x[p \ldots q]$ the sub list of x with elements indexed from p to q. Moreover if u is an element or another list we denote by $x \cdot u$ the concatenation of x and u.

The transitions of the dynamic programming scheme are defined as follows:

If at phase k the system is in state $E = (B, x)$, the decision δ (effort on angle a_k) satisfies $\delta \in \{0, \ldots, \left\lfloor \frac{B}{\kappa(a_k)} \right\rfloor\}$. For such a decision δ, we define $x' = \delta \cdot x[1 \ldots n_k - 2]$, $\delta' = x[n_k - 1]$, and $B' = B - \delta\kappa(a_k)$. The pair (B', x') is the resulting state of the decision δ at phase $k+1$. Notice that $x = x'[2 \ldots n_k - 1] \cdot \delta'$, and $B = B' + x'[1]\kappa(a_k)$. This will be used in Property 4.

The immediate cost of the decision δ is then $W_k(\delta \cdot x)$. Using the previous notations, we observe that $W_k(\delta \cdot x) = W_k(x' \cdot \delta')$.

Let $F_k(B, x)$ be the minimal value of $Part_{k-1}(\zeta)$ where ζ is subject to the two constraints:

$$\sum_{l=1}^{k-1} \zeta_{a_l} \leq C - B ,$$

$$\zeta_{a_{k-1}}, \ldots, \zeta_{a_{k-n_k+1}} = x .$$

We can now express the optimal objective of the stationary search as follows:

$$\min_x F_{|A|+1}(0, x) \tag{9}$$

Property 4. *The values $F_k(B, x)$ satisfy the following recurrence equation:*

$$F_{k+1}(B', x') = \begin{cases} F_k(B' + x'[1]\kappa(a_k), x = x'[2 \ldots n_k - 1]) + W_k(x') \text{ if } k < N \\ \min\limits_{0 \leq \delta' \leq C'} \Big(F_k\big(B' + x'[1]\kappa(a_k), x = x'[2 \ldots n_k - 1] \cdot \delta'\big) + W_k(x' \cdot \delta') \Big) \\ otherwise. \end{cases} \tag{10}$$

where $C' = \left\lfloor \frac{C - B' - \sum_{i=1}^{n_{k+1}-1} x'[i]\kappa(a_{k+1-i})}{\kappa(a_{k-n_{k+1}})} \right\rfloor$.

Proof. Let ζ be the optimal allocation solution of $F_{k+1}(B', x')$. By the decomposition of the values $Part_k$ as a sum of W_l values in (7), we can see that if $k < N$, $n_{k+1} = n_k + 1$ and $x' = \zeta_{a_k}, \ldots, \zeta_1$ contains the effort of all the first angles that led to a remaining budget B'. So for earlier decisions, all the allocation of effort is know and thus $F_{k+1}(B', x') = Part_k(\zeta) = W_k(x') + F_k(B' + x'[1]\kappa(a_k), x = x'[2 \ldots n_k - 1])$.

Assume that $k > N$, so that $n_{k+1} = n_k = N$. Setting $\delta' = \zeta_{a_{k-n_k+1}}$, we have $x' \cdot \delta' = \zeta_{a_k}, \ldots, \zeta_{a_{k-n_k+1}}$. Setting $x = x'[2 \ldots n_k - 1] \cdot \delta'$, the allocation ζ considered for angles a_1

to a_{k-1} defines a feasible solution of the sub problem whose optimal value is $F_k(B' + \kappa(a_k)x'[1], x)$. So $F_{k+1}(B', x') = Part_{k-1}(\zeta) + W_k(x' \cdot \delta')$ is not less than the right hand side of the equality.

Conversely, consider any $\delta' \leq C'$. Set $x = x'[2 \dots n_k - 1] \cdot \delta'$. Let ζ' be the optimal plan solution of $F_k(B' + \kappa(a_k)x'[1], x)$. We can define the plan ζ'' such that $\zeta''_{a_l} = \zeta'_{a_l}$ if $l < k$, and $\zeta''_{a_k} = x'[1]$, and obtain a feasible solution of the sub problem for which $F_{k+1}(B', x')$ is the optimal value. So we get:

$$\begin{aligned} F_{k+1}(B', x') &\leq Part_k(\zeta'') = Part_{k-1}(\zeta') + W_k(x' \cdot \delta') \\ &= F_k(B' + \kappa(a_k) \cdot x'[1], x) + W_k(x' \cdot \delta'). \end{aligned} \tag{11}$$

Hence the reverse inequality holds. □

The forward dynamic programming algorithm issued from Property 4 computes for each phase k from 1 to $|A|$ and for each possible state (B', x') of this phase the value $F_k(B', x')$. Algorithm 1 shows this first part that computes the minimal target missing probability. The computation of the optimal effort allocation ζ is then done using classic dynamic programming approach by searching among the stored values $F_k(B', x')$ which decisions gave the optimal value, with a lower complexity.

Algorithm 1 DP algorithm

```
for all $B \leq C$ do
    $F_1(B, null) = 1$
end for
for $k = 2$ to $|A| + 1$ do
    for all possible couples $(B', x')$, with $length(x') = n_k - 1$ do
        Compute $F_k(B', x')$ using equation (10)
    end for
end for
return $\min_x F_{|A|+1}(0, x)$
```

We can now analyze the complexity of Algorithm 1.

Property 5. *The time complexity of Algorithm 1 is:*

$$\mathcal{O}\left(|A| \cdot N \cdot J \cdot C^N\right).$$

It is pseudo-polynomial for fixed N.

Proof. Let us denote by $z_{max} = \left\lfloor \dfrac{C}{\min_{a \in A} \kappa(a)} \right\rfloor$ the maximum number of observations in an angle. For each iteration of the outer loop, $k \in \{2, \dots, |A| + 1\}$ a state (B', x') is composed of the remaining budget and the at most $N - 1$ last decisions composing x'. The space needed to store each value $F_k(B', x')$ is therefore $\mathcal{O}(|A| \cdot C \cdot z_{max}^{N-1})$. Now for each $F_k(B', x')$, the time complexity of the computation of the recurrence equation (10) depends on the number of different δ' values, which can be bounded by z_{max}. For each δ' the computation of $W_k(x'.\delta')$ is in $\mathcal{O}(N \cdot |R_k|)$. As $\min_{a \in A} \kappa(a) \geq 1$, $z_{max} \leq C$. This gives the whole time complexity of the algorithm. □

Notice that in real applications, N is usually quite small and ranges from 2 to 6 (in the 3D case), and the budget and z_{max} are usually far lower than J, which makes this approach tractable in practice.

IV. Heuristics and bounds

This section is devoted to the presentation of some heuristics methods and a relaxation for the stationary target search.

A. Greedy algorithm

We first present an iterative algorithm that approximates the optimal search effort allocation. This algorithm is an adaptation of the optimal greedy algorithm for the computation of the allocation of discrete effort in the case of disjoint cones of observation and unit costs which was proposed in [12]. Due to the overlapping of the observation cones, the conditions that ensured its optimality vanish. However our experiments, in section VI, shown that it provides a fast and high quality approximation of the optimal solution. We first detail the concept of rate of return of angles on which the algorithm is based and we then present the complete algorithm.

1) Rate of return of angles: Let us first define $p^{(ang)}(a)$ as the probability that the target is in the cone associated with the angle a. We have

$$p^{(ang)}(a) = \sum_{j \in M_a} p(j)\,. \tag{12}$$

We define $b'(j, \zeta, a)$ as the *rate of change* of the function $b(j, \zeta)$ for an additional effort in the angle a by:

$$\begin{aligned} b'(j, \zeta, a) &= b(j, \zeta + \mathbf{1_a}) - b(j, \zeta) \\ &= \alpha_{j,a} \prod_{a' \in Y_j} (1 - \alpha_{j,a'})^{\zeta_{a'}}\,. \end{aligned} \tag{13}$$

Where $b(j, \zeta)$ is defined in Equation (1) and where $\mathbf{1_a}$ is the unit vector of length A such that $\mathbf{1_a}(\mathbf{a}) = \mathbf{1}$ and $\mathbf{1_a}(\mathbf{a'}) = \mathbf{0}$ for $a' \neq a$. As it can be seen in the Equation (13), the value of $b'(j, \zeta, a)$ depends not only on angle a but also on all the angles that cover the region j.

We define the *angular detection function* $\beta(a, \zeta)$ as the conditional probability of detecting the target in one of the regions in the cone associated with an angle a, for an allocation of effort ζ on angles, given the target is located in the cone. The angular detection function $\beta(a, \zeta)$ is:

$$\beta(a, \zeta) = \frac{\sum_{j \in M_a} p(j) \cdot b(j, \zeta)}{p^{(ang)}(a)}\,.$$

We define β' as the *angular rate of change* of the function β for an additional effort in angle a by:

$$\beta'(a, \zeta) = \frac{\sum_{j \in M_a} p(j) \cdot b'(j, \zeta, a)}{p^{(ang)}(a)}\,.$$

We then define $\rho(a, \zeta)$ as the *rate of return* function. It represents, for an allocation ζ, the ratio between the increase of the probability of detection for a new increment of effort in the angle a, and the investment cost generated by the same increment. It equals:

$$\rho(a,\zeta) = \frac{p^{(ang)}(a)\beta'(a,\zeta)}{\kappa(a)} = \frac{\sum_{j\in M_a} p(j)\cdot b'(j,\zeta,a)}{\kappa(a)}. \tag{14}$$

In the overlapping cones case, the value of the *rate of return* $\rho(a,\zeta)$ depends not only on the value of effort ζ_a but also on the allocation of effort ζ to all the angles that cover any region j covered by a. Algorithm 2 summarizes the main steps. Starting from a null allocation of effort, at each iteration an additional unit of effort is added in the angle with the best rate of return until the budget is reached (as we assumed costs not less than 1, there are at most C such iterations). To compute or update the rate of returns, each region j appears at most N times in the sums, and there are at most N terms in the computation of $b'(j,\zeta,a)$. The complexity of this algorithm is thus $\mathcal{O}(C\cdot|A|\cdot J\cdot N^2)$.

Algorithm 2 Greedy algorithm

```
$\zeta = (0,\dots,0), B = 0$, and Compute $\rho(a,\zeta), \forall a \in A$
while $\exists a \in A$ such that $B + \kappa(a) \leq C$ do
    Compute $a* = \underset{a\in A, B+\kappa(a)\leq C}{argmax} \quad (\rho(a,\zeta))$
    $\zeta = \zeta + \mathbf{1}_{a*}$
    $B = B + \kappa(a^*)$
    Update $\rho(a,\zeta)$ for all $a$ such that $M_a \cap M_{a^*} \neq \emptyset$
end while
```

In the non-overlapping cones case and unit costs, the optimisation problem can be exactly solved using this algorithm [12]. In our case this result does not hold. Hence, our heuristic is suboptimal.

B. Adapted Random Permutation Scan Method

We now introduce an algorithm adapted from the conventional *random permutation scan* method as described in [14], referred to as *RPSM* in the following. The original strategy entails a comprehensive sweep of the entire search area in the absence of any prior information about the target's position. During each iteration, the radar conducts an observation in each considered direction following a random order. The algorithm stops when the budget of observations runs out.

In this study, we have adapted this method to scenarios where there is prior knowledge about the target's location. The set of considered angles corresponds to those that cover at least one region j such that $p(j) > 0$ thus we consider angles such that $p^{(ang)}(a) > 0$.

a) The algorithm: A buffer H is used to store the angles that have not been observed yet. The buffer is initialized with the set of angles covering at least one region with a positive prior on target presence (*i.e.* $p^{(ang)}(a) > 0$). When the buffer is empty, it is reinitialized with the same set of angles.

The markovian transition matrix is used to compute the probability of presence of the target at each timestep. Hence at time step t, the probability $p_t(j)$ is computed recursively from the prior probability p_0 and the transition matrices π_t of the Markov chain. The set of angles covering at least one region with a positive prior on target presence is updated consequently at each timestep. An allocation plan is initialised to a null plan and allocations are computed sequentially with Algorithm 3. When the buffer is empty, it is updated with $\{a \in A : p_t^{(ang)}(a) > 0\}$.

Algorithm 3 RPSM adapted to target search with priors

```
$\zeta = (0,\dots,0)$, $B = 0$
$H = \{a \in A : p^{(ang)}(a) > 0\}$
while $\exists a \in H, B + \kappa(a) \leq C$ do
    Sample $a$ from $H$ without replacement
    if $B + \kappa(a) \leq C$ then
        $\zeta = \zeta + \mathbf{1}_a$
        $B = B + \kappa(a)$
    end if
    if $H = \emptyset$ then
        $H = \{a \in A : p^{(ang)}(a) > 0\}$
    end if
end while
```

This algorithm is used as a baseline for the search of a stationary and moving target.

C. Relaxation of the problem for the computation of an upper bound

For the moving target search problem, the FAB algorithm is guaranteed to converge to the optimal plan when the search effort is continuous but not when it is discrete (see [1]). As in [12], we use the value of the optimal payoff obtained in the continuous case as an upper bound on the optimal value of the payoff obtainable in the discrete case. The aim is then to measure the efficiency of the discrete approach with respect to the bound given by the continuous optimal solution.

The detection function and the objective functions (stationary and moving target) have the same formulation both in discrete and continuous cases. These functions can thus be used to compute the optimal allocation of effort in the continuous case.

Computing the optimal allocation of continuous effort for a stationary target search is a quite difficult convex optimization problem. Henceforth, we propose here to further relax the problem in order to compute an allocation on disjoint angles that provides a simpler upper bound for the optimal probability of detection on this problem. This relaxation requires that $\kappa(a) = 1\ \forall a \in A$.

The relaxed instance I' is defined by splitting each angle a into N disjoint cones of same area: $u_1(a),\dots,u_N(a)$. The budget of I' is $C' = C\cdot N$. The visibility coefficients are such that for each region $j = 1,\dots,J$, if a' is an angle of instance I', then $\alpha'_{j,a'} = \alpha_j = \max_{a\in Y_j}\alpha_{j,a}$.

Property 6. *A continuous allocation ζ for the original instance I defines a feasible continuous allocation ζ' for the relaxed instance I', by allocating ζ_a effort to each of the N cones issued from angle a. If a' is an angle of I' then: $\zeta'_{a'} = \sum_{a,\exists i,u_i(a)=a'}\zeta_a$. Moreover $P(\zeta') \geq P(\zeta)$.*

Proof. The allocation ζ' is feasible for the relaxed instance I' because

$$\sum_{a' \in A'} {\zeta'}_{a'} = \sum_{a'} \sum_{a, \exists i, u_i(a)=a'} \zeta_a = \sum_a N \cdot \zeta_a \leq N \cdot C = C'.$$

Let us first observe that for a region j, since for any angle a, $\alpha_{j,a} \leq \alpha_j$,

$$\prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} \geq \prod_{a \in Y_j} (1 - \alpha_j)^{\zeta_a}.$$

Thus,

$$1 - \prod_{a \in Y_j} (1 - \alpha_{j,a})^{\zeta_a} \leq 1 - \prod_{a \in Y_j} (1 - \alpha_j)^{\zeta_a}. \qquad (15)$$

Now consider the new instance I'. For each region j, there exists a unique angle $a'(j)$ that covers j (since in I' we have disjoint observation cones). Hence

$$\begin{aligned} P(\zeta') &= \sum_{j \in J} p(j) \left(1 - (1 - \alpha_j)^{\zeta'_{a'(j)}}\right) \\ &= \sum_{j \in J} p(j) \left(1 - (1 - \alpha_j)^{\sum_{a \in Y_j} \zeta_a}\right) \\ &= \sum_{j \in J} p(j) \Big(1 - \prod_{a \in Y_j} (1 - \alpha_j)^{\zeta_a}\Big). \end{aligned}$$

From (15), we deduce that $P(\zeta') \geq P(\zeta)$. □

We can compute the optimal allocation of continuous effort on disjoint cones of observations for the relaxed instance I' with the algorithm presented in [12]. We can then use it to compute an upper bound on the optimal allocation of continuous effort to overlapping cones for the search of a moving target.

V. Moving target search

We now consider the moving target search problem with objective function given by Eq. (4). The mathematical formulation of this problem is:

$$\begin{aligned} &\max \ \hat{P}_T(Z) && (16) \\ &\text{s.t.} \ \sum_{a \in A} \zeta_a(t) \cdot \kappa(a) \leq C_t \ \text{ for } t = 0, \dots, T\,. \\ &\qquad \zeta_a(t) \in \mathbb{N}\,, \forall a \in A, \ \text{ for } t = 0, \dots, T\,. \end{aligned}$$

Here this problem is solved by using the FAB algorithm [7] which is a generalization of Brown's recursion [15]. Both algorithms adapted to an allocation to disjoint angles have been presented in [12].

Let us recall the principles of the FAB algorithm [1]. Starting from an initial plan Z_0, at each iteration (outer loop) the FAB algorithm computes a new plan, until two successive plans are equal, or the allowed number of iterations is reached.

In an iteration of the outer loop, starting from a plan Z, the computation of a new plan is done iteratively in an inner loop for each time step $t = 0, \dots, T$. At a time-step t of the inner loop, the algorithm computes the values $q(j, Z, t)$ for each region j. To give an intuition, when the objective is the detection probability before horizon T, $q(j, Z, t)$ represents the probability that the target is in region j at time t and is not detected at any time other than t according to the current plan Z. The plan Z is then updated by computing an optimal stationary allocation for time t, using $q(j, Z, t)$ for $j = 1, \dots, J$ as a prior probability distribution of target presence in the regions.

The computation of q values is done by using the markovian target movement, by decomposing q into a product of two functions R and S.

$$q(j, Z, t) = R(j, Z, t) \cdot S(j, Z, t)\,. \qquad (17)$$

When the objective is the detection probability before horizon T, $R(j, Z, t)$ (resp. $S(j, Z, t)$) represents the probability that the target is in region j at time t and has been missed in time steps $t+1, \dots, T$ (resp. $0, \dots, t-1$) withs the plan Z. Those functions can be expressed with a recursive form thanks to the markovian property of the target movement (see details in [1]).

$$\begin{aligned} &R(j, Z, t) = E_{jt}[\gamma_t(\omega)] + \\ &\quad \sum_{i=1}^{J} R(i, Z, t+1) \cdot \big(1 - b(i, \zeta(t+1))\big) \cdot \pi_t(j, i)\,. \qquad (18) \end{aligned}$$

$$\begin{aligned} &S(j, Z, t) = \\ &\quad \sum_{i=1}^{J} S(i, Z, t-1) \cdot \big(1 - b(i, \zeta(t-1))\big) \cdot \pi_{t-1}(i, j)\,. \qquad (19) \end{aligned}$$

The value $E_{jt}[\gamma_t(\omega)]$ is the expectation of the coefficient γ_t of the objective function over trajectories such that the target is in region j at time t. Notice that it only depends on the target movement and does not depend on the plan, so that it can be pre-computed.

The initial values, for $j = 1, \dots, J$, are $S(j, Z, 0) = p_0(j)$ and $R(j, Z, T) = E_{jT}[\gamma_T(\omega)]$.

Notice that in the inner loop of the FAB algorithm, the plan Z is updated iteratively for increasing time steps. So the future of the plan does not change between two consecutive iterations of the innerloop and thus $R(j, Z, t)$ according to (18) can be computed before the inner loop for each region j and each t. At each iteration t of the inner loop, for each region j, $S(j, Z, t)$ can be computed using (19) with the plan updated in the previous iteration.

If we start from a null plan, then the first iteration of FAB computes a myopic plan where the optimal stationary effort allocation of time t is computed from the target move probabilities at time t regardless of the allocation in the previous time steps. But the algorithm can be used with different initial plans.

Unlike the continuous effort case, the algorithm FAB for a discrete effort is not guaranteed to converge to the optimal plan. However, there exists a necessary condition of optimality.

Proposition 1 (From Theorem 3.4 [1])**.** *Assume we have a discrete-effort exponential detection function by region $b(j, \zeta)$.*

If Z is optimal, then for all t the allocation $\zeta(t)$ is an optimal allocation for the stationary target search with prior distribution q.

Therefore, finding, with dynamic programming, the optimal stationary plan at each step t ensures that the plan computed with FAB satisfies the necessary conditions which only guarantees that the returned plan is a local optimum.

In order obtain an optimal plan, we may consider a nonlinear integer program. Indeed, as stated in Section 3.3.2 of [1], when the effort is allocated in each region j independently, one can express the moving target search problem as a nonlinear integer program. However, even in this simple case, this is of a little theoretical and practical use since the objective function is computed by considering all the possible trajectories ω of the target. Hence, this does not scale to our problem with at least 1500 regions.

VI. Experimental results

In this section, we study the efficiency of the algorithms presented in the previous sections, in terms of solution quality and computational effort. We implemented the models and algorithms of the previous sections and designed different realistic scenarios, inducing our experimental parameters, and generated random instances according to these scenarios. The algorithms were implemented in Python and we ran scenarios on an two Intel Xeon E5-2690v3 (24 cores / 2.60 GHz).

Even though the algorithms presented in this paper could be used with a 3D search area, we assumed a stationary airborne platform facing a 2D search area discretized in regions spaced 10km apart, with a total of 1500 regions. The regions are arranged in a honeycomb pattern for an efficient use of 2D space. Using smaller distances between regions would improve the realism of the scenarios. However, it hugely increases the number of regions in the search area (e.g a distance of 1km gives 155000 regions) and then the computation time. In our study, we chose the parameters so that the dynamic programming algorithm could run in a reasonable time. We expect that the behaviour of the algorithms remains consistent for both small and large regions.

The target moves between two adjacent regions at each timestep. Considering the maximum relative speed of fighter jets, we consider a timestep to last roughly 7s in these experiments.

The platform has a radar that can make observations in the angular domain $[-60°; +60°]$. A radar "dwell" (name for an observation) in a direction affects the regions located in a cone of geometric height 250km and angular diameter 3 degrees. We consider set of angles such that they are equally spaced in the angular domain and each associated with an observation cone of angular width 3°.

The maximum number of cones N covering a region is related to the angular separation between angles and the number of angles. When the angular separation between angles is 3° we have $N = 1$ and $|A| = 40$; if the angular separation is 1.5°, $N = 2$ and $|A| = 79$. The radar is considered to use a single mode.

In the experiments, we assumed a fixed cost for making an observation in an angle a to $\kappa(a) = 1$.

In each scenario, an area of interest is initialized. The area of interest is the set of regions that have a nonzero probability of containing the target. This set is fully contained in the part of the search area that is visible from the radar.

The experiments use six parameters for the tested scenarios:

- **FOV_portion**: Determines the size of the area of interest as its portion of the total observable area. It can be 0.05, 0.5, or 1.0.
- **Horizon**: The number of time steps for the search. It can be 1, 2, 5, or 10.
- **Budget**: The number of observations that can be made during a time step. It can be 1, 2, 5, 10, 20, 40, or 50.
- **N**: The maximum number of consecutive cones covering a region (see Figure 2). It can be 1, 2 or 3. Higher values did not lead to significantly different results from what was observed at $N = 3$.
- **visibility**: Indicates the detection model used (see Figure 3) either "*distance*" or "*realistic*".
- **movement_type**: Defines the movement transition matrix. Can be "drone" or "jet". "drone" defines a uniform distribution over all possible direction. "jet" selects a random direction and then defines a distribution where there is 0.9 chance of going in this direction and 0.1 of remaining in the current region. The probability is 0 in all other directions. It approximates the movement model of a fighter jet that does not change direction.

A dwell is considered to last 100ms, allowing for a maximum of 70 observations per time step in different angles. We limited the number of observations to 50 per time steps and the horizon to 10, as larger values do not lead to significantly different results.

In the following experiments, we first focus on the problem of a stationary target search. We evaluate how using a detection model that depends only on the distance can negatively impact the performance of the allocation computed. Then, we evaluate the error of the probability obtained with the greedy algorithm compared with the optimal probability given by the dynamic programming algorithm. Then, we compare the probability of detection obtained with the dynamic programming algorithm with the baseline and the upper bound.

The second part experiments the moving target search. We show how increasing the maximum number N of observation cones covering a region of the search area affects the mean completion time of the plan. Also, we exhibit the performance of the algorithms FAB + greedy algorithm and FAB + dynamic programming for a fixed value of N in terms of the objective function and computation time. Finally, we show how the target movement model affects the mean completion time of the plan.

A. Stationary target search

1) Error caused by the use of the "distance" detection model: In the literature, the detection model used for the radar is often only a function of the distance to the radar. In this section, we evaluate the error caused by the use of such a model compared to the "*realistic*" detection model.

To do so, we compute optimal allocations for stationary target search for the allocation of search effort on overlapping cones of observations such that $N = 3$ on different scenarios. We vary the budget and the FOV_portion. We first compute the optimal allocation for the "realistic" detection model and obtain the optimal probability of detection. Then, we compute the optimal allocation for the "distance" detection model and compute the detection probability of this last allocation by using the "realistic" detection model. We compute the mean absolute error of the probability of detection obtained with the "distance" detection model as compared to the optimal probability. We also show values ranging from the 5th to the 95th percentile of the error.

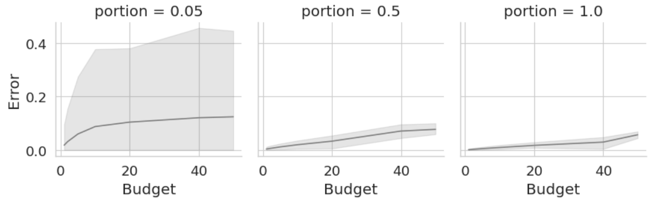


Fig. 4. Error caused by the use of the "distance" detection model.

In Figure 4 we see that the mean error is larger than 0.1 in many cases (especially when the area of interest is small) and the error can even achieve values above 0.4 for some scenarios for large values of budget. The use of the "distance" detection model therefore leads to a significant error in the probability of detection. We will use the more realistic detection model in the remaining experiments.

2) Comparison between the greedy algorithm and the dynamic programming algorithm: Algorithms 1 and 2 are both used as a module of the FAB algorithm, it is therefore interesting to study their behaviour in the simpler context of the stationary target search.

We ran both algorithms on different scenarios. We then computed the mean absolute error of the probability of detection obtained with the greedy algorithm as compared to the optimal probability obtained with dynamic programming.

Figure 5 depicts this error for small (left) to large (right) areas of interest w.r.t the budget of observations and parameter N. We computed the error only for $N = 2$ and 3 because using larger values requires amounts of memory unavailable to us to run the dynamic programming algorithm. The shaded areas around the curves represent the values between the 5th and the 95th percentile of the error.

We observe that the error is small. It is 0 in 81% of the simulations, rarely takes value above 0.02: the maximum error observed was 0.056. There seems to be a relationship between the size of the area of interest, the budget of observations and the error. For small areas of interest, there tends to be a higher error for small budgets. On the contrary for large areas of interest, there tends to be higher error for large budgets.

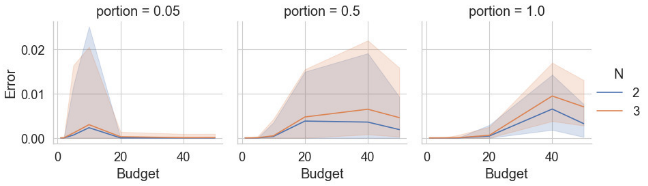


Fig. 5. Error of the greedy algorithm w.r.t. budget for different values of FOV_portion (columns), detection model (rows) and N (color).

Overall, we observed that the greedy algorithm is a good approximation of the optimal plan for the search of a stationary target. It also has a much lower space and time complexity, as shown in Section III-B.

3) Stationary target search: comparison between the optimal solution, RPSM and the relaxation: In Figure 6, we observe the optimal probability of detection computed with the dynamic programming algorithm w.r.t. budget, and we compare it with the one obtained with the random permutation scan method and the upper bound provided by the relaxation of the problem. We show the results for different sizes of the area of interest (columns) and different values of N the maximum number of cones that cover a region (rows). We show the results up to $N = 3$ because larger values do not lead to results significantly different from the ones obtained with $N = 3$.

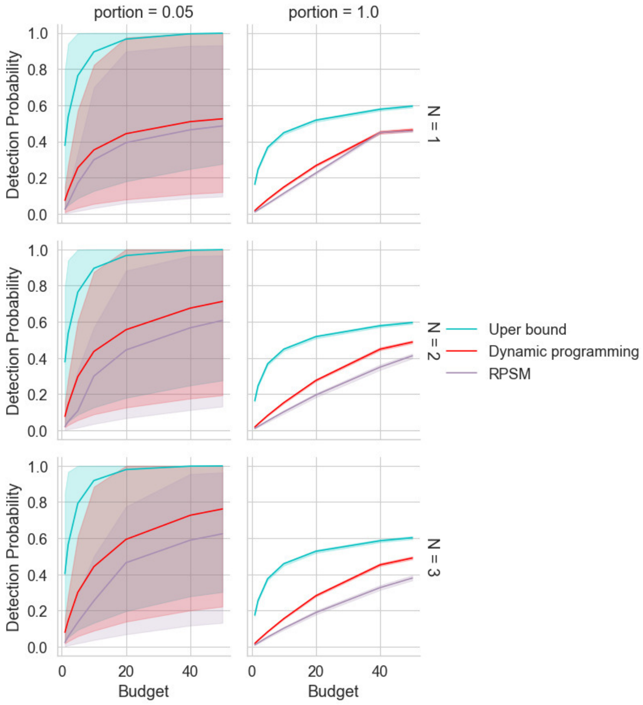


Fig. 6. Probability of detection obtained with three algorithms w.r.t. budget for different values of FOV_portion (columns) maximum number of cones that cover a region (rows).

First, we see that the upper bound provided by the relaxation is far larger than the optimal probability of detection. This bound is not very informative. To get a more informative bound, we should solve the convex program corresponding to the relaxation with gradient techniques.

Our analysis demonstrates a correlation between the magnitude of N and the disparity in performance between the greedy algorithm and RPSM. When applied to a large area of interest, both algorithms exhibit similar performance for $N = 1$. An increment of N to 3 causes a slight enhancement in the greedy algorithm's performance by 4%, whereas RPSM's performance concurrently diminishes.

Conversely, when the area of interest is smaller, the performance difference between the two algorithms becomes greater. Increasing the N value from 1 to 3 in this context brings mutual benefits: a significant 17% increase for the greedy algorithm and a modest 3% improvement for RPSM.

In the context of the search of a stationary target, the increase in N almost does not benefit RPSM due to its inherent constraint of allocating substantial effort to regions with low probability of detection. Conversely, the greedy algorithm derives significant advantages from such increases in N.

B. Moving target search

In this section, we evaluate the performance of our algorithms in the context of the moving target search.

1) Interest of increasing N for the moving target search: In Figure 7, we first observe the mean completion time (referred by ”meantime” in the remaining of this section) obtained with FAB+dynamic programming w.r.t budget for different values of N. We observed comparable results across different values of horizon, and the difference of performance between the algorithms are best observed for larger values of horizon. We therefore isolate the results for a horizon of 10.

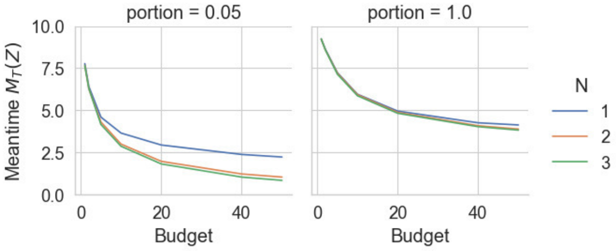


Fig. 7. Meantimes w.r.t. budget obtained with FAB + dynamic programming for N= 1,2,3 for different values of FOV_portion.

In our observations, an increase in the value of N induces a decrease in the meantime. Specifically, for a smaller area of interest, we witness a 6% enhancement in efficiency when N is incremented from 1 to 2, followed by a further 2% improvement as N increases from 2 to 3. This trend shows the diminishing benefits of increasing N.

2) Meantime for different algorithms: In Figure 8, we compare the meantimes obtained with FAB+greedy algorithm, FAB+dynamic programming and RPSM w.r.t. budget, for a fixed horizon = 10. we observe it for different sizes of area of interest.

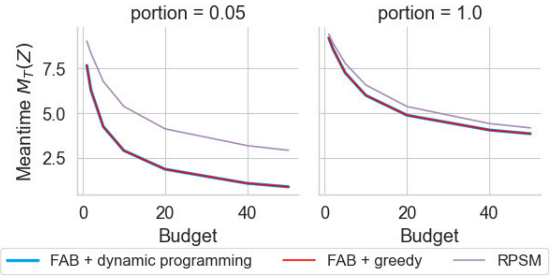


Fig. 8. Comparison of meantimes obtained with different algorithms (the red line is behind the blue line)

Our analysis reveals that the meantimes of plans computed by FAB coupled with dynamic programming and FAB with the greedy algorithm are remarkably similar. Indeed, in 0.45% of the instances, their results are identical.

The mean absolute error in meantime between FAB + greedy algorithm and FAB + dynamic programming is marginal, at 0.002 timesteps (1 timestep = 7 seconds in our experiments, so 0.014 seconds). The peak recorded performance advantage of FAB + dynamic programming over FAB + greedy method amounted to 5.25% of the computed meantime (1.15 seconds gained).

We observed that the adaptation of RPSM to moving target search always performs worse than FAB, especially when the area of interest is small.

3) Comparison of computation times: In Figure 9, we observe the mean computation times of FAB + dynamic programming and FAB + greedy for different values of N, w.r.t. the horizon.

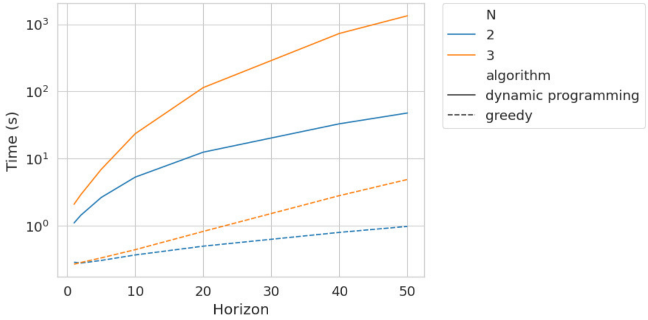


Fig. 9. Computation times for FAB + dynamic programming and FAB + greedy algorithm for different values of N

We observe that FAB + greedy algorithm is several order of magnitudes faster to compute the allocation plan than FAB + dynamic programming. FAB + greedy is often under 1s while FAB + dynamic programming can take several dozens of minutes. Consistently with the worst case time complexity, we also observe that the computation time of the dynamic programming algorithm increases faster with N than the computation time of the greedy algorithm. In a context where allocations must be computed in real time, the greedy algorithm therefore provides a good approximation of the plan computed with FAB + dynamic programming in a very reasonable amount of time. Furthermore, there is a tradeoff to be made between plan quality and computation time as using higher values of N affects both significantly.

4) Search for targets with different movement models: In Figure 10, we observe the meantime of plans for different target moves w.r.t. the budget. We isolate the experiments where horizon = 10 and the area of interest is small. Indeed, these are cases when the difference between the results are best observed.

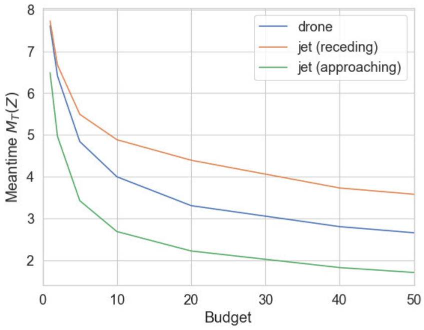


Fig. 10. Meantime for different target types

We can see that when we have a prior on the target direction, it is easier to detect it when it is approaching us and harder when it is receding. When the target moves like a drone and we do not have a prior on its direction, the meantime lies between the meantimes for the two type of jet moves. Note that in the case of a receding target, our model is optimistic because we do not consider ground echoes. Overall, when the target is a jet and we have a prior on its direction, the prior heavily impacts the meantime.

VII. Conclusion

We proposed an optimal algorithm and a fast heuristic for the problem of search of a stationary target with a radar by considering overlapping cones of observation and discrete effort. This allowed us to compute better plans in diverse contexts for moving targets. We also proposed a detection model for the radar which is more realistic than the ones generally used in the literature. Those algorithms were compared in different scenarios for the search of stationary and moving targets. The experiments showed that the realistic detection model gives allocation plans that are significantly different from the one that was previously used in the literature. Using overlapping cones observations led to a notable improvement of the solution obtained. The use of the greedy approximation gave very close to optimal solutions for the stationary search and reduced drastically the computation time for the search of a moving target. Finally, we demonstrated that a prior on the target's direction heavily impacts the mean completion time of the plan.

Future work should consider further theoretical and practical issues. Among them is the complexity of the stationary target search problem. Live experimentations of such algorithms in real conditions remain to be done in order to demonstrate their maturity. Also, other modern important issues are related to the more general problem of searching and tracking multiple targets with multiple assets like e.g. jet fighters and remote carriers embedding different types of active and passive sensors.

References

[1] L. D. Stone, J. O. Royset, and A. R. Washburn, *Optimal search for moving targets*. Springer, 2016.

[2] M. Raap, M. Preuß, and S. Meyer-Nieberg, "Moving target search optimization – a literature review," *Computers and Operations Research*, vol. 105, pp. 132–140, 2019.

[3] F. Delavernhe, P. Jaillet, A. Rossi, and M. Sevaux, "Planning a multi-sensors search for a moving target considering traveling costs," *European Journal of Operational Research*, vol. 292, no. 2, pp. 469–482, 2021.

[4] J. N. Eagle, "The optimal search for a moving target when the search path is constrained." *Operations Research*, vol. 32, pp. 1107–15, 1984.

[5] C. Simonin, J.-P. Le Cadre, and F. Dambreville, "A common framework for multitarget search and cross-cueing optimization," in *International Conference on Information Fusion*, 2008, pp. 1–8.

[6] H. A. L. Thi, D. M. Nguyen, and T. P. Dinh, "A DC programming approach for planning a multisensor multizone search for a target," *Computers and Operations Research*, vol. 41, pp. 231–239, 2014.

[7] A. R. Washburn, "Search for a moving target: The FAB algorithm," *Operations Research*, vol. 31, no. 4, pp. 739–751, 1983.

[8] A. Washburn, "Branch and bound methods for a search problem," *Nav. Res. Logist.*, vol. 45, no. 3, pp. 243–257, 1998.

[9] M. Morin, I. Abi-Zeid, P. Lang, L. Lamontagne, and P. Maupin, "The optimal searcher path problem with a visibility criterion in discrete time and space," in *International Conference on Information Fusion*, 2009, pp. 2217–2224.

[10] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*, ser. Artech House radar library. Boston: Artech House, 1999.

[11] J. L. Williams, "Search theory approaches to radar resource allocation," *7th U.S. / Australia Joint Workshop on Defense Applications of. Signal Processing*, 2011.

[12] H. Vaillaud, C. Hanen, E. Hyon, and C. Enderli, "Target search with a radar on an airborne platform," in *International Conference on Information Fusion*, 2023, to appear in IEEE Xplore.

[13] S. Pérez Carabaza, *Multi-UAS Minimum Time Search in Dynamic and Uncertain Environments*, ser. Springer Theses. Cham: Springer International Publishing, 2021. [Online]. Available: https://link.springer.com/10.1007/978-3-030-76559-0

[14] J. W. Caspers, "Radar random permutation scan method," Patent, oct, 1966.

[15] S. S. Brown, "Optimal search for a moving target in discrete time and space," *Operations Research*, vol. 28, no. 6, pp. 1275–1289, 1980.

DOI: 10.15439/2023F9562
ISSN 2300-5963 ACSIS, Vol. 35

Exploring the Prevalence of Anti-patterns in the Application of Scrum in Software Development Organizations

Michał R. Wróbel*, Dorota Przała* and Paweł Weichbroth*
*Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology, Gabriela Narutowicza 11/12, Gdańsk 80-233, Poland
Email: michal.wrobel@pg.edu.pl, przala.dorota@gmail.com, pawel.weichbroth@pg.edu.pl

***Abstract*—The paper presents a survey-based study that aimed to determine the prevalence of anti-patterns in the Scrum software development methodology. A total of 35 anti-patterns were selected from the literature review, and 42 respondents working in software development organizations located in Poland indicated whether they had encountered each anti-pattern in their organizations. The study found that "Unfinished Tasks" was the most prevalent anti-pattern, highlighting the importance of proper planning and task management within sprints. Additionally, several other common anti-patterns were identified, including daily scrums being extended beyond the recommended time, user stories not being fully refined, and the sprint goal not being defined at the sprint planning meeting. The findings of this study provide valuable insights into the current state of Scrum methodology in software development organizations and highlight areas where there is room for improvement.**

I. Introduction

We are agile! – many organizations proudly proclaim. However, as practice shows, it is always easy to say, but much more harder to accomplish [1]. Since the publication of the Agile Manifesto in 2001 [2], the number of development projects conducted according to the agile approach has been steadily increasing. Nowadays, this approach is also increasingly used outside the domain of software development [3]. However, implementing an agile approach requires more than putting the right processes in place. It requires a change in mindset and attitude toward managing projects.

Scrum has gained significant traction over the years due to its flexibility, iterative approach, and focus on cross-functional team collaboration. It is the most popular software development methodology of all available agile approaches [4], with 9 out of 10 respondents in the State of Agile Report claiming to use it [5]. However, as with any methodology, there are always risks of anti-patterns – common mistakes or misapplications of the methodology that can hinder its effectiveness. A Scrum anti-pattern is defined as a harmful practice within the Scrum framework that may appear convenient at first but proves detrimental in the long run [6].

The aim of this study was to investigate the current state of Scrum implementation in software development organizations located in Poland. For this purpose, a survey was conducted among practitioners to determine the prevalence of Scrum anti-patterns, which are common mistakes or pitfalls in the application of a methodology. They can lead to poor performance, low productivity, and other negative outcomes.

II. Related Work

Based on the interview conducted among 18 respondents, representing 11 IT organizations located in Finland, Eloranta et al. [6] identified ways of potentially harmful mishandling of Scrum. The findings show that a) sprints were too long since the teams could not respond to customer requests and the feedback loop became too long, b) the system testing was performed in the next sprint by a separate team, as well as testing took place at the end of the sprint; in addition, in some cases there was the lack of automation tools employed in the testing, and c) the progress of the work was not made visible with burn-down charts to the teams, thus a phenomenon, termed as "invisible progress" occurred.

Matthies et al. [7] discussed experiences from a classroom project in which a group of 38 students was responsible for developing a single system by using a scaled version of Scrum. In conclusion, the authors argue that the combination of tutor observations, surveys, along with initial testing of automated process analysis leads to better understanding the Scrum adoption, as well as detecting agile practices violations for every team in every sprint, including prolonged and varied duration of sprint, moving testing to the next sprint, and invisible progress.

On a basis of a grounded theory approach, Carew and Glynn [8] investigated how productivity, effectiveness and workflow priorities were influenced by adopting the Scrum. In general, a number of agile anti-patterns were recognized with a negative impact on these three facets, including decision-making incapacity, incomplete deliverables (working code), ad-hoc work requests, and an inability to actually define when work items were completed to the required "Definition of Done", just to name a few.

Through the observation of the two teams, including 14 members who in majority were software developers, as well as acting as a product owner, agile coach, or scrum master, Mortada et al. [9] investigated the cases of four Scrum activities separately practiced among these teams. In total, 13 deviations from Scrum guidelines were identified, including:

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

- four related to the daily scrum event, namely: (1) not all key questions are addressed, (2) not all team members contribute, (3) daily scrum events take longer than 15 minutes, and (4) no fixed time for the daily scrum event;
- seven related to the sprint planning, namely: (5) stories are not refined in the product backlog, (6) no calculation of resources available for the upcoming sprint, (7) sprint goal is defined at the end of the planning meeting, (8) no break down of large stories, (9) no agenda used for the planning meeting, (10) stories are not estimated, and (11) stories not formulated completely;
- two related to the sprint demonstration, namely: (12) sprint does not end with a demonstration, and (13) demonstration to the wrong audience.

Moreover, these agile anti-patterns were also reported to have negative impact on the product quality and team morale.

Based on the interviews conducted with software professionals working in Scrum teams, Çetin and Durdu [10] aimed to explore how the Scrum was practically implemented, and in particular uncover what were the differences between these implementations and the original Scrum model. Their findings show that in some organizations daily Scrum meetings, sprint retrospectives, as well as the burndown charts, were not always implemented.

Having collected data from approximately 40 different software development teams of a large Scrum organization, Heikkila et al. [11] investigated how the requirements were planned and managed, how well the requirements planning and management practices matched Scrum guidelines, and whether the changes were perceived harmful. The findings show that only 30% of user stories were set in progress, while 32% of user stories were closed during the sprint planning day and the sprint review day, respectively. The respondents indicated two reasons why user stories last for multiple sprints, namely the inter-team dependencies and the dependencies to third parties, and the difficulty of splitting user stories into pieces that can be implemented in a single sprint. In conclussion, the authors argue that these two deviations from the Scrum user story management and planning process cannot be categorized as harmful.

McKenzie et al. [12] conducted interviews with eight New Zealand video game development studios with the aim to empirically determine how and why agile frameworks were applied. In this extent, the authors recognized several limitations due to the common misunderstanding in key areas around project management and collaboration, demonstrated by missing retrospectives, and ambiguity related to the status of tasks.

Last but not least, Perry [13] discussed the issue of misunderstanding of the end user role with the ultimate customer. The author concludes that "by focusing on the intermediary rather than on the end user that we do the people who have to work with the system a disservice".

TABLE I
ANTI-PATTERNS SELECTED AFTER LITERATURE REVIEW

Anit-pattern	**Source**
Sprints that are too long	[6], [7]
Variable lengths of sprints	[7]
Unfinished tasks	[8]
Testing in next sprint	[6], [7]
Invisible progress	[6], [7]
Not everyone actively participate in the meeting	[9]
Daily Scrum lasts longer than 15 minutes	[9]
The Daily Scrum does not take place at a fixed time of day	[9]
Not all key issues are addressed	[9]
Daily scrums are not held every day	[10]
Disordered product backlog	[7], [6]
Extensive requirements documentation instead of user stories	[7], [6], [9]
User stories that are too extensive	[11]
PO without authority, has negligible influence on the selection of tasks to be implemented	[7], [6]
Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	[8]
Lack of unanimity whether the task has been completed	[8], [12]
An indirect customer who is service provider of its own customer	[13]
PO delegated by a client and does not understand the role	[7], [6]
Exact estimation instead of relative estimation of items	[7]
Item estimation imposed to the team	[7], [6]
No estimation of resources available for the upcoming sprint	[9]
Sprint planning meeting has no agenda	[9]
Sprint goal set at the end of the meeting or not at all	[9]
Not breaking large user stories into smaller ones during a meeting	[9], [11]
Disruptions in the development process – other areas of the project are developed in turn without one being completed	[8]
Adding items during sprint	[8], [11], [7], [6]
User stories are not fully refined	[9]
A burndown chart is not used	[10]
Semi functional teams	[7], [6]
Insufficient technical knowledge[8]	
Lack of business knowledge	[8]
No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	[7], [12], [9]
The sprint does not end ends with a demonstration	[9]
Demonstration for the wrong targets	[9]
Missing retrospectives	[7], [12]
Combining the two meetings into one	[10]

III. SURVEY DESIGN

To prepare the survey, a literature review was conducted to identify common Scrum anti-patterns. For this purpose, articles in the Web of Science, IEEE Explore, and Scopus databases were searched for the keywords *Scrumbut*, *Scrumfall*, or *Scrum anti-patterns*. A total of 81 articles were found, which were then filtered by title, abstract and content. Finally, 8 articles were identified that described common Scrum anti-patterns [6], [7], [8], [9], [10], [11], [12], [13].

Based on this literature review, 35 Scrum anti-patterns were selected for the survey, which are shown in Table I. To systematize, they were then assigned to one of 7 sections, i.e. Sprint, Daily Scrum, Product Backlog, Product Owner, Sprint Planning, Development Team, and Completing the Sprint. For each of the anti-patterns, survey respondents who indicated that they use or have used Scrum in their development process responded whether and how often they had encountered such a problem in their organizations. A five-point Likert scale was used, along with an additional option of "I don't know". The survey was prepared in Polish.

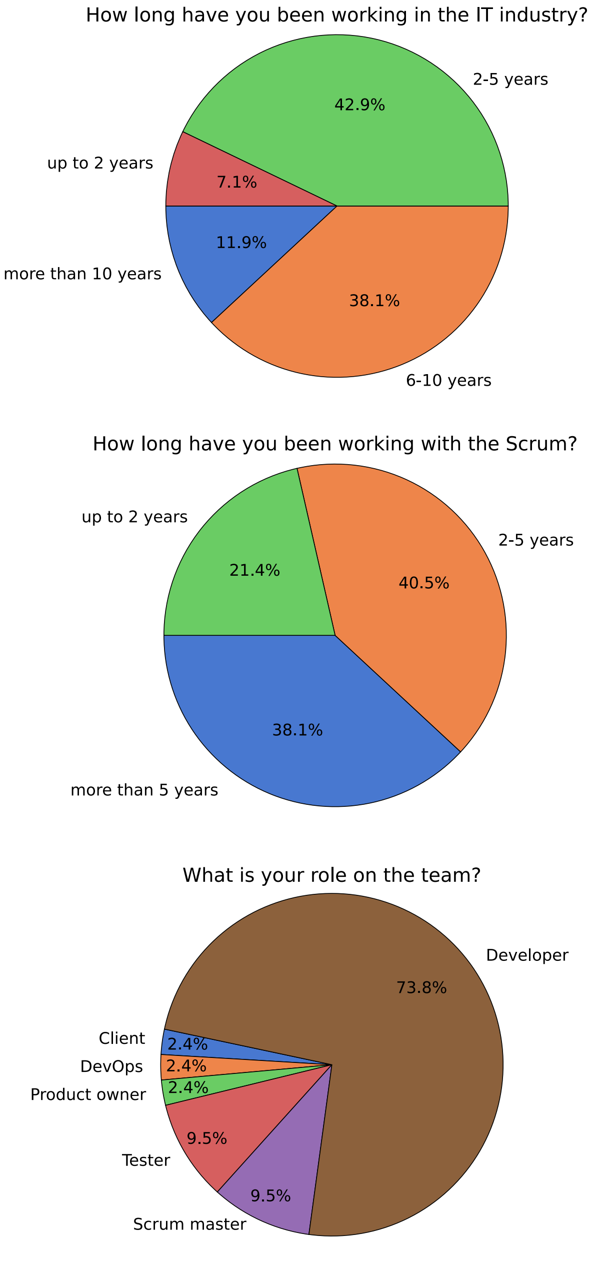


Fig. 1. Experience of survey participants

A. Sprint

The Scrum Guide suggests that sprints should last one month or less and that their length should remain constant throughout the project. Testing and all tasks related to both adding new functionality and testing new code should be completed at the end of the sprint. In addition, progress should be tracked using a burndown chart [14]. The identified anti-patterns that belong to this section are presented in Table II.

B. Daily Scrum

Daily Scrum should take place every day at a fixed time. Each team member should actively participate in the meeting and answer key questions (What did I do yesterday? What will I do today? What obstacles did I encounter?) The meeting should last no longer than 15 minutes [14]. The anti-patterns that have been identified and fall under this section are presented in Table III.

C. Product Backlog

The product backlog should contain items rather than traditional large documentation. The items should be sorted by risk factor and value, and can take the form of user stories that are small enough to be completed in a single sprint [14]. The identified anti-patterns are presented in Table IV.

D. Product Owner

The product owner (PO) represents the interests of the stakeholders and manages the product backlog. Any changes to the backlog can only be introduced by the product owner, and the decisions made should be respected by the entire team [14]. The identified anti-patterns that are part of this section are listed in Table V.

E. Sprint Planning

Sprint planning is the process of deciding what will be worked on in the sprint. Developers should be involved in the selection of tasks, but the product owner has the final decision. Tasks are broken down into smaller ones as needed [14]. The identified anti-patterns are presented in Table VI.

F. Development Team

Programmers in Scrum should form a self-sufficient team with good knowledge of the technical and business knowledge. They should have the competence to both develop and test new versions of software [14]. In this section, 3 anti-patterns were identified and are presented in Table VII.

G. Completing the Sprint

Two meetings should be held at the end of the Sprint. A Sprint Review for the customer, where feedback is received and a demonstration is given, and a Retrospective for the Scrum Team, where they discuss what went well and what went wrong during the Sprint [14]. Table VIII presents the anti-patterns associated with this section.

IV. Results

The survey was conducted in May and June 2022. The invitation to participate was published on LinkedIn and on social media. A total of 42 people took part in the survey.

At the beginning of the survey, information about the participants' experience in IT and Scrum projects was collected. The results are shown in Figure 1. Most of the respondents have commercial IT experience of more than 5 years, and only three have worked in the industry for less than 2 years. Regarding the work with the Scrum methodology, 16 had more than 5 years of experience, 17 between 2 and 5 years, and only 9 less than 2 years. Developers dominated the survey with 31 respondents. In addition, 4 scrum masters and 4 testers participated in the survey, as well as one product owner, one DevOps, and one customer representative.

A. Sprints

The results of the survey regarding sprints are shown in Table II and Figure 2. Of the 5 anti-patterns identified, two were mostly indicated as occurring rarely or never. All respondents indicated that the sprint length is always or mostly in line with the Scrum Guide. In addition, 66% indicated that sprints are always the same length, and only one participant indicated that the opposite is often the case.

In the case of the invisible progress anti-pattern, responses were fairly evenly distributed. Twenty respondents reported that this never or rarely occurs, and 21 reported that it sometimes, often, or always happens.

The results of the survey, on the other hand, confirmed the presence, the last two anti-patterns. None of the respondents stated that there is never a situation where all the tasks that are supposed to be completed in a sprint are completed. In addition, 66% indicated that always or often not all tasks are completed in a given sprint, and 26% that this occurs sometimes. This led to the identification of another anti-pattern – testing in the next sprint. Only 11 respondents reported that this doesn't or rarely happen, while as many as 19 reported that it often or always is the case.

B. Daily Scrum

Table III and Figure 3 shows the results of the daily scrum related anti-patterns. Based on the responses, it can be concluded that daily Scrum in most cases actually take place every day. Only 2 respondents reported that it sometimes happens that a meeting is not held every day, and another two that it is the norm.

For the next two anti-patterns "Not everyone actively participates in the meeting" and "Not all key issues are addressed," the responses indicate that such situations do occur in some organizations. For the latter, only 8 respondents indicated that this never happens.

TABLE II
Sprint related anti-patterns

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Sprints that are too long	1	33	5	3	0	0
Variable lengths of sprints	1	28	10	2	1	0
Unfinished tasks	0	0	3	11	19	9
Testing in next sprint	3	5	6	9	15	4
Invisible progress	1	5	15	13	6	2

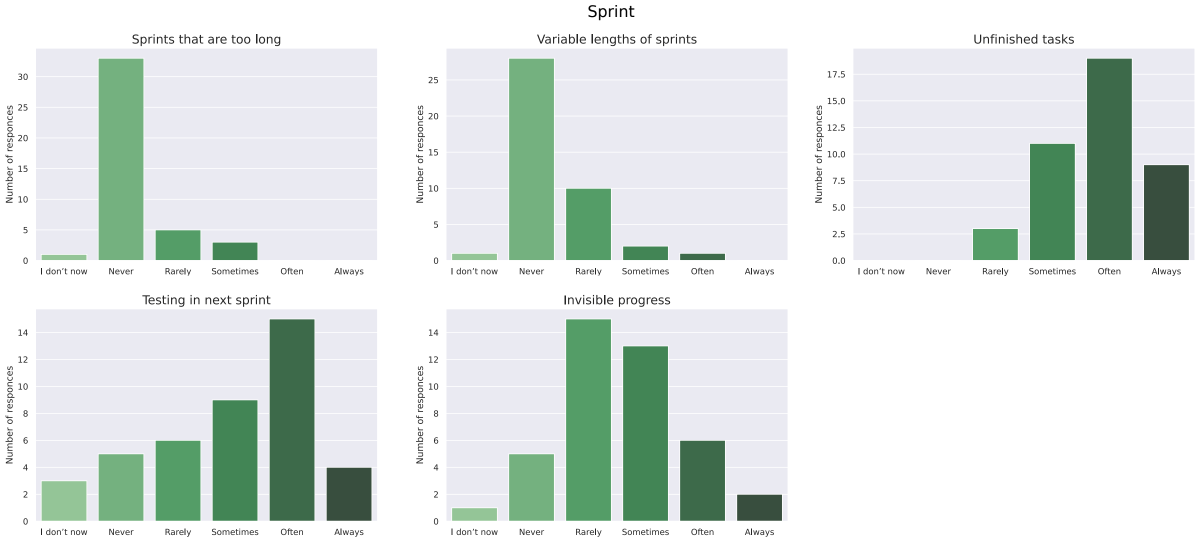


Fig. 2. Sprint related anti-patterns

TABLE III
DAILY SCRUM RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Not everyone actively participate in the meeting	0	15	9	12	3	3
Daily Scrum lasts longer than 15 minutes	0	1	9	9	16	7
The Daily Scrum does not take place at a fixed time of day	0	33	6	3	0	0
Not all key issues are addressed	1	8	16	13	2	2
Daily scrums are not held every day	1	28	9	2	0	2

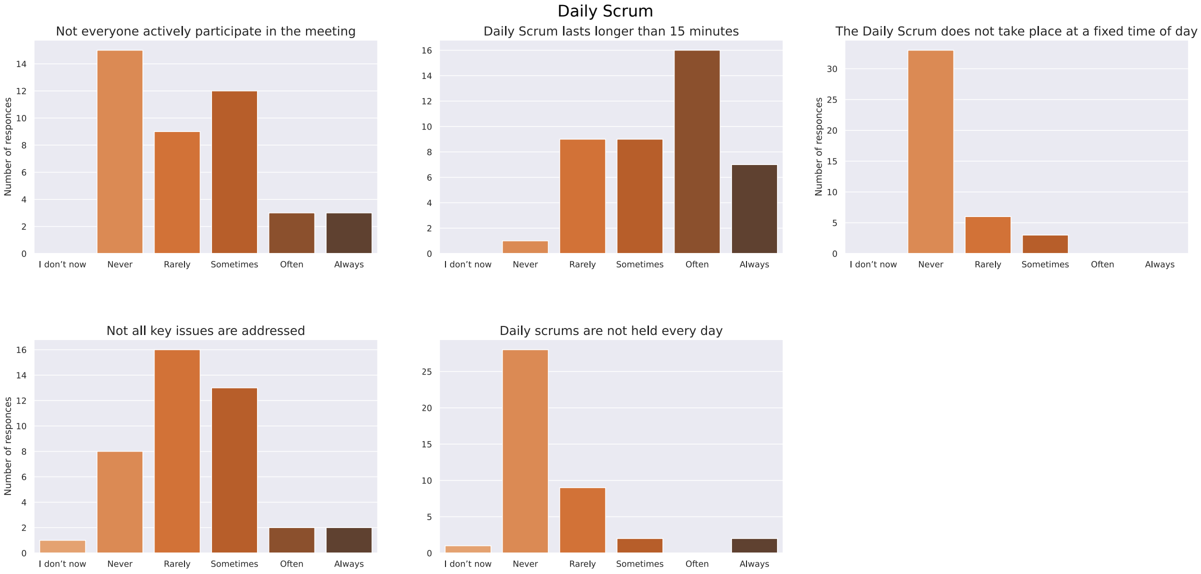


Fig. 3. Daily scrum related anti-patterns

The analysis of the survey results shows that the Daily Scrum goes beyond the recommended 15 minutes in a significant number of cases. Only one respondent reported that it never happens, and as many as 23 that always or often.

C. Product Backlog

The analysis of the responses related to the product backlog showed that the anti-patterns "Extensive requirements documentation instead of user stories" and "User stories that are too extensive" occur rarely in projects. None of the respondents indicated that this situation always occurs, and only 2 and 7, respectively, indicated that it often happens.

On the other hand, in the case of the anti-pattern concerning disordered backlog, more than half of the responses indicated that this situation happens sometimes, often or always. Detailed data with results are shown in Table IV and Figure4.

D. Product Owner

The analysis of the responses related to the product owner showed that only the anti-pattern "Stakeholder indecisiveness" is the common issue among the respondents' organizations. It was reported by 32.5% as often or always occurring. Of the remaining anti-patterns, all received more than 63% of responses that they never or rarely occur. The detailed results of the survey are presented in Table V and Figure 5.

It should be noted that this section of questions was characterized by the highest number of "I don't know" responses. This may indicate that respondents do not fully understand the role of the product owner.

E. Sprint planning

Among the 10 anti-patterns for sprint planning, as many as 2 received 40% or more "Often" or "Always" responses. These are "Sprint goal set at the end of the meeting or not at all", "User stories are not fully refined" and "A burndown chart is not used".

Of the remainder, only the "No estimation of resources available for the upcoming sprint" anti-pattern can be reported as relatively rare, with 64% reported for the "Never" and "Rarely" responses. All others are relatively often identified as occurring in respondents' projects. As can be seen in Table VI and Figure 6, all the others are relatively common in respondents' projects.

F. Development team

The results of the survey with respect to the development team are shown in Table VII and Figure 7. The anti-pattern "Insufficient technical knowledge" was not indicated by any respondent as occurring always and only by four as sometimes.

TABLE IV
PRODUCT BACKLOG RELATED ANTI-PATTERNS

Question	I don’t now	Never	Rarely	Sometimes	Often	Always
Disordered product backlog	1	6	12	12	7	4
Extensive requirements documentation instead of user stories	2	14	13	10	3	0
User stories that are too extensive	1	11	12	11	7	0

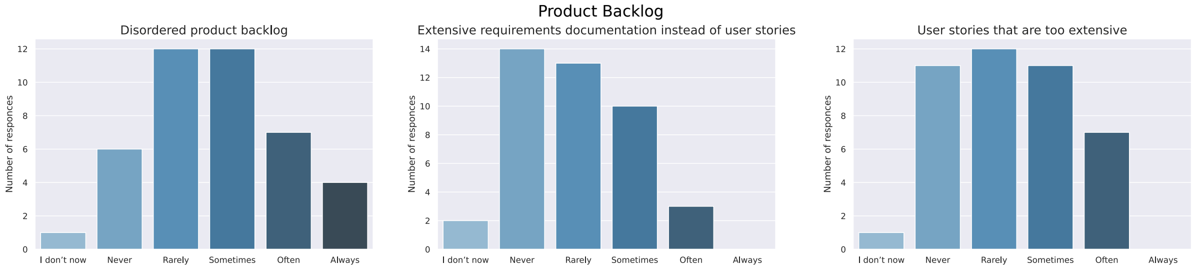


Fig. 4. Product backlog related anti-patterns

TABLE V
PRODUCT OWNER RELATED ANTI-PATTERNS

Question	I don’t now	Never	Rarely	Sometimes	Often	Always
PO without authority, has negligible influence on the selection of tasks to be implemented	1	19	10	6	5	1
Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	1	8	11	7	8	5
Lack of unanimity whether the task has been completed	1	12	16	10	2	1
An indirect customer who is service provider of its own customer	9	14	7	7	4	1
PO delegated by a client and does not understand the role	7	17	8	5	4	0

TABLE VI
SPRINT PLANNING RELATED ANTI-PATTERNS

Question	I don’t now	Never	Rarely	Sometimes	Often	Always
Exact estimation instead of relative estimation of items	2	11	8	10	5	6
Item estimation imposed to the team	0	17	6	6	10	3
No estimation of resources available for the upcoming sprint	0	16	11	7	5	3
Sprint planning meeting has no agenda	3	15	9	9	4	3
Sprint goal set at the end of the meeting or not at all	2	7	8	6	10	10
Not breaking large user stories into smaller ones during a meeting	3	8	12	10	8	2
Disruptions in the development process – other areas of the project are developed in turn without one being completed	2	7	8	17	8	1
Adding items during sprint	0	3	18	11	8	2
User stories are not fully refined	0	2	5	18	10	7
A burndown chart is not used	2	8	5	11	5	11

More than half of respondents indicated that they had never or rarely encountered “Semi functional teams” anti-pattern.

Lack of business knowledge, on the other hand, received the most responses about its occurrence, with more than 65% of the responses indicating that it happens sometimes, often, or always.

G. Completing the sprint

None of the anti-patterns classified in the sprint completion section are present in a significant proportion of respondents’ organizations. Table VIII and Figure 8 shows that only “The sprint does not end ends with a demonstration” was reported as often or always present in 36.8% of the responses.

Among the rest, however, “Missing retrospectives” and “Combining the two meetings into one” can still be distinguished as the ones with a significant proportion of “Sometimes”, “Often” and “Always” answers.

H. Summary of the results

To identify the most common anti-patterns, an prevalence rate was calculated based on the survey results. The weighted sum of responses was calculated using the following weights: “I don’t know” – 0, “Never” – 0, “Rarely” – 1, “Sometimes” – 2, “Often” – 3, and “Always” – 4.

The final score was then calculated for each sum as a fraction of the maximum score possible, i.e. 168. Table IX lists all the anti-patterns ordered by calculated score.

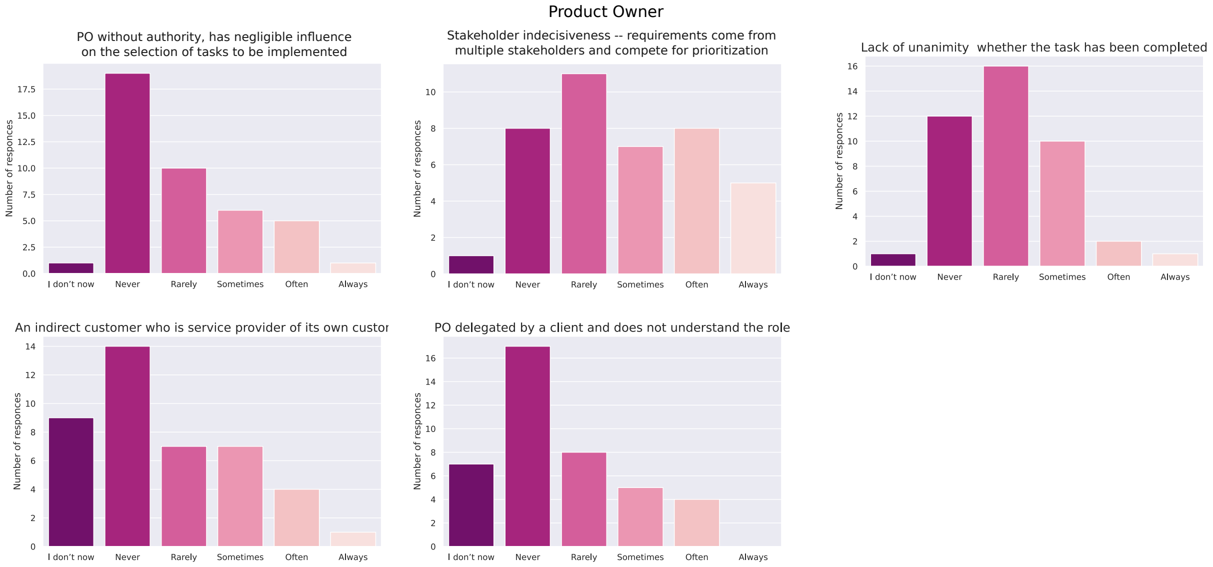


Fig. 5. Product owner related anti-patterns

TABLE VII
DEVELOPMENT TEAM RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
Semi functional teams	2	12	10	7	7	4
Insufficient technical knowledge	0	13	14	10	5	0
Lack of business knowledge	1	8	6	10	15	2

TABLE VIII
COMPLETING THE SPRINT RELATED ANTI-PATTERNS

Question	I don't now	Never	Rarely	Sometimes	Often	Always
No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	2	13	13	4	9	1
The sprint does not end ends with a demonstration	4	12	8	4	8	6
Demonstration for the wrong targets	10	13	6	6	5	2
Missing retrospectives	0	23	5	5	5	4
Combining the two meetings into one	4	19	6	5	3	5

V. DISCUSSION

In terms of the prevalence of anti-patterns studied, "Unfinished Tasks" received the highest score. A situation in which some tasks are not completed within a sprint is in clear contradiction to Scrum's guidelines, which states that "Sprint may be considered a short project" [14]. Of the respondents, only 3 people reported that such a situation rarely happens, and none that it never does. In our opinion, this is a serious violation of the guidelines and indicates not so much poor work by the development team, but rather poor planning of the tasks to be completed within the sprint.

The second most common anti-pattern is extending daily scrums beyond the recommended 15 minutes. Only one respondent reported never having such a situation and 9 rarely. This situation may stem from an incomplete understanding of the Scrum Guide, which openly states that "Daily Scrum is not the only time developers are allowed to adjust their plan" [14]. Daily Scrum should only focus on progress and planning for the next day's work. In practice, in addition to progress reports and planning for the next day's work, this meeting is often used to discuss other issues, such as resolving technical problems that have arisen. We believe this is not a serious problem, but it should be encouraged to limit these meetings to recommended purposes only, and other issues should be resolved in additional meetings, perhaps in a limited group.

Another anti-pattern with the comparable prevalence score is "User stories are not fully refined". This problem stems directly from the problem of requirements engineering. Unlike traditional software development methods, the Scrum methodology does not formally define how requirements are elicited. This can lead to less commitment and dedication to creating comprehensive user stories. Therefore, it is necessary to promote a balanced approach to requirements definition in

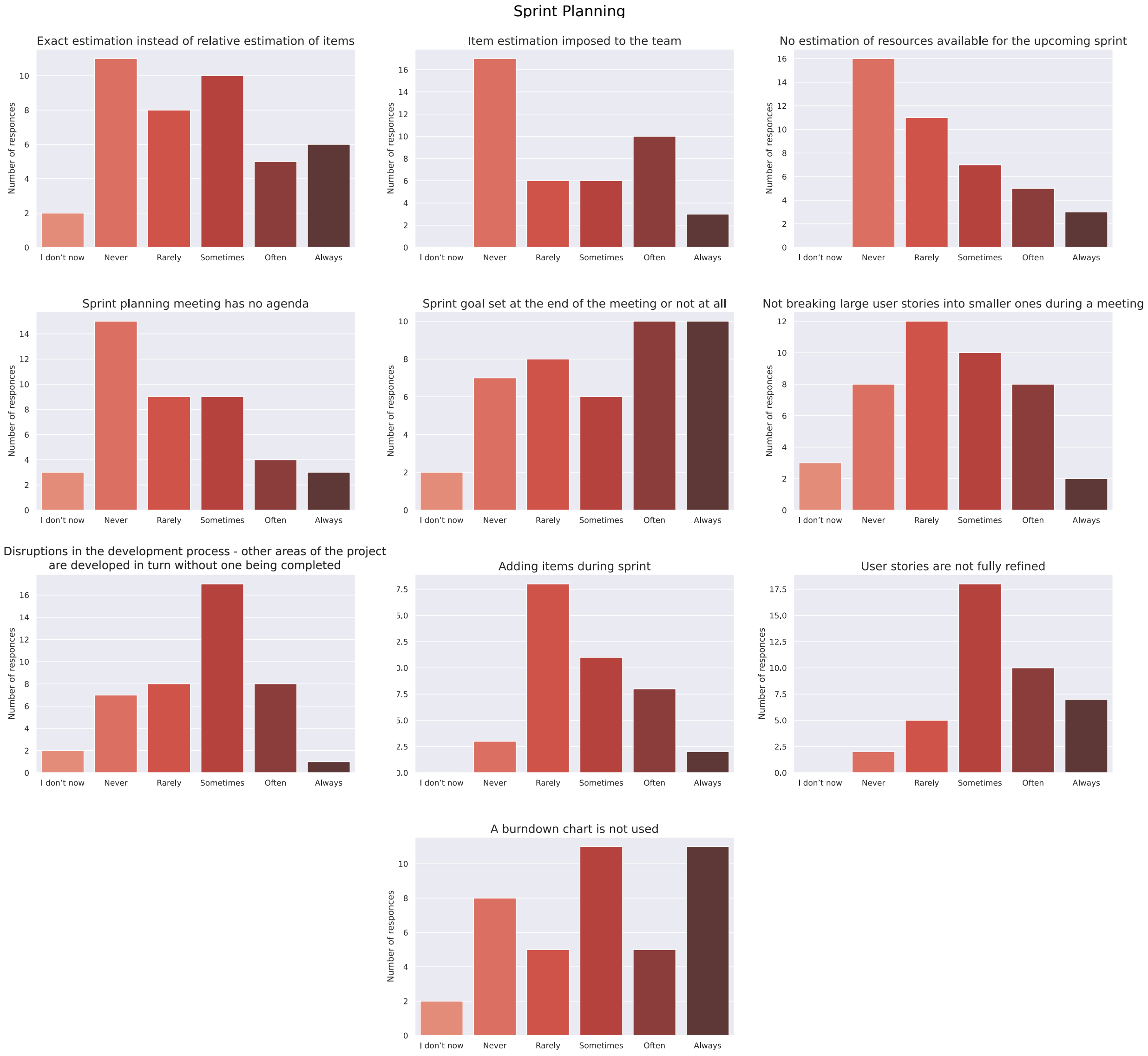


Fig. 6. Sprint planning related anti-patterns

agile software development organizations.

According to the Scrum Guide, the sprint goal should be defined at the sprint planning meeting. However, the survey results show that this is often not the case and that the goal is not defined at all or only at the end of the sprint. In our view, this is a serious violation of Scrum principles. Clear visibility of the purpose of the sprint leads to better coordination of work and greater involvement of team members.

Another anti-pattern in order of the prevalence score relates to not using the burndown chart. However, this is not the only way to visualize the project's progress. Therefore, we cannot draw too far-fetched conclusions here.

The last anti-pattern with prevalence score above 50% is "Testing in next sprint". This can be linked to the previously described prevalence of not completing tasks defined for the sprint. In our opinion, this is also a very important issue. Lack of testing leads to delivery and demonstration of potentially non-working software at the end of Sprint. Scrum guide defines increments that are produced within a single sprint must be usable. Without thorough testing, this cannot be guaranteed. Therefore, it is important to emphasize to the stakeholders of the Scrum project the importance of continuous testing of the delivered functionality.

The least common anti-patterns are related to the length of sprints and Daily Scrums. The respondents confirmed that in their organizations, the length of the sprint is usually within the guidelines and is constant throughout the project. Furthermore, Daily Scrums are held daily and at fixed times.

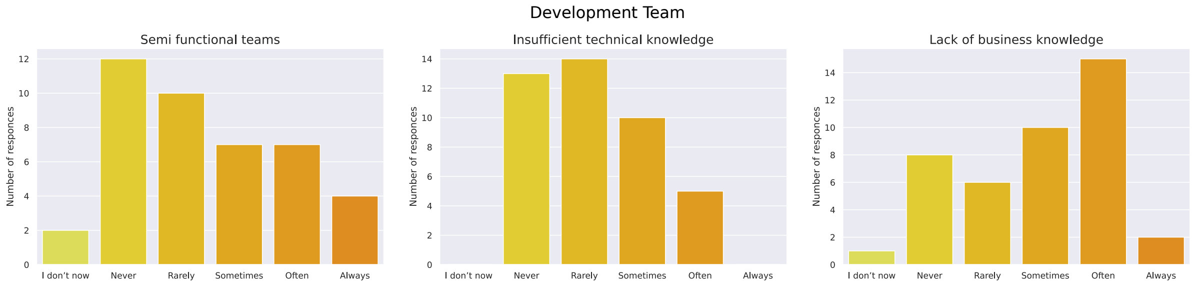


Fig. 7. Development team related anti-patterns

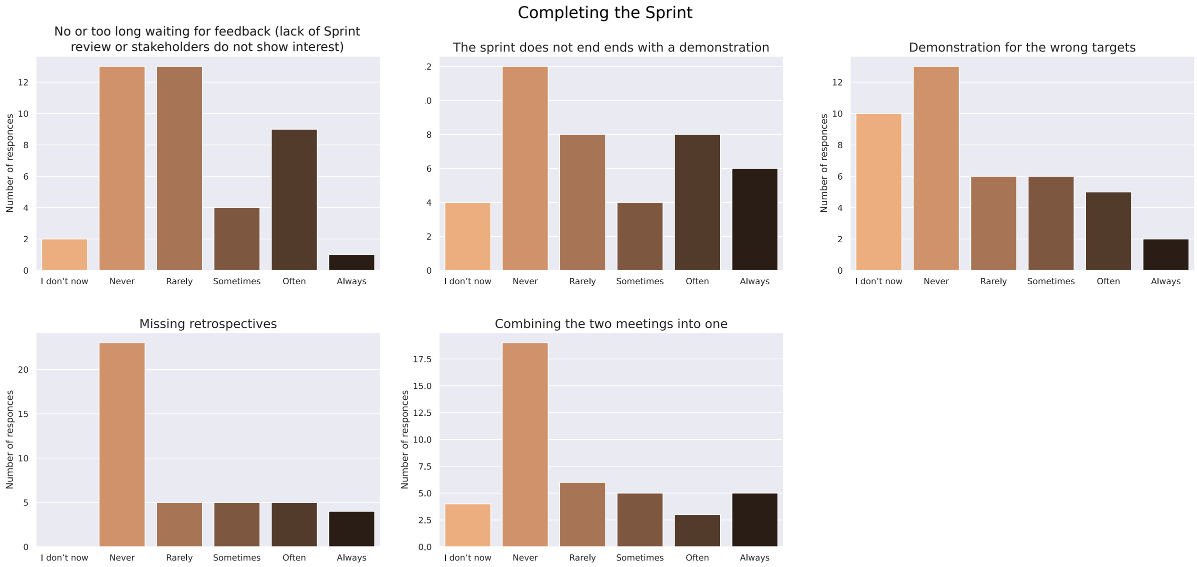


Fig. 8. Completing the sprint related anti-patterns

VI. Conclusion

The aim of the study was to determine the prevalence of 35 anti-patterns in Scrum software development methodology in Polish companies. Through a survey of industry professionals, we identified 6 most common issues in the organizations of the respondents. Among them, we believe that “Unfinished Tasks”, “User stories are not fully refined”, “Sprint goal set at the end of the meeting or not at all” and “Testing in next sprint” are the most critical. Although the presence of the remaining anti-patterns may not be critical to project success, they do indicate areas where teams can improve their adherence to Scrum principles. Overall, the findings of this study provide valuable insights into the current state of Scrum methodology in software development organizations.

While this study has provided valuable insights into the current state of Scrum methodology in software development organizations, further research is needed to understand the underlying causes of these issues. In future research, we aim to focus on investigating why these anti-patterns occur and what factors contribute to their persistence. This could be achieved through in-depth interviews with experienced project managers and Scrum Masters who have encountered these anti-patterns in their work. Such interviews could shed light on the organizational and cultural factors that lead to the breaking of Scrum rules and provide guidance on how to address these issues effectively. By gaining a deeper understanding of the root causes of these anti-patterns, organizations can take more targeted and effective action to improve their Scrum practices and increase the success of their projects.

References

[1] T. Raharjo and B. Purwandari, “Agile project management challenges and mapping solutions: A systematic literature review,” in *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, 2020. doi: 10.1145/3378936.3378949 pp. 123–129.

[2] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries *et al.*, “Manifesto for agile software development,” 2001.

[3] P. Weichbroth, “A case study on implementing agile techniques and practices: Rationale, benefits, barriers and business implications for hardware development,” *Applied Sciences*, vol. 12, no. 17, p. 8457, 2022. doi: 10.3390/app12178457

[4] S. Hassani-Alaoui, A. F. Cameron, and T. Giannelia, “"we use scrum, but ...": Agile modifications and project success,” in *Hawaii International Conference on System Sciences*, 2020. doi: 10.24251/HICSS.2020.765

TABLE IX
PREVALENCE SCORE OF ANTI-PATTERNS

	Anti-pattern	Weighted sum	Score
1.	Unfinished tasks	118	70.24%
2.	Daily Scrum lasts longer than 15 minutes	103	61.31%
3.	User stories are not fully refined	99	58.93%
4.	Sprint goal set at the end of the meeting or not at all	90	53.57%
5.	A burndown chart is not used	86	51.19%
6.	Testing in next sprint	85	50.60%
7.	Lack of business knowledge	79	47.02%
8.	Disordered product backlog	73	43.45%
9.	Adding items during sprint	72	42.86%
10.	Disruptions in the development process – other areas of the project are developed in turn without one being completed	70	41.67%
11.	Stakeholder indecisiveness – requirements come from multiple stakeholders and compete for prioritization	69	41.07%
12.	Invisible progress	67	39.88%
13.	Exact estimation instead of relative estimation of items	67	39.88%
14.	Not breaking large user stories into smaller ones during a meeting	64	38.10%
15.	The sprint does not end ends with a demonstration	64	38.10%
16.	Semi functional teams	61	36.31%
17.	Item estimation imposed to the team	60	35.71%
18.	Not all key issues are addressed	56	33.33%
19.	User stories that are too extensive	55	32.74%
20.	Not everyone actively participate in the meeting	54	32.14%
21.	No estimation of resources available for the upcoming sprint	52	30.95%
22.	No or too long waiting for feedback (lack of Sprint review or stakeholders do not show interest)	52	30.95%
23.	Sprint planning meeting has no agenda	51	30.36%
24.	Insufficient technical knowledge	49	29.17%
25.	Lack of unanimity whether the task has been completed	46	27.38%
26.	Missing retrospectives	46	27.38%
27.	Combining the two meetings into one	45	26.79%
28.	Extensive requirements documentation instead of user stories	42	25.00%
29.	PO without authority, has negligible influence on the selection of tasks to be implemented	41	24.40%
30.	Demonstration for the wrong targets	41	24.40%
31.	An indirect customer who is service provider of its own customer	37	22.02%
32.	PO delegated by a client and does not understand the role	30	17.86%
33.	Daily scrums are not held every day	21	12.50%
34.	Variable lengths of sprints	17	10.12%
35.	The Daily Scrum does not take place at a fixed time of day	12	7.14%
36.	Sprints that are too long	11	6.55%

[5] "State of agile report," https://digital.ai/resource-center/analyst-reports/state-of-agile-report/, 2022, accessed: 2023-07-31.

[6] V.-P. Eloranta, K. Koskimies, T. Mikkonen, and J. Vuorinen, "Scrum anti-patterns–an empirical study," in *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, vol. 1. IEEE, 2013. doi: 10.1109/APSEC.2013.72 pp. 503–510.

[7] C. Matthies, T. Kowark, K. Richly, M. Uflacker, and H. Plattner, "How surveys, tutors, and software help to assess scrum adoption in a classroom software engineering project," in *Proceedings of the 38th International Conference on Software Engineering Companion*, 2016. doi: 10.1145/2889160.2889182 pp. 313–322.

[8] P. J. Carew and D. Glynn, "Anti-patterns in agile adoption: A grounded theory case study of one irish it organisation," *Global Journal of Flexible Systems Management*, vol. 18, pp. 275–289, 2017. doi: 10.1007/s40171-017-0162-8

[9] M. Mortada, H. M. Ayas, and R. Hebig, "Why do software teams deviate from scrum? reasons and implications," in *Proceedings of the International Conference on Software and System Processes*, 2020. doi: 10.1145/3379177.3388899 pp. 71–80.

[10] E. Çetin and P. Onay Durdu, "Blended scrum model for software development organizations," *Journal of Software: Evolution and Process*, vol. 31, no. 2, p. e2147, 2019. doi: 10.1002/smr.2147

[11] V. T. Heikkilä, M. Paasivaara, and C. Lassenius, "Scrumbut, but does it matter? a mixed-method study of the planning process of a multi-team scrum organization," in *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2013. doi: 10.1109/ESEM.2013.27 pp. 85–94.

[12] T. McKenzie, M. Morales-Trujillo, S. Lukosch, and S. Hoermann, "Is agile not agile enough? a study on how agile is applied and misapplied in the video game development industry," in *2021 IEEE/ACM Joint 15th International Conference on Software and System Processes (ICSSP) and 16th ACM/IEEE International Conference on Global Software Engineering (ICGSE)*. IEEE, 2021. doi: 10.1109/ICSSP-ICGSE52873.2021.00019 pp. 94–105.

[13] T. Perry, "The intermediate customer anti-pattern," in *Agile 2008 Conference*. IEEE, 2008. doi: 10.1109/Agile.2008.74 pp. 280–283.

[14] K. Schwaber and J. Sutherland, "The scrum guide," https://www.scrum.org/resources/scrum-guide, 2020, accessed: 2023-04-10.

DOI: 10.15439/2023F3643
ISSN 2300-5963 ACSIS, Vol. 35

Analysis of the Impact of Data Augmentation on the Performance of Deep Learning Models in Multispectral Food Authenticity Identification

Yaru Zhang, Arif Yilmaz, Mirela Popa and Christopher Brewster
Department of Advanced Computing Sciences,
Faculty of Science and Engineering,
Maastricht University, 6200 MD, The Netherlands,
Email: {yaru.zhang, a.yilmaz, mirela.popa, christopher.brewster}@maastrichtuniversity.nl

***Abstract*—Food authenticity is a significant concern in the meat industry, demanding effective detection methods. This study explores the use of multispectral imaging (MSI) and deep learning for meat adulteration detection. We evaluate different deep learning models using transfer learning and preprocessing techniques in a multi-level adulteration classification task. In addition, we propose a novel approach called one-band mixed augmentation for band selection in MSI data, which outperforms traditional reflectance-based feature selection and enhances model robustness. Furthermore, employing the nine-crop approach for dataset augmentation improved the accuracy from 0.63 to 0.74 for DenseNet201 model without transfer learning. This research contributes to advancing food safety assessment practices and provides insights into the application of deep learning for preventing food adulteration. The proposed one-band mixed augmentation approach offers a novel strategy for handling band selection challenges in MSI data analysis.**

I. Introduction

FOOD safety has become a major issue in recent years, garnering significant attention from regulators and industry stakeholders alike. This issue is particularly critical when it comes to minced meat, which lacks distinctive morphological characteristics, making it more susceptible to intentional adulteration. Such fraudulent practices not only pose serious health risks to consumers but also undermine the integrity of the entire food supply chain, eroding public trust in the food industry. Consequently, it is imperative to adopt proactive measures for detecting and preventing food adulteration, ensuring the delivery of safe, reliable, and high-quality food to consumers.

Traditional methods for detecting meat adulteration typically involve destructive sample analysis, such as PCR analysis [1], are time-consuming and require specialized environments and trained professionals. Consequently, the need for effective and efficient techniques to detect food adulteration has become increasingly urgent. To address this challenge, researchers have investigated the use of non-contact technologies to address meat safety concerns, including the detection of fraud in processed meat using non-destructive spectroscopic methods [2]. Additionally, gas sensors have been employed for monitoring meat quality [3], while electronic noses have been utilized for monitoring meat spoilage [4]. These advanced technologies provide cost-effective and rapid alternatives to traditional methods, and their integration into food safety regulations reflects their increasing importance in ensuring the integrity of the food supply [5].

Multispectral imaging (MSI) has received considerable attention in recent years as a fast and non-destructive analytical approach to determining food quality and safety evaluation. MSI captures image data in specific wavelength ranges, providing spatial and spectral information of the object under analysis. Different meat qualities have different reflection intensities under different spectra [6], making it particularly useful for detecting food adulterants. It has been successfully used in various food safety applications, including the evaluation of microbial contamination in ready-to-eat vegetable salad [7], the assessment of cowpea seed health and differentiation of fungal species [8], and the evaluation of ready-to-eat pineapple quality [9]. Additionally, MSI has been used to estimate microbial spoilage in minced pork [10].

Machine learning models, such as partial least squares regression (PLSR) and support vector machines (SVM), have been applied to detect meat adulteration using MSI data [11]–[14]. However, current research on MSI for meat adulteration often utilize only limited attributes of the image data, such as mean and standard deviation, which can be a limitation in terms of accuracy and reliability. This limitation leaves room for improving the accuracy and reliability of meat quality control systems.

To address this limitation and further enhance meat quality control, the combination of MSI with deep learning techniques, such as convolutional neural networks (CNNs), has gained attention. We can improve the accuracy and reliability of meat quality control systems by using these models for image classification tasks in the food domain, specifically for meat adulteration. Although some research has explored the use of CNNs for image classification [15], there is still a gap in the literature on using these models for meat quality control systems.

A previous study developed a framework for coffee maturity classification with 15 bands of multispectral data based on CNNs and achieved a relevant high accuracy on five classes [16], achieved up to 98% accuracy on the dataset and 100% accuracy on cross-validation. However, this approach has not been applied to the problem of meat adulteration and spoilage in MSI data.

By leveraging the comprehensive information provided by MSI and harnessing the capabilities of deep learning, we seek to develop more effective methods for preventing and detecting meat adulteration. In order to facilitate the goals of our study, it was necessary to repreprocess data specifically for meat adulteration, as there was no readily available MSI image dataset for this purpose. We also adapt state-of-the-art CNN models and perform several optimizations to ensure their effectiveness in analyzing the acquired MSI data. Additionally, we explore the potential of leveraging the rich information contained in MSI data by experimenting with different preprocessing approaches.

Through our research, we aim to provide valuable information on the utilization of MSI and deep learning techniques, which can lead to the development of advanced approaches to ensure food safety and preserve the integrity of the meat supply chain. The findings of our study hold great promise for substantial advancements in current practices, leading to the development of more efficient and dependable methods for preventing and detecting food adulteration. As a result, these advancements will play a crucial role in safeguarding the safety and integrity of the food supply chain.

II. Methodology

In this study, we conducted experiments using 180 minced meat samples from 9 adulteration classes. We extracted multi spectral images in 18 bands and encoded and resized them into the required size by a deep learning model. We started with fine-tuning SOTA CNNs models to detect patterns and features indicative of meat adulteration, but the particularities of our image dataset indicated that training from scratch might be a better option for learning relevant features. The best-performing model was selected as the baseline. Additionally, we explored three different pre-processing modalities to assess their impact on the model's performance.

A. Datasets

The data acquisition process followed the pipeline illustrated in Fig. 1. Our study utilized a dataset consisting of MSI images depicting chicken and pork meat samples with varying levels of adulteration. The levels of adulteration spanned from 0% (indicating pure chicken) to 100% (representing pure pork), with nine intervals in between: 0%, 10%, 25%, 40%, 50%, 60%, 75%, 90%, and 100%. Chicken and pork were purchased from four different butcher shops (b_1, b_2, b_3, b_4) in Greece. Samples from each butcher shop contained five instances per adulteration level resulting in 45 samples per butcher shop. In total, the dataset contains 180 samples from four butcher shops.

The images were acquired using the Videometer lab system developed by the Technical University of Denmark and commercialized by "Videometer A/S" (http://www.videometer.com). The MSI images consist of 18 bands for each meat sample. The samples were kept at 4°C and captured after 24 h. There are a total of 180 sample image files of various temperatures and various adulteration levels. The final size of the dataset is 3240 (180x18) grey-scale images from 180 meat samples. The dataset includes nine classes, each representing a different level of adulteration.

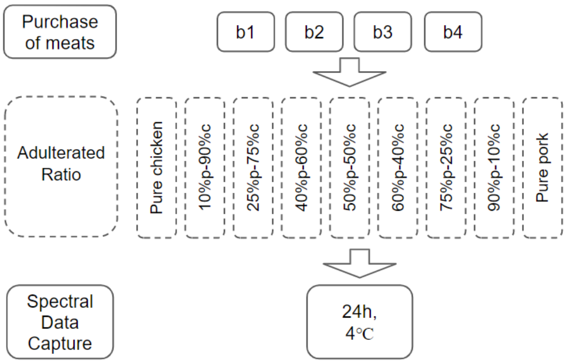


Fig. 1. Multispectral Imaging Acquisition pipeline

B. Data Preprocessing

Proper data preprocessing is crucial as it is the foundation for subsequent data analysis. By performing appropriate data preparation techniques, we can guarantee that the analysis results are reliable and carry significant implications. Additionally, it allows us to address any potential issues or biases in the data and optimize the performance of our machine learning model.

TABLE I
Details of Image Bands and Corresponding Wavelengths Used by Videometer Lab MSI Camera for Capturing MSI Dataset.

the band and wavelength details						
band	band1	band2	band3	band4	band5	band6
region	UV	Violet	Blue	Blue	Cyan	Green
wavelength	405nm	435nm	450nm	470nm	505nm	525nm
band	band7	band8	band9	band10	band11	band12
region	Green	Yellow	Red	Red	Red	Red
wavelength	570nm	590nm	630nm	645nm	660nm	700nm
band	band13	band14	band15	band16	band17	band18
region	NIR	NIR	NIR	NIR	NIR	NIR
wavelength	850nm	870nm	890nm	910nm	940nm	970nm

a) Image Preprocessing: Each sample in the dataset contains 18 grey-scale images of 18 non-uniformly distributed wavelengths with a size of 1200 by 1200 pixels. Each image represents a spectral feature of a sample in a particular band. Table I presents a detailed overview of the chosen wavelength bands utilized in our study. The selected wavelengths cover a spectrum ranging from 405nm to 970nm, comprising a total of 18 bands. Notably, this includes one band in the ultraviolet (UV) region and six bands within the near-infrared (NIR) region. This information provides a comprehensive understanding of the specific wavelengths employed in our research analysis. Fig. 2 shows all 18 bands of a sample which is 10%pork-90%chicken.

In order to accommodate the large extracted images within the proposed models, we first resize the images to a standardized size of 224 by 224 pixels. This resizing ensures uniformity and compatibility across the dataset. Following the resizing step, we employ min-max scaling to encode the images as values ranging from 0 to 1. This preprocessing technique effectively normalizes the pixel values, allowing for efficient handling and analysis of the data

b) Label Extraction: The images were labeled based on the information contained within their names, including the adulteration level, band number, sample name, and storage condition. We converted it into integers ranging from 0 to 8. Specifically, '0' denotes pure chicken, while '8' represents pure pork. For adulteration levels between 10% to 90%, we assigned integer values from 1 to 7 to represent varying ratios of pork and chicken: 10% pork - 90% chicken, 25% pork - 75% chicken, 40% pork - 60% chicken, 50% pork - 50% chicken, 60% pork - 40% chicken, 75% pork - 25% chicken and 90% pork - 10% chicken. This scale indicates the percentage of pork and chicken present in each sample, irrespective of which meat has adulterated the other. In the case of pork-adulterated chicken, a smaller scale number denotes a higher level of adulteration. We chose to utilize a single scale to simplify the paper, instead of employing separate scales for each meat species.

While it is important to label each image accurately, we also wanted to ensure that the labels were practical for the intended application. In this case, we use one-hot encoding to encode the image labels into a numerical format, which assigns a unique numerical value to each category. This approach enabled us to quickly generate statistics and analyze the model performance based on different adulteration levels.

C. Basic Adulteration Classification Pipeline

This study focuses on developing an automated method for detecting adulteration in meat samples using multispectral image analysis. The problem is approached as a classification task, where the performance of various deep-learning models is compared. To ensure consistency, we employ a standardized classification pipeline. This involves inputting an array with dimensions (224, 224, 18), representing the 18 bands of information in each sample, into the models. The models are then trained to predict the degree of adulteration based on the input image array.

To evaluate the effectiveness of different CNN-based models, we compare five models available in the Keras library [17]. The initial selection includes VGG16 and VGG19 [18], which serve as established benchmarks for image classification tasks. Additionally, Inception-ResNetV2 [19] and InceptionV3 [20] are chosen for their superior performance in computer vision tasks. Furthermore, we include DenseNet [21], known for its promising outcomes in similar studies.

To leverage pre-existing knowledge, transfer learning is applied to the selected CNN models. This allows us to explore if pre-trained models can enhance the performance of our task. Fig. 3 illustrates the basic experiment pipeline of the CNN models, providing an overview of the process.

We evaluate the performance of our models using two methods. Firstly, we perform a simple train/test split by partitioning the data into two sets, with the training set containing 80% of the data and the test set containing 20%. Given the limited number of samples in our dataset, we also use 5-fold stratified cross-validation (SCV) to evaluate the models more precisely.

The models are trained using the backpropagation algorithm. This algorithm works by calculating the loss function gradient concerning the network weights and using this gradient to update the weights in a direction that minimizes the loss function. This process is repeated iteratively until the network converges on a set of weights that minimizes the loss function. The specific methods and hyperparameters depend on each model. We evaluate the models using accuracy, precision, recall, and F1 score. We chose these metrics to understand the models' performance thoroughly.

D. Transfer Learning

To enhance the classification performance of our model, we opted to incorporate transfer learning into our training process and to evaluate its effectiveness. Given the limited nature of our dataset, transfer learning was considered a potential solution to optimize the model and improve its accuracy. The base model weights obtained from ImageNet [22], which consists of millions of images and 1000 labels, were used to initialize our model and provide a solid foundation for further training.

When dealing with datasets that contain more than three channels, such as our 18-channel multispectral data, transfer learning requires adjusting the pre-trained weights to accommodate the additional channels. Fine-tuning is performed on the first convolutional layer, configured to enable the neural network to read 18-channel images. In this process, the weights of the first convolutional layer are modified to accept the input of 18 channels. This is done by averaging the pre-trained weights of the first convolutional layer's three channels and replicating the resulting weights 18 times to accommodate all 18 input channels.

To align with our experiment's 9-class classification objective, we modified the fully connected output layer of all five models from 1000 to 9. Furthermore, to adapt transfer learning to our specific task, we made variations in the trainable layers of each utilized network. The trainable and non-trainable layers of the models are specified below.

- **VGG-16** consists of 16 trainable layers. While the first two layers remain fixed, the subsequent 14 layers are trainable. The original fully connected layers of 4096 and 4096 were replaced by adapted layers with sizes of 256 and 128, respectively.
- **VGG-19** comprises 19 trainable layers, where the first seven layers are untrained, while the remaining 12 layers are trainable. Similar modifications were made to the

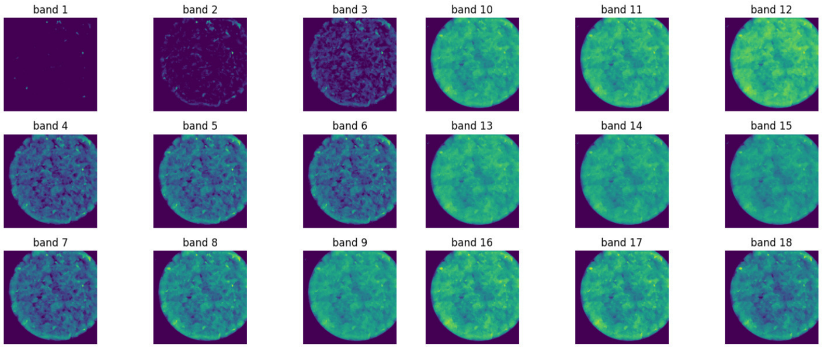


Fig. 2. The 18 bands for a 10% pork-90% chicken sample.

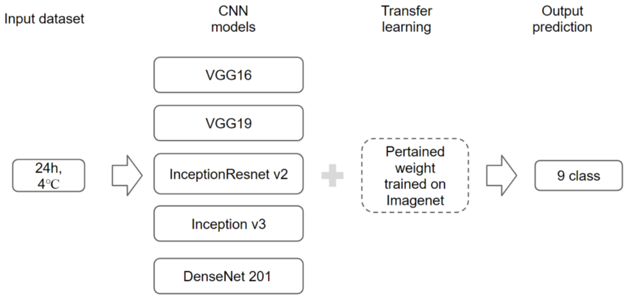


Fig. 3. Baseline CNN Classification Experiment Design

fully connected layers, adjusting their original sizes of 4096 and 4096 to sizes of 256 and 128.

- **InceptionV3** has 310 trainable layers. The first 150 layers are untrained, while the rest are made trainable. The original fully connected layer, initially sized of 2048, is replaced with two new fully connected layers sized of 256 and 128, respectively.
- **Inception-ResNetV2** The first 150 layers are untrained, while the remaining 578 are trainable. The original fully connected layers, sized of 1536, are substituted with two new fully connected layers sized of 256 and 128.
- **DenseNet201** has 706 trainable layers, with the first 150 layers left untrained and the remaining layers made trainable. Two new fully connected layers were introduced, having sizes of 256 and 128, respectively.

By adjusting the trainable layers in the specified way, the neural networks were fine-tuned to better suit our specific problem and data characteristics. The modified models were then used to conduct our experiments and to analyze their performance.

E. Hyperparameter Optimization

After experimenting with several optimizers, including Adam, Adamax, Adamgrad, and SGD, we selected the Adam optimizer for its superior performance. Then, we set the output layer with softmax activation function.

We utilized a learning rate scheduler function to optimize our model's performance. Our approach involved setting an initial learning rate of 0.0001 and employing an exponential decay function that reduced the learning rate by 0.095 at each epoch. The exponential decay scheme provided a smooth decay path, which was particularly effective during the initial stages of training. This strategy helped to improve the learning capacity of the model and yielded better results in our experiments.

F. Cross Validation

Cross-validation (CV) is a widely used technique in machine learning and data analysis to evaluate the predictive performance of models. It helps optimize hyperparameters, identify dataset issues, and prevent overfitting, ultimately improving the effectiveness of models in real-world applications. Stratified cross-validation is an essential variant of CV when working with imbalanced datasets. It ensures that each fold contains representative samples from all classes in the same proportion as the original dataset, mitigating the risk of biased assessments of model performance. By using stratified cross-validation, we can obtain more reliable estimates of the model's generalization capabilities and make better-informed decisions about its suitability for real-world applications.

For our experiments, we used a five-fold stratified cross-validation (5-SCV) approach. The data was divided into five folds, each containing an equal distribution of samples from all classes. The models were then trained on four folds and validated on the remaining one. This process was repeated five times, with each fold used as the validation set once. The stratified aspect of the cross-validation ensured that the class distribution was maintained across all folds, preventing bias in evaluating the models' performance. The final performance metrics were calculated as the average of the five iterations, providing a comprehensive and accurate assessment of the effectiveness of the models.

G. Model Improvements with Various Data Augmentation Configurations

To enhance the classification accuracy, we conducted various experiments on the dataset. We explored several techniques, such as removing uninformative bands, augmenting the training set, and cropping the original image to gather more information. Given the amount of conditions and considered models, we chose to focus on one model (e.g. DenseNet 201 model). Stratified sampling was applied so that 80%-20% of the dataset to be used for training and testing for all experiments. Specifically, for the basic adulteration classification and uninformative bands excluded experiments, the training set consisted of 144 images with 36 testing images. For one band augmentation experiment, the training set was extended to 288 images, while the test set remained at 36 images. The purpose of this was to determine whether augmentation could help the model learn more details about the features needed to classify the original images, rather than the augmented images. In the cropping experiments, the training set is increased to 576 images (for 4 crops) and 1296 (for 9 crops), while the test set size remains constant for consistency reasons.

1) Band Selection: Exclude Uninformative Bands using a Reflectance-based Method: To further refine the dataset and improve the classification performance of CNN models, we conducted a band exclusion approach based on previous research by L.-C. Fengou, P. Tsakanikas, and G.-J. E. Nychas [23]. This previous study compared the mean and standard deviation of the wavelength reflectance of pure chicken and pure pork at different storage times (0 h, 24 h, 48 h) and identified the wavelength from 700 to 940 nm as uninformative, which corresponds to bands 12 to 17 in our dataset. This exclusion was based on the overlap of wavelength reflectance. It is worth noting that the band exclusion approach has improved classification accuracy in previous studies on similar datasets. By excluding uninformative bands, the amount of noise in the data was reduced, and the signal-to-noise ratio was increased, factors which could improve the performance of the models.

To evaluate the impact of the band exclusion on classification performance, we trained the CNN models on a 12-band dataset, where the uninformative bands were excluded. We selected DenseNet201 as our base model without transfer learning and trained it on the 9-class classification task.

2) Optimizing Band Selection: Exclude Uninformative Bands using One-band Mixed Augmentation: Data augmentation is a widespread technique in deep learning used to increase the size and diversity of the training dataset. In our experiments, we applied mixed augmentation to enhance the diversity and size of our training dataset. This technique involves applying various transformations, such as zooming, rotating, shifting, and flipping, to the original data to create new and unique images while preserving the properties of the original data. Zooming allowed us to change the scale of the images, while rotation and shearing enabled us to modify the orientation and shape of the objects within the images. The width and height shifting helped to translate the objects in the images, while the horizontal flipping created a mirror image of the original one. The mixed augmentation method selects a random combination of transformations from the set specified in the augmentation pipeline. The chosen transformations include zooming between 40% and 80% of the original size, rotating the image up to 45 degrees, shifting the width and height of the image by up to 10%, shearing the image up to 20%, and flipping the image horizontally.

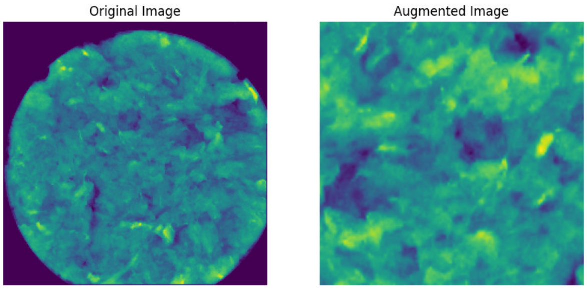


Fig. 4. Example of Augmentation

Fig. 4 shows one augmentation of one 25% pork-75% chicken sample in the dataset. The image underwent several augmentations. First, it was rotated at an angle of -34.97 degrees in a counterclockwise direction. Second, it was translated horizontally by -0.075 and vertically by 11.85 pixels. Third, it was sheared by -0.175, meaning that the object's shape in the image was distorted. Fourthly, the image was zoomed in by a factor of 0.74 along the x-axis and 0.50 along the y-axis. Finally, the image was flipped horizontally.

Multispectral images contain several bands of information, each of which may have varying contributions to the classification performance. In order to determine which bands are more informative for our models, we conduct data selection experiments. To do this, we first apply data augmentation techniques individually to each band in the dataset. Then, we combine the augmented bands and train models on each combination of bands. In this experiment, data augmentation was performed for each band in the dataset, by splitting the data into 144 samples for training and 36 for testing. We applied the described band augmentation process to the training set, which increased the training set size to 288 (for each sample, one band was augmented). Each sample had a shape of (224,224,18), where 224 represents the width and height of the image, and 18 represents the number of bands. Consequently, the final input training shape became (288, 224, 224, 18). Fig. 5 shows the augmentation pipeline of our data.

We evaluate the performance of each model and compare the results to a baseline model (DenseNet 201 without transfer learning) without augmentation. By comparing the performance of all combinations, we can identify the bands that have performed above the baseline and are therefore considered informative. This process allows us to identify the bands that provide the most useful information for classification and can help optimize the selection of bands for future experiments. Furthermore, this approach can be used to investigate the impact of data augmentation on individual bands and could help us understand the effect of each augmentation technique on the overall classification performance.

3) Augmentation by Applying Cropping to All Bands: To strike a balance between the amount of information conveyed

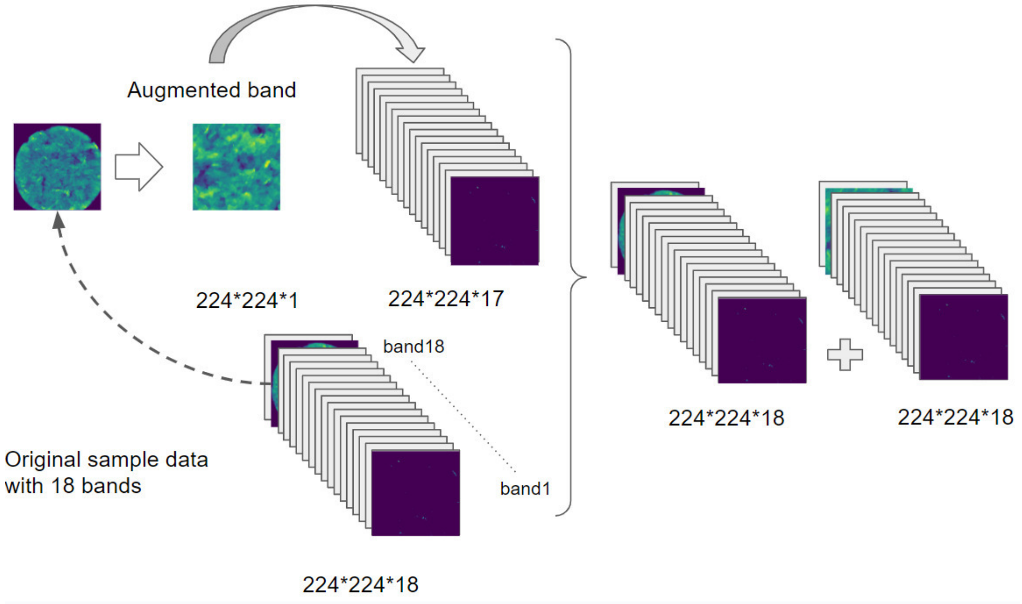


Fig. 5. Pipeline of One-band mixed Augmentation.One band of the original samples is augmented, replacing the original image. The augmented samples are combined with the original samples to form the training set.

by an image and the computational cost required for processing it, image cropping is a beneficial preprocessing technique. By cropping larger images into smaller clips, we can increase the dataset size without sacrificing crucial information, especially when obtaining additional datasets in the same field is challenging.

For the baseline classification, we resized the original images, which had dimensions of (1200 x 1200) pixels, to (224 x 224) pixels. However, this resizing process may result in a loss of information and potentially impact the accuracy of the classification results. To address this challenge, we propose an approach that involves cropping the raw images into four or nine clips. Through experimentation, we determined that four clips, each measuring (600 x 600 pixels), or nine clips, each measuring (400 x 400) pixels, were the most suitable sizes.

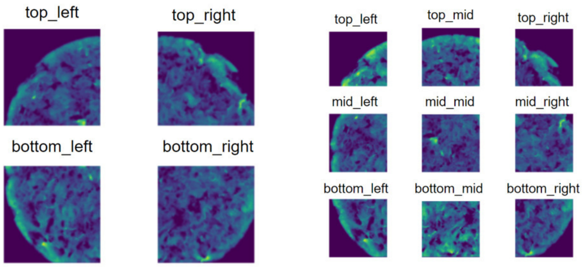


Fig. 6. Example of Image Cropping preprocessing. This image is randomly chosen from the MSI dataset and shows the 4-cropped and 9-cropped versions of the original image.

To maintain the spatial relationship between each cropped image and its original location, we incorporated position information into the extracted CSV file, alongside the corresponding label. This facilitated a clear understanding of the relative location of each cropped clip throughout the data preprocessing pipeline. For the four-cropped approach, we used position information such as lt (top left), lr (bottom left), lb (bottom right), and rb (top right). For the nine-cropped approach, position information included lt (top-left), mt (top-middle), rt (top-right), lm (middle-left), mm (middle-middle), rm (middle-right), lb (bottom-left), mb (bottom-middle), and rb (bottom-right). An example of the cropped images is shown in Fig. 6.

This approach effectively increases the dataset size while preserving the essential information from the original images. Additionally, the inclusion of position information provides valuable context for interpreting and analyzing the cropped clips. Overall, cropping and incorporating position information are effective preprocessing techniques for MSI data, enhancing the analysis quality and facilitating the utilization of these data in machine learning models. In our study, we performed cropping on the original image data, generating four and nine crops, respectively.

III. Results

In this study, we present a comprehensive evaluation of the efficacy of various deep learning models.

We used well-known CNN models including VGG16 and VGG19, Inception-Resnet v2, Inception v3, and DenseNet201 to explore their performance for 9-class adulteration classification task. To further optimize model performance, we leveraged transfer learning techniques.

Furthermore, we experimented with various data augmentation configurations such as rotation, shifting, schearing, flipping and zooming.

We also explored the impact of band selection on model performance, including the exclusion of non-informative bands based on reflectance and augmentation experiments in this study.

By evaluating the performance of various models, our objective was to provide insights into the selection of the most appropriate deep-learning architecture and preprocessing techniques for meat adulteration detection.

A. Best-performing model identification for meat adulteration classification

Table II shows the performance of different deep learning architectures on a 9-class classification task with and without transfer learning. The details of each model is explained in II-D. The results showed that DenseNet201 without transfer learning achieved the best accuracy of 0.63 and a precision of 0.64, while DenseNet201 with transfer learning achieved the best accuracy of 0.62 and the precision of 0.61 on all data sets and combinations.

B. Evaluation of baseline model performance for various data augmentation configurations

As DenseNet201 was the best-performing model, it was selected as a baseline for later experiments. The details of data augmentation are explained in section II-G2. The experiments focus on investigating the effects of band selection and augmentation techniques on the model performance. The results of these experiments are summarized in Table III, demonstrating that excluding non-informative bands improves model accuracy. Furthermore, excluding uninformative bands based on augmentation is shown to enhance model performance, with varying influences observed for different bands. Additionally, the use of all-band cropping augmentation, particularly employing the nine-crop approach, leads to the best results. These findings highlight the significance of band selection and augmentation methods in improving model performance for multispectral imaging data, contributing valuable insights to the field.

1) Band Selection: Exclude Uninformative Bands using a Reflectance-based Method: In this experiment, we trained the baseline model on a 12-band dataset for the 9-class classification task. The bands were selected based on the mean and standard deviation of reflectance for pure classes. The results showed that using 12 bands outperformed models using 18 bands in all metrics, which achieved an accuracy of 0.69 while using 18 bands achieved an accuracy of 0.63.

2) Optimizing Band Selection: Exclude Uninformative Bands using One-band Mixed Augmentation: A mixed band augmentation which is detailed in section II-G2 was applied to the baseline model. Table.IV shows the mean performance of the model under varying band augmentations. Our experimental findings indicate that preprocessing the input images with different band augmentations has a considerable impact on the model's learning capacity. In order to balance computational cost and experimental accuracy, we employed two different random seeds for conducting the experiments.

The average values of the evaluation metrics are presented as the experimental results. In particular, our comparison of the results with the best baseline model introduced in Table II (achieving 0.63 accuracy for the 9-class DenseNet model without augmentation) led to a decrease in performance, including bands: 1, 2, 5, 8, 12, 13, 15 and 16. In contrast, we found that some bands, such as bands: 3, 4, 6, 7, 9, 10, 14, 17, and 18, improved the model's performance.

Fig.7 shows that band 4 (470 nm, blue), band 14 (870 nm, NIR), and band 17 (940 nm, NIR) are the top 3 most informative bands for the dataset using the DenseNet201 model (without transfer learning). The augmentation on band 17 (940nm, NIR) increases the accuracy from 0.72 to 0.81. Therefore, the choice of performing band preprocessing is critical in optimizing the model's accuracy for this classification task.

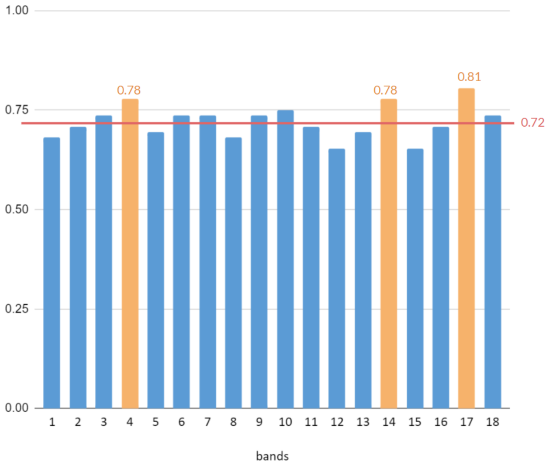


Fig. 7. Comparison of accuracies for various band augmentations on the dataset. The orange columns indicates the top 3 performing augmentation bands, the red line represents the baseline accuracy of the 9-class experiment without augmentation.

Based on the accuracies presented in Table IV, the six lowest performing bands (band1, band5, band8, band12, band13, and band15) were removed. The remaining 12 bands were stacked. The training process for this experiment adhered to the same settings as described in the baseline experiment. The 5-fold cross-validation produced average performance metrics, achieving 0.72 accuracy and 0.72 F1 score.

3) Augmentation by Applying Cropping to All Bands: In order to address the limitations posed by the limited size of the MSI dataset, we used the 4-crop and 9-crop approach to augment the entire 9-class data. Specifically, for the 4-cropped datasets, each class comprised 80 samples, and we used 80% of the dataset for training and 20% for testing, by making sure that none of the cropped versions of the original image would be found in both sets. We trained the baseline model in all these experiments. For the 4-cropped datasets, our experiment achieved 0.71 accuracy for 5-fold stratified cross validation. For the 9-cropped datasets, each class had 180 samples, resulting after augmentation in a total of 1620

TABLE II
PERFORMANCE OF VARIOUS MODELS WITH 5 FOLDS CROSS-VALIDATION WITHOUT AND WITH TRANSFER LEARNING USING 18-BAND 9-CLASS MSI DATA.

CNN models	Without transfer learning				With transfer learning			
	Accuracy(±SD)	F1 score(±SD)	Recall(±SD)	Precision(±SD)	Accuracy(±SD)	F1 score(±SD)	Recall(±SD)	Precision(±SD)
VGG16	0.40 (+/- 0.16)	0.38 (+/- 0.19)	0.40 (+/- 0.16)	0.49 (+/- 0.22)	0.53 (+/- 0.09)	0.52 (+/- 0.10)	0.53 (+/- 0.09)	0.60 (+/- 0.14)
VGG19	0.47 (+/- 0.13)	0.46 (+/- 0.16)	0.47 (+/- 0.13)	0.53 (+/- 0.22)	0.52 (+/- 0.12)	0.52 (+/- 0.12)	0.52 (+/- 0.12)	0.63 (+/- 0.09)
Incep-Res v2	0.61 (+/- 0.07)	0.60 (+/- 0.09)	0.61 (+/- 0.07)	0.69 (+/- 0.08)	0.55 (+/- 0.08)	0.56 (+/- 0.07)	0.55 (+/- 0.08)	0.72 (+/- 0.03)
Inceptionv3	0.56 (+/- 0.07)	0.54 (+/- 0.09)	0.56 (+/- 0.07)	0.62 (+/- 0.12)	0.57 (+/- 0.06)	0.57 (+/- 0.06)	0.57 (+/- 0.06)	0.63 (+/- 0.04)
DenseNet201	**0.63 (+/- 0.12)**	**0.64 (+/- 0.12)**	**0.63 (+/- 0.12)**	**0.75 (+/- 0.09)**	**0.62 (+/- 0.07)**	**0.61 (+/- 0.08)**	**0.62 (+/- 0.07)**	**0.70 (+/- 0.10)**

TABLE III
PERFORMANCE COMPARISON OF DATA CONFIGURATIONS WITH 5-FOLD STRATIFIED CROSS-VALIDATION

	Accuracy(±SD)	F1 score(±SD)	Recall(±SD)	Precision(±SD)
All Bands with Baseline (18 Bands)	0.63 (+/- 0.12)	0.64 (+/- 0.12)	0.63 (+/- 0.12)	0.75 (+/- 0.09)
Uninformative Bands Excluded based on Reflectance	0.69 (+/- 0.06)	0.68 (+/- 0.04)	0.69 (+/- 0.06)	0.76 (+/- 0.01)
Uninformative Bands Excluded based on Augmentation	0.72 (+/- 0.08)	0.72 (+/- 0.08)	0.72 (+/- 0.08)	0.78 (+/- 0.08)
All Bands, Image Cropped to 4 Parts	0.71 (+/- 0.11)	0.71 (+/- 0.13)	0.72 (+/- 0.11)	0.74 (+/- 0.10)
All Bands, Image Cropped to 9 Parts	**0.74 (+/- 0.08**)	**0.73 (+/- 0.09**)	**0.74 (+/- 0.08**)	**0.79 (+/- 0.10**)

TABLE IV
PERFORMANCE OF THE BASELINE MODEL BASED ON ONE-BAND MIXED AUGMENTATION

	band 1	band 2	band 3	**band 4**	band 5	band 6	band 7	band 8	band 9
Accuracy	0.68	0.71	0.74	**0.78**	0.69	0.74	0.74	0.68	0.74
F1 score	0.69	0.70	0.76	**0.80**	0.75	0.76	0.74	0.71	0.77
Recall	0.74	0.75	0.81	**0.83**	0.78	0.79	0.78	0.76	0.84
Precision	0.71	0.71	0.82	**0.84**	0.80	0.78	0.74	0.74	0.81

	band 10	band 11	band 12	band 13	**band 14**	band 15	band 16	**band 17**	band 18
Accuracy	0.75	0.71	0.65	0.69	**0.78**	0.65	0.71	**0.81**	0.74
F1 score	0.76	0.73	0.68	0.74	**0.81**	0.68	0.73	**0.82**	0.76
Recall	0.79	0.78	0.72	0.80	**0.87**	0.72	0.77	**0.85**	0.80
Precision	0.77	0.75	0.70	0.78	**0.83**	0.71	0.76	**0.82**	0.80

samples. We used the same train-test split procedure as above and achieved improved results, with an accuracy of 0.78. The confusion matrices of the three different inputs are visualized in Fig.8. 5-fold stratified cross validation is applied on the 9-crop case, achieving an average accuracy of 0.74.

IV. DISCUSSION

In this study, we examine the effects of various augmentations on the MSI dataset, using CNN based deep learning models for meat adulteration detection. We also inspect the effect of transfer learning and data preprocessing on their performance. The best configuration for the 18- Band, 9 class classification is found as DenseNet201 without transfer learning with an accuracy of 0.63 and F1 score of 0.64.

First we evaluated the performance of different CNN architectures on 9-class classification tasks with and without transfer learning. Our results showed that in the 9-class classification, DenseNet201 achieved the best accuracy with and without transfer learning.

Our findings suggest that the performance of CNN architectures can be influenced by their nature and design. For example, DenseNet201 is composed of densely connected layers, where each layer within a block receives the outputs from all preceding layers within the same block. This architecture promotes feature reuse and information flow, mitigating the vanishing gradient problem.

The number of trainable layers is important because it affects the depth and complexity of the network. A deeper network with more trainable layers has the potential to learn more complex features and patterns in the data. This may explain why Inception-Resnet v2 and DenseNet201 outperformed Inception v3 and VGG architectures in our study. The higher number of trainable layers in these architectures allows them to capture more intricate and nuanced information in the data, leading to improved performance in the classification tasks.

The achieved results indicated that transfer learning did not lead to a significant improvement in performance.This is consistent with the findings of a previous work[24] , they suggested that models trained from scratch can perform just as well as those that are pre-trained, even with substantially less data.

In addition to model architecture, the data configuration was found to have essential impact on the model performance. The experiment excluding uninformative bands chosen by reflectance revealed that the reflectance-based method improves classification performance. As shown in Fig.9, the reflectance of different adulterated samples in the dataset experienced slight changes as the adulteration level increased. By removing non-informative bands (700 to 970 nm), the model achieved better performance using only 12 bands compared to using all 18 bands for the MSI dataset.

Additionally, the experiment demonstrated that the performance improvement achieved by using 12 bands was consistent across different folds of the cross-validation, as indicated by the low standard deviation of the metrics. This consistency

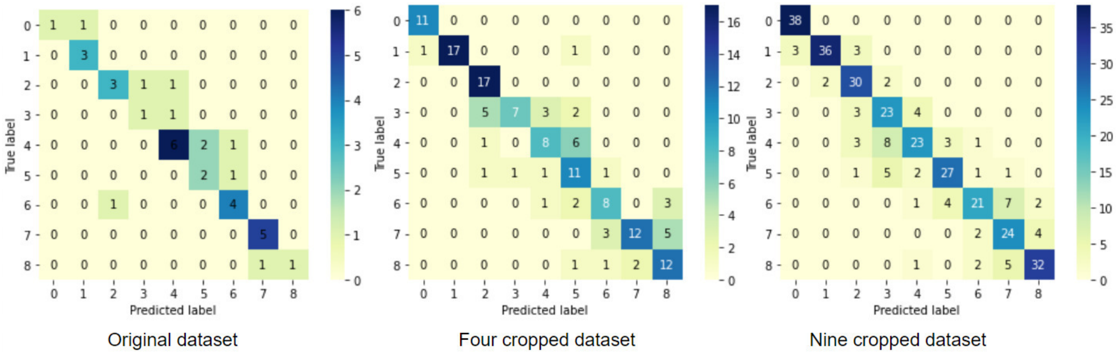


Fig. 8. Comparison of Confusion Matrices for Original, 4-Cropped, and 9-Cropped Images for the baseline model.

suggests that the exclusion of non-informative bands enhances the model's performance, and the results are not dependent on a specific fold.

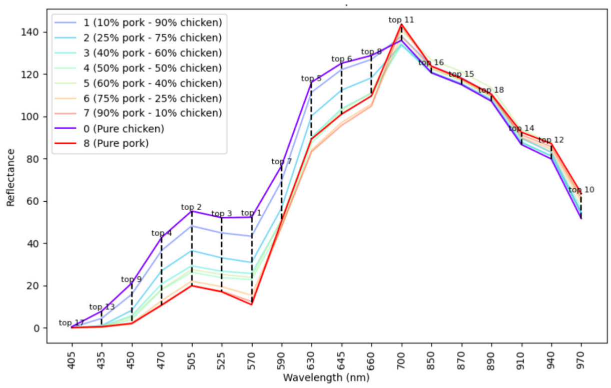


Fig. 9. Mean reflectance of samples of different adulteration level.

In the experiment on optimizing band selection using one-band mixed augmentation, the impact of different bands on the DenseNet201 model's performance without transfer learning was investigated. The results revealed that the choice of bands for augmentation significantly influenced the model's learning capabilities. Specifically, band 4 (470 nm, blue), band 14 (870 nm, NIR), and band 17 (940 nm, NIR) were identified as the most informative bands for the MSI dataset on the DenseNet201 model. Augmenting band 17 further improved the model's accuracy from 0.72 to 0.81, highlighting the importance of band selection and preprocessing in optimizing performance.

Comparing the two methods, one-band mixed augmentation proved to be a better approach for band selection compared to reflectance-based feature selection. Although the performance difference between the two methods was insignificant, the augmented-based 12-band approach slightly outperformed the reflectance-based 12-band approach for the MSI dataset. This finding suggests that one-band mixed augmentation enables a more comprehensive exploration of the feature space, leading to a more robust model.

Both experiments show promising results for selecting informative bands in multispectral image classification tasks. The reflectance-based method provides a straightforward and intuitive approach, while one-band mixed augmentation allows for a more exploratory analysis, potentially uncovering new features beyond spectral characteristics alone. Future research could explore combining these two approaches to leverage their respective advantages and further enhance classification performance.

The size of an original image file, amounting to 103 MB, is a pertinent consideration in the context of the present study, which seeks to identify and analyze meat adulteration in minced chicken-pork samples. Moreover, the extraction of a (224,224,18) numpy array from the original file raises concerns about the optimal utilization of the information contained within the multispectral data. To overcome these limitations, another experiment applied cropping based preprocessing to the original (1200 x 1200) pixels image.

By employing the four-crop and nine-crop approach to augment the entire dataset for the 9-class case, significant improvements were observed. The results indicate that the nine-cropped dataset achieved the highest accuracy of 0.74, outperforming both the uncropped and four-cropped datasets. These findings show the potential of crop augmentation as an effective approach to address the challenges posed by limited dataset sizes and for maximizing the utilization of multispectral data in classification tasks.

V. Conclusion

Our study highlights the potential of CNN models for detecting adulteration in minced meat samples. Among the evaluated models, DenseNet performed the best, showcasing its suitability for this task. We found that transfer learning did not significantly enhance model performance. Preprocessing data augmentation techniques, particularly our proposed one-band mixed augmentation approach, proved crucial in improving the model accuracy. Although our study had limitations, such as a small dataset and focus on a specific type of adulteration, it lays the groundwork for future research in this area. Further exploration of larger datasets and integration of additional data types are encouraged to advance food safety practices.

REFERENCES

[1] C. Yang, G. Zhong, S. Zhou, Y. Guo, D. Pan, S. Wang, Q. Liu, Q. Xia, and Z. Cai, "Detection and characterization of meat adulteration in various types of meat products by using a high-efficiency multiplex polymerase chain reaction technique," *Frontiers in Nutrition*, 2022. doi: 10.3389/fnut.2022.979977

[2] K. Edwards, M. Manley, L. C. Hoffman, and P. J. Williams, "Non-destructive spectroscopic and imaging techniques for the detection of processed meat fraud," *Foods*, vol. 10, no. 2, p. 448, 2021. doi: 10.1016/j.afres.2022.100147

[3] P. F. Pereira, P. H. de Sousa Picciani, V. Calado, and R. V. Tonon, "Electrical gas sensors for meat freshness assessment and quality monitoring: A review," *Trends in Food Science & Technology*, vol. 118, pp. 36–44, 2021. doi: 10.1016/j.tifs.2021.08.036

[4] R. S. Andre, M. H. Facure, L. A. Mercante, and D. S. Correa, "Electronic nose based on hybrid free-standing nanofibrous mats for meat spoilage monitoring," *Sensors and Actuators B: Chemical*, vol. 353, p. 131114, 2022. doi: 10.1016/j.snb.2021.131114

[5] H. J. Marvin, E. M. Janssen, Y. Bouzembrak, P. J. Hendriksen, and M. Staats, "Big data in food safety: An overview," *Critical reviews in food science and nutrition*, pp. 2286–2295, 2017. doi: 10.1080/10408398.2016.1257481

[6] M. Peyvasteh, A. Popov, A. Bykov, and I. Meglinski, "Meat freshness revealed by visible to near-infrared spectroscopy and principal component analysis," *Journal of Physics Communications*, 2020. doi: 10.1088/2399-6528/abb322

[7] P. Tsakanikas, L.-C. Fengou, E. Manthou, A. Lianou, E. Z. Panagou, and G.-J. E. Nychas, "A unified spectra analysis workflow for the assessment of microbial contamination of ready-to-eat green salads: Comparative study and application of non-invasive sensors," *Computers and electronics in agriculture*, vol. 155, pp. 212–219, 2018. doi: 10.1016/j.compag.2018.10.025

[8] C. H. Q. Rego, F. França-Silva, F. G. Gomes-Junior, M. H. D. d. Moraes, A. D. d. Medeiros, and C. B. d. Silva, "Using multispectral imaging for detecting seed-borne fungi in cowpea," *Agriculture*, vol. 10, no. 8, p. 361, 2020. doi: 10.3390/agriculture10080361

[9] S. Younas, Y. Mao, C. Liu, M. A. Murtaza, Z. Ali, L. Wei, W. Liu, and L. Zheng, "Measurement of water fractions in freeze-dried shiitake mushroom by means of multispectral imaging (msi) and low-field nuclear magnetic resonance (lf-nmr)," *Journal of Food Composition and Analysis*, vol. 96, p. 103694, 2021. doi: 10.1016/j.jfca.2020.103694

[10] L.-C. Fengou, E. Spyrelli, A. Lianou, P. Tsakanikas, E. Z. Panagou, and G.-J. E. Nychas, "Estimation of minced pork microbiological spoilage through fourier transform infrared and visible spectroscopy and multispectral vision technology," *Foods*, vol. 8, no. 7, p. 238, 2019. doi: 10.3390/FOODS8070238

[11] M. Kamruzzaman, Y. Makino, and S. Oshita, "Rapid and non-destructive detection of chicken adulteration in minced beef using visible near-infrared hyperspectral imaging and machine learning," *Journal of Food Engineering*, vol. 170, pp. 8–15, 2016. doi: 10.1016/j.jfoodeng.2015.08.023

[12] A. I. Ropodi, E. Z. Panagou, and G.-J. E. Nychas, "Multispectral imaging (msi): A promising method for the detection of minced beef adulteration with horsemeat," *Food Control*, vol. 73, pp. 57–63, 2017. doi: 10.1016/j.foodcont.2016.05.048

[13] ——, "Rapid detection of frozen-then-thawed minced beef using multispectral imaging and fourier transform infrared spectroscopy," *Meat science*, vol. 135, pp. 142–147, 2018. doi: 10.1016/j.meatsci.2017.09.016

[14] L.-C. Fengou, A. Lianou, P. Tsakanikas, F. Mohareb, and G.-J. E. Nychas, "Detection of meat adulteration using spectroscopy-based sensors," *Foods*, vol. 10, no. 4, p. 861, 2021. doi: 10.3390/foods10040861

[15] X. Li, X. Fan, L. Zhao, S. Huang, Y. He, and X. Suo, "Discrimination of pepper seed varieties by multispectral imaging combined with machine learning," *Applied Engineering in Agriculture*, vol. 36, no. 5, pp. 743–749, 2020. doi: 10.13031/aea.13794

[16] M. A. Tamayo-Monsalve, E. Mercado-Ruiz, J. P. Villa-Pulgarin, M. A. Bravo-Ortíz, H. B. Arteaga-Arteaga, A. Mora-Rubio, J. A. Alzate-Grisales, D. Arias-Garzon, V. Romero-Cano, S. Orozco-Arias *et al.*, "Coffee maturity classification using convolutional neural networks and transfer learning," *IEEE Access*, vol. 10, pp. 42 971–42 982, 2022. doi: 10.1109/ACCESS.2022.3166515

[17] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. doi: 10.48550/arXiv.1409.1556

[19] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017. doi: 10.48550/arXiv.1602.07261

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. doi: 10.1109/CVPR.2015.7298594 pp. 1–9.

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. doi: 10.1109/CVPR.2017.243 pp. 4700–4708.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2014. [Online]. Available: https://arxiv.org/abs/1409.0575

[23] L.-C. Fengou, P. Tsakanikas, and G.-J. E. Nychas, "Rapid detection of minced pork and chicken adulteration in fresh, stored and cooked ground meat," *Food Control*, 2021. doi: https://doi.org/10.1016/j.foodcont.2021.108002

[24] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. doi: 10.48550/arXiv.1811.08883 pp. 4918–4927.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 833–842
DOI: 10.15439/2023F7295
ISSN 2300-5963 ACSIS, Vol. 35

Filtering Decision Rules Driven by Sequential Forward and Backward Selection of Attributes: An Illustrative Example in Stylometric Domain

Beata Zielosko*, Urszula Stańczyk† Kamil Jabłoński*
*University of Silesia in Katowice, Institute of Computer Science, Będzińska 39, 41-200 Sosnowiec, Poland
Email: beata.zielosko@us.edu.pl, kjablonski1@us.edu.pl
†Silesian University of Technology, Department of Graphics, Computer Vision and Digital Systems
Akademicka 2A, 44-100 Gliwice, Poland, Email: urszula.stanczyk@polsl.pl

***Abstract*—The paper presents investigations concerning the decision rule filtering process controlled by the estimated relevance of available attributes. In the conducted study, two search directions were used, sequential forward selection and sequential backward elimination. The steps of sequential search were governed by three rankings obtained for variables, all related to characteristics of data and rules that can be induced, as follows, (i) a ranking based on the weighting factor referring to the occurrence of attributes in generated decision reducts, (ii) the OneR ranking exploiting short rule properties, and (iii) the proposed ranking defined through the operation of greedy algorithm for rule induction. The three rankings were confronted and compared from the perspective of their usefulness for the selection of rules performed in the two directions and with two strategies for rule selection. The resulting sets of rules were analysed with respect to the properties of the constituent decision rules and from the point of performance for all constructed rule-based classifiers. Substantial experiments were carried out in the stylometric domain, treating the task of authorship attribution as classification. The results obtained indicate that for all three rankings and search paths it was possible to obtain a noticeable reduction of attributes while at least maintaining the power of inducers, at the same time improving characteristics of rule sets.**

I. Introduction

ONE OF the main goals of data mining is the extraction of useful knowledge from large amounts of data or phenomena described by a high number of attributes. An important element of this process is the determination and selection of the most important attributes related to the described phenomenon [1]. The objective of this step, called feature selection, is to differentiate relevant variables from the entire set of features, while at the same time preserving the descriptive and representative qualities of the original set of attributes [2].

Feature selection can be accomplished by selecting a minimal subset of features that enables obtaining at least the same performance of a classifier as for the entire set of attributes [3]. In this case, feature subset selection requires assessing the quality of each discovered feature subset. Another way of proceeding is to construct a ranking of features based on a specific criterion. Then the variables are ordered from the most to the least important, and the top k features are selected based on a predefined threshold. Feature ranking is also known as feature weighting and involves evaluating individual attributes by assigning weights to them based on their relevance.

A technique used to search space of variables during the attribute selection process is an important factor. Since the problem of locating an optimal subset of features, taking into account all possible variable subsets, is NP-hard, greedy techniques, such as forward selection and backward elimination, are often used instead of exhaustive search. Forward selection begins with an empty set, which is gradually expanded by adding one feature (or a group of features) at a time until specific criteria are met. Sequential backward elimination involves starting with all attributes and progressively discarding them. Depending on the adopted criterion, added or rejected attributes can correspond to the highest positions in the ranking, or, they can be the lowest ranking elements.

One of the disadvantages of sequential selection is that interactions among features are not closely studied and dependencies can be missed when only one path of selection is investigated [4]. This problem can be remedied to some extent by varying the feature selection approach through patterns discovered in the data, such as decision rules, and discarding them only when they are dependant entirely on rejected variables, while keeping under consideration those that refer also to at least one attribute that is contained in the retained set. With this kind of processing, interactions among variables have more influence on the properties of recalled sets of rules.

The aim of the research presented in the paper and its contribution is the investigation and comparison of three influential factors, as follows: (i) two search strategies, i.e. sequential forward selection vs. sequential backward elimination, applied not directly to the variables in the dataset but through the filtering decision rules process, (ii) two approaches to rule selection, i.e., retaining rules that contain conditions only on the variables still in considerations vs. keeping the rules that include conditions on at least one of the attributes contained in the studied set, (iii) three ranking mechanisms, the OneR available in WEKA workbench [5], and two proposed, exploiting the properties of data and patterns discovered in them. One of those referred to the defined weighting factor, which takes into account the number of reducts in which a given attribute occurs and the cardinalities of these reducts [6], while the other was based on the properties of the greedy algorithm for the induction of decision rules, the number of occurrences in the rules and their support.

All experiments were performed on two datasets from the stylometry domain. The writing styles of the considered writers were learnt from available texts through the analysis of quantitative linguistic descriptors and advanced processing. To prevent bias on the observations, the datasets were prepared for the task of binary authorship attribution with balanced classes. The performance for induced rule-based classifiers was estimated with the help of test sets, over which the classification accuracy was averaged.

The results obtained allowed to conclude that all search paths led to increased performance for reduced sets of features while improving the characteristics of constructed rule sets. Backward elimination with keeping the rules referring to any attributes in the considered set allowed for reduction of more attributes than forward selection with limiting conditions in rules only to still present variables. The three investigated rankings produced close maximal predictions but for different numbers of attributes and rules. Greedy ranking held its ground when pitted against the other two, it even led to the one case of perfect recognition. These observations proved the merits of the described research works and again validated the methodology for ranking-driven rule selection.

The structure of the paper is organised as follows. Section II presents background information related to feature selection and induction of decision rules. Section III provides a description of stylometric analysis of texts, as the application domain. Section IV contains the explanation for the experiments performed and comments on the results obtained. Conclusions and future research plans are given in Section V.

II. Background information

In this section, aspects related to feature selection and decision rules are provided. Search strategies were described in the context of feature selection, and the main approaches for induction of rules were presented. Finally, the processing steps of rule filtering driven by feature selection were given.

A. Feature selection

During recent years, due to increasing demands for dimensionality reduction, extensive efforts in feature selection research have been made. It can be realised as a stage of data mining, related to data pre-processing, and then it affects such elements as visualisation, learning algorithms, and performance of classifiers. The main task of feature selection is to remove irrelevant or redundant variables so that their elimination from the set of attributes will not affect the performance of the learning algorithms [7]. The process of feature selection allows for data reduction and lowering of storage requirements. Furthermore, since the goal is to find the most relevant variables, it is possible to strive to improve data quality by enhancing data mining algorithms, that is, reducing learning time and improving predictive capabilities.

A feature selection procedure can be considered to contain three stages: (i) search for potential subsets of variables, (ii) evaluation of the subset of attributes based on some criteria, and (iii) setting the stop condition for the search. The final stage is closely linked to the initial one, as the search is repeated iteratively until the stopping criterion is met.

Due to the large search space, feature selection is also perceived as a combinatorial problem—for a dataset with N attributes, the search space is 2^N. Searching for an optimal subset of features taking into account all possible variable subsets is NP-hard problem [8]. An exhaustive search can be performed only if the number of attributes is relatively small. Instead, greedy [9] or meta-heuristics [10] approaches can be used.

To select a subset of variables from the input data, different search strategies can also be applied, including genetic algorithms, evolutionary computation techniques, heuristic search algorithms, and various hybrid strategies. Among greedy techniques, the sequential search performed as forward selection and backward elimination can be distinguished [11]. The sequential backward elimination method starts with all the variables, and then gradually features are removed from the set, either one by one, or in groups. In each step, the eliminated variable or variables contribute the least to the criterion function. Forward selection starts with the empty set to which sequentially features are added, again either one at a time or in groups, until certain criteria are met.

Both search strategies are heuristic and cannot guarantee the optimality of the selected features. Among the alternatives to these approaches, floating, branch-and-bound, and randomised can be mentioned [12]. Random search methods, for example, genetic algorithms, add some randomness to the search procedure to help escape from a local optimum. In certain cases, especially when dealing with high-dimensional datasets, an individual search is performed. Such methods evaluate each feature individually based on a specific criterion or condition. The branch-and-bound algorithm finds the optimal feature subset if the criterion function used is monotonic [3]. Floating search methods prevent the situation where the variable is deleted in backward elimination, and then it cannot be reselected, and also when a feature is added in forward selection and cannot be deleted once it was selected [11].

B. Ranking construction

Feature selection can be performed in two different ways, by selecting a subset of attributes or by creating a ranking of variables [13]. In the latter case, the variables are ordered according to the adopted criterion or evaluation function from the most important to the least important and the top k attributes are selected from the ranking, with k being some pre-selected threshold number. Feature ranking plays an important role in directing the search process in different machine learning tasks, especially when an exhaustive search is computationally unfeasible and a heuristic search approach is necessary. It determines the order in which the variables are explored by the algorithms within the feature space.

Feature ranking methods use different measures, for example, based on similarity score, statistics, information theory, or on some functions of the classifier's outputs [1]. Traditional ranking approaches evaluate variables without incorporating any learning algorithm. This category typically consists of filter-based feature selection methods, such as referring to information gain, correlation, or Relief algorithm. However, there are also some studies on wrapper techniques, which involve methods such as recursive feature elimination [14], and the classifier-aided feature ranking approach [15].

In the paper, three ranking mechanisms were studied, related to the properties of the data, and discovered patterns in the form of decision reducts and decision rules. One ranking was based on the defined weighting factor calculated through reducts, another was related to the OneR algorithm, and the third ranking was proposed by the authors and based on the properties of the greedy algorithm for rule induction. All the rankings obtained were used as filters for sets of induced rules.

1) Ranking of attributes based on reducts: Reduct is one of the key notions in rough sets theory [16] and refers to feature selection performed within the framework of rough sets. There are many definitions of a reduct because they deal with different criteria related to the selection of attributes and computing the most relevant sets of variables, for example, decision and local reducts for decision tables, reducts for information systems, reducts based on the generalised decision, or fuzzy decision reducts.

A reduct can be defined as a minimal set of attributes that preserves the degree of dependency of the entire set of attributes. Taking into account the performance, the reduct is such a minimal subset of attributes that has the same classification power as the complete set of available attributes [17].

The problem of calculating reducts is NP-hard, therefore, different heuristic approaches are used for its construction, for example finding reducts through sampling data from a decision table [18], heuristics based on discernibility matrix [19], greedy algorithms [9], Boolean reasoning, and many others [20]. In the investigation presented in the paper, the genetic algorithm [21], implemented in the Rough Sets Exploration System (RSES) [22], was used to construct the reducts. It is a binary genetic algorithm where every binary individual encodes one subset of attributes that is a potential reduct. The fitness function of a subset R has the form:

$$F(R) = \frac{n - L_R}{n} + \frac{2C_R}{m^2 - m}, \quad (1)$$

where n is the length of bit strings equal to a number of attributes, and m gives a number of objects. L_R denotes a number of "1"-s in the subset R, and C_R denotes the number of object pairs (with different decision values) discerned by the attribute subset R. Calculating C_R is the most time-consuming operation. It is accelerated by the "distinction table", a binary matrix of size $(n+1)\times(m^2-m)/2$. Each column corresponds to one attribute (the last column corresponds to the decision), and each row corresponds to one pair of different objects. The value "1" denotes an attribute with a different value on the pair of objects. Finding a reduct means finding the minimal subset of columns that cover the matrix.

The described genetic algorithm allows to generate a satisfactorily high number of reducts in relatively short time. The resulting reducts may contain different attributes and may also have different cardinalities. For the set of induced reducts, the weighting factor for features was proposed that takes into account the number of reducts in which a given attribute exists, and cardinalities of these reducts [6],

$$W_F(G_{Red}, a) = \sum_{i=k_{min}}^{k_{max}} \frac{card\,(RED(G_{Red}, a, i))}{card\,(G_{Red}) \cdot i}, \quad (2)$$

where k_{min} and k_{max} are respectively the minimal and the maximal reduct cardinalities detected for the group G_{Red}. $RED(G_{Red}, a)$ denotes the set of all reducts from the group G_{Red} that include the attribute a, and $RED(G_{Red}, a, k)$ is the set of reducts of length k that contain the attribute a. Then $card\,(RED(G_{Red}, a, k))$ returns for the group G_{Red} the number of reducts with specific length equal to k that contain the given attribute a. The values of W_F range from 0 (the attribute a is not included in any of the reducts in this group) to $1/k_{min}$, when the attribute is included in all the reducts and all the reducts have the same cardinality (then $k_{min} = k_{max}$).

A higher value of the weighting factor presented indicates that the attribute appears in more reducts with lower cardinalities, and low values of W_F are obtained for attributes that are included in fewer reducts containing more variables. All attributes included in a group can be ordered by the scores calculated for them, and a ranking is obtained as a result.

The described weighting factor promotes reducts with a small number of attributes. This way of reasoning follows from the fact that in a situation where we have two reducts and one of them has a smaller number of attributes, according to the definition of a reduct, this smaller number of attributes is sufficient to protect the performance of the system. Moreover, it complies with the Minimum Description Length principle [23]: "the best hypothesis for a given set of data is the one that leads to the largest compression of data". Additionally, reducts with smaller numbers of attributes are preferred from a knowledge representation perspective.

2) OneR algorithm: The OneR (One Rule) algorithm is a simple classification algorithm that is used in the field of machine learning. Its purpose is to select the most conclusive feature from all available features in the dataset, in order to create a simple classification model. This is done by calculating the number of occurrences of particular class labels for each value of a given attribute in the dataset. After this process, the OneR algorithm selects the feature for which the value is the most discriminating in the context of predicting class labels. In practice, for the selected feature, a single condition is created in a decision rule that is used to classify new instances. The algorithm generates one rule per unique attribute value of the selected best feature.

The main strength of the OneR algorithm is its ability to select the most relevant feature in the context of class prediction [24]. Although the OneR algorithm is simple and does not take into account interdependencies between features, it often allows to obtain satisfactory classification accuracy. In addition, this algorithm tends to choose the value of attribute that occurs the most frequently, and in this way it allows to ignore noise existing in the data. OneR is also called one-level decision tree algorithm. It selects attributes from a dataset one by one and generates a different set of rules based on the error rate from the training set. Finally, it chooses the attribute that offers rules with minimum error [25].

3) Ranking of attributes based on greedy algorithm properties: In the research, the authors propose a ranking mechanism exploiting the properties of the greedy algorithm for the induction of decision rules [26]. Such an algorithm constructs a decision rule for each row of a decision table. In each iteration, attributes are selected to form the conditions of the rules. The selected attribute separates the maximum number of rows from a set of rows with a different class label, so a decision table is divided into sub-tables as dictated by given attribute and corresponding value. The partitioning of a table is completed when all rows in the sub-table, corresponding to the selected attribute, have the same class labels.

As shown in previous research [27], given certain assumptions about the NP class, the greedy algorithm used to induce decision rules produces results that are not far from the best approximate polynomial algorithms for minimising the length of the rules, which is important for knowledge representation. Short rules can be considered as more general so they allow to reflect patterns hidden in the data and prevent overfitting, which is important for the classification process.

During research focused on the greedy algorithm, it was observed that in the majority of cases, when constructing decision rules, the greedy algorithm at each iteration selects an attribute that separates at least 50% of the remaining rows with different decisions.

The proposed ranking was based on the attributes contained in the decision rules, the percentage of separated rows with decisions different from the decision attached to a given rule, and the support of the rule. The latter element is an important factor in assessing the quality of decision rules. In order to construct the ranking, the decision rules were induced by the greedy algorithm and duplicate rules were removed from the entire set of rules. Then, for each attribute, the number of its occurrences in the rules was determined, assigning the highest positions in the ranking to the attributes with the highest number of occurrences. If the number of occurrences was the same for several attributes, then the percentage of rows separated by the given attribute was taken into account. The third factor that played a role in determining the score for each attribute was the support of the rule in which the attribute appeared, which led to the assignment of higher positions in the ranking to attributes from the rules with higher support.

C. Decision rules

Decision rules belong to popular forms used for data representation. They are induced from datasets very often presented as a decision table $T = (U, A \bigcup \{d\})$ [16], where U is a non-empty, finite set of objects, $A = \{a_1, \ldots, a_m\}$ is a set of condition attributes i.e., $a_i : U \to V_a$, where V_a is the set of values of attribute a_i called the domain of a_i, and $d \notin A$ is a distinguished attribute called a decision, with values $V_d = \{d_1, \ldots, d_{|V_d|}\}$. The decision rules take the form:

$$(a_{i_1} = v_1) \wedge \ldots \wedge (a_{i_k} = v_k) \to d = v_d,$$

where $a_{i_1}, \ldots, a_{i_k} \in \{a_1, \ldots, a_m\}$, $v_i \in V_{a_i}$, and $v_d \in V_d$. Pairs $(a_{i_1} = v_1)$ are called descriptors or conditions. The number of conditions in a premise part of a rule is its length. Short rules are preferred from the point of view of knowledge representation and with regard to the MDL principle. They are easier to understand and interpret. When assessing the quality of decision rules, support is another important factor. It is a number of such objects from the decision table whose attribute values satisfy the premise part of the rule, and they have the same decision as the one attached to the rule. This measure allows to discover major patterns present in the data.

There are a wide variety of approaches for induction of decision rules. Among the exact ones, Boolean reasoning and extensions of dynamic programming should be mentioned [28]. The construction of decision rules with maximum support or minimum length is considered an NP-hard problem, so different heuristics are used. They are based on modifications of exact approaches, different kinds of greedy algorithms, methods relying on sequential covering, genetic algorithms, and many others. In the rough set theory, the popular approach is also induction of rules based on a reduct. Then each rule has length equal to the cardinality of the reduct, and each object from a decision table has assigned values corresponding to condition attributes included only in this reduct.

Apart from using decision rules as a form of knowledge representation, they are very often used as classifiers. In this situation, the rule filtering process can be treated as a method of pruning the rule set to fine-tune the classifier by reducing the number of rules. The use of filtering rules in the framework of the feature selection process often leads to improved classification accuracy.

In the experiments performed, the decision rules were induced by the exhaustive algorithm implemented in the RSES system. It constructs all minimal decision rules, i.e. rules with minimal numbers of descriptors (pairs attribute = value) in their premise parts. Then, they were filtered sequentially, according to the search strategy added or removed, driven by the studied rankings of attributes.

III. Stylometric data

A writing style is an individual characteristic, based to some extent on social and cultural background, education, lifetime experiences, elements that are learnt, but also on personal linguistic preferences and habits. To obtain a definition of an authorial profile, access to some representative samples of writing is needed. Comparative analysis and stylometric data mining lead to the discovery of patterns specific to writers and the construction of approximating descriptions that can be applied to text samples of unknown or unconfirmed authorship to find the closest match. This way of carrying out the authorship attribution task means solving a classification problem [29], therefore, a dataset to be prepared needs to include some training and test samples, all relying on a set of selected efficient style-markers [30].

Stylometric descriptors that work best refer to common language elements as they are used almost subconsciously, so they are less prone to forgery or imitation. Lexical and syntactic markers are often employed for the task [31]. They provide quantitative characteristics through frequency of occurrence for function words and punctuation marks, which results in real-valued features. In the experiments reported, the set of markers contained 24 elements with values calculated over text samples obtained by partitioning long novels by four acclaimed writers into smaller chunks. The authors studied, Edith Wharton, Mary Johnston, Jack London, and James Oliver Curwood, were paired according to gender [32], in order to form two datasets with binary authorship attribution.

The division of long texts into smaller parts resulted in imposing a specific stratification of the input space [33]. To avoid bias when evaluating the performance of a classifier, the datasets (the male writer dataset and the female writer dataset) prepared included one train set and two test sets. The samples contained in sets of different types were based on separate novels. With binary classification, balanced data and the same importance of all classes, classification accuracy was used as a measure of performance, providing information on the average portion of correctly attributed text samples from test sets.

Among popular data mining approaches, those that involve induction of decision rules belong to the most advantageous. They not only enable assigning authors to samples, but also enhance understanding of the stylometric domain by providing an inside view on linguistic patterns detected for authors by the transparent form of discovered rules. Short rules, with a few conditions in their premises, are preferred over long rules [24]. The former are more general, while the latter with their too detailed definitions can cause over-fitting.

The datasets were discretised with the Fayyad and Irani algorithm [34]. It is one of the top-down supervised methods, which starts with assigning one large interval to represent in the discrete domain all the values of a transformed variable. Then, referring to the MDL principle and calculation of entropy [23], candidates for cut-points are evaluated to discover which are most supportive to distinction of classes. If further partitioning is disadvantageous to entropy, the processing stops. As a consequence, it is possible that some variables are removed from consideration in the discrete domain when they have a single categorical representation. In the experiments, for the female writer dataset 20 out of the total of 24 features received more than a single bin, and for the male writer dataset the set of attributes was reduced to 22.

IV. Performed experiments

The experimental process of the research works consisted of the following stages:

- Preparation of two datasets (female writers and male writers), which included discretisation by Fayyad and Irani algorithm applied to all condition attributes;
- Construction of three rankings of attributes:
  - *Reducts*—based on reducts and the proposed weighting factor;
    Using a genetic algorithm implemented in the RSES system, one group of 150 reducts was generated. Obtained reducts consisted of different attributes from the whole set of available features and had different cardinalities. The weighting factor defined in Eq. (2) took into account all these elements and returned scores for the variables. The ordering of attributes by their scores resulted in the ranking.
  - *OneR*—based on the OneR algorithm implemented in WEKA software [5];
  - *Greedy*—based on the properties of the greedy algorithm for induction of decision rules, that is, the number of rules in which a given attribute occurs, the percentage of separated rows with different decisions and the support of decision rules.
- Induction of decision rules by exhaustive algorithm, for the input datasets;
- Filtration of sets of rules accordingly to sequential forward selection and sequential backward elimination driven by attributes included in a given ranking;
- Evaluation of performance for rule-based classifiers with test sets;
- Assessment of the quality of rule sets from the point of view of knowledge representation, i.e., taking into account the number of rules, average length and average support;
- Comparative study of results, for two search directions, two rule selection strategies, and three rankings.

Details of all steps are provided below, along with comments on the results obtained.

A. Rankings

For the female and male writer datasets, the rankings obtained were presented in rows of Table I (where the letters F and M indicate the female and male writer datasets, respectively). The row *Position* denotes the position of the given attribute in a ranking, and 1 is considered the highest ranking position, assigned to the most important feature.

For the female writer dataset, in the case of the ranking constructed through reducts and the OneR algorithm, the entire set of attributes was used, with the exception of attr14, attr16, attr18, and attr21. It resulted from the situation that these attributes had only 1 bin allocated by the supervised discretisation process. For the male writer dataset, there was a similar situation, i.e. instead of 24 attributes, only 22 were used to create the ranking since attr11 and attr14 were the

Table I
RANKINGS OF ATTRIBUTES FOR FEMALE AND MALE WRITERS

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Reducts-F	attr23	attr1	attr22	attr17	attr12	attr3	attr2	attr10	attr9	attr5	attr6	attr8	attr4	attr13	attr7	attr20	attr11	attr15	attr0	attr19		
Reducts-M	attr23	attr6	attr3	attr7	attr5	attr20	attr16	attr1	attr15	attr8	attr9	attr17	attr22	attr4	attr12	attr0	attr18	attr21	attr13	attr10	attr2	attr19
OneR-F	attr23	attr1	attr17	attr22	attr20	attr2	attr13	attr6	attr8	attr4	attr9	attr12	attr3	attr7	attr10	attr15	attr11	attr0	attr19	attr5		
OneR-M	attr23	attr17	attr3	attr1	attr6	attr16	attr13	attr0	attr18	attr9	attr7	attr2	attr8	attr22	attr12	attr5	attr4	attr20	attr21	attr15	attr19	attr10
Greedy-F	attr1	attr23	attr2	attr17	attr3	attr13	attr11	attr22	attr10	attr8	attr6	attr0	attr19	attr5	attr7	attr4						
Greedy-M	attr23	attr1	attr3	attr0	attr21	attr10	attr16	attr18	attr8	attr2	attr22	attr7	attr6	attr12	attr19	attr15	attr9					

attributes to which only 1 bin was assigned in the Fayyad and Irani discretisation process.

From the ranking created based on the accuracy of the greedy algorithm, in addition to the attributes with a single categorical representation, other variables were also excluded because they did not appear in the induced decision rules. Therefore, these rankings were shorter and contained 16 attributes for the female dataset and 17 for the male dataset.

It is worth noting that for the male writer dataset, all three rankings assigned the highest position to the same attribute: attr23. In the case of the female set, this attribute was ranked second only in the ranking related to the greedy algorithm. Furthermore, the features disregarded by the greedy ranking (attr9, attr12, attr15, and attr20 for female writers, and attr4, attr5, attr13, attr17, attr20 for male writers) were not recognised as irrelevant or close to irrelevant by other rankings, for example, for the male writers attr17 was found as the second ranking for the OneR algorithm.

B. *Strategies employed in decision rule filtering*

Forward selection was performed by sequentially filtering and increasing the set of decision rules. Starting with the highest ranking attribute, from the entire set of rules those were selected that contained conditions (in their premises) relating only to this attribute. Then, in the second step, a subset of recalled attributes was extended to the top two positions, and such rules were selected that relied only on these two variables as conditions. Next, three top ranking features were studied, and so on. In each step of the sequential search, the conditions in the rules were limited only to the currently selected subset. The forward rule filtering process continued until all available features and rules were included in the set considered.

The backward elimination was achieved by sequentially decreasing the set of decision rules. Starting with the attribute in the lowest ranking position, those rules were selected from the entire set of rules, which contained in their premises the condition referring to this very attribute. If a rule included some other attributes that worked as conditions, then that rule was not removed from the set of rules. The second step of backward reduction meant rejection of rules with conditions limited to the two lowest ranking variables, and so on, until the set of rules was exhausted. The difference between the two strategies involved is shown in the illustrative small example.

Let us assume a set of five condition attributes, for simplicity ranked as follows, where 1 is considered the top ranking position, and 5 the bottom of the ranking:

Position 1: attr1
Position 2: attr2
Position 3: attr3
Position 4: attr4
Position 5: attr5

The set of rules, subject to filtering driven by ranking, consists of eight elements.

Rule 1: with condition on attr1
Rule 2: with conditions on attr2 and attr3
Rule 3: with conditions on attr1 and attr5
Rule 4: with conditions on attr1 and attr4
Rule 5: with condition on attr3
Rule 6: with conditions on attr3 and attr5
Rule 7: with conditions on attr2 and attr5
Rule 8: with condition on attr4

For backward elimination, the processing starts with all rules included in the recalled set. Then, for the filtering steps, the resulting sets are as follows.

Step 1: Rejected attributes: attr5, recalled rules: all rules
Step 2: Rejected attributes: attr5, attr4, recalled rules: 1, 2, 3, 4, 5, 6, 7
Step 3: Rejected attributes: attr5, attr4, attr3, recalled rules: 1, 2, 3, 4, 7
Step 4: Rejected attributes: attr5, attr4, attr3, attr2, recalled rules: 1, 3, 4
Step 5: Rejected attributes: all, recalled rules: no rules

For forward selection at the starting point, the set of recalled rules is empty. It is next gradually expanded as listed below.

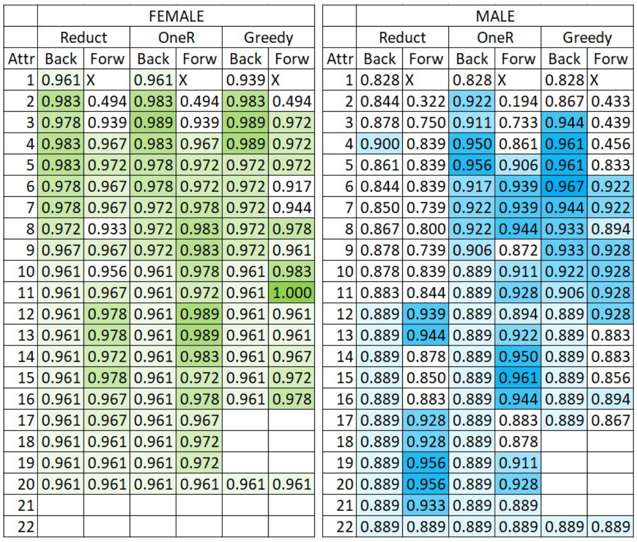
Step 1: Selected attributes: attr1, recalled rules: 1
Step 2: Selected attributes: attr1, attr2, recalled rules: 1
Step 3: Selected attributes: attr1, attr2, attr3, recalled rules: 1, 2, 5
Step 4: Selected attributes: attr1, attr2, attr3, attr4, recalled rules: 1, 2, 4, 5, 8
Step 5: Selected attributes: all, recalled rules: all rules

The process of rule filtering carried out for the greedy rankings was slightly different than for the other two rankings, because the rule sets induced by the exhaustive algorithm included rules with conditions on such features that were absent in the greedy ranking. Therefore, the first step of rule elimination was to remove rules containing only attributes that did not appear in these rankings, while the last step for forward selection was to add these rules.

C. *Performance of rule-based classifiers*

For all rule-based classifiers obtained in the decision rule filtering process, performance was evaluated with test sets. Fig. 1 presents the average classification accuracy obtained.

Decision rules induced by the exhaustive algorithm were selected through the backward elimination (column Back) and forward search (column Forw) strategies, with the conditions for recalling rules governed by the three rankings (groups of columns, Reduct, OneR, Greedy). The results shown in the bottom row provide the reference point, because they correspond to the case where the entire sets of attributes and rules were taken into account. The X mark denotes the situation where no rules were included in the set of recalled rules. The coloured cells indicate where the classification accuracy exceeded the reference point. The intensity of cell colour depends on how much the accuracy was improved. For each step of the rule filtering process, the columns Attr indicate the ranking position considered (which for forward search corresponds to the number of variables taken into account).



	FEMALE					
	Reduct		OneR		Greedy	
Attr	Back	Forw	Back	Forw	Back	Forw
1	0.961	X	0.961	X	0.939	X
2	0.983	0.494	0.983	0.494	0.983	0.494
3	0.978	0.939	0.989	0.939	0.989	0.972
4	0.983	0.967	0.983	0.967	0.989	0.972
5	0.983	0.972	0.978	0.972	0.972	0.972
6	0.978	0.967	0.978	0.972	0.972	0.917
7	0.978	0.967	0.972	0.978	0.972	0.944
8	0.972	0.933	0.972	0.983	0.972	0.978
9	0.967	0.967	0.972	0.983	0.972	0.961
10	0.961	0.956	0.961	0.978	0.961	0.983
11	0.961	0.967	0.961	0.972	0.961	1.000
12	0.961	0.978	0.961	0.989	0.961	0.961
13	0.961	0.978	0.961	0.989	0.961	0.961
14	0.961	0.972	0.961	0.983	0.961	0.967
15	0.961	0.978	0.961	0.972	0.961	0.972
16	0.961	0.967	0.961	0.978	0.961	0.978
17	0.961	0.967	0.961	0.967		
18	0.961	0.961	0.961	0.972		
19	0.961	0.961	0.961	0.972		
20	0.961	0.961	0.961	0.961	0.961	0.961
21						
22						

	MALE					
	Reduct		OneR		Greedy	
Attr	Back	Forw	Back	Forw	Back	Forw
1	0.828	X	0.828	X	0.828	X
2	0.844	0.322	0.922	0.194	0.867	0.433
3	0.878	0.750	0.911	0.733	0.944	0.439
4	0.900	0.839	0.950	0.861	0.961	0.456
5	0.861	0.839	0.956	0.906	0.961	0.833
6	0.844	0.839	0.917	0.939	0.967	0.922
7	0.850	0.739	0.922	0.939	0.944	0.922
8	0.867	0.800	0.922	0.944	0.933	0.894
9	0.878	0.739	0.906	0.872	0.933	0.928
10	0.878	0.839	0.889	0.911	0.922	0.928
11	0.883	0.844	0.889	0.928	0.906	0.928
12	0.889	0.939	0.889	0.894	0.889	0.928
13	0.889	0.944	0.889	0.922	0.889	0.883
14	0.889	0.878	0.889	0.950	0.889	0.883
15	0.889	0.850	0.889	0.961	0.889	0.856
16	0.889	0.883	0.889	0.944	0.889	0.894
17	0.889	0.928	0.889	0.883	0.889	0.867
18	0.889	0.928	0.889	0.878		
19	0.889	0.956	0.889	0.911		
20	0.889	0.956	0.889	0.928		
21	0.889	0.933	0.889	0.889		
22	0.889	0.889	0.889	0.889	0.889	0.889

Figure 1. Accuracy of rule-based classifiers, for female and male writers

As can be observed in Fig. 1, in the case of the Greedy column and the backward elimination strategy, rejecting rules only with attributes not included in the ranking resulted in the same classification accuracy as for the entire set of attributes. For forward selection, the results for the last step of selecting features included in the rankings differed slightly from the ones given in the bottom row, after adding the rules with attributes that did not appear in this ranking. For female writers, a small improvement was noted, and for male writers, a small decrease was visible.

When the backward elimination strategy was combined with the Reduct and OneR rankings and applied to the female writer dataset, it should be noted that for all ranking positions considered the classification accuracy was always at least at the reference level, even in the last step of filtering for the attribute in the first position in the rankings. The highest value of the classification accuracy of 0.989 was obtained for the Greedy and OneR rankings, and was related to the third position in these rankings. For ranking based on reducts this value was slightly smaller (0.983) and happened in processing of the fourth position in the ranking.

In the case of forward selection executed for the female writer dataset, the highest possible classification quality equal to 1.0 existed for 11 attributes placed at top positions in the Greedy ranking. For the OneR algorithm the maximum was equal to 0.989 and for the Reduct ranking 0.978, and both were detected when the twelfth positions were processed.

For the male writer dataset for the top position in the three rankings, backward elimination obviously returned the same results. The highest classification accuracy of 0.967 was obtained for the Greedy ranking related to the sixth top position in the ranking. It was also the highest improvement noted for this dataset. Apart from the top two ranking positions, for all the rest of filtering steps, the classification was either the same or improved over the reference point. For the OneR ranker, the best performance (0.956) referenced the fifth top ranking position. With the exception of the top ranking position, for the OneR ranking in the entire rule filtering path, the reported performance was at least as good as for the entire sets of rules and attributes considered. The ranking based on reducts brought the worst results among the three rankings, however, even here they were still detected cases of maintaining or increasing performance for the reduced sets of rules.

In the forward search applied to the male dataset, the OneR ranking was most advantageous: for the fifteenth ranking position the maximal classification accuracy 0.961 was recorded. The second best level of predictions (0.956) was obtained for the nineteenth position of the Reduct ranking. The Greedy algorithm came last with the highest accuracy of 0.928, however, it resulted from processing the ninth ranking position, so more decision rules and features were discarded than for the other two cases.

D. Characteristic of rule-based models

The entire process of rule filtering driven by rankings involved two search directions, two strategies for rule selection, and three rankings. For all the sets and subsets of decision rules constructed, their characteristics were observed, as shown in Table II. These observations included the number of rules (NoR column), average rule length (Len column), and average rule support (Supp column). The column Attr points to the ranking position considered.

As could be expected, analysis of the rule sets showed that as the number of rules in the set decreased, their average lengths tended to decrease, and the average supports increased. This was particularly evident in the rows at the bottom or close to the bottom of the tables. The average values relating to the shortest rules with the highest support were marked in bold.

In the case of forward selection, the differences regarding the number of rules, their length, and support were more visible than in the case of backward elimination. It was due to the nature of how the strategies employed for rule selection in each case worked, as they were not the same.

With forward as a search direction, the processing started with the empty set of rules and then, gradually, in each step some recalled rules were added. These rules could include

Table II
CHARACTERISTICS OF RULE SETS WITH FILTERING DRIVEN BY RANKINGS

	Reducts - Female						OneR - Female						Greedy - Female					
	Back			Forw			Back			Forw			Back			Forw		
Attr	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp
20	4121	4.8	6.6	4121	4.8	6.6	4121	4.8	6.6	4121	4.8	6.6	4121	4.8	6.6	4121	4.8	6.6
19	4120	4.8	6.6	3830	4.7	6.8	4120	4.8	6.6	3935	4.8	6.4						
18	4119	4.8	6.6	3660	4.7	6.7	4119	4.8	6.6	3644	4.7	6.6						
17	4119	4.8	6.6	2557	4.5	7.6	4118	4.8	6.6	3491	4.7	6.5						
16	4118	4.8	6.6	2071	4.4	8.1	4118	4.8	6.6	2804	4.5	6.9	4117	4.8	6.6	805	4.0	10.2
15	4117	4.8	6.6	1608	4.4	8.1	4117	4.8	6.6	1957	4.4	7.8	4113	4.8	6.6	548	3.8	11.2
14	4112	4.8	6.6	1204	4.2	8.7	4116	4.8	6.6	1412	4.2	8.5	4101	4.8	6.6	414	3.7	11.7
13	4107	4.8	6.6	902	4.0	9.2	4113	4.8	6.6	1097	4.0	8.8	4097	4.8	6.6	387	3.6	11.2
12	4097	4.8	6.6	624	3.9	9.6	4109	4.8	6.6	688	3.8	10.1	4093	4.8	6.6	369	3.6	11.4
11	4077	4.8	6.6	369	3.6	11.0	4102	4.8	6.6	442	3.5	11.3	4091	4.8	6.6	346	3.6	11.0
10	4042	4.8	6.5	232	3.4	12.8	4080	4.8	6.5	276	3.3	13.2	4051	4.8	6.6	206	3.3	13.1
9	4033	4.8	6.5	215	3.3	12.4	4027	4.8	6.5	182	3.1	14.5	3985	4.8	6.5	127	2.9	15.2
8	3957	4.8	6.5	132	3.0	14.5	3914	4.8	6.5	114	2.9	16.5	3890	4.8	6.5	99	2.7	15.6
7	3860	4.9	6.5	97	2.8	15.4	3705	4.8	6.5	73	2.6	18.3	3719	4.9	6.3	68	2.5	15.8
6	3675	4.9	6.6	47	2.7	22.3	3445	4.8	6.6	56	2.6	19.0	3563	4.9	6.3	55	2.5	16.6
5	3365	4.9	6.5	30	2.5	27.5	3015	4.8	6.5	26	2.4	29.0	3252	4.8	6.3	42	2.5	16.8
4	2651	4.8	6.8	18	2.3	32.1	2651	4.8	6.8	18	2.3	32.1	2656	4.8	6.3	27	2.4	18.1
3	2351	4.8	6.6	10	2.2	**37.6**	1932	4.7	6.6	9	2.1	**36.4**	2382	4.8	6.0	16	2.3	16.5
2	1548	4.6	6.6	4	**2.0**	34.0	1548	4.6	6.6	4	**2.0**	34.0	1548	4.6	6.6	4	**2.0**	**34.0**
1	378	**3.9**	**7.2**	0	0.0	0.0	378	**3.9**	**7.2**	0	0.0	0.0	1224	4.8	6.3	0	0.0	0.0

	Reducts - Male						OneR - Male						Greedy - Male					
	Back			Forw			Back			Forw			Back			Forw		
Attr	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp	NoR	Len	Supp
22	15283	5.1	5.9	15283	5.1	5.9	15283	5.1	5.9	15283	5.1	5.9	15283	5.1	5.9	15283	5.1	5.9
21	15283	5.1	5.9	11850	5.0	6.2	15283	5.1	5.9	12597	5.0	6.1						
20	15282	5.1	5.9	10067	4.9	6.2	15283	5.1	5.9	9835	4.9	6.3						
19	15282	5.1	5.9	8520	4.8	6.4	15283	5.1	5.9	7565	4.8	6.8						
18	15281	5.1	5.9	6155	4.7	6.7	15282	5.1	5.9	6115	4.8	7.2						
17	15278	5.1	5.9	4780	4.6	7.2	15272	5.1	5.9	4224	4.6	7.8	15270	5.1	5.9	2574	4.6	7.6
16	15271	5.1	5.9	3777	4.6	7.1	15256	5.1	5.9	3094	4.6	8.3	15246	5.1	5.9	1768	4.4	8.0
15	15256	5.1	5.9	2537	4.4	7.7	15229	5.1	5.9	2086	4.4	9.3	15212	5.1	5.9	1270	4.2	9.1
14	15231	5.1	5.9	1625	4.2	8.5	15184	5.1	5.9	1247	4.3	10.6	15150	5.1	**5.9**	902	4.1	10.0
13	15188	5.1	5.9	1125	4.2	8.8	15129	5.1	5.9	931	4.2	11.4	15035	5.1	5.9	554	3.9	11.2
12	15133	5.1	5.8	789	4.1	9.9	15023	5.1	5.9	609	4.0	12.5	14826	5.1	5.8	333	3.7	11.1
11	14972	5.1	5.8	486	4.0	10.1	14974	5.1	5.9	517	3.9	12.6	14549	5.1	5.8	210	3.5	12.3
10	14791	5.1	5.8	307	3.8	11.8	14717	5.1	5.9	359	3.8	13.3	14074	5.1	5.8	157	3.5	14.5
9	14484	5.1	5.8	199	3.6	12.2	14403	5.1	5.8	221	3.6	15.4	13871	5.1	5.8	132	3.3	14.0
8	14118	5.1	5.8	121	3.5	15.3	14104	5.1	5.8	147	3.7	15.7	13188	5.1	5.7	87	3.1	14.2
7	13454	5.1	5.7	62	3.3	17.1	13562	5.1	5.8	69	3.3	21.5	12563	5.2	5.7	51	3.0	17.1
6	12952	5.1	5.7	33	2.9	19.9	12543	5.1	**5.9**	42	3.2	22.3	11791	5.2	5.5	30	2.8	17.9
5	11807	5.2	5.7	18	2.6	28.2	11695	5.1	5.8	21	2.8	24.9	10844	5.2	5.6	23	2.9	18.2
4	9876	5.1	5.7	11	2.5	36.2	10043	5.1	5.6	9	2.1	37.9	9887	5.2	5.6	11	2.5	30.4
3	7492	5.1	**6.0**	5	2.2	50.8	7782	5.1	5.7	4	2.3	47.8	7485	5.2	5.4	7	2.0	**39.0**
2	5338	5.1	5.8	2	**2.0**	**61.5**	5522	5.1	5.5	1	**2.0**	**60.0**	5240	5.1	4.9	4	**1.8**	36.5
1	1596	**4.7**	4.3	0	0.0	0.0	1596	**4.7**	4.3	0	0.0	0.0	1596	**4.7**	4.3	0	0.0	0.0

conditions limited to the variables in the subset considered. In the first step only the top ranking attributes were taken into account, in the second step the top two were accepted, and so on. If a rule also contained conditions on other features (placed somewhere lower in the ranking), then it was not included in the recalled set. Therefore, always a ranking position that was processed directly gave the number of variables studied, and a subset of features present in the rules was explicitly visible.

The strategy applied in the backward elimination of decision rules started with the entire set of rules and then the groups of rules were gradually excluded, taking into account conditions on attributes from the lowest positions in a ranking. In this case, the assumption was that the eliminated rules should contain only attributes considered and discarded so far in the ranking. If a rule also included conditions on other features that were higher ranking, then such a rule was kept in the remaining set. This processing resulted in operation on higher numbers of rules for the same ranking position than when compared to the strategy applied in the forward selection. In fact, for each ranking position a set of rules recalled by forward selection was a subset of rules retained by backward elimination. It was especially striking in the case of the Greedy ranking and the number of rules obtained as characteristics for the constructed rule sets.

The advantage of this strategy was visible in the classification results, in particular for the female data set, where for almost every position of the ranking, the accuracy of rule-based classifiers was at least as good as the reference level considered for all variables from the set. Thus, this direction and the filtering rule strategy contributed to enhancing the power of the classifier. The drawback of such processing lies in keeping in considerations the higher numbers of attributes, and a lack of clear specification of their subset taken into account in each step. If there was a rule referring to all features, such rule would be kept to the very end, to the last step of filtering process, despite its significant length that indicates too close definition to be of any practical use, as the probability of exactly the same detailed pattern among test samples is low.

When the characteristics of the obtained rule sets were analysed, the classification accuracy of the constructed classifier was treated with extra attention. To provide the general look on the rule filtering process, over the entire run of feature and rule selection, the average performance was calculated, and the corresponding standard deviation (per sample), as shown in Table III. For both datasets and all rankings, backward elimination always brought better results than forward selection when combined with their strategies for rule selection. When the averaged performance is compared with the reference points of accuracy for the entire set of attributes available, it is clear that the backward search resulted in the improvement for all rankings for female writer dataset, while for male writers that was true for the OneR and Greedy rankings. For female writers the highest average classification accuracy was obtained for the Reduct-based ranking and for male writers for Greedy ranking. On the other hand, for female writers standard deviation reflected almost only direction and not ranking, yet for male writers the highest (but still rather small, only fractional) values were obtained for Greedy ranking.

Table III
SUMMARY OF OBTAINED ACCURACY OF RULE-BASED CLASSIFIERS

	Ranking and search direction					
	Reduct		OneR		Greedy	
	Back	Forw	Back	Forw	Back	Forw
	Female					
Average	0.989	0.939	0.968	0.949	0.968	0.937
St.dev.	0.01	0.11	0.01	0.11	0.01	0.12
	Male					
Average	0.877	0.840	0.899	0.870	0.910	0.817
St.dev.	0.02	0.14	0.03	0.16	0.04	0.18

The rule characteristics can be treated as dimensions in an optimisation space. Among them, the performance and the ranking position for which undiminished performance was reported could also be included. Such a summarising look was given in Table IV, where the lowest values are preferred for: ranking position, number of rules, average rule length. The highest values are preferred for: classification accuracy and average rule support. No overall Pareto points were detected, but for each dimension, some maxima and minima can be observed, or groups of characteristics could be analysed. When the same values of the observed criterion were recorded more than once, the occurrence for the highest ranking position was selected as the best, since it corresponded to the most extensive reduction, as long as it happened while the observed performance was not lower than the reference point.

As the forward selection strategy quantitatively enlarged the set of rules, it can be noted that these were moderately short rules with relatively large supports, and the number of such rules was small. For the female writer dataset, the highest value of average rule support was 37.6 for an average length of 2.2 and the number of rules equal to 10. For the male writer dataset, the highest value of average rule support was 61.5 for an average length of 2.2 and the number of rules equal to 2.

Table IV
SUMMARY OF OBTAINED BEST RESULTS

Optimality criterion	Female	Male
	Other characteristics	
Acc: **1.0** - F 0.967 - M	Ranking: Greedy, Pos: 11 Direction: Forw NrR: 346, AvgL: 3.6, AvgS: 11.0	Ranking: Greedy, Pos: 6 Direction: Back NrR: 11791, AvgL: 5.2, AvgS: 5.5
NrR: **16** - F 21 - M	Ranking: Greedy, Pos: 3 Direction: Forw Acc: 0.973, AvgL: 2.3, AvgS: 16.5	Ranking: OneR, Pos: 5 Direction: Forw Acc: 0.906, AvgL: 2.8, AvgS: 24.9
AvgL: **2.3** - F 2.8 - M	Ranking: Greedy, Pos: 3 Direction: Forw Acc: 0.972, NoR: 16, AvgS: 16.5	Ranking: OneR, Pos: 5 Direction: Forw Acc: 0.906, NoR: 21, AvgS: 24.9
AvgS: **32.1** - F 24.9 - M	Ranking: Reducts, OneR, Pos: 4 Direction: Forw Acc: 0.967, NoR: 18, AvgL: 2.3	Ranking: OneR, Pos:5 Direction: Forw Acc: 0.906, NoR: 21, AvgL: 2.8
Position: **1** - F 2 - M	Ranking: Reducts, OneR Direction: Back, Acc: 0.961 NoR: 378, AvgL: 3.9, AvgS: 7.2	Ranking: OneR Direction: Back, Acc: 0.922 NoR: 5522, AvgL: 5.1, AvgS: 5.5

In the case of the backward elimination strategy, the cut in the number of rules was generally smaller than for the forward search. The smallest reduction occurred for the Greedy ranking, for the female set. For the male set, the number of rules corresponding to the attribute in the highest ranking position was the same for all rankings, similarly the average rule length. Furthermore, for this dataset, the number of rules decreased about 10 times under this search strategy. For the Reduct and OneR rankings and female writers it was even greater.

The experiments carried out with varying search directions and strategies for rule selection enabled studying the effectiveness of the three rankings in the rule filtering process. The proposed Greedy ranking held its ground against the other two, leading to noticeably improved predictions for rule sets of decreased cardinalities, which is evidenced by the fact how often it led to the best results given in Table IV, and which clearly illustrates its merits.

V. CONCLUSIONS

The paper provides an illustrative example for the proposed research methodology dedicated to decision rule filtering governed by attribute rankings. The process of rule selection was executed with sequential backward reduction, where an entire set of induced rules is available at the beginning and then some elements from this set are discarded; and with sequential forward search, where the processing starts with the empty set to which recalled elements are added gradually. Along with two search directions, two strategies for rule selection were used, one with recalling rules including conditions only on variables from the currently considered subset, and the other with finding rules dependent on at least one of the attributes in the studied set.

In the investigations, three rankings of attributes were employed. The proposed ranking based on the percentage of separated rows and the properties of the greedy algorithm was confronted with the previously defined ranking referring to decision reducts, and the OneR ranker available in the popular WEKA environment. For the three rankings, the selection of rules was performed in the two directions, and the resulting rule sets were analysed with respect to the properties of constituent decision rules, such as their numbers, average length, and average support, but also from the point of evaluation of performance for all constructed rule-based classifiers when applied for labelling of samples from test sets.

The results from the experiments indicate that for all three rankings and search paths it was possible to obtain a noticeable reduction of attributes while at least maintaining the power of inducers, at the same time improving characteristics of rule sets. The special focus on Greedy ranking enabled to discover that it not only led to discarding some variables from the available sets, treating them as irrelevant, but also proved effective for rule filtering.

Future research will include application of the Greedy ranking in the feature selection process for other types of inducers, with different mathematical backgrounds and modes of operation. Also, the influence of discretisation step will be studied, as one of the factors greatly influencing representation of data and the patterns present in it.

Acknowledgment

The research works presented in the article were carried out at the Institute of Computer Science, University of Silesia in Katowice, Sosnowiec, Poland, and within the statutory project of the Department of Graphics, Computer Vision and Digital Systems (RAU-6, 2023), at the Silesian University of Technology, Gliwice, Poland.

References

[1] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, “Feature selection: A data perspective,” *ACM Comput. Surv.*, vol. 50, no. 6, 2017.

[2] I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, Eds., *Feature Extraction: Foundations and Applications*, ser. Studies in Fuzziness and Soft Computing. Physica-Verlag, Springer, 2006, vol. 207.

[3] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artificial Intelligence*, vol. 97, no. 1, pp. 245–271, 1997.

[4] U. Stańczyk, “Weighting of features by sequential selection,” in *Feature Selection for Data and Pattern Recognition*, ser. Studies in Computational Intelligence, U. Stańczyk and L. Jain, Eds. Berlin, Germany: Springer-Verlag, 2015, vol. 584, pp. 71–90.

[5] I. Witten, E. Frank, and M. Hall, *Data Mining. Practical Machine Learning Tools and Techniques*, 3rd ed. Morgan Kaufmann, 2011.

[6] B. Zielosko and U. Stańczyk, “Reduct-based ranking of attributes,” in *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES-2020, Virtual Event, 16-18 September 2020*, ser. Procedia Computer Science, M. Cristani, C. Toro, C. Zanni-Merk, R. J. Howlett, and L. C. Jain, Eds., vol. 176. Elsevier, 2020, pp. 2576–2585.

[7] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. CRC Press, 2007.

[8] E. Amaldi and V. Kann, “On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems,” *Theoretical Computer Science*, vol. 209, no. 1, pp. 237–260, 1998.

[9] B. Zielosko and M. Piliszczuk, “Greedy algorithm for attribute reduction,” *Fundam. Informaticae*, vol. 85, no. 1-4, pp. 549–561, 2008.

[10] M. M. Mafarja and S. Mirjalili, “Hybrid whale optimization algorithm with simulated annealing for feature selection,” *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[11] P. Pudil, J. Novovièová, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[12] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[13] U. Stańczyk, B. Zielosko, and L. C. Jain, “Advances in feature selection for data and pattern recognition: An introduction,” in *Advances in Feature Selection for Data and Pattern Recognition*, ser. Intelligent Systems Reference Library, U. Stańczyk, B. Zielosko, and L. C. Jain, Eds. Springer, 2018, vol. 138, pp. 1–9.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[15] W. Altidor, T. M. Khoshgoftaar, and J. V. Hulse, “An empirical study on wrapper-based feature ranking,” in *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 2009, pp. 75–82.

[16] Z. Pawlak and A. Skowron, “Rough sets and boolean reasoning,” *Information Sciences*, vol. 177, no. 1, pp. 41–73, 2007.

[17] A. Janusz and D. Ślęzak, “Utilization of attribute clustering methods for scalable computation of reducts from high-dimensional data,” in *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2012, pp. 295–302.

[18] Y. Yang, D. Chen, H. Wang, E. C. Tsang, and D. Zhang, “Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving,” *Fuzzy Sets and Systems*, vol. 312, pp. 66–86, 2017.

[19] Y. Liu, L. Zheng, Y. Xiu, H. Yin, S. Zhao, X. Wang, H. Chen, and C. Li, “Discernibility matrix based incremental feature selection on fused decision tables,” *International Journal of Approximate Reasoning*, vol. 118, pp. 1–26, 2020.

[20] J. Henzel, A. Janusz, M. Sikora, and D. Ślęzak, “On positive-correlation-promoting reducts,” in *Rough Sets*, R. Bello, D. Miao, R. Falcon, M. Nakata, A. Rosete, and D. Ciucci, Eds. Springer International Publishing, 2020, pp. 213–221.

[21] J. Wróblewski, “Ensembles of classifiers based on approximate reducts,” *Fundam. Informaticae*, vol. 47, no. 3–4, p. 351–360, 2001.

[22] J. Bazan and M. Szczuka, “The rough set exploration system,” in *Transactions on Rough Sets III*, ser. Lecture Notes in Computer Science, J. F. Peters and A. Skowron, Eds. Berlin, Heidelberg: Springer, 2005, vol. 3400, pp. 37–56.

[23] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[24] R. Holte, “Very simple classification rules perform well on most commonly used datasets,” *Machine Learning*, vol. 11, pp. 63–91, 1993.

[25] S. Ali and K. A. Smith, “On learning algorithm selection for classification,” *Applied Soft Computing*, vol. 6, no. 2, pp. 119–138, 2006.

[26] M. J. Moshkov, M. Piliszczuk, and B. Zielosko, “Greedy algorithm for construction of partial association rules,” *Fundam. Informaticae*, vol. 92, no. 3, pp. 259–277, 2009.

[27] ——, “On construction of partial reducts and irreducible partial decision rules,” *Fundam. Informaticae*, vol. 75, no. 1-4, pp. 357–374, 2007.

[28] B. Zielosko, “Sequential optimization of γ-decision rules,” in *Federated Conference on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proceedings*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2012, pp. 339–346.

[29] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the American Society for Information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.

[30] M. Eder, “Style-markers in authorship attribution a cross-language study of the authorial fingerprint,” *Studies in Polish Linguistics*, vol. 6, no. 1, pp. 99–114, 2011.

[31] H. Wu, Z. Zhang, and Q. Wu, “Exploring syntactic and semantic features for authorship attribution,” *Applied Soft Computing*, vol. 111, p. 107815, 2021.

[32] S. G. Weidman and J. O’Sullivan, “The limits of distinctive words: Re-evaluating literature’s gender marker debate,” *Digital Scholarship in the Humanities*, vol. 33, pp. 374–390, 2018.

[33] U. Stańczyk and G. Baron, “On heterogeneity or sub-classes aspect in construction of stylometric input datasets,” in *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 26th International Conference KES-2022, Verona, Italy, 7-9 September 2022*, ser. Procedia Computer Science, M. Cristani, C. Toro, C. Zanni-Merk, R. J. Howlett, and L. C. Jain, Eds. Elsevier, 2022, vol. 207, pp. 2526–2535.

[34] U. M. Fayyad and K. B. Irani, “Multi-interval discretization of continuousvalued attributes for classification learning,” in *13th International Joint Conference on Articial Intelligence*, vol. 2. Morgan Kaufmann Publishers, 1993, pp. 1022–1027.

DOI: 10.15439/2023F3017
ISSN 2300-5963 ACSIS, Vol. 35

Automatic Colorization of Digital Movies using Decolorization Models and SSIM Index

Andrzej Śluzek, Marcin Dudziński, Tomasz Świsłocki
0000-0003-4148-2600, 0000-0003-4242-8411, 0000-0002-7679-7533
Warsaw University of Life Sciences - SGGW,
Institute of Information Technology,
Nowoursynowska 159, bld.34,
02-776 Warsaw, Poland
Email: {andrzej_sluzek, marcin_dudzinski, tomasz_swislocki}@sggw.edu.pl

***Abstract*—Re-colorization of images or movies is a challenging problem due to the infinite RGB solutions for a monochrome object. In general, the process is assisted by humans, either by providing colorization hints or relevant training data for ML/AI algorithms. Our intention is to develop a mechanism for fully unguided (and with no training data used) colorization of movies. In other words, we aim to create acceptable colored counterparts of movies in domains where only monochrome visualizations physically exist (e.g. IR, UV, MRI, etc. data). Following our past approach to image colorization, the method assumes arbitrary *rgb2gray* models and utilizes a few probabilistic heuristics. Additionally, we maintain the temporal stability of colorization by locally using *structural similarity* (SSIM) between adjacent frames. The paper explains the details of the method, presents exemplary results and compares them to the *state-of-the art* solutions.**

NOTE: ***All figures are best viewed in color and high resolution.***

I. Introduction and Motivation

COLORIZATION of monochrome objects is an ill-posed problem due to the infinite number of RGB solutions for given grayscale data. Nonetheless, this topic holds notable practical and commercial significance, particularly in the restoration of historical photos and movies, e.g. [1], [2].

In general, the development of colorization techniques involves incorporating more and more human knowledge and expectations into the algorithms [1]. To that end, earlier methods involved providing reference color images [3], [4] or manually *scribbling* important fragments [5], [6]. Recently, AI-based techniques have dominated, with architectures designed to learn color patterns suitable for specific domains, semantics, and/or contents, such as [7], [8].

Some works consider recognizing/learning the image domain (or specific objects within images) to further improve the results, e.g., [9], [10].

In other works, inter-domain transfer learning is considered [11], or multiple alternative colorizations are proposed by exploiting learned probability densities [12], [13].

Colorization methods for monochrome movies basically follow the same principles. In some works (for example, [2]), there is no distinction between the colorization of still images and movies. That is, each frame of a movie is considered a separate item and is re-colorized individually. However, such an approach may introduce temporal colorization discontinuities. Therefore, in recent works, including [14], [15] or [16], the issue of colorization continuity between adjacent frames is addressed, mainly by combining spatial and temporal consistencies of colors assigned to pixels with similar intensity levels.

Nevertheless, in all typical papers on re-colorization of monochrome objects, the algorithms are assisted by human knowledge/experience/expectations, even if the assistance is disguised as *training datasets* of relevant images/videos.

Therefore, the solution discussed in this paper may seem audacious. We propose a mechanism for fully automatic video colorization without additional metadata, manual assistance, learning processes, or domain identification. In other words, we aim to create plausible colored representations of *gray worlds* using only monochrome videos as the data source. By "plausible," we mean results that are visually attractive, statistically repeatable, and deliver convincingly rich sensations of colors.

It should be noted that we exclude simple pseudo-coloring techniques that use a limited number of colors corresponding to the number of intensity levels, resulting in a limited richness and diversity of coloristic effects.

The problem addressed appears to be relevant and practical because there are numerous domains, such as infrared, ultraviolet, ultrasound, MRI, X-ray, and others, where only single-channel visual data physically exist. Then, realistic-looking automatic colorizations can be synthesized for various reasons, even if it is just for aesthetic purposes (see examples in Fig. 1).

In our recent papers [17], [18], we overviewed the results obtained for unguided colorization of monochrome images, and the method for colorizing videos is a natural extension of those results. Therefore, Section II revisits a number of assumptions and techniques adopted in monochrome image colorization and repeated in video colorization.

Section III presents steps that are specific to the colorization of videos. A summary of the experimental results and corresponding comments is provided in Section IV. Finally, the concluding Section V discusses some supplementary details (mainly related to limiting the number of solutions) and outlines prospective directions for future research.

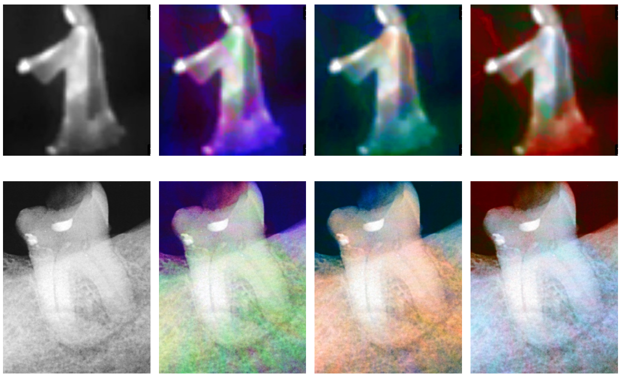


Fig. 1. Examples of monochrome images from non-visual domains (IR and X-ray), and their color-rich and visually plausible colorizations.

II. Principles of Automatic Colorization

A. Significance of Decolorization Models

Generally, colorization methods assume that the intensity of grayscale objects defines the luminance channel of the colored outputs, so only two channels of chrominance need to be reconstructed. It seems like a matter of arbitrary choice which chrominance models are used (e.g., CIELab in [8], [9] or YUV in [2], [5]). Sometimes, the model is not even specified, and only the final outcomes in RGB format are discussed.

However, almost no papers on re-colorization address the opposite question: **How was the original (real or hypothetical) RGB object decolorized to obtain a grayscale object?**

Typically applied decolorization (*rgb2gray*) models, i.e. YUV or YIQ, assume linear functions of primary colors:

$$I = k_R R + k_G G + k_B B \tag{1}$$

where $k_R = 0.299$, $k_G = 0.587$ and $k_B = 0.114$ (or, based on [19], $k_R = 0.2126$, $k_G = 0.7152$ and $k_B = 0.0722$).

Nevertheless, given that we assume that colored objects only exist hypothetically, any combinations of non-negative coefficients k_R, k_G, and k_B can be formally used (subject to the straightforward condition $k_R + k_G + k_B = 1$).

Therefore, as shown in Fig. 2, a colored object can be converted into a variety of its monochrome counterparts, which may significantly differ depending on the adopted *rgb2gray* model.

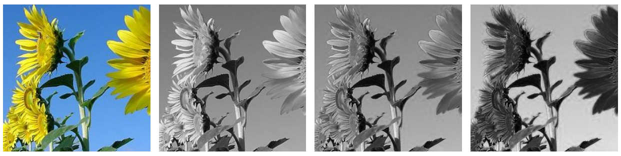


Fig. 2. A colored image and its three monochrome variants obtained using three different *rgb2gray* models, namely $[0.299, 0.587, 0.114]$, $[0.44, 0.14, 0.42]$ and $[0.14, 0.11, 0.75]$).

Correspondingly, by assuming a certain *rgb2gray* model, we can restrict the re-colorization results in a specific way. Any pixel with intensity I can only be assigned colors that satisfy Eq. 1 with the adopted $[k_R, k_G, k_B]$ values. Fig. 3 shows examples of the same image re-colorized using the same algorithm, but with different *rgb2gray* models adopted.

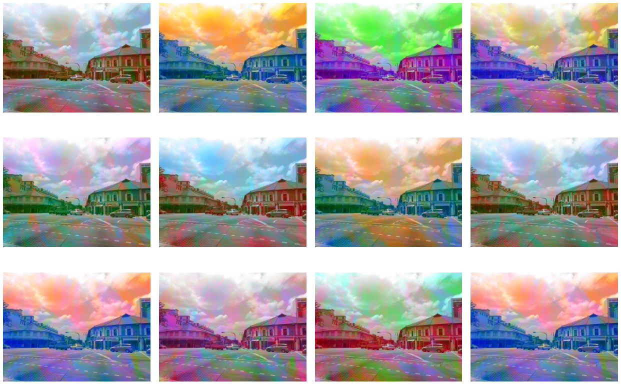


Fig. 3. The same monochrome image colorized (by the method discussed in [17], [18]) with different *rgb2gray* models adopted.

B. Colorization of Individual Pixels

In coloring monochrome objects depicting some non-existent *color worlds*, we are not restricted by the propertiess of YUV or YIQ models, which provide the best consistency between the brightness of the color and its monochrome counterpart, as perceived by human observers. Therefore, in the developed colorization scheme, we assume that:

Monochrome visual objects are derived from (only hypothetically existing) color counterparts by an *rgb2gray* model with arbitrarily selected k_R, k_G, and k_B coefficients.

If we assume that the coefficients are uniformly sampled into n values each (for example, $n = 101$ for a 0.01 stepsize), it can be easily obtained that the total number of available *rgb2gray* models is:

$$N = \frac{n(n+1)}{2} \quad or \quad N = \frac{(n-3)(n-2)}{2} \tag{2}$$

The first variant includes zero-coefficient models (e.g., $[0.4, 0, 0.6]$ or $[0, 1, 0]$). These models are excluded in the second variant, where each primary color must have a non-zero contribution to the intensity levels.

In the experiments, we use 0.05, 0.01 or 0.002 stepsizes, so that the total numbers of *rgb2gray* models are $231(171)$, $5151(4851)$ or $125,751(124,251)$. However, in the end, we only consider a limited number of $20 - 40$ models (see Sections IV and V for more details), resulting in the same number of alternative colorization outputs.

In rendering digital images, we adopt finite numbers of intensities and colors. Thus, given a monochrome intensity level I_p from the discrete range 0 to 255, it can only be assigned colors from a pool of colors that satisfy (subject to color discretization) Eq. 1. The size of those pools varies, depending on the intensity level. As an example, the numbers of colors assigned to intensity levels in two arbitrarily selected *rgb2gray* models are shown in Fig. 4.

The figure shows that the largest pools of colors are for mid-range intensities, with the number of colors gradually decreasing to a single choice for extremely dark or light intensities. (Monochrome white/black should remain white/black in color.)

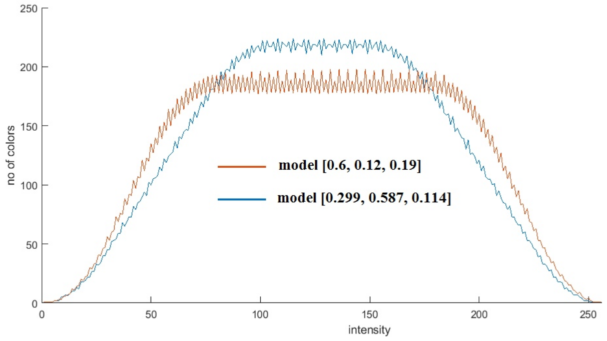
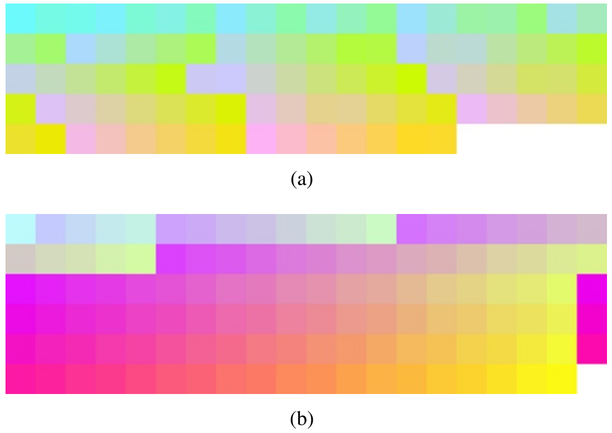


Fig. 4. Numbers of RGB colors assigned to intensities in two exemplary *rgb2gray* models.

As an example, Fig. 5 displays the pool of colors for intensity level 208 under two *rgb2gray* models. Note that the first model is actually YUV, and all colors are perceived as having almost the same brightness. In contrast, the perceived brightness of colors varies significantly for the second model.


(a)

(b)

Fig. 5. Colors assigned to 208 intensity in (a) $[0.299, 0.587, 0.114]$ and (b) $[0.69, 0.12, 0.19]$ *rgb2gray* models.

With no prior information provided, all colors available to the I_p level can be assigned to a pixel of that intensity with the same probability. However, if the pixel has an adjacent pixel with an intensity I_1 and its already assigned color C_1, the probabilities of colors that could be assigned to I_p should be influenced by the intensity and color of the neighbor.

Therefore, we propose a simple but (as shown later) surprisingly effective heuristic rule:

The greater the difference in brightness between adjacent pixels, the higher the likelihood that their assigned colors will also differ significantly.

Under this rule, we prioritize colors from the pool available to the I_p level, which are at distances from the C_1 color proportional to the difference in intensity levels $\|I_p - I_1\|$.

Let's assume a pixel with I_p intensity, which has an already colored neighbor with I_1 value and C_1 color. Let $\mathbf{C} = \{C_{p_1}, ..., C_{p_N}\}$ be the list of colors assigned to I_p in the adopted *rgb2gray* model.

The neighbor (with I_1 value and C_1 color) contributes a color from the above **C** list. First, the list is ordered by the distances of its colors from C_1, i.e. $\mathbf{Cmod} = \{C_{p_{i1}}, ..., C_{p_{iN}}\}$, where

$$\|C_{p_{in}} - C_1\| \leq \left\|C_{p_{i(n+1)}} - C_1\right\| \tag{3}$$

It should be noted that inter-color distances are measured in the HSV space, as differences in this space are sufficiently close to represent perceived color similarities, e.g. [20].

Then, a uniform distribution is used to randomly select a color from a specific sub-range of the **Cmod** list. The location of this sub-range depends on the difference $\|I_p - I_1\|$. In general, for smaller differences, the sub-range is narrower and shifted to the left of **Cmod**, while for larger differences, it is wider and shifted towards the end of **Cmod**). Detailed description of this step can be found in [18].

In particular, if the neighboring pixel has the same intensity, i.e. $I_p = I_1$, the I_p intensity pixel would be (preliminarily) assigned the same color C_1.

Actually, images are colored incrementally (details in the following subsection), and it may happen that an uncolored pixel has several already-colored neighbors. Then, the color selection can be performed several times for that pixel, and the final choice is the mean of the colors obtained from all colored neighbors.

$$C_p = \frac{1}{K}\sum_{j=1}^{K} C_j \tag{4}$$

where $K = 1, 2, 3$ or 4.

This step allows us to generate a larger variety of colors than from the (possibly limited) pools of colors that are assigned to individual intensity levels.

C. Incremental Colorization

Colorization of monochrome images is performed incrementally, starting from the darkest and brightest pixels, for which the unique color choice exists (see Fig. 4). They are considered the initial list of already colorized pixels for the algorithm, which is actually a randomized variant of a popular *flood-fill* method.

Pseudo-code of the colorization algorithm is provided below.

Step 4 (which does not exist in the standard flood-fill method) is introduced to randomize the expansion of colorized patches, i.e. to avoid unnecessary regularities in the colorization process.

Algorithm 1 Monochrome image colorization

```
Require: Initial list $L$ of colored pixels
Ensure: Colorized images
remove from $L$ pixels which do not have uncolored neighbors;
copy $L$ to $Q$ queue
while $Q$ is not empty do
    move a random pixel from $Q$ to the front of $Q$;
    get the front pixel $p$ of $Q$;
    if $ps$ (south neighbor of $p$) exists and is uncolored then
        colorize $ps$ (Eq. 4);
        append $ps$ to the end of $Q$;
    end if
    repeat Steps 6-9 for $pn$, $pe$, $pw$ (north, east, west neighbors of $p$);
    remove $p$ from $Q$;
end while
```

D. Other Remarks

Despite the heavy presence of randomizing factors, the results produced by the outlined method are surprisingly repeatable, depending only (as expected) on the adopted *rgb2gray* model. Actually, all images shown in Figs 1 and 3 are generated by the method, and many more examples can be found in [17], [18].

The visual plausibility of the results can be further improved by projecting the colors of each pixel p onto the corresponding YUV plane $I_p = 0.299R + 0.587G + 0.114B$ (or $I_p = 0.2126R + 0.7152G + 0.0722B$). This is because colors assigned to the same intensity level can vary significantly in terms of their perceived brightness, see Fig. 5b. Projecting the colors on the YUV planes unifies the perceived brightness of colors with intensities of the original monochrome image, although it may alter the colors somewhat, as shown in Fig. 6.

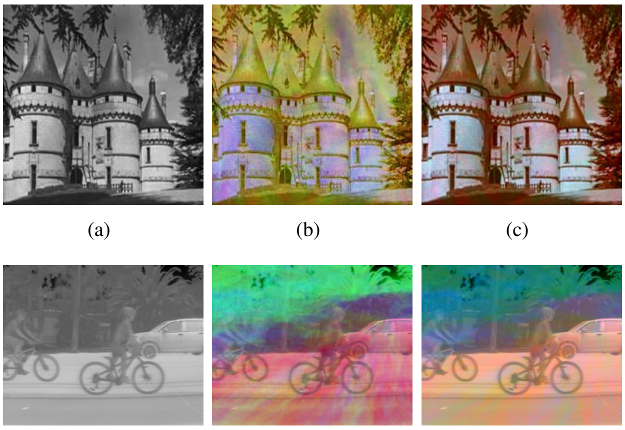


Fig. 6. Original monochrome images (a) and their colorizations before (b) and after (c) projections on the YUV planes.

III. From Images to Movies

Formally, the colorization of monochrome movies does not differ from the colorization of images, and both operations are often considered to be almost equivalent, e.g, [21]. Each frame can be independently colorized to provide the corresponding frame of the color movie. Such a simplified approach may be used if image colorization performs nearly flawlessly, which can only be achieved in specific and well-defined domains. Nevertheless, this method has been successful in a number of works, including the commercial system described in [2].

In our application, we cannot use this approach due to randomizing factors in the image colorization scheme. In such cases, even if the same *rgb2gray* model is used, adjacent frames with nearly identical content may have noticeably different colorization. An example is given in Fig. 7. Thus, the temporal continuity/stabilization of rendered colors over sequences of similar frames is particularly important for our problem.

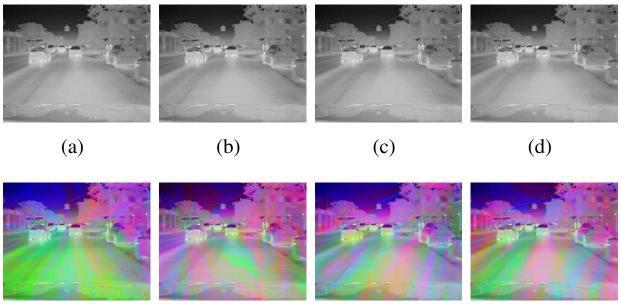


Fig. 7. Four subsequent monochrome frames individually colorized (using the same *rgb2gray* model) by the method outlined in Section II.

Recently, more attention has been paid to the temporal continuity of colors in the colorized movies. Several papers, e.g., [14], [15], [16], consider the regularization or stabilization of colors in adjacent video frames. In [22], a more general problem of stabilizing any visual properties in processed video files is considered.

Nonetheless, in all these papers, the results are obtained by training dedicated neural networks on sufficiently representative ground-truth data. Therefore, these approaches are not applicable to the considered problem of colorizing monochrome videos for which the color counterparts never existed.

In the proposed solution, we assume that the colorization continuity should reflect not only intensity similarities between adjacent frames but also content similarities, particularly the local ones.

After evaluating several popular metrics of image quality and similarity, e.g. [23], [24], the *structural similarity index measure* (SSIM) has been identified as the top candidate, [25]. SSIM is particularly suitable for monochrome images and is applicable to images of any content.

In general, the SSIM is defined by a weighted combination of three measures that broadly represent statistical similarities between the intensity, contrast, and structure of two image samples, X and Y:

$$SSIM(X,Y) = i(X,Y)^{\alpha} \times c(X,Y)^{\beta} \times s(X,Y)^{\gamma} \quad (5)$$

where $i(X,Y) = \frac{2\mu_X\mu_Y + c_1}{\mu_X^2 + \mu_Y^2 + c_1}$, $c(X,Y) = \frac{2\sigma_X\sigma_Y + c_2}{\sigma_X^2 + \sigma_Y^2 + c_2}$ and $s(X,Y) = \frac{\sigma_{XY} + c_3}{\sigma_X\sigma_Y + c_3}$.

Depending on the area over which means and standard deviations are computed, SSIM can indicate image similarities either globally (i.e., between whole images) or locally (i.e., between small neighborhoods of the same coordinates (m,n) in both images).

When coloring the current frame in the context of the previous one, we actually utilize both aspects of SSIM.

First, we calculate the global SSIM measure between the current monochrome frame (to be colored) and its monochrome predecessor (already colored). If the value is too low (the recommended threshold is 0.5), we assume no perceptual similarity between the frames, and the current frame is colorized independently.

In practice, such situations occur infrequently, mainly when assembling longer movies from shorter unrelated fragments, and high values of global SSIM similarity between adjacent frames can typically be expected. For example, the SSIM values between the neighboring pairs of monochrome frames in Fig. 7 are as follows: 0.9511 (frames *a,b*), 0.9756 (frames *b,c*), and 0.9508 (frames *c,d*).

Therefore, we normally define colors of the colorized frame I_k as weighted combinations of the independent colorization of I_k and colors of the previous frame I_{k-1}, i.e., for given pixel coordinates (m,n) we use the *local* values of SSIM:

$$Col_k(m,n) = sim \times Col_{k-1}(m,n) + (1 - sim) \times Col_k(m,n) \quad (6)$$

where $sim = SSIM(I_k(m,n), I_{k-1}(m,n))$.

Additionally, the colors computed by Eq. 6 are projected onto the corresponding YUV planes (defined by pixel intensities in frame I_k) as explained in Section II-D.

Fig. 8 provides exemplary effects of the proposed color regularization over neighboring frames. First, we display the local SSIM indexes between the monochrome frames from Fig. 7 in Figs 8(a-c). Then, the lower row of Fig. 8 shows the colored frames after the regularization.

IV. Experiments

The proposed method of monochrome movie colorization involves heuristic assumptions, arbitrary model selection, and probabilistic computational schemes. Furthermore, the obtained results cannot be objectively assessed since reference or ground-truth results are assumed to be non-existent.

Therefore, the performance of the method and quality of its outputs can only be evaluated through extensive experimentation. In particular, the final results are typically assessed using subjective criteria such as *visual plausibility*, *coloristic attractiveness*, *aesthetic value*, etc.

One of the main challenges in conducting such experiments is the large number of potential *rgb2gray* models, as discussed in Subsection II-B. The sheer quantity of alternative colorizations may be overwhelming for human evaluators. Therefore, it is necessary to reduce the number of effectively considered *rgb2gray* models.

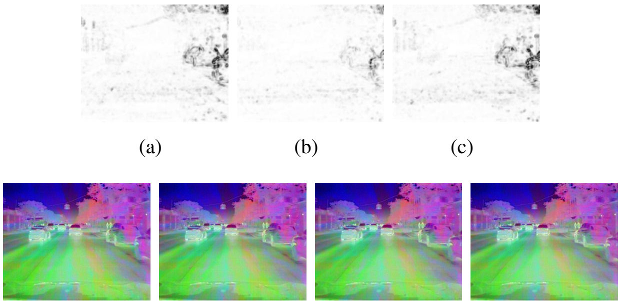


Fig. 8. SSIM maps (intensities proportional to the numerical values) for Fig. 7 pairs of monochrome frames: (a) for frames *a,b*, (b) for frames *b,c* and (c) for frames *c,d*. The bottom row displays the results after the color regularization.

For moderate numbers of adopted *rgb2gray* models, e.g., $231(171)$ models with 0.05 stepsize, we found that the most plausible solutions are normally obtained from the $10 - 15\%$ of results with the lowest value of *colorfulness* (details of this metric are provided in [1], [26]). Specifically, when the monochrome objects depict scenes from the real world, this subset typically includes solutions that vaguely align with human coloristic expectations (more information in [17]).

With a large number of models, e.g., $5151(4851)$ or $125,751(124,251)$, the models are preliminarily clustered into a recommended number of classes ($20 - 40$), and only the cluster medoids are used. Details of the clustering algorithm are outlined in Subsection V-A.

In any case, the users are presented with a limited number of suggested colorizations, from which they can select the most satisfactory option. Subject to the quality constraints of the original monochrome movies, the results always appear attractive and convincing, resembling scenes from 'fairytale lands'. Therefore, the preferred colorized version becomes a matter of personal choice.

In the experiments, the non-visual (IR) movies mainly come from FLIR ADAS[1] and CAMEL [27] datasets. The visual-frequency movies, which are used to better highlight the differences between our approach and the 'traditional' re-colorization expectations, primarily come from personal collections.

This section includes short representative frame sequences as illustrations. For example, Fig. 9 once again confirms that colorization supported by SSIM-based regularization provides a natural-looking continuity of colored frames, free of flickering and artifacts.

Fig. 10 showcases a rather unusual (but occasionally possible) result of re-colorization of a real-world movie. It can be observed that the selected colorization option appears even more natural than the original color movie!

[1] https://www.flir.eu/oem/adas/adas-dataset-form/

Similar effects can be observed in Fig. 11, where the re-colorized frames of an underwater movie appear more authentic than the original shots.

Nevertheless, in typical cases, even if the monochrome movie has ground-truth colors, there is no correspondence between the original colors and their re-colorized versions. In other words, there is no distinction between colorizing monochrome movies with or without existing color originals. The results appear visually plausible, but they may or may not meet the ground-truth coloristic expectations, with the latter being more typical (see examples in Fig. 12).

Finally, Fig. 13 provides exemplary colorization results (arbitrarily selected from a number of alternative results) for a sequence of frames extracted from an outdoor IR movie.

Overall, the experimental results confirm that plausible video colorization can be achieved fully automatically, without the need for learning from relevant training data, human assistance, or supplementary metadata. In other words, it appears possible to synthesize realistically-looking colorful immersion into the 'gray worlds' of monochrome visual data. With the mechanisms provided for pre-selecting the $20-40$ most promising *rgb*2*gray* models, users can choose their preferred colorization version from a limited yet sufficiently diverse number of alternative solutions.

However, considering that only subjective assessment criteria are currently utilized, presenting the actual video clips would be a more suitable approach to report the experimental results.

A. Comparing to SOTA

It is generally assumed that AI-based methods deliver *state-of-the-art* (SOTA) results for the colorization of monochrome objects. In particular, some works on video colorization, including [14], [16], use individual frame colorization by SOTA AI methods as benchmarks for the proposed algorithms.

Following the same approach, we colorized sequences of frames from the tested videos using the publicly available (at https://deepai.org/machine-learning-model/colorizer) tool which applies one of the most advanced re-colorization methods (outlined in [2]).

The results shown in Fig. 14 are utterly disappointing. While *visual plausibility* is basically unchanged compared to the monochrome images, the other subjective criteria, such as *coloristic attractiveness* or *aesthetic value* fall far below expectations. The color outputs are almost direct replicas of the grayscale values from the monochrome images. Apparently, when facing unfamiliar contents, the algorithm decides to keep the original monochrome colorization. In other words, satisfactory and visually attractive AI-based colorization is not possible for monochrome objects for which no coloristic knowledge or experiences are available. Therefore, the practicality of the proposed approach is somewhat boosted.

TABLE I
COEFFICIENTS $[k_R, k_G, k_b]$ OF 32 ADOPTED *rgb*2*gray* MODELS.

$[0.04, 0.77, 0.19]$	$[0.05, 0.18, 0.77]$
$[0.05, 0.05, 0.90]$	$[0.25, 0.13, 0.62]$
$[0.83, 0.05, 0.12]$	$[0.59, 0.11, 0.30]$
$[0.12, 0.71, 0.17]$	$[0.75, 0.21, 0.04]$
$[0.50, 0.20, 0.30]$	$[0.25, 0.70, 0.05]$
$[0.42, 0.44, 0.14]$	$[0.04, 0.91, 0.05]$
$[0.14, 0.39, 0.47]$	$[0.58, 0.38, 0.04]$
$[0.40, 0.33, 0.27]$	$[0.70, 0.04, 0.26]$
$[0.14, 0.55, 0.31]$	$[0.04, 0.37, 0.59]$
$[0.52, 0.04, 0.44]$	$[0.71, 0.15, 0.14]$
$[0.34, 0.04, 0.62]$	$[0.41, 0.55, 0.04]$
$[0.26, 0.44, 0.30]$	$[0.13, 0.24, 0.63]$
$[0.28, 0.27, 0.45]$	$[0.04, 0.58, 0.38]$
$[0.40, 0.14, 0.46]$	$[0.12, 0.83, 0.05]$
$[0.57, 0.28, 0.15]$	$[0.26, 0.58, 0.16]$
$[0.16, 0.06, 0.78]$	$[0.91, 0.06, 0.03]$

V. CONCLUDING REMARKS

A. Limiting the number of alternative solutions

The main practical obstacle in prospective applications of the proposed approach is (as highlighted in Subsection II-B and Section IV) the large number of available *rgb*2*gray* models. Human observers are unable to assess all possible colorization outputs, so limiting the number of models to a limited (but sufficiently diversified in terms of the produced results) is an important issue. This is the outline of the proposed remedy.

As an alternative to the representations given in Figs 4 and 5, the colors assigned to the selected intensity I can be visualized as the polygonal intersection of the $k_R R + k_G G + k_B B = I$ plane with the RGB color space cube. An example is provided in Fig. 15 (note locations of the centers of gravity of the depicted polygons).

Thus, the 256×3 matrix of gravity center coordinates of such polygons for $I = 0, ..., 255$ can be considered a compact representation of the adopted *rgb*2*gray* model. The matrix can be nicely visualized by a (discrete) curve winding from *black* to *white* in the RGB cube (see examples in Fig. 16). Those curves will be referred to as *mean-color curves*.

Then, the *rgb*2*gray* models can be clustered by clustering their *mean-color curves* (i.e. 256-dimensional arrays of $3D$ coordinates). Eventually, *medoids* of the obtained clusters are identified, and only the models corresponding to those medoids are used for colorization.

Therefore, we adopt a limited number of models (for example, only 32 clusters are built regardless the total number of models) which are as diversified as possible in terms of their statistical coloristic properties.

For the total number of 4851 *rgb*2*gray* models (i.e., 0.01 stepsize), the list of adopted model is given in Table I. It can be noted that one of the models, namely $[0.26, 0.58, 0.16]$, is quite similar to the standard YUV model with $[0.299, 0.587, 0.114]$ coefficients.

B. Summary

In this paper, we propose a method for addressing the ill-posed problem of colorizing monochrome movies without

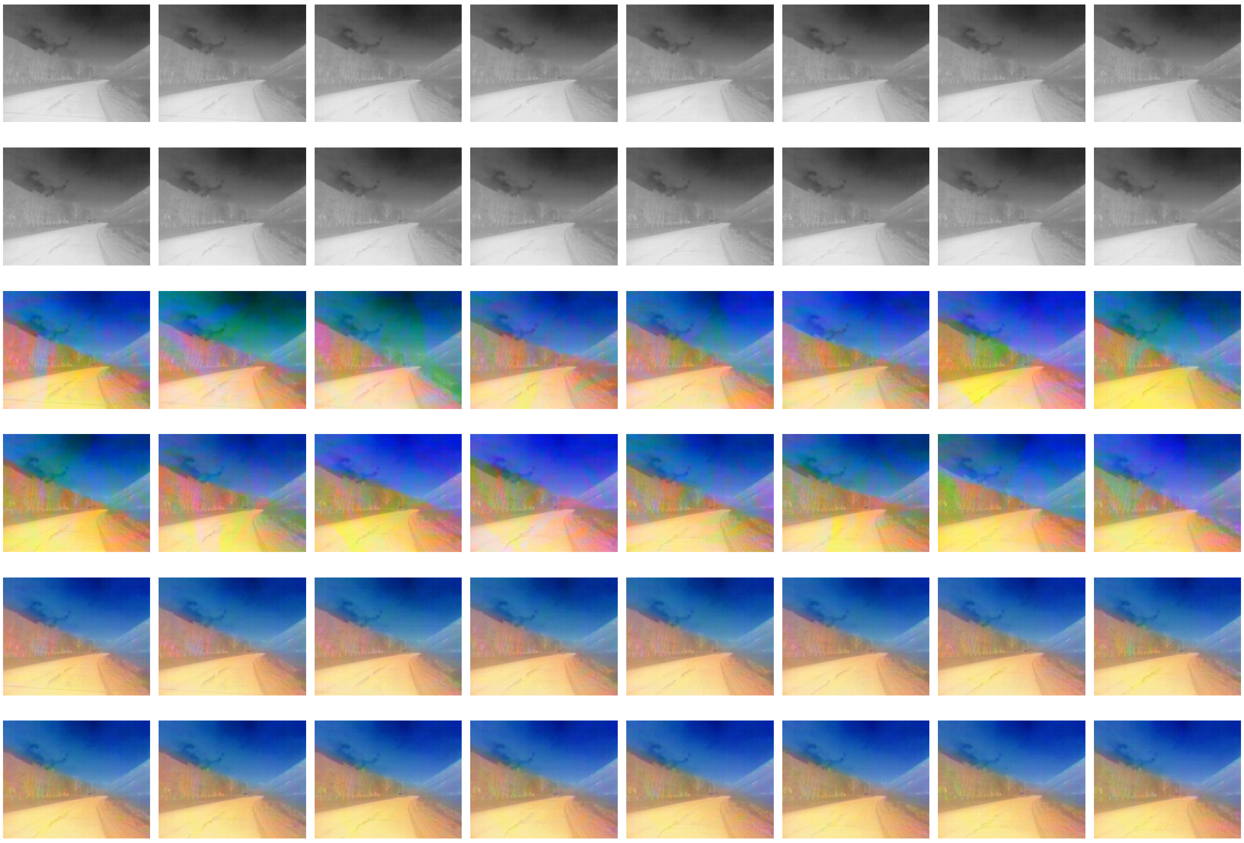


Fig. 9. Monochrome frames of an IR movie (two top rows), frames colorized individually (two middle rows), and frames colorized with the color regularization (two bottom rows).

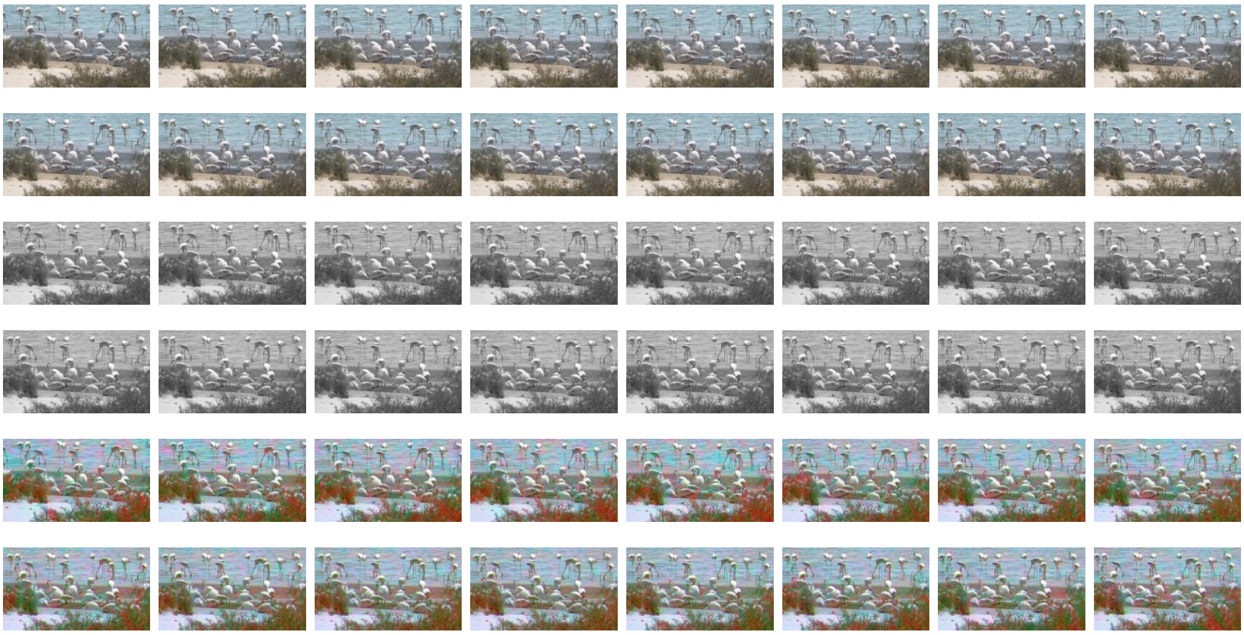


Fig. 10. The original color frames of a movie (two top rows), their decolorized variants (two middle rows), and one of the achieved re-colorization options (two bottom rows).

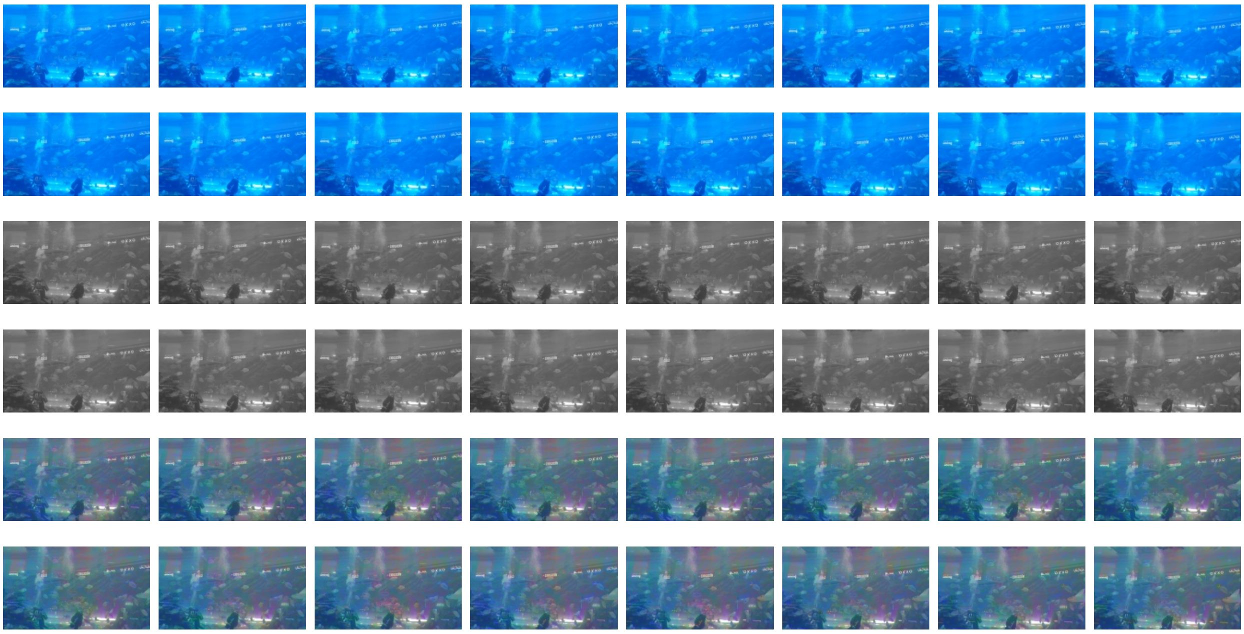


Fig. 11. The original color frames of an underwater movie (two top rows), their decolorized variants (two middle rows), and one of the achieved re-colorization options (two bottom rows).

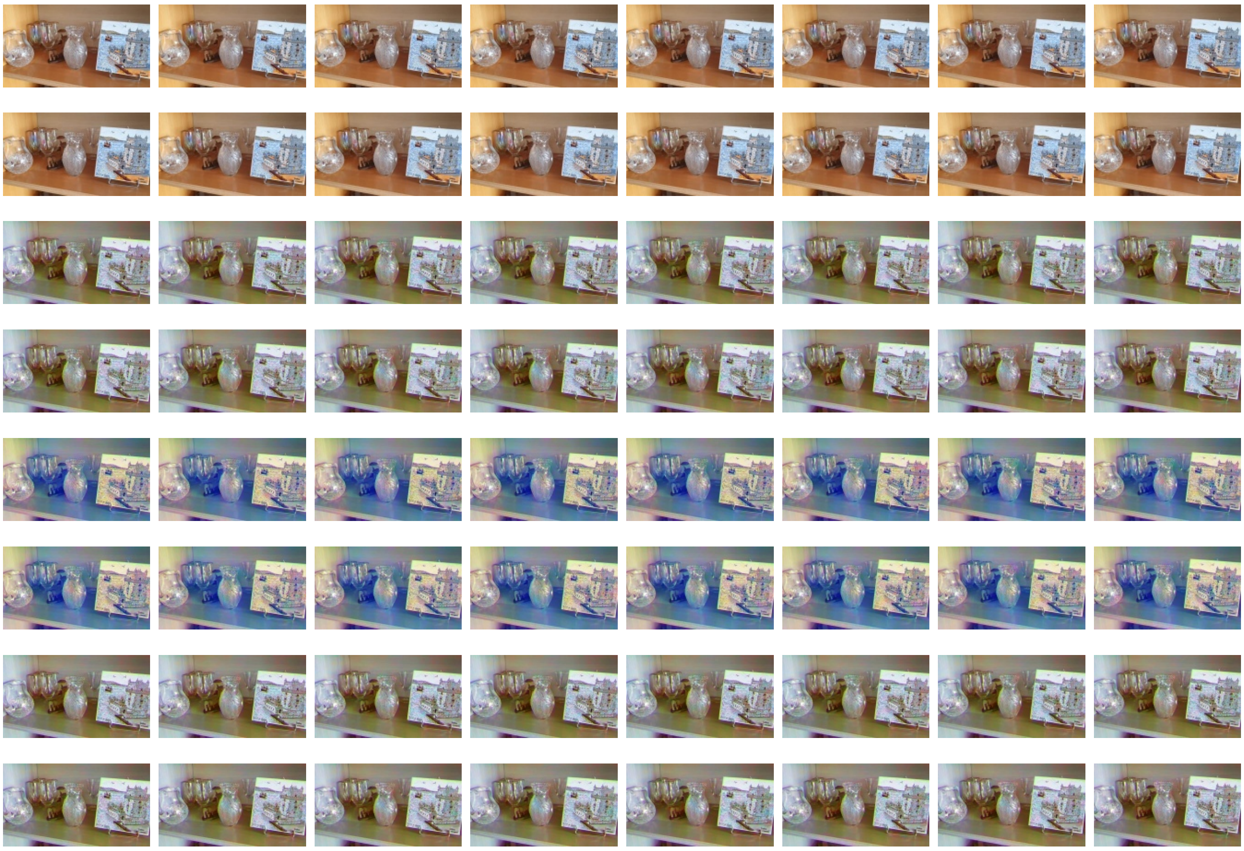


Fig. 12. The original color frames (two top rows), and their exemplary re-colorization options (pairs of lower rows).

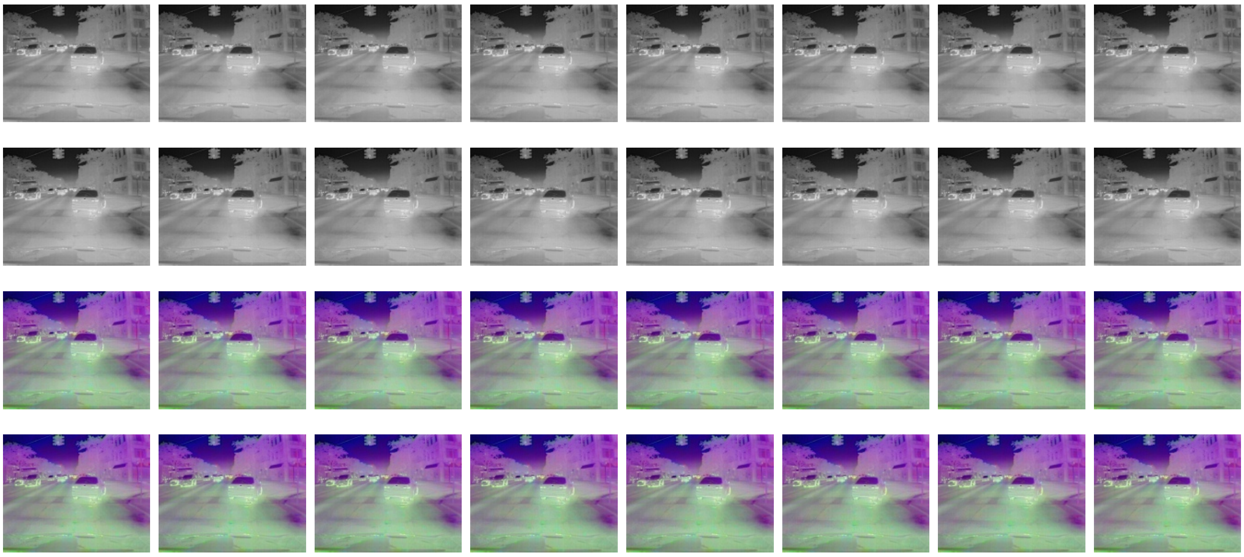


Fig. 13. An exemplary sequence of monochrome frames from an IR movie (top two rows) and its arbitrarily selected colorization (bottom two rows).

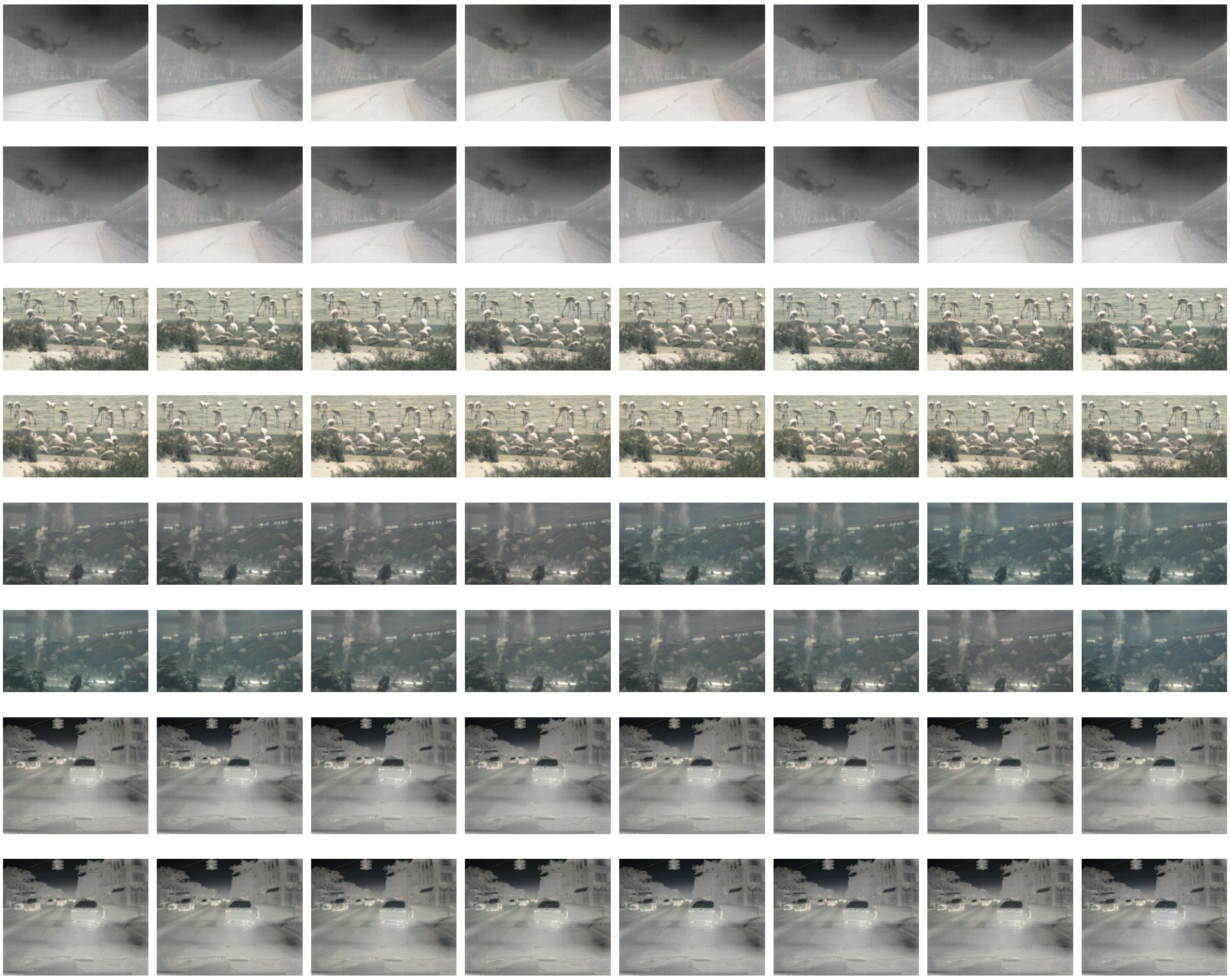


Fig. 14. Sequences of monochrome frames from (top to bottom) Figs 9, 10 , 11 and 13 colorized by an AI-based SOTA algorithm.

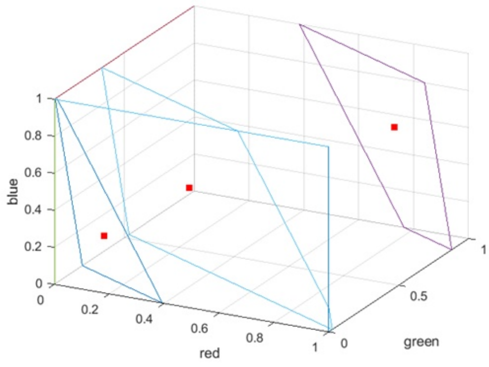
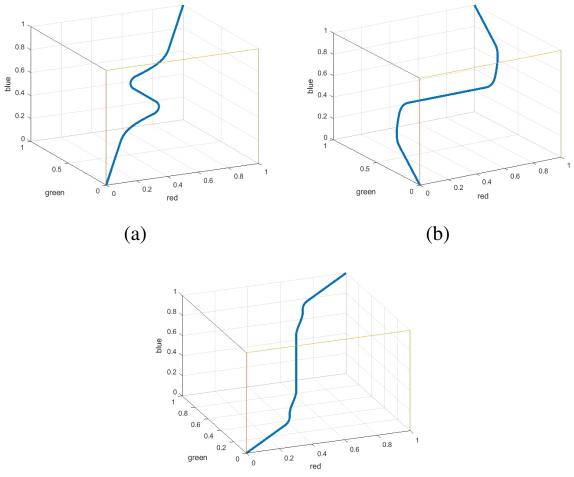


Fig. 15. Distribution of colors assigned to three exemplary intensities 30, 80 and 208 under the $I = 0.299R + 0.587G + 0.114B$ *rgb2gray* model. Red dots indicate the centers of gravity of the intersection polygons.



(a)

(b)

(c)

Fig. 16. *Mean-color curves* representing locations of polygon centers for intensity levels ranging from 0 to 255 for three *rgb2gray* models: (a) $[0.299, 0.587, 0.114]$, (b) $[0.69, 0.12, 0.19]$ and (c) $[0.13, 0.175, 0.695]$.

any direct or indirect human assistance. Our method builds upon our recent results in colorization of monochrome images, where we assume the use of arbitrary decolorization (*rgb*2*gray*) models.

The movie colorization process involves two operations: image colorization and temporal stabilization of rendered colors. First, individual frames are colored using simple probabilistic heuristics and a randomized *flood-fill* technique, starting from the initial queue of darkest/brightest pixels with deterministic color choices. In the second operation, we utilize the SSIM similarity index to determine whether and to what extent color continuity should be maintained between adjacent frames.

While a large number of *rgb*2*gray* models can hypothetically be used, we can pre-select a limited number of sufficiently diverse variants. Users can then choose their preferred colorization from these options, typically based on personal preference.

The method is primarily designed for colorizing monochrome movies in domains where no actual color data exists, such as IR, UV, MRI, etc. In other words, our goal is to transform the monochrome data into convincingly realistic color versions of these *gray worlds*. This may be necessary for various reasons, including aesthetic considerations.

In future work, our intention is to focus on the following problems that have not yet been adequately addressed:

- Analysis of the mathematical properties of the method, which includes exploring alternative probability distributions used in the adopted heuristics, investigating the local (individual frames) and global (movies) convergence of colorization results, etc.
- Developing metrics for the objective evaluation of colorization results, including the selection of appropriate assessment criteria.
- Optimizing the code, including parallelization techniques, optimizing data structures, and other strategies. The ultimate objective may involve achieving real-time performance.
- Integrating the method with selected AI techniques to enhance its capabilities and explore potential synergies.

References

[1] I. Zeger, S. Grgic, J. Vukovic, and G. Sisul, "Grayscale image colorization methods: Overview and evaluation," *IEEE Access*, vol. 9, pp. 113 326–113 346, 2021. doi: 10.1109/ACCESS.2021.3104515

[2] A. Salmona, L. Bouza, and J. Delon, "Deoldify: A review and implementation of an automatic colorization method," *Image Processing On Line*, vol. 12, pp. 347–368, 2022. doi: 10.5201/ipol.2022.403

[3] R. Irony, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Eurographics Symposium on Rendering (2005)*. The Eurographics Association, 2005. doi: 10.2312/EGWR/EGSR05/201-210. ISBN 3-905673-23-1. ISSN 1727-3463

[4] R. Gupta, A. Chia, D. Rajan, E. Ng, and Z. Huang, "Image colorization using similar images," in *20th ACM Int. Conf. on Multimedia (MM'12)*, 2012, pp. 369–378.

[5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Transactions on Graphics*, vol. 23, pp. 689–694, 06 2004. doi: 10.1145/1015706.1015780

[6] A. Popowicz and B. Smolka, "Fast image colourisation using the isolines concept," *Multimedia Tools and Applications*, vol. 75, pp. 15 987–16 009, 2017. doi: 10.1007/s11042-016-3892-2

[7] R. Zhang, P. Isola, and A. Efros, "Colorful image colorization," in *Computer Vision – ECCV 2016*. Springer, 2016. doi: 10.1007/978-3-319-46487-9_40 pp. 649–666.

[8] E. Farella, S. Malek, and F. Remondino, "Colorizing the past: Deep learning for the automatic colorization of historical aerial images," *Journal of Imaging*, vol. 8, p. 269, 10 2022. doi: 10.3390/jimaging8100269

[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," *ACM Transactions on Graphics*, vol. 35, pp. 1–11, 07 2016. doi: 10.1145/2897824.2925974

[10] J. Su, H. Chu, and J. Huang, "Instance-aware image colorization," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2020. doi: 10.1109/CVPR42600.2020.00799 pp. 7965–7974.

[11] H. Lee, D. Kim, D. Lee, J. Kim, and J. Lee, "Bridging the domain gap towards generalization in automatic colorization," in *Computer Vision – ECCV 2022*. Springer, 2022. doi: 10.1007/978-3-031-19790-1_32 pp. 527–543.

[12] A. Deshpande, J. Rock, and D. Forsyth, "Learning large-scale automatic image colorization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. doi: 10.1109/ICCV.2015.72 pp. 567–575.

[13] A. Royer, A. Kolesnikov, and C. Lampert, "Probabilistic image colorization," in *Proc. British Machine Vision Conference (BMVC)*. BMVA Press, September 2017. doi: 10.5244/C.31.85 pp. 85.1–85.12.

[14] C. Lei and Q. Chen, "Fully automatic video colorization with self-regularization and diversity," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.00387 pp. 3748–3756.

[15] M. Hofinger, E. Kobler, A. Effland, and T. Pock, "Learned variational video color propagation," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds. Cham: Springer Nature Switzerland, 2022. ISBN 978-3-031-20050-2 pp. 512–530.

[16] M. G. Blanch, N. O'Connor, and M. Mrak, "Scene-adaptive temporal stabilisation for video colourisation using deep video priors," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds. Cham: Springer Nature Switzerland, 2023. ISBN 978-3-031-25069-9 pp. 644–659.

[17] A. Śluzek, "Do we always need ai for image colorization?" *Proceedings of the 4rd Polish Conference on Artificial Intelligence, Lodz, Poland*, April 2023, in print.

[18] A. Śluzek, "On unguided automatic colorization of monochrome images," in *Proc. 31 Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2023, Plzen, Czechia.*, May 2023, in print.

[19] ITU-R, "Parameter values for the hdtv standards for production and international programme exchange," Geneva, Recommendation BT.709-6, 2015.

[20] E. Jessee and E. Wiebe, "Visual perception and the hsv color system: Exploring color in the communications technology classroom," *Technology Teacher*, vol. 68, no. 1, pp. 7–11, 2008.

[21] S.-Y. Chen, J.-Q. Zhang, Y.-Y. Zhao, P. L. Rosin, Y.-K. Lai, and L. Gao, "A review of image and video colorization: From analogies to deep learning," *Visual Informatics*, vol. 6, no. 3, pp. 51–68, 2022. doi: 10.1016/j.visinf.2022.05.003

[22] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018. ISBN 978-3-030-01267-0 pp. 179–195.

[23] K.-H. Thung and P. Raveendran, "A survey of image quality measures," in *2009 International Conference for Technical Postgraduates (TECHPOS)*, 2009. doi: 10.1109/TECHPOS.2009.5412098 pp. 1–4.

[24] P. Jagalingam and A. V. Hegde, "A review of quality metrics for fused image," *Aquatic Procedia*, vol. 4, pp. 133–142, 2015. doi: https://doi.org/10.1016/j.aqpro.2015.02.019

[25] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. doi: 10.1109/TIP.2003.819861

[26] D. Hasler and S. Suesstrunk, "Measuring colorfulness in natural images," in *Human Vision and Electronic Imaging VIII*, vol. 5007. SPIE, 2003. doi: 10.1117/12.477378 pp. 87–95.

[27] E. Gebhardt and M. Wolf, "Camel dataset for visual and thermal infrared multiple object detection and tracking," in *2018 15th IEEE Int. Conf. AVSS*, 2018. doi: 10.1109/AVSS.2018.8639094 pp. 1–6.

DOI: 10.15439/2023F3625
ISSN 2300-5963 ACSIS, Vol. 35

IoT for the Maritime Industry: Challenges and Emerging Applications

Sheraz Aslam[*], Herodotos Herodotou[*], Eduardo Garro[†], Álvaro Martínez-Romero[†], María A. Burgos[†],
Alessandro Cassera[‡], George Papas[‡], Petros Dias[‡], and Michalis P. Michaelides[*]
[*]Cyprus University of Technology, Limassol 3036, Cyprus.
[†]Prodevelop S.L, 46001 VALENCIA, Spain.
[‡]Eurogate Container Terminal, Limassol, Cyprus.

***Abstract*—The Internet of things (IoT) ecosystem provides a platform for the connectivity of interrelated smart devices to automate manual processes and reduce labor costs. IoT has brought significant benefits to all industries, including maritime, as various objects (e.g., ports, ships, agents, etc.) are connected to gather and share information within the maritime ecosystem. The innovative technological aspects of IoT are promoting the effective collaboration between the research community and the maritime industry, for enhancing the performance of maritime transportation systems. Therefore, this study discusses recent advances delivered by the IoT and other emerging technologies, like machine learning (ML) and computer vision (CV), for smart maritime transportation systems (SMTSs). In particular, this paper presents two specific use cases of SMTSs, namely, predictive maintenance and container damage/seal inspection. Moreover, the key benefits of integrating IoT with ML and CV are highlighted for the above-mentioned use cases. Finally, a discussion is presented to highlight key opportunities along with foreseeable future challenges in adopting these new technologies by the maritime industry.**

I. Introduction

MARITIME transportation is fundamental to the global supply chain. Conventional transportation systems are not inter- or intra-connected. However, with the advancements of information and communication technologies, the concept of connecting everything into smart maritime transportation systems (SMTSs) has emerged [1]. SMTSs use computing, telecommunication, radio-location, internet of things (IoT), and/or automation technologies to improve the performance, management, and safety of transportation systems [2]. In addition, SMTSs communicate information to maritime stakeholders about congestion levels, alternative modes of transportation, or alternate routes. Smart containers are also an essential component of SMTSs, as they enable fine-grained tracking of goods across continents [3]. A terminal is considered “digital” or “smart” if all of its objects are fully interconnected via different communication technologies, e.g., IoT, cloud, etc., to exchange information. A combination of smart sensors and actuators, wireless devices, data centers, and other IoT-based port services form the main infrastructure of smart ports, enabling port authorities to provide more reliable information and various new services to their customers.

A real-time smart port at Xiamen Ocean Gate has been implemented [4] as the first automated container terminal in China that follows the standards of a global automated terminal handling system. Furthermore, it is considered as the first terminal in the world to deploy 5G mobile networks and cloud technologies in its operations [5]. The authors of [6] also proposed several solutions for digitalization in global supply chains, connecting cities and ports via maritime informatics, and sustainable shipping. Furthermore, they also argue that the future of shipping relies on exploiting collaboration via digital data sharing. European ports have also taken initiatives to become smart ports [7]. For example, the Port of Rotterdam is using IoT-based sensors to enable advanced intelligence and create digital twins, the Port of Hamburg is using 5G-based networks to monitor the critical infrastructure by enabling virtual reality, while the Port of Seville is using mobile network technology to monitor goods and traffic in the port in real-time.

Moreover, the sustainability of short sea shipping (SSS) is central to a clean, safe, and efficient European Union (EU) transport system, researched by related Sea Traffic Management (STM) EU projects (such as STMV and STEAM). The study presented in [8] reports on key challenges for advancing reliability, quality, and safety, and removing unnecessary costs and delays at SSS hubs, with a particular focus on Eastern Mediterranean. Specifically, it considers the effect of port-2-port (P2P) communication on port efficiency by investigating the factors influencing the various waiting times at the Port of Limassol, Cyprus, both qualitatively and quantitatively. Finally, measures are proposed for improving agent performance based on the principles of Port Collaborative Decision Making, including P2P communication, data sharing, and transparency among all stakeholders involved in a port call process, and open dissemination of agent-specific key performance indicators (KPIs).

Cargo handling at the Port of Limassol is currently performed by several quay cranes and straddle carriers. These cranes are heavy machines that have multiple subsystems running internally that are managed by different programmable logic controllers (PLCs). The PLCs take input directly from

This research was funded by the European Union’s Horizon Europe program for Research and Innovation through the aerOS project under Grant No. 101069732.

the cranes, execute specific programmed logic, and produce output that allows terminal personnel to operate the crane. PLCs are therefore the most accurate representation of a port crane's status [9]. However, Big Data, artificial intelligence (AI)/ machine learning (ML), and IoT technologies, currently rely on remote servers and/or cloud technologies. So there is a gap between where the data is most accurate, and where the analysis and predictions are performed. This mismatch impacts real-time observability at the upper levels of the terminal system due to higher latency between the central server and the source from which the data originates. These constraints limit the efficiency of the terminal, as it cannot prevent possible unforeseen problems such as outages. Furthermore, there is a need to combine latest technologies with the IoT paradigm in order to enhance the performance of maritime ports. For instance, combination of IoT and ML/deep learning (DL) can enhance the lifetime of maritime equipment and reduce costs through accurate predictive maintenance (PdM). In another scenario, IoT with computer vision (CV) can help reduce human intervention in risky environments, e.g., security seal inspection and damage detection for containers during loading/unloading.

Therefore, this study focuses on how IoT can benefit the maritime industry, especially when combined with other emerging technologies. First, we highlight the research challenges and future directions for the adoption of IoT by the maritime industry. Then, we discuss two specific use cases of SMTSs, namely, PdM and container damage/seal inspection, where IoT is combined with ML/AI and computer vision technologies, respectively.

II. Challenges for IoT adoption

In the shipping industry, a huge amount of data is generated from various sources and formats. The various sources include traffic data, weather data, port call data, cargo data, water level data, and maritime equipment data. In this section, the main open issues related to the deployment of IoT in the maritime industry are presented below.

Real-time data collection and transfer: Data collection and transmission are considered two of the main issues in the use of IoT, ML, and CV technologies in the maritime industry. This is because the data collected by current technologies may be incorrect, unreliable, or incomplete in certain locations or at certain times due to the frequent movement of ships at sea. In addition, a ship is usually equipped with multiple sensors that generate a huge amount of data, and transmitting the data to data centers for further processing is inefficient, creating a major challenge and uncertainty. For example, each sensor requires a particular bandwidth to transmit the data to the database. Hence, new sensor technologies are needed to improve data quality, and high-tech IoT-based communication systems to speed up the data transmission speed [10]. Also, automating data acquisition instead of manual input will improve the quality of data.

Security and privacy concerns: Although modern maritime transportation systems benefit greatly from IoT and other communication technologies, security and reliability risks have increased significantly [11]. The involvement of various maritime stakeholders in planning and managing maritime traffic flows further exacerbates these challenges. Therefore, there is an urgent need to develop an IoT-based collaborative processing system that unifies the modular structure and integrates multiple modules involved in maritime transportation systems. In addition, a common and controlled access mechanism that cannot be manipulated or tampered with by unauthorized parties is also an essential requirement for SMTSs. Moreover, intercommunication and the integration of heterogeneous technologies into IoT-enabled SMTSs offer opportunities not only for the industries but also for cyber criminals. Cyber threat intelligence is an effective security strategy that uses AI models to understand cyber-attacks and can effectively protect IoT-enabled data [12]. In addition, the research community must take advantage of the latest technologies (e.g., ML, CV, blockchain, etc.) by combining them with IoT-assisted systems to find secure and reliable solutions for SMTSs at sea.

Big Data: In the maritime industry, huge amounts of data are generated from various sources every day. For example, Marine Traffic, an automatic identification system (AIS) vessel tracking website, reports collecting 520 million AIS messages daily involving 180 thousand distinct vessels from 3000 active AIS stations worldwide [13]. Port authorities and various port stakeholders (e.g., cargo terminals, tug operators) also collect data on the arrival, berthing, loading, unloading, relocation, anchoring, and departure of ships from ports [8], [14]. Various sensors are also deployed at sea to record data on various oceanographic, environmental, and meteorological parameters of interest, with data volumes reaching up to 5 GB per day [15]. Therefore, developing and designing a reliable and suitable storage architecture to meet the requirements of Big Data in the maritime industry has become a major challenge. Another challenge is the timeliness of processing maritime big data. In the maritime industry, fast and accurate decisions are needed to avoid hazards that relies on fast data processing. Hence, big data processing systems should be able to handle diverse and huge amounts of data, which are constantly increasing. There are several general big data frameworks for processing real-time data, such as Apache Kafka, Elasticsearch, OpenSearch, MongoDB, etc. So, developing an efficient big data framework that can process large maritime data sets is still an open challenge.

Emerging technologies: To ensure higher efficiency of SMTSs, new and emerging technologies must be adopted. In this regard, frugal AI, CV, ML, and IoT, especially when integrated with each other, can play a crucial role in reducing costs and increasing efficiency in smart ports. By using frugal AI models, the system can be lighter and less computationally intensive, which can be beneficial in situations where resources are limited. To prevent cyberattacks on SMTSs, satellite IoT and high-altitude platform solutions can be deployed. With increased GPS jamming and spoofing attacks on maritime systems, satellite IoT can serve as a complementary long-range

Fig. 1: A view of the container terminal at the Port of Limassol, Cyprus

solution and thus be beneficial in many ways [16].

III. IoT for Predictive Maintenance

Container handling equipment (such as ship-to-shore (STS) cranes, gantry cranes, straddle carriers, or reach stackers) require an optimal maintenance process due to the heavy loads, long operating times, and diverse weather conditions, motivating the need for a precise PdM system. Figure 1 shows the container terminal with several cranes and other equipment at the Port of Limassol, Cyprus. Preventive maintenance has traditionally been performed using supervisory control and data acquisition (SCADA) systems set up with human-coded thresholds, warning rules, and configurations to determine when a machine's condition requires repair or even replacement. However, this semi-manual approach does not take into account the more complex dynamic behavior of the machines, nor the contextual data that relates to the operational process as a whole. For this reason, and thanks to recent advances in AI and IoT, the implementation of ML-based solutions is seen as the next functional step that can lead to significant cost savings, higher predictability, and better availability. PdM results in up-to-the-minute knowledge of health status, which allows neither waiting for equipment to shut down (i.e., reactive maintenance) nor performing maintenance when it is not required (i.e., preventive maintenance). In PdM, the IoT can play an important role in predicting faults/failures at an early stage. For example, a number of appropriate sensors can be deployed in machinery/engines to minimize the risk of negligent failures of key components and increase their efficiency [17]. More comprehensive monitoring provides real-time information such as cargo temperature, gas emissions, and other important data that can help optimize operations, reduce maintenance costs, and increase the safety of the entire ecosystem [18].

Based on the current literature, a significant amount of work has already been carried out regarding ML and IoT approaches to solve PdM problems in various industrial environments. This, along with previous research on how the effectiveness of quay container cranes is affected by reduced speed and breakdowns [19] lies the basis for developing PdM approaches (proposed by combining IoT with ML models) applied to port machinery (such as STS cranes and straddle carriers). In addition, most ML projects require sufficient historical data to help understand past failures. This includes general characteristics such as mechanical properties, average usage, and operating conditions. However, even when sufficient data is available, the selection, training, and inference process of the most appropriate ML model for the computational capabilities of the pilot project's infrastructure elements is paramount. Although cloud computing can support predictive analytics solutions, running these models on a remote server introduces potential latency issues that can lead to delayed response times, depending on the quality of connectivity to send data from the device to the cloud over the internet. Edge technology can be considered as a way to optimize the speed and performance of predictive analytics by running ML models locally. Frugal AI models are seen as the optimum approach to (i) predict the remaining useful life of assets with regression models and (ii) predict failures within a given time window with classification models for STS cranes and straddle carriers. In the envisioned architecture, far-edge devices could be used for collecting data from sensors and the machine's PLC, and performing basic pre-processing of the data. Edge processing will be performed for local ML training and inferencing for identifying the need for PdM. Finally, cloud resources could be used for more intense ML training and testing.

IV. IoT for Container Seal Inspection

Containers play an important role in the maritime industry worldwide. Containerization has improved the way cargo is transported around the world by ensuring the safety of cargo in transit. To ensure the security of containers, they are sealed with security seals that prevent an unauthorized entry (a pictorial example of security seals is presented in Fig. 2). During transit, customs officers need to inspect security seals when containers pass through the container terminal gate. The current mechanism for checking the seals relies on visual inspection by humans, which can be time-consuming, labor-intensive, and potentially dangerous. Within the digitalization journey, CV is a rapidly growing field of AI that has many potential advantages in various industries [20], particularly in manufacturing and industrial settings, where visual inspections are critical to ensure product quality and safety. One significant advantage of CV is its ability to automate visual inspection processes. With the help of CV algorithms, it is possible to automate quality control, defect detection, and inspection processes, resulting in increased accuracy, speed, and cost-effectiveness.

One of the most important sources of data for the terminal is the camera system. Thanks to the explosion of customizable internet protocol television (IPTV) cameras, the captured video streams can be used in combination with CV to improve various aspects of the port. In particular, it is important to track the continuous flow of incoming and outgoing containers. Therefore, the development of CV and IoT-based systems will allow the terminal to automatically identify containers

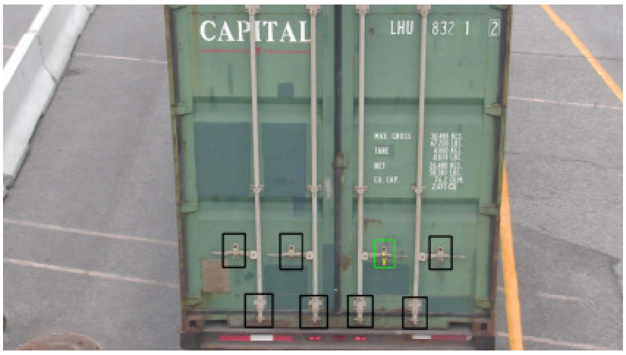


Fig. 2: A container with the security seal shown in a green square; the possible alternative positions for the security seals are shown in black squares.

with damage and verify the presence of proper container seals without requiring human intervention. These two features will be offered to container terminals' customers, providing added value from the quality checkpoint of view. In addition, frugal AI models can provide the automated and low-latency analyses of those video streams on the edge, avoiding human mistakes and at the same time reducing safety risks, thus achieving a secure, and trustable environment for workers.

Based on current literature, several studies exist that are employing IoT and CV for visual inspection processes in different industrial settings. However, we found only a few studies that deal with the maritime industry using CV, IoT, and edge computing based solutions. In the maritime industry, edge technology is considered the way of optimizing the speed and performance of CV analytics by performing ML locally, since despite cloud computing can support CV solutions, performing the inference process on a remote server may lead to potential latency issues, which may cause an unsafe environment for terminal workers. Therefore, an edge-based architecture of container damage/seal inspection via CV and ML technologies would be the most suitable solution. In the envisioned architecture, far-edge devices (i.e., IPTV customizable cameras) could be used for collecting video streams and performing basic pre-processing. Edge processing will be performed for local ML training and inference for identifying the presence of container damages or the absence of safety seals. Finally, cloud resources could be used for more intense ML training and testing.

V. Conclusion

This paper reviews the current status of IoT in SMTSs. The main research gaps identified in this study are: 1) the need to deploy IoT in the maritime industry, 2) the integration of IoT with other emerging technologies such as ML, edge computing, and CV, 3) data security and privacy, and 4) big data collection and storage, while the edge technology-based solution is proposed to optimize the speed and performance of predictive analytics by running ML and CV models locally. Furthermore, these challenges can also be addressed by combining emerging technologies, i.e. IoT, ML, edge computing, and CV, in the maritime industry. This paper explores two such use cases of SMTSs, while combining IoT with other emerging technologies; i.e., ML/DL for PdM, and CV for container damage/seal inspection.

References

[1] S.-J. Chang, G.-Y. Hsu, J.-A. Yang, K.-N. Chen, Y.-F. Chiu, and F.-T. Chang, "Vessel Traffic Analysis for Maritime Intelligent Transportation System," in *Proc. of the 71st Vehicular Technology Conference*. IEEE, 2010, pp. 1–4.

[2] A. Bahnasse, A. Badri, M. Talea, F. E. Louhab, A. Al-Harbi, A. Khiat, and S. Broumi, "WIMAX Technology for Maritime Intelligent Transport Systems Communication," in *Proc. of the 2nd Intl. Conf. on Future Networks and Distributed Systems*. ACM, 2018, p. 10.

[3] S. Aslam, M. P. Michaelides, and H. Herodotou, "Internet of ships: A survey on architectures, emerging applications, and challenges," *IEEE Internet Things J*, vol. 7, no. 10, pp. 9714–9727, 2020.

[4] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu, and O. Postolache, "Internet of Things for Smart Ports: Technologies and Challenges," *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.

[5] "World First 5G Container Port in XIAMEN to Explore Driverless Tech," November 2019, https://www.yicaiglobal.com.

[6] M. Lind, M. Michaelides, R. Ward, and R. T. Watson, *Maritime informatics*. Springer, 2021.

[7] M. Ozturk, M. Jaber, and M. A. Imran, "Energy-Aware Smart Connectivity for IoT Networks: Enabling Smart Ports," *Wireless Communications and Mobile Computing*, 2018.

[8] M. P. Michaelides, H. Herodotou, M. Lind, and R. T. Watson, "Port-2-port communication enhancing short sea shipping performance: The case study of cyprus and the eastern mediterranean," *Sustainability*, vol. 11, no. 7, p. 1912, 2019.

[9] H. S. Bedi and K. Arora, "Monitoring and controlling of industrial crane using programmable logic controllers," *Ind. J. of EEI (IJEEi)*, vol. 3, no. 2, pp. 115–118, 2015.

[10] I. Zaman, K. Pazouki, R. Norman, S. Younessi, and S. Coleman, "Challenges and opportunities of big data analytics for upcoming regulations and future transformation of the shipping industry," *Procedia Eng*, vol. 194, pp. 537–544, 2017.

[11] P. Zhang, Y. Wang, G. S. Aujla, A. Jindal, and Y. D. Al-Otaibi, "A blockchain-based authentication scheme and secure architecture for iot-enabled maritime transportation systems," *IEEE trans Intell Transp Syst*, 2022.

[12] P. Kumar, G. P. Gupta, R. Tripathi, S. Garg, and M. M. Hassan, "Dltif: Deep learning-driven cyber threat intelligence modeling and identification framework in iot-enabled maritime transportation systems," *IEEE trans Intell Transp Syst*, 2021.

[13] "MarineTraffic – A day in numbers," March 2020, https://www.marinetraffic.com/blog/a-day-in-numbers/.

[14] S. Aslam, M. P. Michaelides, and H. Herodotou, "Berth allocation considering multiple quays: A practical approach using cuckoo search optimization," *J Mar Sci Eng*, vol. 11, no. 7, p. 1280, 2023.

[15] I. Lytra, M.-E. Vidal, F. Orlandi, and J. Attard, "A big data architecture for managing oceans of data and maritime applications," in *Intl. Conf. on Engineering, Technology and Innovation*. IEEE, 2017, pp. 1216–1226.

[16] D. Yang, Y. Zhou, W. Huang, and X. Zhou, "5g mobile communication convergence protocol architecture and key technologies in satellite internet of things system," *Alexandria Eng J*, vol. 60, no. 1, pp. 465–476, 2021.

[17] A. Kamolov and S. Park, "An iot-based ship berthing method using a set of ultrasonic sensors," *Sensors*, vol. 19, no. 23, p. 5181, 2019.

[18] M. A. Ben Farah, E. Ukwandu, H. Hindy, D. Brosset, M. Bures, I. Andonovic, and X. Bellekens, "Cyber security in the maritime industry: A systematic survey of recent advances and future trends," *Information*, vol. 13, no. 1, p. 22, 2022.

[19] A. H. S. Mufti, "Quay container crane productivity effectiveness analysis: Case study pt jakarta international container terminal," *Int J Innov Sci Res Technol*, vol. 7, no. 8, August 2022.

[20] F. K. Konstantinidis, S. G. Mouroutsos, and A. Gasteratos, "The role of machine vision in industry 4.0: an automotive manufacturing perspective," in *Intl. Conf. on Imaging Systems and Techniques (IST)*. IEEE, 2021, pp. 1–6.

DOI: 10.15439/2023F1703
ISSN 2300-5963 ACSIS, Vol. 35

Binary Classification of Agricultural Crops Using Sentinel Satellite Data and Machine Learning Techniques

1st Paolo Bertellini
R&D - Data science
Abaco Group
Mantua, Italy
p.bertellini@abacogroup.eu

2nd Gianluca D'Addese
Department of FIM
UNIMORE
Modena, Italy
gianluca.daddese@unipr.it
0000-0001-7755-0893

3rd Giorgia Franchini
Department of FIM
UNIMORE
Modena, Italy
giorgia.franchini@unimore.it
0000-0001-9082-8087

4th Simone Parisi
R&D - Data science
Abaco Group
Mantua, Italy
s.parisi@abacogroup.eu

5th Carmelo Scribano
Department of FIM
UNIMORE
Modena, Italy
carmelo.scribano@unimore.it
0000-0003-1006-7826

6th Daniele Zanirato
Department of Statistical Sciences
University of Bologna
Bologna, Italy
daniele.zanirato@studio.unibo.it

7th Marko Bertogna
Department of FIM
UNIMORE
Modena, Italy
marko.bertogna@unimore.it
0000-0003-2115-4853

***Abstract*—The automated process of determining the crop type carried on plots of land, leveraging data provided by earth observation satellites, represents a highly valuable ability that can serve as a foundation for subsequent analyses or as input for calibrating models, such as Decision Support Systems. This paper presents a study on the task of crop classification starting from indices derived from imagery data provided by ESA Satellites Sentinel 1 and 2. We create a valuable tool to verify farmers' claims, especially in relation to state subsidies for specific crops of interest. To this purpose, we focus on perfecting a binary classification for each of five crops of interest (Tomatoes, Soy, Sugar Beet, Rice, and Wheat), aimed to accurately discern the target crop against any other possible crop. The paper investigates various preprocessing techniques to create a dataset suitable for traditional machine learning methods, which presumes that each land plot to classify is represented by a fixed set of features. To deal with inevitable missing observations caused by clouds or other environmental factors, we investigate different imputation strategies (linear interpolation and constant value filling). Complementary, we study the impact of imbalanced classification labels and evaluate the effectiveness of standard balancing techniques. The findings offer practical implications for monitoring and optimizing agricultural practices in the context of precision farming and sustainable agriculture.**

***Index Terms*—Crop Classification, AgriAI, Machine Learning for Agriculture, Decision Support Systems, Sentinel, Copernicus**

I. Introduction

IN LINE with the objectives for the period 2023-2027 of the European Union's Common Agricultural Policy (CAP), Italy has allocated funds for the production of protein crops (€544 million per year). The aim is to incentivize local agricultural production of crops such as Tomatoes, Soy, Sugar Beet, Rice, and Wheat in local production through the granting of reimbursements by the state. Consequently, there is a pressing need for methods to verify the authenticity of indigenous crops. On-site inspections prove to be costly and inadequate given the scale of the problem, meanwhile satellite remote sensing is a promising technology to classify crops since it can provide periodically large-scale observations of ground objects [11], [13]. The objective of this study is to automate such verification procedures by using Machine Learning (ML) systems applied to satellite data of the relevant geographical areas. Specifically, we focus on the central area of region Emilia-Romagna (Italy) . The problem can be cast as a binary classification: given the data pertaining to a particular field and the farmer's statement regarding the crop, our goal is to verify the veracity of the information given.

The usage of machine learning algorithms for similar tasks is explored in many recent contributions. Among them, Random Forest [3], [7], [12] or decision tree-based [3], [9] classifiers are among the most commonly employed methods for handling this type of data. Data for this paper has been sourced from Sentinel satellites, deployed within the Copernicus program, managed by the European Space Agency (ESA). Sentinel-2 images, have already proven to be a valuable data source for crop mapping in different regions and countries like Central Europe [7], Spain [3], [10], Lebanon [9] and China [12].

The remainder of this paper is structured as follows. Section II, provides a description of the composition of the dataset, highlighting the type of utilized data, the characteristics of the satellite observations, and the distribution of the different crop types. Section III provides e a theoretical overview of the Machine Learning method employed: Random Forests

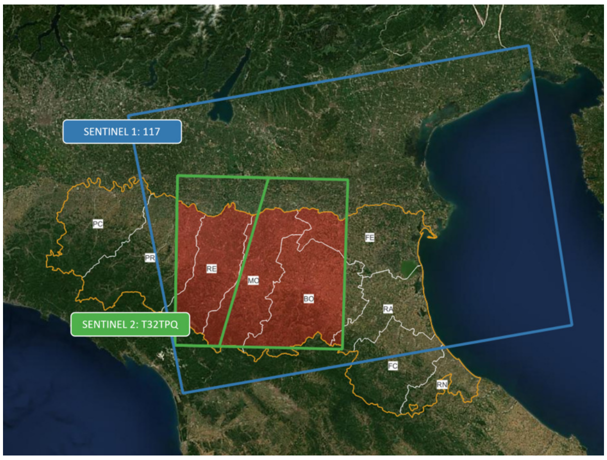


Fig. 1: Study area and locations of ground truth samples (red area). The analyzed fields are located in the central part of the region. This specific area of interest tends to prioritize the production of Wheat, Soy, and Sugar Beet over Rice and Tomatoes.

(RF) with an emphasis on the data pre-processing work, from handling missing dates due to adverse weather conditions to addressing the dataset's imbalance. Section IV presents the performance of the proposed pipeline for crop classification. The aim of this section to assess the accuracy of our model using the complete time frame of the planting process, from seeding to product harvest. In Section V we briefly go through the main result of this work to provide possible branches of further research on the topic.

II. Dataset and Preprocessing

All the data used for the remote sensing purpose in this work is acquired from the earth observation satellites deployed by ESA during the missions Sentinel-1 and Sentinel-2.[1]. Each of these missions deployed in a near-polar sun-synchronous orbit a twin pair of satellites (named Sentinel-1A and Sentinel-1B, Sentinel-2A and Sentinel-2B), which provide sensor observation capabilities depending on the objective of the mission. In particular, Sentinel-1 satellites are equipped with C-band synthetic-aperture radar (C-SAR), while Sentinel-2 satellites are instead equipped with passive Multi-spectral camera operating in 13 distinct bands spanning the spectrum of visible, near-infrared and short wave infrared. Sentinel-1A has a revolution period, hence a temporal revolution, of 12 days, whereas Sentinel-2A and Sentinel 2B of 3 to 5 days, depending on the area. Spatial resolution of Sentinel-1 observations is 10 meters, hence the crop field is discretized in squares of $100m^2$. Sentinel-2 data has a different spatial resolution according to the sensor by which they are collected, either 10 or 20 meters. From a qualitative standpoint, Sentinel-1 active radar sensors imply it always collects the data independently of atmospheric conditions, while Sentinel-2 satellites, relying on passive optical sensors, can produce missing or fragmentary observations due to clouds presence. Starting from the raw observations obtained from the Sentinel satellites, we leverage a total of 16 numerical indices, 12 of them are obtained by Sentinel-2A and Sentinel-2B, and 4 came from Sentinel-1A and Sentinel-1B. Sentinel-1B stopped working in December 2021 and is currently unavailable. Overall, the time frame of the whole dataset spans across 14 months.

A. Data Preparation

Each of the 2 indexes coming from Sentinel-1, named *backscatter* and *coherence*, is further divided in the two polarization VV (co-polarized) and VH (cross-polarized). Backscatter defines the portion of the radar signal that get reflected from the earth's surface straight to the radar antenna, while coherence is defined as the normalized value of the complex cross-correlation between a pair of SAR observation spaced by a period of 12 days. Intuitively, a very low value of the coherence, might indicate a big change in how the field presents itself in 12 days time-difference. To prioritize the number of available observations over the homogeneity of those observations, we also leverage the multiple observations of fields that are visible from partially overlapping orbits This is acceptable since we use pixel statistics (as detailed below), not raw observations, to analyze our data. The differences between observations from different orbits are therefore considered negligible.

The most popular index obtainable from Sentinel-2 observations is the NDVI (Normalized Difference Vegetation index), eq. (1), defined as the ratio of the difference and the sum of the reflected radiation in the near infrared and red, which is a good indicator of the amount of chlorophyll in a field.

$$NDVI := \frac{NIR - RED}{NIR + RED} \tag{1}$$

In order to produce a suitable time series representation for each field, we devise a simple yet effective post-processing strategy of the raw satellite observations. For each record, using its vector geometry, the corresponding patch is cut out for each available observation. Then, each patch is reduced to a set of five statistics (mean, mode, standard deviation, maximum and minimum). The full set of observation obtained over the growing season constitute the field time series that we aim to classify, and it's labeled with a single class identifying the crop being grown.

Our dataset consists of 49 different types of crops and 16,684 sample fields. Among those 49 crops only five are subject to crop-specific subsidies and only for them it is necessary to verify the truthfulness of the farmer's declarations. Getting more specific, the dataset consists of 12,496 samples with specific target crop, divided in: 743 Tomatoes, 63 Rice, 5,214 Wheat, 2,974 Sugar Beet and 3,502 Soy, and 4,188 samples with crops not subject to specific subsidies and therefore considered as ”OTHER” class.

[1] https://sentinel.esa.int/web/sentinel/missions

TABLE I: Amount of crop types for the available fields

Crop	**Amount**
Tomatos	*743*
Soy	*3502*
Sugar Beet	*2974*
Rice	*63*
Wheat	*5214*
Others	*4188*
Total	*16684*

B. Analysis of the used Dataset

Therefore, we propose a qualitative analysis of the available data, this is helpful in order to build an intuition for the possibility to accurately classify the observed crop based on the data gathered. The bar plots (Figure 2a-Figure 2e) depict the variation of the NDVI value across all the fields during the season. The black line represents the behavior of the NDVI average across the fields, while the green vertical bands represent one standard deviation above and one below the mean. The sowing and harvesting phases of the different crops (See Table II) are delimited by the dashed blue and red vertical lines, respectively. The behavior of the NDVI appears to be highly indicative of the growth trend: the blue area (sowing) is usually followed by an increase in the value of the index, whereas the red bands (harvesting) correspond to a steep decrease of the NDVI value. This has a straightforward interpretation given the meaning of the NDVI (indicative of the amount of chlorophyll in the field): after the sowing period, the amount of chlorophyll increases during the vegetative growth, and decreases rapidly with the harvesting. Looking at the width of the green error bars, we can also appreciate the variability changes during the different phases and among different crop types.

III. Method

Since ultimately the objective is to verify if the crop is the one declared by the farmer (among the 5 ones of interest) versus any other different crop type, we propose to train a binary classifier for each of the crop of interest. After the common data preparation steps detailed above, a binary classification dataset is prepared for each of the target crops, by assigning to the samples labeled as it the label positive (1) and to any other sample the label negative (0). At deployment stage, the binary classifier is fed a new unseen crop that should belong to a specific class and assess the truthfulness of the declared crop.

It is known in the literature that Random Forest has great performances in remote sensing classification tasks [1]. A Random Forest (RF) [2] is an ensemble method whose base estimators are decision trees. This method reduces bias and variance thanks to the introduction of multiple uncorrelated voters, because each of the trees has the chance to learn a different pattern in the data and then this knowledge is combined.

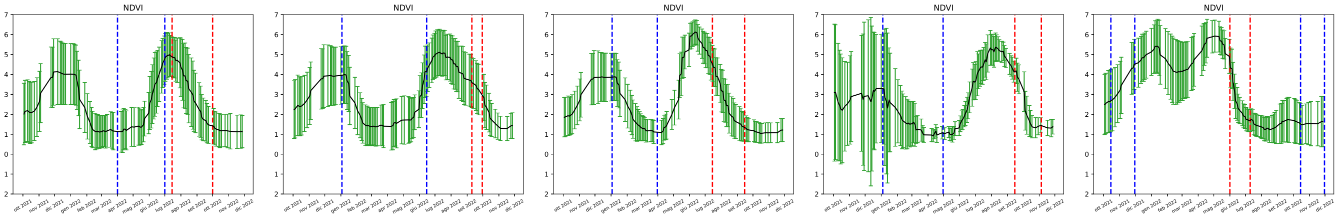
A. Missing data management

Due to variable atmospheric conditions and the nature of Sentinel-2 passive optical sensors, we have a lot of images that were totally or partially covered, hence we opted to disregard covered units up to a certain threshold during preprocessing. In order to represent each field as a fixed set of features, we define a common set of observation dates. While other strategies are possible, we prefer to preserve all the remaining observations by considering all the timestamps that corresponds to at least a single observation in the whole dataset, resulting in 110 valid timestamps. Complete observations of all the 16 indices are usually not available for all the 110 timestamps for each field, Figure 3 illustrates the approximate distribution of the amount of observations per field. We experiment with two standard techniques to fill the missing values. The first is to insert an out-of-scale value in all empty dates. The other technique we evaluate is linear interpolation [8]. For each field, we linearly interpolated all the statistics (mean, mode, minimum, maximum, standard deviation) in each of the empty dates.

B. Data imbalance management

The creation of the five binary classification dataset inherently causes large data imbalance that is of potential harm when training a classifier, as it could learn a bias towards the dominant class. To tackle this problem, we investigate 3 popular ways to obtain a balanced dataset before training the classifier: undersampling, random oversampling, SMOTE (Synthetic Minority Oversampling Technique) [5].

Undersampling consists in balancing the training set by randomly eliminating units from the majority class. The main drawback of this is the information loss, as the classifier may lose the chance to see some different and valuable examples of the adversary class. On the other hand, this approach leads to having a smaller dataset, which makes the training phase less time-consuming and energy-demanding, also reducing the environmental impact in the perspective of GreenAI. On a different note, random oversampling consists of balancing the training set by randomly sampling more units of the minority class.While this approach does not lose any information, it presents other drawbacks.The dataset becomes larger without actually adding new information: this translates in an increased computational cost and training time, without a corresponding increase in the algorithm performance. Furthermore, we run the risk to induce overfitting, reducing the classifier's ability to generalize with respect to undersampling, likely leading to many false positives [6]. Finally, SMOTE is a balancing technique that focus on generating synthetic samples for the minority class, this is achieved by interpolation in feature space between two neighbor samples from the minority class.This approach mitigates the risk of overfitting, but it is often unclear whether the newly generated units are actually realistic, therefore the risk is to create noise or introducing a bias in the training sample. For this reason, we refrain from leveraging SMOTE in the following experimental section.



(a) Tomato (b) Soy (c) Sugar Beet (d) Rice (e) Wheat

Fig. 2: Bar plots showing NDVI index across time for different crops. Vertical green error bars represent one standard deviation above and one below the average, computed among the fields. Vertical dashed blue lines enclose the sowing periods, red ones enclose the harvesting ones (See Table II)

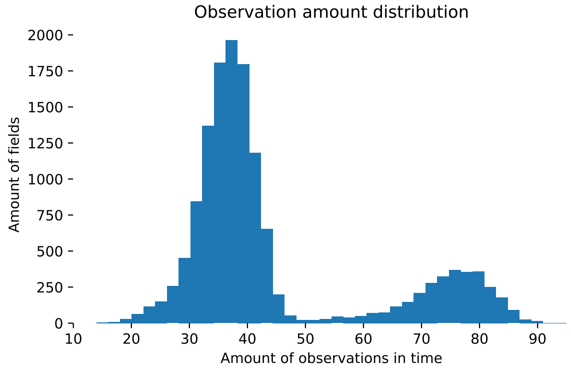


Fig. 3: Distribution of the amount of observations.

Intuitively, we could prospect that undersampling could be the best option for our case, given the dataset to be large enough for the model to see enough variability in the adversary class. Moreover, since false declarations are a rare occurrence (<5%), we could foresee that among the units flagged as suspicious (classified as negative for the class of interest), there would be a predominance of false positive. False positive could trigger and unnecessary verification, while false negatives could mean a false declaration going undetected, hence the trade-off should be carefully evaluated in the deployment scenario.

IV. Experiments

In this section, we present and discuss the results of the methodology detailed in the first part of this manuscript. To restate, the goal of our research is to leverage the observations provided by the ESA Copernicus satellites in order to provide the regulatory agency with feedback on the truthfulness of the stated crop for a given plot of land. A negative outcome of the automatic classification might trigger an on-site inspection, hence it is crucial to be able to provide a reliable classification.

A. Experimental Setup

For the purpose of this work, five crops have been considered, which are reported in Table II along with the most relevant information of the production's life cycle. Our training data include all the available observation for the 2022 season (completed), we train a binary classifier for each of the five crops of interest, leveraging the datapoints labeled with the target class as positive examples (class 1), and the remaining datapoints as negative examples (class 0), those include the remaining 4 classes along with other crop classes. In this evaluation, we use as training data all the data available between the beginning of the sowing phase to the end of the harvesting for the target crop. This analysis is relevant for two aspects: (a) It allows for a controlled setup to experiment with multiple options for dealing with the problems of extremely unbalance between positive and negative examples (Section III-B) and to evaluate the two options for dealing with missing observations (Section III-A). (b) It allows defining a baseline for the classifier's performance in a best-case scenario.

TABLE II: Start and end of sowing and harvesting operations for the crops of interest, along with the minimum and maximum observed days of duration of the crop cycle. The information is obtained from the data for the 2022 growth season. Dates are expressed in (mm/dd) format.

Crop	**Sowing**		**Harvesting**		**Cycle duration (d)**	
	start	end	start	end	min	max.
Tomato	04/01	07/01	07/15	10/01	95	110
Soy	01/03	06/15	09/10	09/30	120	150
Sugar Beet	01/03	03/31	07/15	09/15	60	90
Rice	01/04	04/30	09/15	11/05	160	180
Wheat	10/15	11/30	06/01	07/10	210	230

To assess the effectiveness and reliability of the classification models we use the standard metrics in the literature for binary classification: Precision, Recall and F1-Score, defined below.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Given the extreme unbalance of the dataset uses for testing, it is best to avoid relying on the Accuracy, since a high accuracy may be achieved by simply predicting the majority class.

For each binary classifier, we isolate a uniformly sampled 25% of the available data for evaluation, while the remaining 75% is used for training. The very same train-test subset have been used for all the experiments, to ensure a fair comparison.

Before training each classifier, we further refine the training data by narrowing the observation window to only include observations between the start of the sowing season to the end of the harvesting season for the target crop. This is especially important since the same field might be grown with different crops through the year, which might induce a bias in the classifiers.

B. Results

TABLE III: Comparison of Interpolation with out-of-range placing for filling statistics of missing observation dates.

Crop	***Strategy***	**Precision**	**Recall**	**F1-Score**
Tomato	*Interp.*	0.89	0.59	0.71
	Fill	0.89	0.62	0.73
Soy	*Interp*	0.915	0.86	0.89
	FIll	0.92	0.86	0.89
Sugar Beet	*Interp*	0.99	0.93	0.96
	Fill	0.99	0.92	0.96
Rice	*Interp*	1.0	0.75	0.85
	Fill	1.0	0.375	**0.54**
Wheat	*Interp*	0.93	0.93	0.93
	Fill	0.93	0.93	0.93

a) Management of missing observations: The first fundamental experiment involves comparing different strategies for filling missing observations in the statistics, as introduced in Section III-A. The results obtained with two strategies, linear interpolation and filling missing values with an out-of-scale value (-1000), are presented in Table III. The choice of these strategies appears to have minimal impact on the evaluated metrics, except for the *Rice* crop, which exhibits a significantly low recall when using the constant value strategy. However, it is challenging to precisely attribute this result solely to the interpolation strategy, due to the crop's extreme under-representation across the dataset. For the above reasons, in the remainder of this analysis, we opted to use the Interpolation strategy.

b) Dataset balancing: The second part of the experimental evaluation focuses on determining the optimal strategy for handling the highly unbalanced dataset used to train the binary classifier. We compare the performance of the baseline classifier trained on the original imbalanced training set with the same model trained using two different resampling techniques: undersampling and oversampling. Undersampling involves reducing the number of instances from the majority class to achieve a more balanced representation of the classes, conversely oversampling involves increasing the number of instances in the minority class to address the class imbalance. These and other resampling strategies are further described in Section III-B.

By analyzing the results presented in Table IV, noteworthy observations can be made regarding the two resampling techniques. The application of the undersampling strategy resulted in an increase in the number of false positives, where crops are mistakenly classified as the target crop. Consequently, this led to a notable decrease in the Precision metric, reaching drastic levels for certain cases as highlighted in the results table. This issue can be attributed to the significant reduction in the training data caused by the undersampling procedure. As a result, the negative effect is particularly pronounced for the most underrepresented crops, such as Rice and Tomato. It's worth noticing that False positive are possibly the least desirable outcome in our application scenario, while a false negative leads to further investigations on the effective crop being carried out, a false negative means that bogus declaration are more likely to be undetected.

TABLE IV: Effects of training data balancing on the unbalanced test data.

Crop	***Train Data***	**Precision**	**Recall**	**F1-Score**
Tomato	Unbalanced	**0.89**	**0.59**	0.71
	Undersampling	**0.39**	**0.92**	0.54
	Oversampling	0.83	0.72	0.77
Soy	Unbalanced	0.915	0.86	0.89
	Undersampling	0.832	0.93	0.88
	Oversampling	0.9	0.91	0.91
Sugar Beet	Unbalanced	0.9	0.93	0.96
	Undersampling	0.95	0.96	0.95
	Oversampling	0.99	0.99	0.99
Rice	Unbalanced	1.0	0.75	0.85
	Undersampling	**0.12**	0.87	0.22
	Oversampling	**1.0**	0.75	0.86
Wheat	Unbalanced	0.93	0.93	0.93
	Undersampling	0.88	0.97	0.92
	Oversampling	0.92	0.95	0.93

On the other end the oversampling strategy shown promising results, with a noticeable reduction in false negatives with respect to the baseline with only a manageable increase in false positive, the overall superiority of this approach is hence validated by an increase in F1 score for all the crops. To conclude, while we can't recommend relying on an undersampling strategy, we are confident in suggesting the oversampling approach in order to reduce the number of occurrences of an investigation being triggered by mistake.

c) Choice of the classifier: The previous results were all obtained with a Random Forest classifier, to demonstrate the validity of our choice, therefore hereafter we evaluate the alternative usage of the very powerful and popular binary classifier SVM [4] This classifier builds a separation hyperplane by choosing support vectors, those are the defined as the harder to classify points. For a simple yet meaningful comparison, we compare the results obtained without train dataset balancing, using the interpolation strategy for the management of the missing observation.

In Table V we report the results. It's clear that RF slightly outperforms SVM for all the crops, of particular interest it is the Rice crop, that never gets correctly classified, hence scoring zero in all the metrics. The reason lies in the very small number of fields labeled as Rice (Table I), with SVM clearly requiring a larger training set. Another advantage of RF is its superior interpretability compared to SVM. With each decision

TABLE V: Comparison of the results obtained with Random Forest and SVM.

Crop	*Classifier*	**Precision**	**Recall**	**F1-Score**
Tomato	**RF**	0.89	0.59	0.71
	SVM	0.876	0.55	0.68
Soy	**RF**	0.915	0.86	0.89
	SVM	0.87	0.87	0.87
Sugar Beet	**RF**	0.99	0.93	0.96
	SVM	0.97	0.93	0.95
Rice	**RF**	1.0	0.75	0.85
	SVM	0.0	0.0	0.0
Wheat	**RF**	0.93	0.93	0.93
	SVM	0.9	0.94	0.92

tree acting as a transparent flowchart, it becomes easier to comprehend how the model classifies data points. The consensus-based decision-making further enhances stability and improves generalization abilities. In contrast, SVM's optimal hyperplane may lack clear interpretability.

V. Conclusion and future works

In conclusion, this work provides a significant contribution to agricultural research by demonstrating the applicability of machine learning techniques and the utility of satellite data for crop classification. The proposed methodology can be applied to verify the authenticity of farmers' claims, especially regarding state subsidies for specific crops of interest.

Our work lays the foundation for further research in the field of agricultural field classification using satellite imagery and ML techniques. Several avenues for future work can be explored to enhance and extend the findings presented in this paper.

1) Multi-class classification could provide a more comprehensive understanding of crop distribution and facilitate more accurate crop monitoring and yield estimation.
2) Incorporating additional data sources, such as weather data, soil composition, or historical crop records, could improve the accuracy and robustness of the classification models.
3) Investigating the development of dynamic classification models that can adapt to changing environmental conditions and crop phenology could enable real-time monitoring and detection of crop changes, disease outbreaks, or other significant events that affect agricultural fields.

By pursuing these future research directions, we can advance the field of agricultural field classification and contribute to the development of more accurate, efficient, and sustainable agricultural practices.

Acknowledgment

This work has been partially supported by the INdAM research group GNCS.
The publication was created with the co-financing of the European Union-FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062/2021.
This work has been partially supported by the industrial collaboration between ABACO Group SpA and UNIMORE.

References

[1] Belgiu, M., Drăguţ, L.: Random forest in remote sensing: A review of applications and future directions. ISPRS journal of photogrammetry and remote sensing **114**, 24–31 (2016)

[2] Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)

[3] Campos-Taberner, M., García-Haro, F.J., Martínez, B., Sánchez-Ruíz, S., Gilabert, M.A.: A copernicus sentinel-1 and sentinel-2 classification framework for the 2020+ european common agricultural policy: A case study in valència (spain). Agronomy **9**(9) (2019). https://doi.org/10.3390/agronomy9090556, https://www.mdpi.com/2073-4395/9/9/556

[4] Chapelle, O., Haffner, P., Vapnik, V.N.: Support vector machines for histogram-based image classification. IEEE transactions on Neural Networks **10**(5), 1055–1064 (1999)

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)

[6] Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on learning from imbalanced datasets II. vol. 11, pp. 1–8 (2003)

[7] Immitzer, M., Vuolo, F., Atzberger, C.: First experience with sentinel-2 data for crop and tree species classifications in central europe. Remote Sensing **8**(3) (2016). https://doi.org/10.3390/rs8030166

[8] Lepot, M., Aubin, J.B., Clemens, F.H.: Interpolation in time series: An introductive overview of existing methods, their performance criteria and uncertainty assessment. Water **9**(10), 796 (2017)

[9] Nasrallah, A., Baghdadi, N., Mhawej, M., Faour, G., Darwish, T., Belhouchette, H., Darwich, S.: A novel approach for mapping wheat areas using high resolution sentinel-2 images. Sensors **18**(7) (2018). https://doi.org/10.3390/s18072089

[10] Piedelobo, L., Hernández-López, D., Ballesteros, R., Chakhar, A., Del Pozo, S., González-Aguilera, D., Moreno, M.A.: Scalable pixel-based crop classification combining sentinel-2 and landsat-8 data time series: Case study of the duero river basin. Agricultural Systems **171**, 36–50 (2019). https://doi.org/https://doi.org/10.1016/j.agsy.2019.01.005

[11] Wardlow, B., Egbert, S., Kastens, J.: Analysis of time-series modis 250 m vegetation index data for crop classification in the u.s. central great plains. Remote Sensing of Environment **108**, 290–310 (06 2007). https://doi.org/10.1016/j.rse.2006.11.021

[12] Yi, Z., Jia, L., Chen, Q.: Crop classification using multi-temporal sentinel-2 data in the shiyang river basin of china. Remote Sensing **12**(24) (2020), https://www.mdpi.com/2072-4292/12/24/4052

[13] You, N., Dong, J.: Examining earliest identifiable timing of crops using all available sentinel 1/2 imagery and google earth engine. ISPRS Journal of Photogrammetry and Remote Sensing **161**, 109–123 (2020). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.01.001

DOI: 10.15439/2023F363
ISSN 2300-5963 ACSIS, Vol. 35

Experiments on Software Error Prediction Using Decision Tree and Random Forest Algorithms

Ilona Bluemke
0000-0002-2894-5976
Warsaw University of Technology,
Institute of Computer Science,
Nowowiejska 15/19 00-665
Warsaw, Poland
Email: Ilona.Bluemke@pw.edu.pl

Paweł Borsukiewicz
00000-0002-2934-6115
Warsaw University of Technology,
Institute of Computer Science,
Nowowiejska 15/19 00-665
Warsaw, Poland
Email:
pborsukiewicz99@gmail.com

***Abstract*—Machine learning algorithms are widely used in the assessment of error-proneness in software. We conducted several experiments with error prediction on public PROMISE repository. We used Decision Tree and Random Forest algorithms. We also examined techniques aiming at the improvement of performance and accuracy of the model – such as oversampling, hyperparameter optimization or threshold adjustment. The outcome of our experiments suggests that Random Forest algorithm, with 100 – 1000 trees, can be used to obtain high values of evaluation parameters such as accuracy and balanced accuracy. However, it has to be implemented with a set of techniques countering imbalance of the datasets used to assure high values of precision and recall that correspond with correct detection of erroneous software. Additionally, it was shown that the oversampling and hyperparameter optimization could be reliably applied to the algorithm, while threshold adjustment technique was not found to be consistent.**

***Index Terms*—error prediction, error proneness, decision tree, random forest, PROMISE repository, machine learning.**

I. Introduction

HIGH quality of software is essential in software production. Software testing is a key part of the software development process, especially for quality assurance, but it requires a lot of time and resources. It is estimated that testing activities consume more than half of the cost of the whole software development process [1], [2]. Application of software error prediction is almost 30% cheaper than testing, according to recent studies [3]. Ability to predict faulty components in the early phase may highly increase cost-effectiveness. There is abundant literature on software defect prediction, some works are mentioned in section II. There are also several systematic literature reviews on this subject eq. [4], [5], [6].

Contemporary solutions put strong emphasis on the usage of machine learning algorithms in software defect prediction. However there are many papers on software error prediction we decided to conduct some experiments on the usage of machine learning algorithms – Decision Tree and Random Forest on public PROMISE repository. We wanted to examine the relationship between computational effort and accuracy of obtained results. Techniques aiming at the improvement of performance and accuracy of the model – such as oversampling, hyperparameter optimization or threshold adjustment were also analyzed and addressed.

Paper is structured as follows. Section II introduces related work. Methodology is stated in section III. The experiment and its analysis are presented in section IV. Section V concludes the paper, highlighting some issues.

II. Related Work

Lately various studies have been conducted in order to compare algorithms used to determine error-proneness. Many of them were based on wide variety of machine learning algorithms [7] and used formerly mentioned PROMISE [8] repository as its dataset.

Random Forest, Naive Bayes, J48, Immunos 1 & 2, CLONALG, AIRS (Artificial Immune Recognition System algorithm) 1 & 2 and AIRS 2 Parallel were algorithms studied by Catal et al. [9],[10]. When AIRS algorithm was taken into consideration the researchers concluded that the best results were obtained when CK metrics and LOC metric were combined.

AIRS is a system inspired by human's immunological system with B-cells and T-cells as our guardians. In the past the main application was supervised learning, however, in 2001 it was demonstrated that algorithms based on this system can be also used for classification domain [9]. Other slightly different algorithms using AIS include CLONALG [11] and Immunos [12].

As previously mentioned, Catal et al. [9] have also directed their research towards Random Forests (RF). This algorithm is based on existence of high number of so called "trees". Each tree is independent, but at the end the majority voting result of all the trees in the forest is taken as a final result. Model can be trained with various performance enhancing techniques. One of them is bootstrap sampling or bagging (bootstrap aggregating), which means randomly taking only small part of the dataset for each training process and repeating it multiple times. In case of singular trees one may also consider pruning – technique based on removal of some of the nodes that are not essential and could result in overfitting. Nonetheless, as it was stated in study by L. Breiman [13] that procedure is not needed in case of RF algorithm when bagging is applied. According to the study by Mundada et al. [14] RF is the best algorithm for NASA datasets, which are a part of the PROMISE repository.

J48 is an algorithm that implements C4.5 decision tree learning [9], [15].

Mundada et al. [14] directed their study towards Artificial Neural Network (ANN) and Resilient Back Propagation (RBP) using JM1 dataset. As a result of this experiment, researchers concluded that the better accuracy of ANN algorithm was reached, when compared with already existing analytical models.

Bishnu et al. [16] studied performance of QUAD Tree-Based K-Means Clustering Algorithm using AR3, AR4 and AR5 datasets. It was concluded that error rates of this algorithm are comparable to the ones obtained with other algorithms. In order to obtain the best values, data sets partitioning has to ensure that the sum of distances within the clusters is properly reduced [17].

Okutan and Taner [18] used 9 datasets from PROMISE dataset to research Bayesian Networks. The results of the study stated that the LOC, RFC and LOCQ metrics are the best choice due to their effectiveness when this algorithm is considered. An important advantage of this network is the fact that it can be used even when the metrics are incomplete for some sets.

Kumudha et al. [19] have introduced a significant development in the field. Their research focused on conventional Radial Basis Function Neural Network (RBFNN) and the novel Adaptive Dimensional Biogeography Based Optimization Model (ADBBO). Having based the research in CM1, JM1, KC1, KC2, and PC1 datasets, results obtained during this study showed that newly proposed method is more effective when compared with already existing algorithms.

Gupta and Gupta. [20] have used derived metrics from PROMISE repository datasets to determine fault classification. In this study, the emphasis was put on the data distribution and skewness rather than the algorithms itself.

Erturk and Akcapinar [21] have used projects from PROMISE repository to conduct research on Fuzzy Inference Systems (FIS) [22] and Adaptive Neuro Fuzzy Inference System (ANFIS). Those new methods deploy iterative software error-proneness prediction to automatically detect fault prone sections.

Alighardashi et al. [23] have used ten PROMISE and NASA datasets to test feature selection method. Five filter methods were used during this study. Weighted filter (WF) method was determined to be able to detect best features that would allow the fault prediction accuracy to be the increased in the fastest way possible.

After preparation of the above related work recent publications in this domain appeared e.g. [24], [25], [26], [27]. These works are not included in the above text.

III. Methodology

For the purpose of the experiment Decision Tree and Random Forest algorithms [9] were selected. Random Forest, being composed of Decision Trees, is a flexible algorithm that can be applied both in classification and regression problems. As the purpose of the experiment is to assess error-proneness of the samples within the datasets, one can consider the problem primarily as classification problem. One can also assume that there are two classes of results - code either is correct or incorrect. However, as it is possible to apply regression version of Random Forest algorithm in this particular scenario and to some extend treat its values as a probability of existence of an error, it was used and compared against its classification counterpart.

Python [28] was used for implementation and functions from NumPy [29], Pandas [30], scikit-learn [31] and imbalanced-learn [32] libraries were a basis for the implementation of the algorithm and the evaluation of performance such as accuracy, recall, precision, etc.

Hyperparameter optimization was not initially performed as some hyperparameters, such as forest size, were the focus of the study and in order to better understand what are the disadvantages of the basic model. Optimization of multiple hyperparameters would result in largely extended training times, especially if larger forests were to be considered. In further parts optimization techniques were used in order to increase the performance and assess the full potential of Random Forest in error-prediction field.

Similarly, SMOTE [33] oversampling was another technique that was not used initially, but was introduced later in order to improve the performance of the model. By default bootstrapping was enabled throughout the whole experiment and pruning was not performed as recommended by Breiman [13]. The order of processes is presented in Fig1.

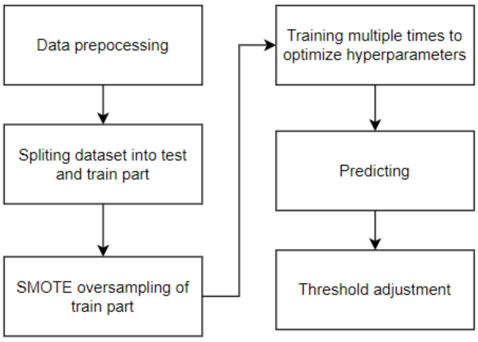

Data prepocessing
Spliting dataset into test and train part
SMOTE oversampling of train part
Training multiple times to optimize hyperparameters
Predicting
Threshold adjustment

Fig 1. Phases of experiment

IV. Experiments

Public NASA datasets were used, including those available within the PROMISE repository. The scope of the tests includes CM1, KC1, KC2, PC1, PC2 and PC3 sets.

Experiments were conducted on two devices: personal laptop and virtual machine provided by the Warsaw University of Technology.

A. Experiment results

Experiment was divided into a series of incremental steps. Each step introduced new technique or method, or combined those previously assessed. The final outcome was the process presented in Fig 1.

Initially the datasets were analyzed and some results of analysis are presented in Table I. It can be seen that datasets are highly imbalanced.

Table I.
Dataset properties

Dataset	Dataset size	Error-free software in dataset [%]	Number of metrics
CM1	498	90.2	22
KC2	522	79.5	22
PC1	1109	93.1	22
PC3	1563	89.8	38
KC1	2109	84.6	22
PC2	5589	99.6	37

Before introduction of any enhancement mechanism, it was assumed that the optimal number of trees for the experiment should be in the range from 100 to 1000. This observation was confirmed throughout the experiment. Above 1000 trees any substantial improvement to the evaluation metrics was not observed as shown in Table II. It is worth noting that the training time grows almost linearly with the forest size, therefore lower forest sizes are generally preferred when training time is limited. Even though, all experiments were performed for forest sizes of 1 (decision tree), 10, 100, 1000, 10000, 25000 and 50000, results provided in this paper were obtained for forests with 1000 trees unless stated otherwise.

Table II.
Basic classification accuracy

Number of trees	1	10	100	1000	10000
Dataset	Accuracy				
CM1	0.81	0.87	0.86	0.86	0.86
KC1	0.81	0.84	0.85	0.85	0.85
KC2	0.74	0.82	0.82	0.80	0.82
PC1	0.90	0.91	0.92	0.92	0.92
PC2	0.99	0.99	0.99	0.99	0.99
PC3	0.85	0.90	0.90	0.90	0.90

Even though, accuracy score is rather high, it is usually very close to the percentage of error-free samples in the dataset. Analyzing metrics such as recall, precision, and balanced accuracy it was clearly visible that model tends to classify vast majority of samples as error-free, thus making from few to even no useful detections (true positive values) as in case of dataset CM1, as shown in Table III.

Table III.
Basic classification for CM1 dataset

Accuracy	0.86
Recall	0.00
Precision	0.00
Balanced accuracy	0.49
F1 score	0.00

First technique aiming to improve error-prone software detection that was assessed was threshold adjustment. As this problem deals with two classes – erroneous software and error-free software it is primarily a classification problem. However, one may take a regression approach with the 0.5 threshold as a default one. Adjustment of that selection threshold, either its lowering or increasing, could potentially lead to better classification. Exemplary outcome of the experiment for CM1 dataset is presented in Fig 2.

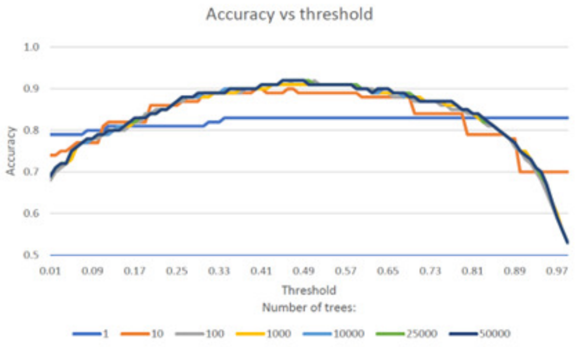


Fig 2. Threshold adjustment for CM1 dataset

Slight change of the threshold around the 0.5 mark in some cases, provided some minor improvements. Nonetheless, no direct pattern could be established observing various datasets and forest sizes. Moreover, it did not help in any way to tackle the problem of datasets imbalance, still being strongly biased towards error-free classes. Given method was also combined with subsequently described methods, however, at each step it was too unpredictable and, as a result, it was discarded.

Further studies were done on SMOTE oversampling technique. As presented in Table IV, it aimed to reduce the learning bias resulting from dataset imbalance by equalizing proportions via creating artificial samples.

Table IV.
Dataset sizes before and after oversampling

	Before SMOTE		After SMOTE	
	Faulty	Not faulty	Faulty	Not faulty
CM1	45	428	428	428
KC1	307	1696	1696	1696
KC2	96	399	399	399
PC1	73	980	980	980
PC2	21	5288	5288	5288
PC3	151	1333	1333	1333

As presented in Table V and Table VI, significant improvements could be noticed. Not only, was the accuracy improved, but more importantly precision and recall values also, which indicate that the true error-free software was now properly detected and classified.

TABLE V.
DATASET SIZES BEFORE AND AFTER OVERSAMPLING

	Accuracy	
	Before	After
CM1	0.86	0.94
KC1	0.85	0.91
KC2	0.80	0.85
PC1	0.92	0.96
PC2	0.99	1.00
PC3	0.90	0.93

TABLE VI.
OVERSAMPLED CLASSIFICATION FOR CM1 DATASET

Accuracy	0.94
Recall	0.98
Precision	0.88
Balanced accuracy	0.91
F1 score	0.93

In the final part of the experiment hyperparameter optimization for the previously analyzed oversampling technique was added. Similarly to the forest size selection, this step becomes more and more computationally intensive with the increase in number of the combinations that have to be considered. Performing grid search and cross-validation proved to be successful in improving results, as can be seen in Table VII. Comparison of balanced accuracies obtained throughout the experiment was presented in Fig 3.

TABLE VII.
RESULTS AFTER HYPERPARAMETER OPTIMIZATION

	Dataset					
	CM1	KC1	KC2	PC1	PC2	PC3
Accuracy	0.95	0.91	0.87	0.98	1.00	0.94
Recall	0.98	0.91	0.89	0.98	1.00	0.95
Precision	0.91	0.91	0.86	0.97	0.99	0.93
F1 score	0.95	0.91	0.87	0.98	1.00	0.94
Balanced accuracy	0.95	0.91	0.86	0.98	1.00	0.94

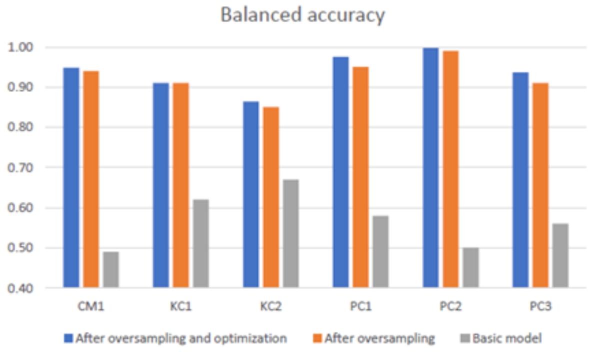


Fig 3. Balanced accuracy comparison

B. *Comparison with other studies*

Comparing obtained results with other studies within the domain, one can reference AUC obtained in a study by Catal et al. [10]. As presented in Fig. 4, results for all of the datasets that were covered by both experiments have significantly improved.

Nonetheless, when comparing the results with studies based on neural networks, such as the ones obtained via implementation of ADBBO by P. Kumudha et al. [19], presented in Fig. 5, it can be observed that the Random Forest provided better results for PC1 and CM1 datasets, while it was outperformed by Neural Network in case of KC1 and KC2 datasets.

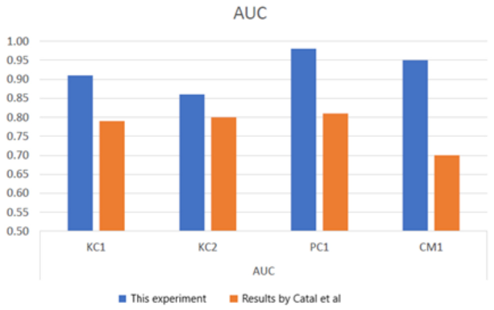


Fig 4. AUC comparison with prior study

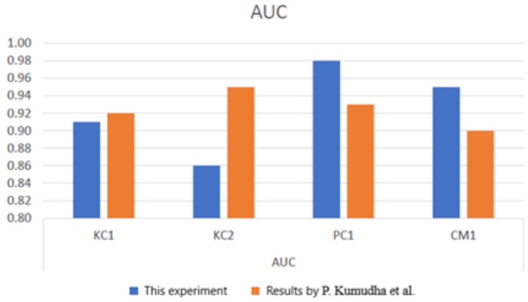


Fig 5. AUC comparison with prior study

A recent study by T.F. Husin et al. [26] was analyzing Least Square Support Vector Machine (LSSVM) combined with the use of SMOTE technique. Even though, it was concluded that SMOTE significantly improved obtained results, as presented in Fig. 6, those results were not close to the results obtain in this or any of two previously mentioned studies.

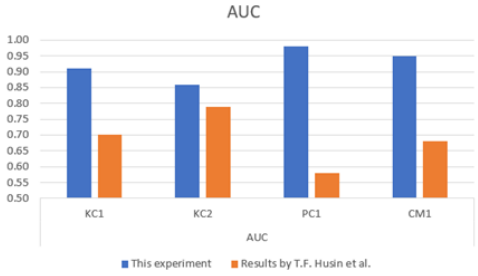


Fig 6. AUC comparison with new study

V. CONCLUSIONS

The aim of our study was to assess the viability of application of Decision Tree and Random Forest algorithms within the scope of error-proneness detection field. The series of experiments was conducted for six different datasets and total algorithm training time was approximately 150 hours with the majority of this time spent on the final version. Therefore, due to vastness of collected data detailed results presented in section IV focused only one of them – CM1. Study was performed on the data acquired from PROMISE repositories started with the analysis of the most basic models, which turned out to be insufficient due to the bias towards error-free classification resulting from dataset imbalance. Subsequently, a set of techniques was deployed in order to improve its performance. They included hyperparameter optimization, basic feature selection, threshold adjustment and SMOTE oversampling technique.

As a result it was possible to observe that implementation of mechanisms aiming at improvement of performance of algorithms resulted in models being able to quite accurately classify samples present within PROMISE repository. High values of precision and recall, in most cases above 90%, may assure one that software errors can be well detected using Random Forest algorithm. It was also shown that usually random forests of sizes between 100 and 1000 are the most appropriate as above that values accuracy does not seem to improve, while computation time does. Nonetheless, it is also worth mentioning that single decision trees also provided useful results, however, they cannot quite compete with the anti-overfitting properties of the forest. Further, if predictions trained on PROMISE datasets are to be reasonable, one shall counter negative effects of imbalanced dataset – oversampling was proved to be a viable solution that significantly increased values of evaluation parameters such as balanced accuracy. Additionally, if training time is not limited, hyperparameter optimization may further improve obtained results. Finally, there has not been found any reason to use regression instead of classification it this particular classification problem. Throughout the study, it was found that threshold adjustment technique could result in slight improvements, however, it could not be reliably used.

In order to further improve results obtained by Random Forest, one may consider application of more advanced feature selection methods. Similarly, it would be reasonable to use MOOD and QMOOD object metrics. Further, one could consider creation of their own datasets, based on publicly available repositories. Performing a training on data gathered from projects in the same language, technology or domain as the target test set could also make prediction algorithm more sensitive to crucial aspects of assessing error-proneness for a given case.

References

[1] F. Elberzhager, A. R. Rosbach, Eschbach, J. Münch, "Reducing Test Effort: A Systematic Mapping Study on Existing Approaches", Information and Software Technology, vol. 54, no. 10, 1092-1106, 2012.

[2] K. Bareja, A. Singhal, "A Review of Estimation Techniques to Reduce Testing Efforts in Software Development", http://dx.doi.org/ 10.1109/ACCT.2015.110, 2015.

[3] J. Hryszko, L. Madeyski, "Cost Effectiveness of Software Defect Prediction in an Industrial Project", http://dx.doi.org/ 10.1515/fcds-2018-0002, 2018.

[4] Y.Z. Bala, P.A. Samat, K.Y. Sharif, N. Manshor, "Current Software Defect Prediction: A Systematic Review", http://dx.doi.org/ 10.1109/AiIC54368.2022.99114586, 2022

[5] F. Matloob et al., "Software Defect Prediction Using Ensemble Learning: A Systematic Literature Review", http://dx.doi.org/ 0.1109/ACCESS.2021.3095559, 2021.

[6] Y. Zhao, K. Damevski, H,Chen, "A Systematic Survey of Just-in-Time Software Defect Prediction", http://dx.doi.org/ 10.1145/ 3567550, 2023.

[7] T. Menzies , J. DiStefano, A. Orrego , R. Chapman, " Assessing predictors of software defects", in *Proc Predictive software models workshop*, pp. 1-5, 2004.

[8] G. Boetticher, T. Menzies, T. Ostrand, PROMISE Repository of Empirical Software Engineering Data, West Virginia University, Department of Computer Science 2007.

[9] C. Catal, B. Diri, B. Ozumut, "An artificial immune system approach for fault prediction in object oriented software", pp. 238-245, http://dx.doi.org/ 10.1109/DEPCOS-RELCOMEX, 2007.

[10] C. Catal, B. Diri, "Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem", http://dx.doi.org/ 10.1016/j.ins.2008.12.001, 2009.

[11] J. Brownlee, "Clonal selection theory & CLONALG. The clonal selection classification algorithm", in *Technical Report 2-02*, Swinburne University of Technology, 2005.

[12] J. H. Carter, "The immune system as a model for pattern recognition and classification", http://dx.doi.org/10.1136/jamia.2000.0070028, 2001.

[13] L. Breiman, "Bagging predictors.", *Mach Learn* 24, pp.123–140, https://doi.org/10.1007/BF00058655Y, 1996.

[14] D. Mundada, A. Murade, O. Vaidya, and J. N. Swathi, "Software Fault Prediction Using Artificial Neural Network And Resilient Back Propagation", *Int. J. Comput. Sci. Eng*., vol. 5, no. 03, pp. 173–179, 2016.

[15] Z. Xiang, L. Zhang, "Research on an Optimized C4.5 Algorithm Based on Rough Set Theory", http://dx.doi.org/ 10.1109/ ICMeCG.2012.74, 2012.

[16] P. Bishnu and V. Bhattacherjee, "Software Fault Prediction Using Quad Tree-Based K-Means Clustering Algorithm", pp. 1146–1150, http://dx.doi.org/ 10.1109/TKDE.2011.163, 2012.

[17] P. Bishnu and V. Bhattacherjee, "Outlier Detection Technique Using Quad Tree" in *Proc Int'l Conf. Computer Comm. Control and Information Technology*, pp. 143-148, 2009.

[18] A. Okutan and O. Taner, "Software defect prediction using Bayesian networks", http://dx.doi.org/ 10.1007/s10664-012-9218-8, 2014.

[19] P. Kumudha, R. Venkatesan, "Cost-Sensitive Radial Basis Function Neural Network Classifier for Software Defect Prediction", http://dx.doi.org/ 10.1155/2016/2401496, 2016.

[20] S. Gupta, D. Gupta, "Fault Prediction using Metric Threshold Value of Object Oriented Systems", *International Journal of Engineering Science and Computing*, vol. 7, no. 6, pp. 13629–13643, 2017

[21] E. Erturk, E. Akcapinar, "Iterative software fault prediction with a hybrid approach", http://dx.doi.org/ 10.1016/j.asoc.2016.08.025, 2016.

[22] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system", http://dx.doi.org/ 10.1109/21.256541, 1993.

[23] F. Alighardashi, M. Ali, Z. Chahooki, "The Effectiveness of the Fused Weighted Filter Feature Selection Method to Improve Software Fault Prediction", pp. 5, http://dx.doi.org/10.22385/jctecs.v8i0.96, 2016.

[24] C. Lakshmi Prabha, Dr.N. shivakumar "Software Defect Prediction Using Machine Learning Techniques" , *Proc. of the Fourth International Conference on Trends in Electronics and Informatics*, IEEE Xplore Part Number: CFP20J32-ART; ISBN: 978-1-7281-5518-0, 2020.

[25] Y. Shen, S. Hu, S, Cai, M. Chen, "Software Defect Prediction based on Bayesian Optimization Random Forest", http://dx.doi.org/ 10.1109/DSA56465.2022.00149, 2022.

[26] T.F. Husin, M.R. Pribadi, Yohannes, "Implementation of LSSVM in Classification of Software Defect Prediction Data with Feature Selection", *9th Int. Conf. on Electrical Engineering, Computer Science and Informatics (EECSI2022)*, pp.126-131, 2022.

[27] MD.A. Jahangir, MD. A.Tajwar, W. Marma, "Intelligent Software Bug Prediction: An Empirical Approach", http://dx.doi.org , 101109/ ICREST57604.2023.10070026, 2023.

[28] Python Core Team, "Python: A dynamic, open source programming language", Python Software Foundation, accessed 28.04.2022, https://www.python.org/

[29] C.R. Harris, K.J. Millman, S.J. van der Walt et al. "Array programming with NumPy", Nature 585, pp. 357–362, http://dx.doi.org/ 10.1038/s41586-020-2649-2, 2020.

[30] W. McKinney, "Data structures for statistical computing in python", *Proc. of the 9th Python in Science Conference*, vol 445, pp. 56-61, http://dx.doi.org/ 10.25080/Majora-92bf1922-00a, 2010.

[31] Pedregosa et al., "Scikit-learn: Machine Learning in Python", *Journal of Machine Learning Research 12*, pp. 2825-2830, 2011.

[32] G. Lematre, F. Nogueira, C. K. Áridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning", *Journal of Machine Learning Research* 17, pp. 1-5, http://dx.doi.org/ 10.48550/arXiv.1609.06570, 2017.

[33] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique", *Journal of artificial intelligence research*, pp. 321-357, 2002.

DOI: 10.15439/2023F722
ISSN 2300-5963 ACSIS, Vol. 35

Diagnosing Machine Learning Problems in Federated Learning Systems: A Case Study

Karolina Bogacka, Anastasiya Danilenka
0000-0002-7109-891X, 0000-0002-3080-0303
Warsaw University of Technology
Plac Politechniki 1, 00-661 Warszawa, Poland
Email: {karolina.bogacka, anastasiya.danilenka}.dokt@pw.edu.pl

Katarzyna Wasielewska-Michniewska
0000-0002-3763-2373
Systems Research Institute, Polish Academy of Sciences
Newelska 6, 01-447 Warszawa, Poland
Email: katarzyna.wasielewska@ibspan.waw.pl

***Abstract*—The proliferation of digital artifacts with various computing capabilities, along with the emergence of edge computing, offers new possibilities for the development of Machine Learning solutions. These new possibilities have led to the popularity of Federated Learning (FL). While there are many existing works focusing on various aspects of the FL process, the issue of the effective problem diagnosis in FL systems remains largely unexplored. In this work, we have set out to artificially simulate the training process of four selected approaches to FL topology and compare their resulting performance. After noticing concerning disturbances throughout their training process, we have successfully identified their source as the problem of exploding gradients. We have then made modifications to the model structure and analyzed the new results. Finally, we have proposed continuous monitoring of the FL training process through the local computation of a selected metric.**

I. Introduction

FEDERATED Learning (FL, [1], [2]) is a relatively novel approach to Distributed Machine Learning (DML). It allows a system to take full advantage of the data locality and computing power of distributed devices. In a standard scenario, the goal is to train a global model using local models trained by the clients on their local data. Clients periodically send parameter updates to an aggregator node. The new version of the global model is established, communicated back to clients and the process repeats until stopping criteria are met.

Currently, a common way to prepare for an FL training process begins with the centralized construction and training of an initial ML model. This preliminary phase allows the developer to utilize a plethora of already established techniques in order to develop the best ML solution possible. The data preprocessing steps, architecture and hyperparameters from that solution are then used as a basis for the local models trained by the FL clients. However, this approach also relies on the existence of a global dataset with a distribution and format that sufficiently resembles that of the client data. This global dataset may sometimes be impossible to create due to the client data being very localized, client-specific and inaccessible because of privacy concerns. Of course, such a model can also be developed as an FL model from scratch by conducting multiple FL training runs and selecting the best performing training parameters and architectures. Unfortunately, the diagnosis of problems such as vanishing or exploding gradients based on the learning curve would necessarily be hindered by the existence of other destructive factors, like client dropout and differing local data distributions. The development of the final model would therefore necessitate a large number of completed FL training processes and as such be both very resource and time consuming.

Additionally, many current research trends in FL result in solutions that may undergo a very different training process from the centralized baseline. For example, a common goal of trying to achieve scalability while preserving the stability of the training is often mitigated through the appropriate choice of topology. Here, topology refers to the network topology of the FL system, which indicates how clients communicate with the server and with each other [3]. The additional communication on the global and local level may cause the training curve to undergo periodic spikes and drops in accuracy, which can then be hard to distinguish from other ML problems.

We have encountered the problems mentioned above throughout our work on the Assist-IoT project [1]. We were conducting tests in order to select the FL topology best suited to use in the Assist-IoT project in the pilots focusing on: (1) construction workers' health and safety assurance, (2) vehicle exterior condition inspection [4]. The purpose of the FL solution was effective fall detection of construction workers in the case of (1) and automatic vehicle damage detection in the case of (2). In an effort to determine the best topology for the aforementioned use cases, we have analyzed different approaches to the problem [5] and selected 4 most "promising" and representative to further test their behaviour. Our experiments have revealed concerning instabilities in the training processes of some of them. Through additional trials and further examination of the existing result, we have identified the source of the problem as exploding gradients. The problem of exploding gradients here describes a situation in which the gradient backpropagated through a neural network grows exponentially during training, causing the neural network performance to stall or even deteriorate [6].

The work of Karolina Bogacka and Anastasiya Danilenka was funded in part by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

[1] https://assist-iot.eu

We have then modified the hyperparameters to avoid this problem and achieve better results. We would like to share our process as a case study, finishing it with a proposal of an additional procedure that could enable easier identification of the exploding gradient problem in FL systems.

II. Related works

A. The state of FL diagnostic tools

There are few existing works concerning the design and development of FL diagnostic tools. Some of them focus on system monitoring through the identification of badly-performing clients while maintaining security and privacy [7] [8]. This approach to diagnostics is appropriate for FL systems, however, it is often unable to sufficiently recognize any problems stemming from the model architecture or training configuration. The most comprehensive attempt at providing systematic problem diagnosis for FL, FedDebug, offers the possibility of setting breakpoints and replaying previous rounds through a continuous collection of metrics, such as response time, training and validation loss as well as other performance indicators. Those metrics are then used to construct a simulation of the training, which enables easier identification of a faulty client. Despite the narrow focus of this solution, the broad functionalities of the system should be easily extended to other problems [9].

One of the rare works to not focus on finding underperforming clients but on mitigating issues concerning FL models is Fed-DNN-Debugger [10]. It consists of two modules: one is responsible for nonintrusive metadata capture (NIMC), which produces data that is then used for automated neural network model debugging (ANNMD). Local models are then repaired through retraining on specially selected samples. However, its main focus lies in training bugs, which are caused by misconducted training processes, such as biased data, noisy data or insufficient training. It does not enable easy identification of structure bugs, which stem from inappropriate model architecture or hyperparameters.

B. The exploding gradient problem

The exploding gradient problem describes a phenomenon in which the gradient backpropagated through a neural network grows exponentially from layer to layer [6]. Unfortunately, the maximal depth of many popular ML architectures is limited by the existence of this phenomenon. There are existing techniques such as weight scaling or batch normalization which can be used to mitigate these problems. However, they are not always effective [6]. It is possible to use architectures that avoid the exploding gradient problem [11] such as fully connected ReLU networks. Nevertheless, due to the limited functionality of those architectures, it is not a commonly employed practice.

C. Advances in research on Topology of Federated Learning

Although a typical FL system follows a simple centralized topology, with a single server node, often located in the cloud, communicating directly with a federation of clients, this is not necessarily the most optimal, or efficient, solution for many use cases [3]. Interest in network topology in the context of FL stems from the evidence that its impact can be extremely effective in mitigating data heterogeneity. Some types of topologies can also either fully eliminate the need for a central cloud server or greatly reduce its importance [12]. This is significant since the main roadblock for the full production deployment of many FL systems involves communication inefficiency. Other works try to balance these two approaches, by combining nodes in various ways, for example by organizing the clients into groups [13].

A broad classification of current trends in FL topology-related research can be found in [3], which classifies FL topology types into centralized (referred to also as star) [2], tree [14], hybrid [15], gossip [16], grid [17], mesh [3], clique [12] and ring [18]. Classic Federated Averaging [2], can be counted as an example of the centralized topology, involving only a single server independently communicating with each FL-participating client.

The TornadoAggregate algorithm, described in [15], combines star and ring topologies to form STAR-rings and RING-stars. One involves a central server performing periodic federated averaging combined with ring-based groups, while the other consists of a ring with star-based groups. Surprisingly, the first approach is significantly more successful, outperforming the RING-stars with regard to performance, while maintaining the same scalability as described in the aforementioned paper. This process does not require the setup of additional devices and so seems suitable for later reuse.

Many of the above-mentioned topologies use client grouping to manage the problems with heterogenous data. Moreover, this kind of mitigation method can also be used in combination with a centralized topology, in the form of centralized training with dynamic clustering implemented as IFCA in [19]. IFCA involves the simultaneous training of a given number of clusters, allowing for the dynamic creation of client clusters and models personalized for that cluster. However, some of the reported results were subpar due to the necessity of beginning the training with a warm start and accurate knowledge on the number of clusters present in the dataset [20]. In order to limit the occurrence of these issues, an improved version was developed. The new algorithm, SR-FCA, periodically reclusters the clients in a manner that leads it to be both more robust and less resource-intensive for edge clients.

In summary, there are many approaches to FL topology which result in different benefits and drawbacks. Some solutions focus on providing additional robustness to the system at the expense of decreased privacy and a more cumbersome setup. Others accept a communicational and computational overhead in exchange for the ability to use FL without selecting a single centralized server.

D. Scalability in FL

In our experiments, we have concentrated on the systems that are potentially easy to set up, scalable and able to withstand perturbations present in edge environments. Here, a scalable FL system should be able to maintain high and effective performance in an massively distributed environment, that is, one with a very large number of clients [21]. Many topologies achieve scalability through the creation of local groups, which minimize the necessary frequency of the global aggregation rounds and, by extension, the communicational strain on the server [15].

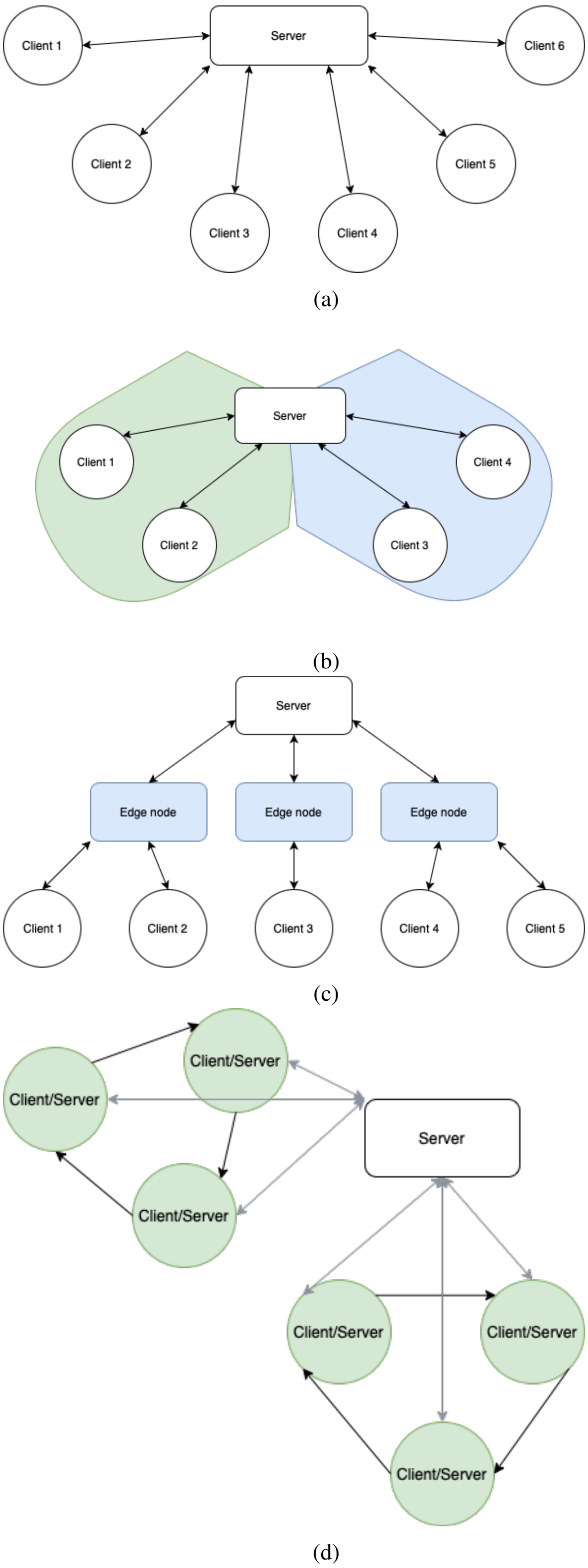
III. Experimental Setup

In order to determine the best topology (both in terms of achieving the best possible performance and maintaining robustness to issues such as heterogenous data or client dropout) for the Assist-IoT pilots, our tests have been conducted on four potential solutions representative of the general trends. Those solutions are described visually in Figure 1. Their short summations can be found below.

1) Centralized: The centralized topology closely adheres to the original process of FL training from [2]. The server sends the model parameters to the clients, where they are trained for multiple iterations and subsequently aggregated on the server. We have decided to include this topology as a baseline for comparison with other, more sophisticated methods.

2) Centralized with dynamic clusters: The structure involves a server communicating with multiple clients. The main difference between this approach and a classic, centralized topology lies in the existence of multiple models (each of which is meant to be suitable just for a subset of the clients). After a given amount of training rounds, the server reclusters the clients based on the similarity of their weights (here estimated using Euclidean distance). The models are then aggregated separately for a given cluster [20]. Importantly, this implementation of SR-FCA (which stands for Successive Refine Federated Clustering Algorithm) aggregates the models using TrimmedMeanGD [22] instead of FedAvg, which provides additional robustness to the training process by removing outlier weights before aggregation. This method does not necessitate any previous knowledge about the number of clusters in the dataset nor additional computation on the client. The dynamic nature of its clustering algorithm causes this method to easily adapt to changes in client number and distribution. Unfortunately, SR-FCA does not necessarily increase the scalability of the solution in its current form.

3) Hierarchical: A hierarchical topology, known also as a tree topology [23], introduces a third type of node, apart from the client and server node to a centralized system: an intermediate (edge) node. In this case, the FL process begins with the server sending the model parameters to the edge nodes, which in turn send them to their clients. The clients train the model for a single iteration and send the results to the edge node, which aggregates those intermediate results. After this process repeats a set number of times, all the aggregated parameters from all of the edge nodes are once again aggregated on the main server [24]. As for the grouping of the clients to a given edge node, this simulation follows the heuristic introduced in [14] by spreading out groups of clients with similar data distribution between various clients.



(a)

(b)

(c)

(d)

Fig. 1: A visualization of the FL system topologies investigated in this work

We have decided to test it as a more sophisticated and scalable FL topology, that nevertheless does not need computationally-intensive client clustering and does not introduce additional aggregation algorithms to Federated Averaging.

4) Hybrid: Tornadoes, or STAR-rings, is an especially promising approach presented in [15]. It combines a centralized solution with the existence of local, ring-based client groups. After the global server supplies the clients with starting parameters, the clients train the model and pass it on to the next client in their ring. In the next iteration, they accept an appropriate model from the previous client in the chain, train it for a given iteration on their own data and pass it on to the next client. After a set amount of inter-node iterations, all the model parameters from all of the clients are aggregated by the centralized server. Interestingly, since ring-based groups can be very susceptible to catastrophic forgetting in groups with high variance [15], a specialized clustering algorithm has been proposed by the authors. This algorithm requires access to client data distributions to compute the most optimal arrangement of ring groups, which may not be suitable for more private use cases. Additionally, it tends to be quite resource-intensive and frequently returns rings with significantly unequal numbers of nodes, which in some cases may complicate system maintenance. Although this FL topology requires using both an exhaustive client grouping algorithm and a more elaborate communication schema, the reported scalability of this method in environments with a large number of nodes is promising enough that we have decided to examine it further in our research.

A. Experiment Design

The experiments were conducted using the German Traffic Sign Recognition Benchmark Dataset [25], developed for a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks in 2011. The dataset incorporates 43 distinct classes divided into batches of size 16. The data was thoroughly shuffled and divided equally between the clients. Later, 80% of that dataset was used as the training data and 20% as the testing data. Independently of the local datasets, a global test set was placed on the server containing 12630 out of all the examples, with the remaining 39209 being divided between all the clients. In order to minimize the computations necessary for the simulation, the dataset has been rescaled to the size of 32 by 32 pixels.

The model used for the experiments consisted of 2 convolutional layers and a single dense layer. It was initially trained using the Adam optimizer with categorical cross-entropy loss without any gradient clipping. After the analysis of first experiments, gradient clipping was introduced for weights exceeding the value of 1.0. The clients were trained for 25 global rounds with 20 local iterations (the exact manner of conducting local iterations differed from topology to topology).

1) Client Grouping and Communication Schema: For the centralized training, no grouping of the clients was involved. Instead, the clients locally trained the model for one epoch on their own data and then sent those models to the server for aggregation, which constituted a full round. 25 of such training rounds have been conducted, with metrics such as aggregated loss, aggregated accuracy, global test set loss, and global test set accuracy being gathered after each of those rounds.

In the case of the centralized topology with dynamic clusters, the threshold λ of 5, size parameter t of 3 and β of 0.1 have been used. The Euclidean distance served as a metric to compute the differences between local weights. The clients have trained for 20 local iterations before each global round, and every 4 global rounds the clients were reclustered.

For the training of hierarchical FL a total of 5 intermediate nodes have been simulated, each managing 20 clients assigned to it. To sum up, the training was conducted on a 100 FL clients. Hierarchical FL involves a more intense communication protocol in the relation between the clients and the intermediate nodes: FedSGD [14]. For this reason, while the global communication schema between the edge nodes and the server has been maintained, the communication between the clients and edge nodes was much more frequent. Although this schema does increase the intensity of communication between nodes, it does not additionally overwhelm the server and, instead, maintains constant contact with edge nodes, which are presupposed to be much closer geographically located to the clients than the server.

Finally, for the hybrid topology, the number of 33 clusters was determined to be the most appropriate. This decision was influenced mainly by the suggestion placed in the original paper, highlighting the importance of small rings [15]. The original paper was also the source of the algorithm used for grouping the clients into clusters. The lengths of the resulting clusters vary from 1 to 8 clients per cluster. Additionally, the decision to conduct the experiments using a larger number of clusters did not influence the computational intensity of the process for the clients. It also did not add any overhead to the necessary communications between the clients and the server. Similarly to the hierarchical FL, the schema used here maintained a set number of global rounds with a set number of local rounds of training in between, here involving the clients accepting a new model, training it for one batch, and then passing it down the chain.

IV. Results and the diagnostic process

First tests conducted on IID data can be seen in Figure 2. Each one was conducted three times and averaged in order to obtain a smoother, more informative curve. Although two topologies, centralized (yellow) and centralized with dynamic clusters (blue), seem to be converging smoothly, there are suspicious perturbations that can be spotted both in the case of the hybrid topology (green) and hierarchical topology (purple). A potential explanation of the similar results of the centralized topology and centralized with dynamic clustering may stem from the fact, that in highly IID environments centralized topologies with dynamic clustering form just a single cluster and therefore are reduced to a simple centralized topology. Further examination reveals that each drop in the aggregated accuracy for the hierarchical topology happened in a different run.

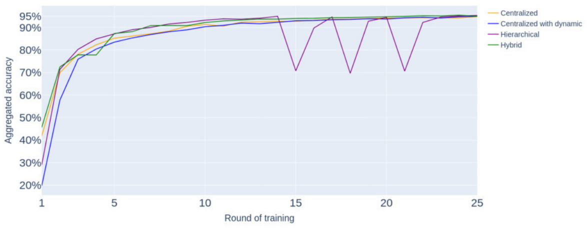


Fig. 2: The accuracy training curves for initial experiments

The issue has been further investigated in Figure 3a. The figure shows the mean aggregated loss measured for the clients belonging to a given cluster (differentiated with a color) after each local iteration and shown on a logarithmic scale. A cluster in our implementation of hierarchical FL includes all the clients performing their local aggregation on the same edge node, which means that all of the clients involved in the simulation are divided into 5 clusters with 20 clients each. The rise in aggregated test loss starts in iteration 201 with a global aggregation round (marked as a pink cross on the figure), and begins to drop after 221, so after another global aggregation round. It can be then observed that the sudden rise in aggregated was correlated with the adherence to a given cluster. Perhaps the frequent local aggregation rounds in hierarchical FL minimize the effectiveness of the Adam optimizer and cause gradient explosions to spread more effectively.

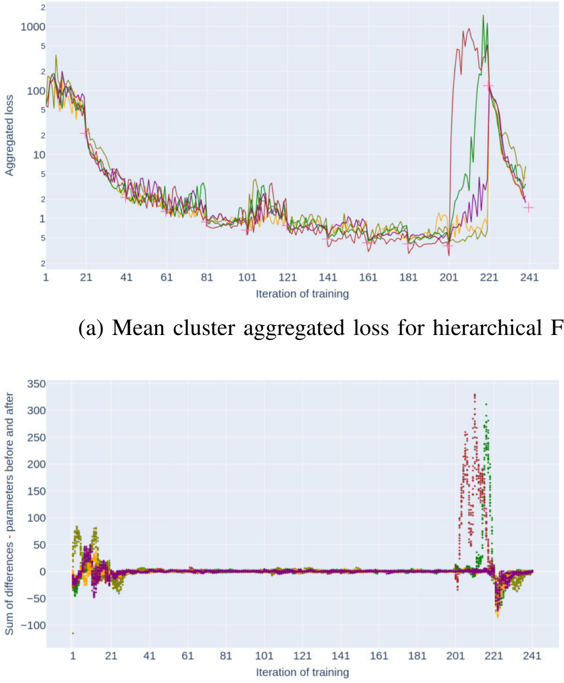
Figure 3b shows a makeshift metric, measuring the sum of the differences between obtained and trained weights for each client after each local iteration. The colors differentiating the adherence to a given node are maintained. The sudden increases in the weight differences correlate with the rise in aggregated loss both in the cluster affiliation and the iteration. These results confirm the existence of a gradient explosion problem.

After diagnosing the issue, an additional precaution of gradient clipping was applied to the model. New trials were then conducted to see if the learning curve improved. Figure 4a shows improvement in the form of a significantly smoother training process. Analogously, the difference in weights as shown on Figure 4b has stabilized and decreased significantly.

Repeating the first trial yields on Figure 5 slightly better, significantly smoother performance for all of the already mentioned FL topologies. An especially significant difference is visible for the hierarchical FL performance (purple), which suggests it to be an especially vulnerable topology to gradient explosions.

V. Discussions

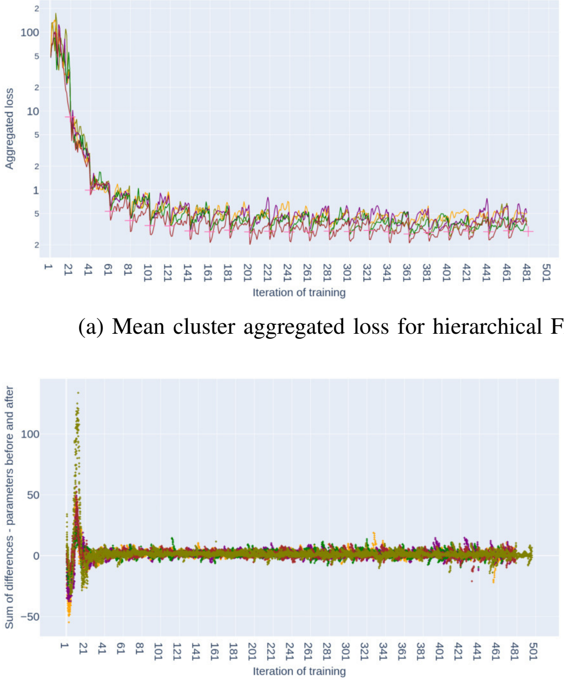
Based on the usefulness of the additional metrics collected throughout the training in the diagnostic process, we propose continuous monitoring of the gradient scale of the local



(a) Mean cluster aggregated loss for hierarchical FL

(b) Client weight differences for hierarchical FL

Fig. 3: Initial experiments



(a) Mean cluster aggregated loss for hierarchical FL

(b) Client weight differences for hierarchical FL

Fig. 4: Improved experiments (after the addition of gradient clipping)

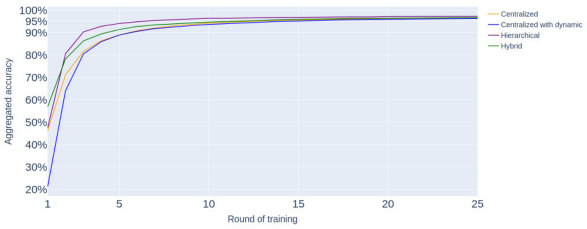


Fig. 5: The accuracy training curves for improved experiments

models through the regular computation of the gradient scale coefficient on the clients. The gradient scale coefficient is defined as follows.

$$GSC(k,l,f,\theta,x,y) = \frac{||J_k^l||_{qm}||f_k||_2}{||f_l||_2} \tag{1}$$

It measures the relative sensitivity of layer l with regards to random changes in layer k, measuring the size of the gradient flowing background relative to the size of the activations growing forward. A detailed explanation of this metric and how to use it to can be found in [6]. Its usefulness stems from robustness to network scaling, which introduces the possibility of result standardization.

We would like to measure the GSC of each of the client models after every iteration in order to use it to detect large, sudden shifts on the global level. These sudden shifts could then indicate the possibility of gradient explosion and prompt the developer to quickly recognize the problem without wasting needless resources for unsuccessful ML training.

VI. Conclusions

Although FL system and algorithm design remain popular research areas, the question on how to effectively enable the debug and maintenance of those systems is still largely unanswered. Our case study presents how a problem commonly encountered in classical ML may present in more complex FL topologies. We have also proposed a potential monitoring method for the early detection of such problems. All in all, we would like to stress the importance of the inclusion of such tools in distributed environments, where issues like client dropout or diverging distributions may be masking more fundamental problems.

References

[1] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *CoRR*, vol. abs/2009.13012, 2020. [Online]. Available: {https://arxiv.org/abs/2009.13012}

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. [Online]. Available: {http://arxiv.org/abs/1602.05629}

[3] J. Wu, S. Drew, F. Dong, Z. Zhu, and J. Zhou, "Topology-aware federated learning in edge computing: A comprehensive survey," 2023. [Online]. Available: https://arxiv.org/abs/2302.02573

[4] "Pilot Scenario Implementation – First Version," 2022. [Online]. Available: {https://assist-iot.eu/wp-content/uploads/2022/05/D7.2_Pilot_Scenario_Implementation-First_Version.pdf}

[5] *Introducing Federated Learning into Internet of Things ecosystems – preliminary considerations*, 07 2022.

[6] G. Philipp, D. Song, and J. G. Carbonell, "The exploding gradient problem demystified - definition, prevalence, impact, origin, tradeoffs, and solutions," 2018.

[7] A. Li, L. Zhang, J. Wang, F. Han, and X.-Y. Li, "Privacy-preserving efficient federated-learning model debugging," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2291–2303, 2022.

[8] Y. Liu, W. Wu, L. Flokas, J. Wang, and E. Wu, "Enabling sql-based training data debugging for federated learning," *CoRR*, vol. abs/2108.11884, 2021. [Online]. Available: https://arxiv.org/abs/2108.11884

[9] W. Gill, A. Anwar, and M. A. Gulzar, "Feddebug: Systematic debugging for federated learning applications," 2023.

[10] S. Duan, C. Liu, P. Han, X. Jin, X. Zhang, X. Xiang, H. Pan *et al.*, "Fed-dnn-debugger: Automatically debugging deep neural network models in federated learning," *Security and Communication Networks*, vol. 2023, 2023.

[11] B. Hanin, "Which neural net architectures give rise to exploding and vanishing gradients?" in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/13f9896df61279c928f19721878fac41-Paper.pdf

[12] A. Bellet, A. Kermarrec, and E. Lavoie, "D-cliques: Compensating noniidness in decentralized federated learning with topology," *CoRR*, vol. abs/2104.07365, 2021. [Online]. Available: {https://arxiv.org/abs/2104.07365}

[13] L. Chou, Z. Liu, Z. Wang, and A. Shrivastava, "Efficient and less centralized federated learning," *CoRR*, vol. abs/2106.06627, 2021. [Online]. Available: https://arxiv.org/abs/2106.06627

[14] N. Mhaisen, A. A. Abdellatif, A. Mohamed, A. Erbad, and M. Guizani, "Optimal user-edge assignment in hierarchical federated learning based on statistical properties and network topology constraints," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 55–66, 2022.

[15] J. Lee, J. Oh, S. Lim, S. Yun, and J. Lee, "Tornadoaggregate: Accurate and scalable federated learning via the ring-based architecture," *CoRR*, vol. abs/2012.03214, 2020. [Online]. Available: {https://arxiv.org/abs/2012.03214}

[16] I. Hegedűs, G. Danner, and M. Jelasity, "Gossip learning as a decentralized alternative to federated learning," in *Distributed Applications and Interoperable Systems*, J. Pereira and L. Ricci, Eds. Cham: Springer International Publishing, 2019, pp. 74–90.

[17] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Federated learning for distributed spectrum sensing in nextg communication networks," 2022. [Online]. Available: https://arxiv.org/abs/2204.03027

[18] H. Eichner, T. Koren, H. B. McMahan, N. Srebro, and K. Talwar, "Semi-cyclic stochastic gradient descent," *CoRR*, vol. abs/1904.10120, 2019. [Online]. Available: {http://arxiv.org/abs/1904.10120}

[19] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," 2021.

[20] Harshvardhan, A. Ghosh, and A. Mazumdar, "An improved algorithm for clustered federated learning," 2022.

[21] M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and privacy," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3790–3804, 2021.

[22] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a.html

[23] J. Wu, S. Drew, F. Dong, Z. Zhu, and J. Zhou, "Topology-aware federated learning in edge computing: A comprehensive survey," *arXiv preprint arXiv:2302.02573*, 2023.

[24] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Edge-assisted hierarchical federated learning with non-iid data," *CoRR*, vol. abs/1905.06641, 2019. [Online]. Available: http://arxiv.org/abs/1905.06641

[25] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, 2012, selected Papers from IJCNN 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608012000457

DOI: 10.15439/2023F6381
ISSN 2300-5963 ACSIS, Vol. 35

Use of traffic sampling in anomaly detection for high-throughput network links

Marek Bolanowski, Andrzej Paszkiewicz
0000-0003-4645-967X
0000-0001-7573-3856
Rzeszów University of Technology, Rzeszów, Poland
Email: {marekb, andrzejp}@prz.edu.pl

Hubert Mazur
0009-0003-6784-8428
Rzeszów University of Technology, Rzeszów, Poland
Email: hub.mazur99@gmail.com

***Abstract*—Currently, anomaly detection is an increasingly important issue in terms of research work and applications in production systems. Information about system malfunction allows the implementation of precise diagnostic and corrective actions. Two main approaches based on statistical analysis and machine learning techniques are used in anomaly detection systems, which are computationally complex, especially when dealing with high traffic volumes in computer network. In this paper, the limitation of the sampling frequency for network traffic parameters is proposed as a technique to reduce the computational complexity of anomaly detection methods. The proposed approach has been verified in a real network link monitoring system for a medium-sized ISP. The results obtained are promising and can be used to build a production system that enables the development of early warning systems in the area of security incident detection dedicated to high-speed access links.**

I. Introduction

DISTRIBUTED information systems are becoming increasingly prevalent in critical areas of human life. For instance, they are used to control traffic in the city [1], [2], monitor patients' vital signs [3], or manage technological processes in smart factories [4]. This information systems are exposed to a number of new types of cyber security threats. The market offers ready-made tools for executing attacks, which affects the constant increase in the number of security incidents. During the pandemic period alone, cybercrime increased by 600% [5], and the average cost of a data security breach in the U.S. in 2022 was 4.35 million [6]. There is no single effective system of protection against these threats. Nowadays threat detection and elimination systems have a cascade structure. In other words, we have many interconnected layers in which IDS, IPS, ACL, etc. function. Each type of layer is sensitive to different types of attacks. In the case of carrier access links, such as those used for Internet Service Provider (ISP) companies, simple Access Control List (ACL) rules that filter network traffic based on source and destination addresses are generally applicable. Even in the case of such a simple mechanism, the implementation of a larger number of ACLs, or the implementation of a mechanism for logging information (what flow and by what ACL was blocked) can bring significant delays in the transmission path. Therefore, the authors posed the question during their research: is it possible to detect anomalous behavior without introducing additional delay while reducing the computational complexity of detecting process? Anomaly detection is an important data analysis task that detects anomalous or abnormal data from a given data set. Preliminary research has shown that a conducted cyberattack can affect the change of statistical characteristics of network traffic in the access link. Therefore, the analysis of descriptive link parameters, statistical techniques or artificial intelligence can be used in the area of an access link on the border of the protected network to detect the threat. Anomaly detection is widely used in myriad fields such as medical, public health, fraud detection, intrusion detection, industrial damage, image processing, sensor networks, robot behavior and astronomical data [7]. Current research is concerted around speeding up the detection process reducing the computational complexity of the entire process and identifying not only the occurrence of a given anomaly but also eliminating its causes.

At present, there is a clear trend related to identifying the best AI models for anomaly detection in ISP links in order to achieve the best possible detection performance. Applications in this area include both supervised and unsupervised methods [8], [9], [10], [11]. Of course, previously, network traffic sampling methods [12] were used for anomaly detection using traditional IDS probes. Such methods were applied, for example, in the work [9], and the obtained results look promising. Their applications allow for preliminary verification in terms of detecting anomalies in large volumes of network traffic. However, it should be noted that a large body of work in this field is based on previously prepared test datasets [13], [14], [15] or on data obtained from real links with low throughputs [16]. Preliminary results of conducted research have shown that, in addition to data sampling, the proper preparation of acquired data and flow aggregation have a positive impact on detection outcomes. Of course, data preprocessing can also be computationally complex, but it can be easily parallelized and computed distributed among system nodess [17], [18]. The analysis of available literature clearly demonstrates the pursuit of increasing the accuracy of predictive models, but we must not forget about their applicability in real computer networks. In this study, the authors decided to investigate the impact of data set impoverishment (sampling) on the sensitivity of the anomaly detection model and whether it is possible to limit the number of processed traffic samples while maintaining the detection level. The entire study was conducted in a production network of an ISP (Enf sp. z o.o). The developed detection layer at the ISP access link can serve as an additional layer of protection against cyber-attacks in cascade anomaly detection systems [19]. If the detection effectiveness of the model slightly decreases with decreasing traffic sampling frequency, it will positively contribute to reducing the amount of necessary measurement data to be transmitted and the processing time required, thus increasing the applicability of the solution in real networks.

The article has the following structure: Chapter 2 presents the network structure of the ISP access node and the architecture of the data acquisition and processing system. In Chapter 3, the data aggregation and sampling process are discussed in detail. Chapter 4 describes the model used for anomaly detection. Chapter 5 presents the obtained results, including the accuracy of detection in relation to the sampling frequency. In Chapter 6, the obtained results were summarized, and directions for further research were indicated.

II. ISP EDGE NODE TOPOLOGY

As mentioned earlier, the research was conducted in the environment of a medium-sized ISP. Real network traffic from end customers was analyzed. In order to carry out the research, it was necessary to modify the structure of the access node used in the system. The system structure is shown in Figure 1. The access router (Extreme MLX-4) connect the entire network segment to the Internet using the BGP protocol. The core of the access network was built based on two switches: Extreme 690 (CORE switch) and Extreme 670 (S1 switch). Policy shaping and NAT for the LAN segment were implemented through a software router (TC + IPTables) built on a Dell R710 server. Two additional hosts, PC1 and PC2, were introduced into the network. PC1 was connected to the LAN network using a Dasan switch, while PC2 was connected through a TP-Link switch in the demilitarized zone of the access node. Its task was to emulate an attack on PC1. All traffic transmitted to the LAN is directed through port P1. Using the port mirroring mechanism, the traffic from port P1 is copied to the Dell PowerEdge R940 server, where calculations related to anomaly detection are performed. This server had the following specifications: Intel(R) Xeon(R) Gold 624 CPU @ 2.60GHz processor; 128 GB of RAM; NVIDIA Tesla V100-PCIE-16GB GPU; HDD 4.5 TB. The 'PowerEdge R940' server hosted a virtual machine based on the Debian OS, which collected traffic (bidirectional) using tcpdump. The laboratory setup allowed for data collection in the infrastructure from the layer 1 to layer 7 of the ISO/OSI model, capturing individual packets for specific network flows using the tcpdump sniffer. Such an environment allowed for testing various data processing techniques and AI algorithms to determine the optimal sampling frequency at which the created models would effectively detect abnormal periods in the packet flow in the investigated network.

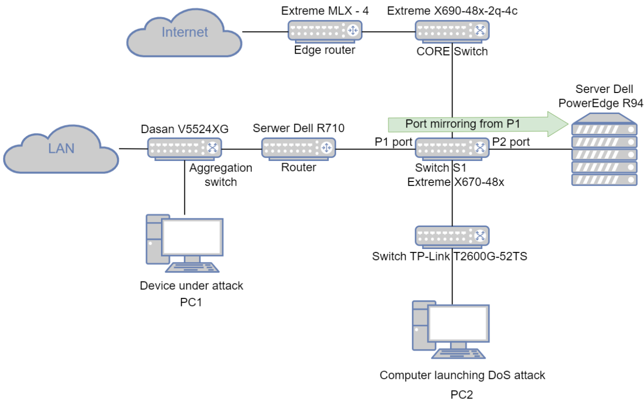


Fig. 1. ISP network edge node architecture with testbed elements

In the next step, a system was built to allow smooth frequency sampling changes. It should be noted that during the conducted research, the entire traffic from port P1 was collected. The entire sampling process was performed on the PowerEdge R940 server, enabling repeated tests for different sampling frequencies. Ultimately, in production systems, the sampling frequency can be set on a specific probe installed in the network. This not only reduces the amount of processed data but also limits the amount of data transmitted between the probe and the detection system. Additionally, initial data preprocessing can also be performed on the measurement probe (in the test system port P1 acts as the probe). Sequential packet selection with a fixed period between consecutive samples was used in the sampling process. In other words, all collected packets were labeled with consecutive natural numbers, and only those packets whose indexes were multiples of a selected natural number s, such as $s = 2$ (sampling every other packet), were chosen for further analysis. Of course, it is possible to apply a different statistical distribution of samples, which will be the subject of further research. The data received from the ISP network was saved in .dump file format. Subsequently, it was divided into equal time intervals (windows). Each window represents a short time of network operation that is evaluated by the machine learning model to classify the entire window as either anomalous or not. The order of the sampling and windowing processes is interchangeable. In the next step, CICFlowMeter software [20] was used for feature extraction. As a result of its operation, CSV files containing feature vectors describing each analyzed packet were obtained. These files were used in further analysis for feature selection and aggregation, which will be described in detail in the subsequent part of the article. The data processing process is described in Figure 2.

In order to describe the process of windowing, i.e., to divide packets into windows depending on the time of their capture, let us make the following assumptions:

$$T = \{t_0, t_1, t_2, \ldots, t_z\}$$
$$t_k = k \cdot f, \quad k \in \mathbb{N}_0, \quad k \leq z$$

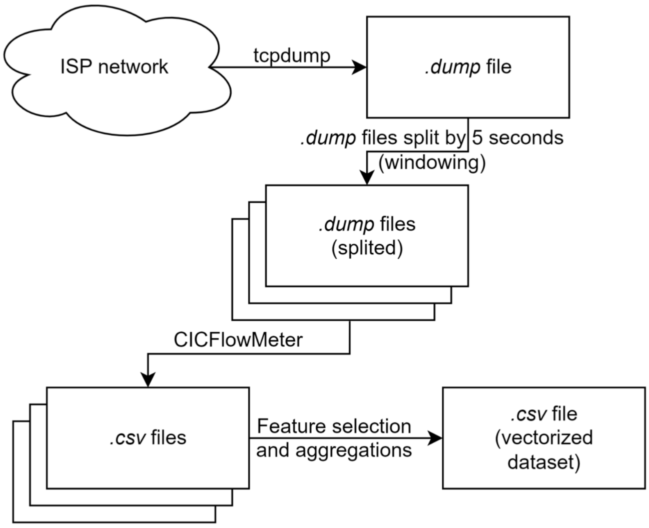

ISP network
tcpdump
.dump file
.dump files split by 5 seconds (windowing)
.dump files (splited)
CICFlowMeter
.csv files
Feature selection and aggregations
.csv file (vectorized dataset)

Fig. 2. Data processing scheme

$$z = \left\lfloor \frac{t_{test}}{f} \right\rfloor,$$

where: t_{test} – total duration of the test; k – the number of the given window; P - set of all packages; T – set of all the moments of time in which the windows begin; f – the length of the window within which the packages will be aggregated.

In view of this, we can assume that the set of packages contained in a given window can be described as follows:

$$O_u = \{p \in P : t_u \leq p^{(t)} < t_{u+1}\}$$

$$u = 0, 1, \ldots, z-1$$

where: O – set of all windows; $p^{(t)}$ - packet capture time p.

The data was divided into two sets:

1) The training set represented normal network traffic and was collected for one hour under standard network operating conditions. It consisted of traffic from LAN clients and PC2 (see Figure 1). These data will be used to train a model for the purpose of identifying normal traffic. The anomaly detection model used in the further part of this work will be based on a set of unsupervised algorithms. This approach was chosen because in case of supervised learning model, staff would have to label which packets belonged to normal traffic and which were considered anomalous. This process is extremely time-consuming. Naturally, in the case of unsupervised learning, during the training period, it is essential to ensure that the network is not under attack. Therefore, the training time of the models must be closely monitored by the technical personnel. After training the model on attack-free traffic, it should be able to determine whether incoming packets grouped in windows O_u will contain flows characterized by parameter values deviating from the characteristics of normal traffic. The training dataset contained information on 1,182,566,238 packets.
2) The test set aimed to verify the performance of the model based on the training set. The packets in the windows represented network traffic in two states: normal and anomalous. The anomaly was a 5-minute long Denial-of-Service (DoS) attack. The test dataset contained information on packets captured over a period of 45 minutes, out of which 20 minutes represented normal traffic, the next 5 minutes included the anomaly, and the remainder consisted of normal traffic again. For this dataset, window labeling was performed to mark them as either anomalous or non-anomalous in order to assess the quality of the trained model. The test set contained information on 697,871,782 packets.

It should be noted that during the conducted research, a series of experiments related to DoS and DDoS attacks were carried out, and repeatability of the obtained results was achieved. The DoS attack was identified by the ISP operator as the most common type of attack that the network encounters during its normal operation. Of course, the model shows sensitivity to other types of anomalies not related to DoS attacks, but research in this area needs to be continued.

III. Pre-processing of data

The data collected during the experiments were continuously subjected to the process of cleaning and preparation for further stages of processing related to model training and anomaly detection. According to the scheme presented in Figure 2, all extracted windows O_u had to undergo a vectorization process, so that each window represented independent feature vectors. The vectorization method used in this work is an aggregation approach of selected flow features obtained through feature extraction using the CICFlowMeter software for unique source and destination IP address pairs. A flow represents the packet flow between two network devices, defined by source and destination IP addresses, as well as used ports and network protocols. For the purpose of this work, the notations $p^{(s)}$ and $p^{(r)}$ were adopted to denote the source and destination IP addresses of a given packet, respectively. Therefore, the vectorization process can be described as follows:

$$D^{(u)} = \left\{ F_2\left(F_1\left(R_i^{(u)}\right)\right) : i = 0, 1, \ldots, \left|R^{(u)}\right| \right\},$$

$$R^{(u)} = \left\{ \left\{ p \in O_u : \left\{p^{(s)}, p^{(r)}\right\} = \bar{U}_j^{(u)} \right\}, \right.$$
$$\left. j = 0, 1, \ldots, \left|\bar{U}^{(k)}\right| \right\},$$

$$U^{(k)} = \left\{ \left\{p^{(s)}, p^{(r)}\right\} : p \in O_k \right\},$$

where: $D^{(u)}$ - the aggregated feature vectors of window flows u; $R^{(u)}$ - a set of packet collections with unique destination and recipient IP addresses; $U^{(u)}$ - a set of all destination and source IP address pairs in the window k; $\bar{U}^{(u)}$ - a subset contained in $U^{(u)}$ composed only of its unique elements; F_1 - the first aggregation function, its task is to aggregate packet features for each unique flow; F_2 - the second aggregation function, its task is to aggregate flow features for each unique destination and recipient IP address pair.

In the first stage (aggregation F_1), the characteristics of each flow occurring in the processed window were aggregated. The set of packets in the window is divided into subsets, where each subset contains the set of packets responsible for the creation of a particular flow. In the second stage, the aggregated characteristics obtained in stage F_1 were further aggregated for each unique destination and recipient IP address pair $p^{(s)}, p^{(r)}$ in the processed window O_u. Additionally one dimension describing the number of flows for unique destination and recipient IP address pairs was added to the final vectors $D^{(u)}$. This type of aggregation allows for a complete vector representation of flow data for a given window, which directly translates into reducing the computational complexity of the detection process by reducing the number of features to 25. These features were selected through experimental work aimed at identifying characteristics that maximize the effectiveness of anomaly detection. The list of all used features is presented in Table I, which also indicates the actions performed in the individual aggregation stages F_1 and F_2.

TABLE I
FEATURES USED IN FEATURE EXTRACTION PROCESS

ID	Feature Description	Aggregation F_1	Aggregation F_2
0	Number of flows		Count
1	Flow duration		Average
2	Number of packets sent	Count	Sum
3	Number of packets received	Count	Sum
4	Total length of packets sent	Sum	Sum
5	Total length of packets received	Sum	Sum
6	Minimum length of packets sent	Minimum	Average
7	Maximum length of packets sent	Maximum	Average
8	Average length of packets sent	Average	Average
9	Standard deviation of length of packets sent	Standard deviation	Average
10	Minimum length of packets received	Minimum	Average
11	Maximum length of packets received	Maximum	Average
12	Average length of packets received	Average	Average
13	Standard deviation of length of packets received	Standard deviation	Average
14	Packets per second	Average	Sum
15	Bytes per second	Average	Sum
16	Packets sent per second	Average	Sum
17	Packets received per second	Average	Sum
18	Minimum packet length	Minimum	Average
19	Maximum packet length	Maximum	Average
20	Average packet length	Average	Average
21	Standard deviation of packet length	Standard deviation	Average
22	Average packet size	Average	Average
23	Average segment size of sent packets	Average	Average
24	Average segment size of received packets	Average	Average

IV. MODEL DESCRIPTION

To test the performance of the sampling frequency's impact on anomaly detection accuracy, a densely connected neural network based on an autoencoder architecture was used [21]. The application of this model for anomaly detection is well-known in the literature, and its effectiveness for the complete dataset was experimentally confirmed in the initial stage of the conducted research. The operation of the adopted model can be divided into two main stages:

1) The forward propagation stage of the neural network, which consists of two key components:
   a) Compression of the input feature vector into fewer dimensions (encoding).
   b) Reconstruction of the compressed feature input vector (decoding).
2) The stage of calculating the reconstruction error based on the comparison of the input vector with the output of the neural network. Based on the reconstruction error, a decision is made to classify the sample into normal or containing an anomaly.

Let M denote the reconstruction error for a single vector w. It can be observed that as a result of applying aggregation F_2, we obtain a set of vectors describing the features of all unique sender and receiver IP address pairs. Therefore, the reconstruction error for a single vector w can be expressed as follows:

$$M_w^{(u)} = \frac{\sum_{i=0}^{24} (D_i^{(u,w)} - m(D^{(u,w)})_i)^2}{25}$$

To calculate the reconstruction errors for all vectors in a given window O_u, the above formula should be applied to each $w = 0, 1, \ldots, \left|D^{(u)}\right|$.

The classification of a window can be expressed as follows:

$$a_u = \begin{cases} \text{anomaly} & \text{if } \max(M^{(u)}) > y \\ \text{no anomalies} & \text{otherwise,} \end{cases}$$

where: y – classification threshold; a_u – window classification decision u; $m(D^{(u,w)})$ - vector reconstructed using autoencoder.

Table II presents the detailed architecture of the utilized autoencoder, which was developed based on conducted experiments aiming to maximize the effectiveness of anomaly detection. The dimensions of the input data to each of the layers is marked as follows: the first dimension marked "-" is the number of feature vectors, which can be arbitrary. The second dimension is the size of the input vectors. The output dimension column describes the dimension of the vectors after calculating the total excitation of each neuron and applying the activation function.

The model was trained using windows from the training dataset. It was trained for 30 epochs using the ADAM[22] optimization method and mean squared error (MSE)[23] as the

TABLE II
AUTOENCODER ARCHITECTURE

Layer	Input dimension	Number of neurons	Output dimension	Activation function.
densely connected densely connected	(-, 25) (-, 13)	13 6	(-, 13) (-, 6)	RELU RELU
densely connected densely connected	(-, 6) (-, 13)	13 25	(-, 13) (-, 25)	RELU no activation function

reconstruction loss for window characteristics. Additionally, to improve the weight fitting process, the data underwent standardization[24] using the mean and standard deviation of the features from the windows in the training dataset.

V. RESULTS

The combination of processing data using aggregation of unique sender and receiver IP address pairs, along with a model based on maximum reconstruction error of processed feature vectors in each pair's window, yielded good results in anomaly detection task. The windows where anomalies occurred showed significantly higher maximum reconstruction error compared to those characterized by normal traffic. Table III presents the results of anomaly detection quality on the test dataset. The performance of the developed model was evaluated on the test dataset for different sampling frequencies $s = 10, 25, 50$. The obtained results are presented in Figures 3 to 5. The maximum reconstruction error for the non-anomalous sender and receiver IP address pair is indicated in blue color, while the reconstruction error for the attacking device's IP address and the target IP address is shown in red color.

TABLE III
RESULTS OF MODEL EVALUATION ON THE TEST SET

Sampling frequency (s)	Window size in seconds	Detection accuracy
1	5	100.0%
2	5	100.0%
5	5	100.0%
10	5	100.0%
25	5	99.8%
50	5	87.6%

The results show that satisfactory performance is achieved even in the case of $s = 25$, which means checking every 25th network traffic sample. It is important to note that in the experiments, the window length was 5 seconds and the entire attack lasted 5 minutes. It is assumed that for longer-lasting attacks with higher network traffic intensity, such as DDoS attacks, the sampling frequency can be further reduced. The sampling threshold should be determined individually based on the characteristics of the specific network and the sensitivity of the system expected by the ISP operator.

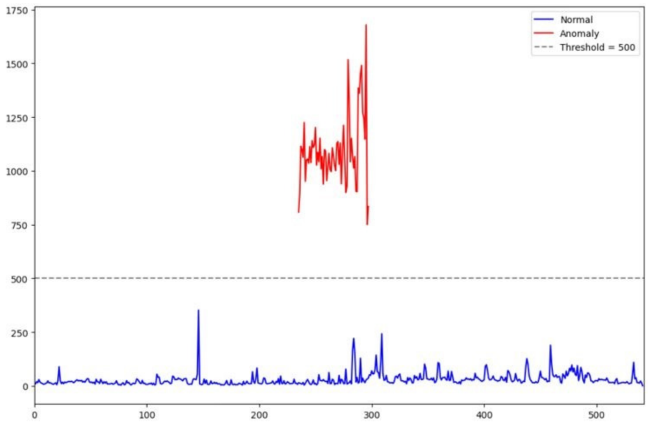


Fig. 3. Model evaluation on test set for sampling every 10 packet

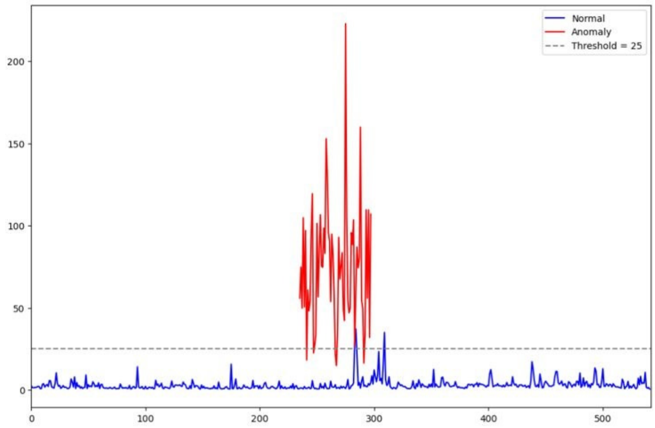


Fig. 4. Model evaluation on test set for sampling every 25 packet

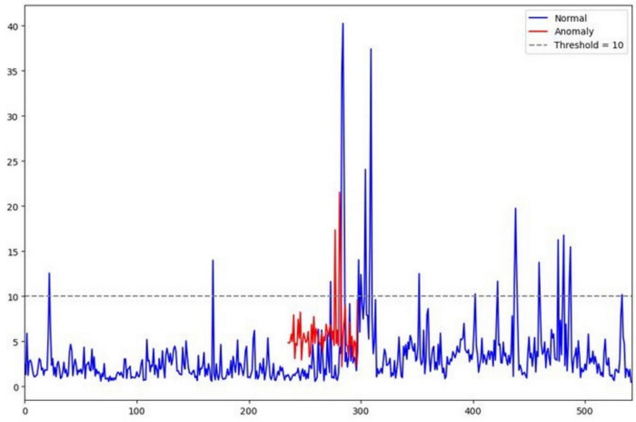


Fig. 5. Model evaluation on test set for sampling every 50 packet

VI. Summary

The paper presents the results of research related to the possibilities of applying a data sampling mechanism for anomaly detection on high bandwidth network links. The research work was carried out in a medium ISP environment in a production infrastructure. The anomaly detection approach proposed in the work taking into account windowing and data sampling allowed to reduce the data needed for anomaly detection (DoS Attack) by 25 times. This makes it possible to reduce the bandwidth of IDS and IPS probes detecting threats, which will directly translate into the cost of implementing cybersecurity systems. Further research concert around the use of non-uniform sequential sampling of traffic, e.g. by using different frequencies and statistical distributions depending on the time of day or network activity. In addition, preliminary studies have shown that the designed system is also effective in detecting other types of anomalies, e.g. data generated by faulty network interfaces. It should be noted that the proposed approach makes it possible to monitor high-throughput access links of ISPs and thus introduce another layer of protection for the entire ICT system against cyber attacks. Thanks to the use of traffic copies, the proposed architecture itself does not bring delays to the end user traffic forwarding process, and once a threat is detected, a given flow can be redirected for further inspection using policy-based routing mechanisms.

Acknowledgment

Work of Marek Bolanowski and Andrzej Paszkiewicz is financed by the Minister of Education and Science of the Republic of Poland within the "Regional Initiative of Excellence" program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN. The research was carried out in cooperation with "Centrum Badawczo-Rozwojowym Inteligentnych Sieci CBRIS" Enf Sp. z o.o.

References

[1] B. Pawłowicz, M. Salach, and B. Trybus, "Infrastructure of RFID-based smart city traffic control system," in *Automation 2019*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Springer International Publishing, 2020, vol. 920, pp. 186–198. ISBN 978-3-030-13272-9 978-3-030-13273-6 Series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-030-13273-6_19

[2] B. Pawłowicz, M. Salach, and B. Trybus, "Smart city traffic monitoring system based on 5g cellular network, RFID and machine learning," in *Engineering Software Systems: Research and Praxis*, P. Kosiuczenko and Z. Zieliński, Eds. Springer International Publishing, 2019, vol. 830, pp. 151–165. ISBN 978-3-319-99616-5 978-3-319-99617-2 Series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-319-99617-2_10

[3] S. Dash, S. Biswas, D. Banerjee, and A. U. Rahman, "Edge and Fog Computing in Healthcare – A Review," *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 191–206, 2019. doi: 10.12694/scpe.v20i2.1504. [Online]. Available: https://www.scpe.org/index.php/scpe/article/view/1504

[4] M. Kostolani, J. Murin, and S. Kozak, "An effective industrial control approach," 2019-09-26. doi: 10.15439/2019F187 pp. 911–914. [Online]. Available: https://fedcsis.org/proceedings/2019/drp/187.html

[5] "Cyber security statistics the ultimate list of stats data, and trends for 2023," https://purplesec.us/resources/cyber-security-statistics/, accessed: 2023-05-02.

[6] "Cost of a data breach 2022 a million-dollar race to detect and respond," https://github.com/ahlashkari/CICFlowMeter, accessed: 2023-05-02.

[7] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. doi: 10.1016/j.jnca.2015.11.016

[8] S. Saha, A. Haque, and G. Sidebottom, "Towards an ensemble regressor model for ISP traffic prediction with anomaly detection and mitigation," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2022. doi: 10.1109/ISNCC55209.2022.9851774. ISBN 978-1-66548-544-9 pp. 1–6.

[9] M. Shajari, H. Geng, K. Hu, and A. Leon-Garcia, "Tensor-based online network anomaly detection and diagnosis," *IEEE Access*, vol. 10, pp. 85 792–85 817, 2022. doi: 10.1109/ACCESS.2022.3197651

[10] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014. doi: 10.1109/SURV.2013.052213.00046

[11] G. Fernandes, J. J. P. C. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019. doi: 10.1007/s11235-018-0475-8

[12] B. Tellenbach, D. Brauckhoff, and M. May, "Impact of traffic mix and packet sampling on anomaly visibility," in *2008 The Third International Conference on Internet Monitoring and Protection*. IEEE, 2008. doi: 10.1109/ICIMP.2008.18. ISBN 978-0-7695-3189-2 pp. 31–36.

[13] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications, 2018. doi: 10.5220/0006639801080116. ISBN 978-989-758-282-0 pp. 108–116.

[14] W. Lu and A. A. Ghorbani, "Network anomaly detection based on wavelet analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 837601, 2008. doi: 10.1155/2009/837601

[15] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. ACM, 2020. doi: 10.1145/3416013.3426457. ISBN 978-1-4503-8120-8 pp. 37–45.

[16] D. Hulskamp and C. Cappo, "Effectiveness assessment of time series models for anomalies detection in real network traffic," in *2022 41st International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2022. doi: 10.1109/SCCC57464.2022.10000354. ISBN 978-1-66545-674-6 pp. 1–8.

[17] X. Larriva-Novo, M. Vega-Barbas, V. A. Villagrá, D. Rivera, M. Álvarez Campana, and J. Berrocal, "Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets," *Applied Sciences*, vol. 10, no. 10, p. 3430, 2020-05-15. doi: 10.3390/app10103430

[18] A. Bhandari, K. Kumar, A. L. Sangal, and S. Behal, "An anomaly based distributed detection system for DDoS attacks in tier-2 ISP networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1387–1406, 2021. doi: 10.1007/s12652-020-02208-3

[19] A. Bădică, C. Bădică, M. Bolanowski, S. Fidanova, M. Ganzha, S. Harizanov, M. Ivanovic, I. Lirkov, M. Paprzycki, A. Paszkiewicz, and K. Tomczyk, "Cascaded anomaly detection with coarse sampling in distributed systems," in *Big-Data-Analytics in Astronomy, Science, and Engineering*, S. Sachdeva, Y. Watanobe, and S. Bhalla, Eds. Springer International Publishing, 2022, vol. 13167, pp. 181–200. ISBN 978-3-030-96599-0 978-3-030-96600-3 Series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-96600-3_13

[20] "Cicflowmeter," https://www.ibm.com/security/data-breach, accessed: 2023-05-02.

[21] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2020. doi: 10.48550/ARXIV.2003.05991 Publisher: arXiv Version Number: 2.

[22] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," vol. 2, no. 1, p. 41, 2019. doi: 10.17977/um018v2i12019p41-46

[23] Y. Liu, "Mean square error of survey estimates," in *Encyclopedia of Quality of Life and Well-Being Research*, F. Maggino, Ed. Springer International Publishing, 2021, pp. 1–3. ISBN 978-3-319-69909-7

[24] M. Gal and D. L. Rubinfeld, "Data standardization," 2018. doi: 10.2139/ssrn.3326377

Proceedings of the 18th Conference on Computer
Science and Intelligence Systems pp. 883–888

DOI: 10.15439/2023F3368
ISSN 2300-5963 ACSIS, Vol. 35

L1-Norm Principal Component Analysis Using Quaternion Rotations

Adam Borowicz
0000-0003-0320-5530
Faculty of Computer Science, Bialystok University of Technology
Wiejska str. 45A, 15-351 Bialystok, Poland
Email: a.browicz@pb.edu.pl

***Abstract*—Principal component analysis (PCA) based on L1-norm has drawn growing interest in recent years. It is especially popular in the machine learning and pattern recognition communities for its robustness to outliers. Although optimal algorithms for L1-norm maximization exist, they have very high computational complexity and can be used for evaluation purposes only. In practice, only approximate techniques have been considered so far. Currently, the most popular method is the bit-flipping technique, where the L1-norm maximization is viewed as a combinatorial problem over the binary field. Recently, we proposed exhaustive, but faster algorithm [1] based on two-dimensional Jacobi rotations that also offer high accuracy. In this paper, we develop a novel variant of this method that uses three-dimensional rotations and quaternion algebra. Our experiments show that the proposed approach offers higher accuracy than other approximate algorithms, but at the expense of the additional computational cost. However, for large datasets, the cost is still lower than that of the bit-flipping technique.**

I. Introduction

PRINCIPAL component analysis (PCA) is a method for multivariate data analysis with various uses, including dimensionality reduction, feature extraction and noise reduction [2]. The PCA tries to identify orthogonal directions, along which the data exhibit the greatest variability. The projections of the data on these directions are viewed as principal components. This technique is also referred as L2-PCA, because the data variability is measured using Frobenius norm (L2-norm on matrices). It can be easily implemented using, for example, singular value decomposition (SVD) of the observation data matrix [3]. However, it is also sensitive to the presence of outliers, i.e., data points that differ significantly from the other observations. In order to mitigate this drawback, several PCA techniques have been proposed that are based on L1-norm [4], [5], [6], [7]. Interestingly, the L1-norm criterion can also be used to perform independent component analysis (ICA) after data whitening [8], [9]. The L1-norm optimization problem can be formulated in several ways [5], but unlike in the case of the L2-PCA, these formulations are not equivalent. In this paper, we consider the following maximization:

$$\mathbf{Q}_{\text{L1}} = \underset{\substack{\mathbf{Q}=[\mathbf{q}_1,\ldots,\mathbf{q}_k]\in\mathbb{R}^{d\times k}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_k}}{\operatorname{argmax}} \sum_{i=1}^{k} \|\mathbf{X}^T\mathbf{q}_i\|_1, \tag{1}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ is a data matrix of rank $r_x \leq \min\{d, n\}$, consisting a sequence of observation vectors $(\mathbf{x}_i)_{i=1}^n$, and $\|.\|_1$ denotes L1-norm that return the sum of the absolute values of the individual entries. The parameter k denotes the number of the L1 principal components. Please note that the problem (1) is not scalable, i.e. it can not be translated into a sequence of the one-unit problems simply by projecting the data-matrix onto the null-space of the previous solution as in the L2-PCA algorithms. Furthermore, absolute value function is non-differentiable. For these reasons, obtaining the exact solution is a rather challenging task. In [5] it was shown that, if $\mathbf{X}\mathbf{B}_{\text{opt}} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and

$$\mathbf{B}_{\text{opt}} = \underset{\mathbf{B}\in\{\pm 1\}^{n\times k}}{\operatorname{argmax}} \|\mathbf{X}\mathbf{B}\|_*, \tag{2}$$

where $\|.\|_*$ denotes nuclear norm, then $\mathbf{Q}_{\text{L1}} = \mathbf{U}\mathbf{V}^T$ is the optimal solution to (1). Therefore, the L1-norm maximization can be viewed as a combinatorial problem over the binary field. Unfortunately, the exhaustive search algorithm [5] has complexity $\mathcal{O}(n^{dk-k+1})$ and is difficult to use in practice. A faster, yet suboptimal, version of this approach is based on consecutive bit-flipping operations [6]. Its time complexity is of order $\mathcal{O}(nd\min\{n,d\} + n^2(k^4 + dk^2) + ndk^3)$, which can still be prohibitive for large data sizes. The most computationally efficient algorithm based on the fixed-point (FP) iterations was developed earlier in [4]. Unfortunately, it is rather inaccurate.

Recently, we proposed two L1-PCA algorithms [1] based on the Jacobi estimation framework. This framework is commonly used for diagonalizing symmetric matrices [3], [10] through the two-dimensional (plane) rotations. It also found applications in data-driven algorithms [11], [12] for iterative transformations of multi-dimensional data. It was shown in [1] that the Jacobi-based L1-PCA approaches provide high accuracy as compared to the existing suboptimal algorithms. They are also considerably faster than currently the most accurate method based on bit-flipping. In this paper, we propose to replace the conventional Jacobi rotations with higher-dimensional quaternion-based rotations. It is expected that, in this way, the convergence properties of an algorithm can be improved. A similar approach has been proposed in work [13] where we used quaternionic factorization of the 4×4 orthogonal matrices and Newton-Raphson iterative

This work was supported by Bialystok University of Technology under the grant WZ/WI-IIT/5/2023

scheme to solve the ICA problem. Here, we present a simpler approach based on three-dimensional rotations to solve the L1-norm maximization problem. When compared to our previous method [1], a novel algorithm offers a higher probability of finding a solution that is closer to the optimal one at the expense of the additional computational cost. However, our experiments show that for large datasets, this cost is still lower than that of the bit-flipping method.

II. Preliminaries on quaternions

A quaternion $Q \in \mathbb{H}$ can be represented using the rectangular form as follows [14]:

$$Q = q_0 + \boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3, \quad q_0, q_1, q_2, q_3 \in \mathbb{R}, \tag{3}$$

where $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ denote imaginary units. The real part of Q is q_0 and the pure quaternion part is $\boldsymbol{i}q_1 + \boldsymbol{j}q_2 + \boldsymbol{k}q_3$. The multiplication of quaternions is determined by the following rules:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1. \tag{4}$$

It is associative and distributes over vector addition, but it is not commutative.

When describing properties of the quaternions it is convenient to express them as the combination of a scalar part, $q_0 \in \mathbb{R}$, and a vector part, $\mathbf{q} = [q_1, q_2, q_3]^T \in \mathbb{R}^3$: $Q = [\![q_0, \mathbf{q}]\!]$. For example, the conjugate of Q can be written as $\bar{Q} = [\![q_0, -\mathbf{q}]\!]$, and the norm (modulus), is given by:

$$|Q| = \sqrt{q_0^2 + \|\mathbf{q}\|^2}. \tag{5}$$

For non-null quaternion, the inverse is defined as follows:

$$Q^{-1} = \bar{Q}|Q|^{-2}, \quad QQ^{-1} = Q^{-1}Q = 1. \tag{6}$$

Since

$$e^Q = \sum_{k=0}^{\infty} \frac{Q^k}{k!} = e^{q_0} \left[\!\!\left[\cos\|\mathbf{q}\|, \frac{\mathbf{q}}{\|\mathbf{q}\|} \sin\|\mathbf{q}\| \right]\!\!\right] \tag{7}$$

every quaternion Q can also be expressed in an exponential (polar) form:

$$Q = |Q|e^{\theta\mathbf{q}/\|\mathbf{q}\|} = |Q| \left[\!\!\left[\cos\theta, \frac{\mathbf{q}}{\|\mathbf{q}\|} \sin\theta \right]\!\!\right], \tag{8}$$

where $0 \leq \theta < 2\pi$ is an angle such that:

$$\cos\theta = \frac{q_0}{|Q|}, \quad \sin\theta = \frac{\|\mathbf{q}\|}{|Q|}. \tag{9}$$

This form is especially useful, because it allows us to express rotation in SO(3), The notation SO(d) denotes special orthogonal group in a d-dimensional Euclidean space, consisting all orthogonal matrices of determinant 1. Let $P = [\![0, \mathbf{p}]\!]$ be a pure quaternion that corresponds to a vector $\mathbf{p} \in \mathbb{R}^3$ and

$$U = u_0 + \boldsymbol{i}u_1 + \boldsymbol{j}u_2 + \boldsymbol{k}u_3 = \left[\!\!\left[\cos\frac{\theta}{2}, \mathbf{u}\sin\frac{\theta}{2} \right]\!\!\right], \tag{10}$$

be a unit-norm quaternion, where $\mathbf{u} = [u_1, u_2, u_3]^T \in \mathbb{R}^3$ is a unit vector indicating the direction of an axis of rotation, and an angle $0 \leq \theta < 2\pi$ is the magnitude of the rotation about the axis. Then the rotation of the vector $\mathbf{p}$ with an angle θ around a vector $\mathbf{u}$ can be expressed as follows:

$$P' = [\![0, \mathbf{p}']\!] = UPU^{-1} = UP\bar{U}. \tag{11}$$

Above operation can be expressed equivalently using matrix/vector multiplication by $\mathbf{p}' = \mathbf{R}(U)\mathbf{p}$, where:

$$\mathbf{R}(U) = \tag{12}$$

$$\begin{bmatrix} 1-2u_2^2-2u_3^2 & 2u_1u_2+2u_0u_3 & 2u_1u_3-2u_0u_2 \\ 2u_1u_2-2u_0u_3 & 1-2u_1^2-2u_3^2 & 2u_2u_3+2u_0u_1 \\ 2u_1u_3+2u_0u_2 & 2u_2u_3-2u_0u_1 & 1-2u_1^2-2u_2^2 \end{bmatrix},$$

is a rotation matrix. Theoretically, any rotation matrix can also be constructed using Euler angles, as a product of the three rotation matrices about the axes of the fixed coordinate system. Unfortunately, the Euler angles differing in many ways can give the same rotation matrix. In our case, this leads to multiple cost function calculations for the same point. Since the representation (10) corresponds almost uniquely to a given rotation matrix, such ambiguities can easily be avoided when working with quaternions.

III. Methods

A. Jacobi-based estimation framework

In the conventional Jacobi estimation framework [11], [1] the solution matrix is considered to be a product of the rotations in SO(2). These rotations are applied successively to the data matrix so that some objective function is optimized. For instance, in our previous work [1], the L1-norm metric is maximized as follows:

$$\mathbf{X}^{(t)} = \mathbf{G}(p_t, q_t, \theta_t)\mathbf{X}^{(t-1)}, \quad t = 1, 2, ..., \tag{13}$$

with $\mathbf{X}^{(0)} = \mathbf{W}\mathbf{X}$, where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is an arbitrary orthonormal matrix defining initialization point. The matrix $\mathbf{G}(p, q, \theta)$ represents Jacobi/Givens rotation [3] by the θ angle in the (p, q) plane, i.e.:

$$\mathbf{G}(p,q,\theta) = \begin{bmatrix} \mathbf{I}_{p-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cos\theta & \mathbf{0} & \sin\theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{q-p-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\sin\theta & \mathbf{0} & \cos\theta & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d-q-1} \end{bmatrix}, \tag{14}$$

where p, q are two integers such that such $1 \leq p < q \leq d$. Thus, the solution matrix $\hat{\mathbf{Q}}_{\mathrm{L1}} \in \mathbb{R}^{d \times k}$ is given by:

$$\hat{\mathbf{Q}}_{\mathrm{L1}} = \mathbf{W}^T \left[\overset{\curvearrowright}{\prod_t} \mathbf{G}(p_t, q_t, \theta_t)^T \right]_{*1:k}, \tag{15}$$

where $[\,.\,]_{*1:k}$ denotes the first k columns of an argument matrix. All possible rotations represented by pairs (p_t, q_t) are arranged in so-called sweeps. These sweeps are repeated cyclically until the maximum number of iterations is reached or when, for all rotations in the current sweep, we have $|\theta_t| \approx 0$. In fact, any rotation order is allowed [12], [15], but most frequently a row-cycling ordering is used as presented in Tab. I. For a fixed arrangement of the plane rotations, each

TABLE I: Row-cycling ordering for $d = 3$.

sweep no.	1			2			...
t	1	2	3	4	5	6	...
(p_t, q_t)	(1,2)	(1,3)	(2,3)	(1,2)	(1,3)	(2,3)	...

transformation in (13) depends on a single parameter θ_t, and hence d-dimensional optimization problem can be reduced to the sequence of $d(d-1)/2$ simpler one-dimensional subproblems per sweep. Let us denote by $\hat{x}_{ij}^{(t)}(\theta)$ the (i, j)-th entry of the data matrix (13) evaluated for the angle $\theta_t = \theta$. Then, the 'local' L1-norm maximization problem at tth rotation can be defined as follows:

$$\theta_t = \underset{-\pi/2 \le \theta < \pi/2}{\operatorname{argmax}} \sum_{\substack{i \in \{p_t, q_t\} \\ i \le k}} \sum_{j=1}^{n} \left| \hat{x}_{ij}^{(t)}(\theta) \right| . \tag{16}$$

Since we are interested in finding only the first k principal components, the outer summation range in (16) covers only indices less than or equal to k. In this way, the rotations that would have to be performed entirely in the null-space can simply be omitted. Also note that, the matrix (14) modifies only the rows p_t, q_t of the data matrix $\mathbf{X}^{(t-1)}$, so that the summation coefficients can be computed directly as

$$\hat{x}_{p_t j}^{(t)}(\theta) = x_{p_t j}^{(t-1)} \cos\theta + x_{q_t j}^{(t-1)} \sin\theta, \tag{17}$$

$$\hat{x}_{q_t j}^{(t)}(\theta) = x_{q_t j}^{(t-1)} \cos\theta - x_{p_t j}^{(t-1)} \sin\theta. \tag{18}$$

In work [1], we proposed two methods for solving (16). The first one performs exhaustive angle search, and the second one uses a differentiable approximation for absolute value function and calculates the rotation angles using the simplified Newton method. In this paper, we consider only the exhaustive algorithm due to its simplicity and high accuracy. Namely, the objective function in (16) is evaluated at the set of equidistant points, i.e.: $\{-\pi/2 + i\pi/m : i = 0, 1, ..., m-1\}$. We call this set the dictionary. The parameter m is an integer value controlling an angular resolution, i.e., the smallest non-zero angle that is used to represent rotation. Theoretically, greater the value of m, the higher angular resolution and better accuracy of the optimization. However, by increasing this value, we do not prevent the method from falling into local optima.

B. Proposed method

Key idea of the proposed method is to modify the Jacobi estimation framework by replacing rotations in SO(2) with rotations in SO(3). Please note that the conventional approach can guarantee a global convergence only for $d = 2$. For higher-dimensional problems, the Jacobi rotations are performed sequentially. Therefore, we may easily get trapped in local maximum due to non-convexity of the cost function. Similarly, rotations in SO(3) do not guarantee finding a global optimum for $d > 3$, however, such replacement can increase frequency with which the method finds an optimal solution. Furthermore, with the higher-dimensional rotations, more data samples are used when computing the local objective functions, and thus these functions should be smoother, which may result in a faster convergence. Namely, we propose to replace (14) with a quaternion based matrix:

$$\mathbf{R}(p, q, r, U) = \tag{19}$$

$$\begin{bmatrix} \mathbf{I}_{p-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & u_{11} & \mathbf{0} & u_{12} & \mathbf{0} & u_{13} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{q-p-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & u_{21} & \mathbf{0} & u_{22} & \mathbf{0} & u_{23} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{r-q-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & u_{31} & \mathbf{0} & u_{32} & \mathbf{0} & u_{33} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{d-r-1} \end{bmatrix},$$

where $1 \le p < q < r \le d$ and the coefficient u_{ij} is (i, j)-th entry of the matrix (12) computed for the quaternion:

$$U(\lambda, \phi, \theta) = \left[\!\left[\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\mathbf{u}(\lambda, \phi) \right]\!\right]. \tag{20}$$

Since the matrix (12) is orthogonal, the matrix (19) is also orthogonal. For convenience, the rotation axis is factorized using spherical coordinates, i.e.:

$$\mathbf{u}(\lambda, \phi) = [\cos\lambda \sin\phi,\ \sin\lambda \sin\phi,\ \cos\phi]^T, \tag{21}$$

where $0 \le \lambda < 2\pi$ and $0 \le \phi \le \pi$ denotes azimuthal and polar angle, respectively. More formally, our optimization problem can be stated as follows:

$$(\lambda_t, \phi_t, \theta_t) = \underset{\substack{0 \le \lambda < 2\pi \\ 0 \le \phi \le \pi \\ 0 \le \theta < 2\pi}}{\operatorname{argmax}} \sum_{\substack{i \in \{p_t, q_t, r_t\} \\ i \le k}} \sum_{j=1}^{n} \left| \hat{x}_{ij}^{(t)}(\lambda, \phi, \theta) \right|, \tag{22}$$

where $\hat{x}_{ij}^{(t)}(\lambda, \phi, \theta)$ denotes the (i, j)-th entry of transformed data matrix $\mathbf{R}(p_t, q_t, r_t, U)\mathbf{X}^{(t-1)}$. The coefficients $\hat{x}_{ij}^{(t)}(\lambda, \phi, \theta)$ for $i \in \{p_t, q_t, r_t\}$ can be stacked in the vector representing imaginary part of the following quaternion:

$$[\![0, \mathbf{x}_j^{(t)}(\lambda, \phi, \theta)]\!] = U(\lambda, \phi, \theta) X_j^{(t-1)} U(\lambda, \phi, \theta)^{-1}, \tag{23}$$

$$X_j^{(t-1)} = \left[\!\left[0, [x_{p_t j}^{(t-1)}, x_{q_t j}^{(t-1)}, x_{r_t j}^{(t-1)}]^T \right]\!\right]. \tag{24}$$

As before, the simplest solution to (22) is to use an exhaustive search method. Please note that there is not necessary to discretize the entire sphere because for a given vector (21) and an angle θ there is an opposite vector $-\mathbf{u}(\lambda, \phi)$ that generates the same rotation matrix for the angle $2\pi - \theta$. Let us denote by $\mathbb{A} = \{i\pi/m : i = 0, 1, ..., m-1\}$ and $\mathbb{B} = \{j\pi/m : j = 0, 1, ..., 2m-1\}$ the sets of equidistant points on interval $[0; \pi)$ and $[0; 2\pi)$, respectively. Then our spherical coordinate search dictionary can be defined as the following set:

$$\begin{aligned} \mathbb{D} = \{(\lambda, \phi, \theta) :\ & (\lambda, \phi, \theta) \in \mathbb{A} \times \mathbb{A} \times \mathbb{B} \wedge \\ & (\phi > 0 \vee \lambda = 0) \wedge (\phi = 0 \vee \theta > 0)\}, \end{aligned} \tag{25}$$

where × denotes Cartesian product. The condition in the second line removes from the set redundant coordinates and those for which the rotation matrix is equal to the identity matrix (for $\theta = 0$), except the point at the north pole.

TABLE II: Arrangements of rotation subspaces for various signal dimensionalities.

	$d=4$	
t	(p_t, q_t)	(p_t, q_t, r_t)
1	(1, 2)	(1, 2, 3)
2	(1, 3)	(1, 2, 4)
3	(1, 4)	(2, 3, 4)
4	(2, 3)	...
5	(2, 4)	...
6	(3, 4)	...

	$d=5$	
t	(p_t, q_t)	(p_t, q_t, r_t)
1	(1, 2)	(1, 2, 3)
2	(1, 3)	(1, 4, 5)
3	(1, 4)	(2, 4, 5)
4	(1, 5)	(3, 4, 5)
5	(2, 3)	...
6	(2, 4)	...
7	(2, 5)	...
8	(3, 4)	...
9	(3, 5)	...
10	(4, 5)	...

It can be verified that the cardinality of the set (25) is $l = 2m^3 - 3m^2 + 3m$. Obviously, for large m searching for the solution exhaustively may be unpractical, as the sequence length l grows rapidly with m. However, as we will show in the experimental section, the rotations in SO(3) can be represented with a much lower resolution than rotations in SO(2). In other words, the parameter m can be much smaller than that of the conventional Jacobi-based framework. Therefore, we can still use the exhaustive method at reasonable runtime.

Similarly to the conventional method, the consecutive rotations are organized in sweeps and repeated cyclically until convergence. However, since we deal with rotations in SO(3), a three-dimensional subspace must be defined for each rotation. Theoretically, for $d > 3$, the rotations can be performed in all possible subspaces defined as the 3-combinations of the row indices. However, we observed that to have convergence, not all combinations are needed. Namely, any combination (x, y, z) can be removed if all three pairs (x, y), (x, z) and (y, z) can also be found in other combinations. As presented in Tab. II, for $d = 4$ the combination $(1, 3, 4)$ was removed because there are the combinations $(1, 2, 3)$, $(1, 2, 4)$ and $(2, 3, 4)$ containing the pairs $(1, 3)$, $(1, 4)$ and $(3, 4)$, respectively. Such subsets of the combinations can also be obtained by joining plane rotations sharing the same dimensions. For example, two pairs (1,2) and (1,3) can be joined in the triple $(1, 2, 3)$, and the pair $(2, 3)$ can be removed as it is already present in the triple. In our simulations, we use this algorithm to generate arrangements presented in Tab. II for $d = 4, 5$. It can be verified that for $d \geq 3$ we have $n_r = \lceil d(d-2)/4 - \mathrm{mod}(d, 2) + 1 \rceil$ 3D rotations per sweep, where $\lceil . \rceil$ denotes ceiling operation. Similarly to Jacobi-based framework, there may exist other arrangements of the rotation subspaces, and some of them may be better than others. However, this issue is out of scope of this paper, and will be studied in a future work.

The pseudo-code of the proposed method is presented in Alg. 1. The sequence of triples defined in line 5 is defined according to the Tab. II. Please note that, at tth rotation, the matrix (19) modifies only the rows p, q, r. Therefore it is not necessary to compute it explicitly. In fact, only the matrices (12) are needed. Furthermore, they can be pre-computed for a given dictionary $\mathbb{D}$ once and used in subsequent iterations.

Algorithm 1 Pseudo-code of the proposed algorithm

Require:
$\mathbf{X} \in \mathbb{R}^{d\times n}, \mathbf{W} \in \mathbb{R}^{d\times d}, \mathbf{W}\mathbf{W}^T = \mathbf{I}_d, k \leq \mathrm{rank}(\mathbf{X}), \mathbb{D}$

Ensure: $\mathbf{Q} \in \mathbb{R}^{d\times k}, \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$

```
1:  $\mathbf{X}^{(0)} \leftarrow \mathbf{W}\mathbf{X}$
2:  $t \leftarrow 1$
3:  for $sweepNum = 1 : maxSweepNum$ do
4:      $encore \leftarrow 0$
5:      for $(p, q, r) = \{(1, 2, 3), ...\}$ do
6:          $(\lambda_t, \phi_t, \theta_t) = \underset{(\lambda,\phi,\theta)\in\mathbb{D}}{\mathrm{argmax}} \sum_{\substack{i\in\{p,q,r\} \\ i\leq k}} \sum_{j=1}^{n} \left|\hat{x}_{ij}^{(t)}(\lambda, \phi, \theta)\right|$
7:          $\mathbf{R}_{\text{opt}} \leftarrow \mathbf{R}(p, q, r, U(\lambda_t, \phi_t, \theta_t))$
8:          $\mathbf{W} \leftarrow \mathbf{R}_{\text{opt}}\mathbf{W}$
9:          $\mathbf{X}^{(t)} \leftarrow \mathbf{R}_{\text{opt}}\mathbf{X}^{(t-1)}$
10:         $t \leftarrow t + 1$
11:         if $\mathbf{R}_{\text{opt}} \neq \mathbf{I}$ then $encore \leftarrow 1$
12:     if $encore = 0$ then break
13: $\mathbf{Q} \leftarrow [\mathbf{W}^T]_{*1:k}$
```

IV. Experiments

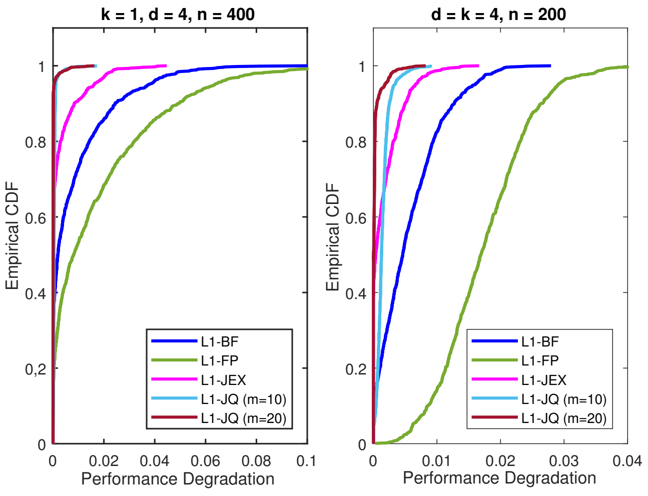
The proposed method has been implemented and evaluated in the Matlab environment. For convenience, it was denoted as L1-JQ, which stands for Jacobi method with quaternion rotations. For comparative purposes we also evaluated three other approximate methods for maximization of the L1-norm: bit-flipping algorithm [6] (L1-BF), fixed-point iterations [4] (L1-FP), and Jacobi exhaustive method with SO(2) rotations [1] (L1-JEX). In the case of the L1-JEX method, the parameter m was set to 512. We verified empirically that, for this method, rotations at a smaller angle than $\pi/512$ are not statistically significant. In order to explore how the estimation error is affected by the angular resolution, the algorithm L1-JQ was evaluated for two different values of the parameter $m \in \{10, 20\}$. For all approaches, the identity matrix was used as the initialisation point.

A. Accuracy

The performance degradation ratio attained by the algorithms was measured using a similar procedure to that in [6]. Namely, the following metric was considered:

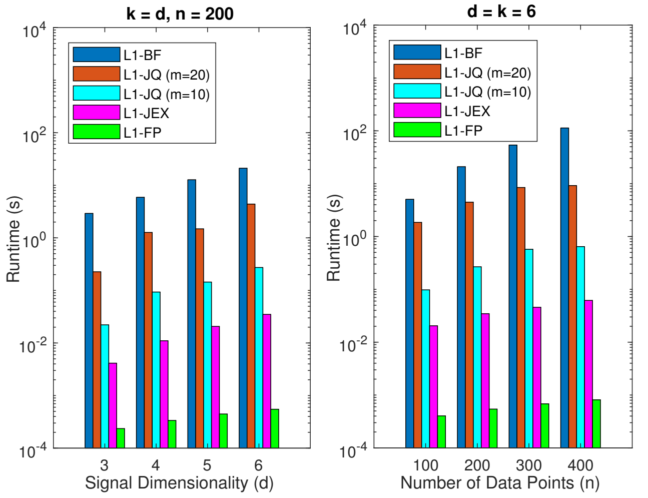
$$\Delta(\mathbf{Q}, \mathbf{X}) = \frac{\|\mathbf{X}^T\mathbf{Q}_{\text{L1}}\|_1 - \|\mathbf{X}^T\mathbf{Q}\|_1}{\|\mathbf{X}^T\mathbf{Q}_{\text{L1}}\|_1}, \tag{26}$$

where $\mathbf{Q}$ is the orthonormal matrix estimated using an evaluated method. Ideally, the matrix $\mathbf{Q}_{\text{L1}}$ should be the matrix obtained by an optimal L1-PCA algorithm [5], for the same data matrix. Unfortunately, the computational complexity of the optimal method is extremely high. Thus, such an approach is possible only for very small data sizes ($n \ll 100$). The presented method is intended for larger data sets. For these reasons, in this experiment, we replaced the matrix $\mathbf{Q}_{\text{L1}}$ with the matrix representing the best solution among all methods. In order to measure which method gives the best result most



(a) (b)

Fig. 1: Empirical CDF of performance degradation ratio estimated for various L1-PCA algorithms



(a) (b)

Fig. 2: Comparison of average runtimes (in seconds) measured for all methods and various data sizes across 100 Monte Carlo runs. (a) Runtime vs signal dimensionality. (b) Runtime vs number of observation samples.

frequently, the empirical cumulative distribution functions (ECDFs) were computed. The ECDFs are the fractions of the measurements (26) that are less than or equal to the specified values. Thus, the higher the value of the ECDF, the better accuracy. We considered two scenarios: the first one with $d = 4$, $k = 1$, $n = 400$, and the second one with $d = k = 4$, $n = 200$. In both scenarios, 1000 random data matrices were randomly generated with entries drawn independently from a Gaussian distribution $\mathcal{N}(0, 1)$ as in [6], [1]. The results are presented in Fig. 1. In the first scenario (on the left), the proposed approach with $m = 20$ gives a zero or close to zero value of the degradation ratio in about 95 percent of runs. This is the best score among all methods. In the case of the L1-JEX, L1-BF and L1-FP methods, these fractions are 65, 30, and 15 percent, respectively. The performance loss due to smaller dictionary size is rather not noticeable in this scenario. Both versions of the proposed method perform equally well regardless of the value of the parameter m. In the second scenario (Fig. 1b), we see that each method attains the best result less frequently. However, once again, the L1-JQ algorithm for $m = 20$ achieves lower values of the metric (26) more frequently than any other method. It gives the best or close to best result in about 75 percent of runs, while for the L1-JEX and L1-BF methods, these frequencies are 40 and 10 percent, respectively. The most significant performance loss can be seen for the L1-FP method. It comes from the fact that the L1-FP method for $k > 1$ is based on successive null-space projections that violate the non-scalability principle of the L1-PCA. We also see that the reduction in accuracy of the L1-JQ method due to the decrease in a value of the parameter m is more prominent. This reduction is especially noticeable in a frequency with which the method obtains the best solution. Please note that even if the proposed approach does not give the best solution most frequently, the metric (26) usually takes relatively small values. Namely, the degradation ratio attained by L1-JQ method with $m = 10$, computed with respect to the best solution, is with empirical probability 1 less than 0.008. Other methods attain significantly greater values of the degradation ratio. For example, the largest values of the metric (26) returned by the L1-JEX and L1-FB methods were 0.016 and 0.027 respectively.

B. Execution time

In order to compare the computational performance of the proposed algorithm with other methods, we measured their average execution times for various data sizes. The experiments were carried out on the system with AMD Ryzen 5 3550H processor. Once again, two scenarios have been considered. In the first one (Fig. 2a), we examined how the dimensionality of the signal affects the computation time. Here, we assumed that the number of data samples $n = 200$ and $k = d$. It can be seen that even for the higher angular resolution ($m = 20$), the proposed method is a faster than L1-BF algorithm. On the other hand, it is slower than the method based on SO(2) rotations, even when the angular resolution is low ($m = 10$). None of the methods can compete with the L1-FP method, which turns out to be the fastest approach. Also note that the execution time of the all rotational methods increases quite fast with the dimension number. It is not surprising, as the number of rotations per sweep increases quadratically with d.

In the second scenario (see Fig. 2b), we assumed that $k = d = 6$ and checked how the computation time is affected by the number of samples n. The results are similar to that of the previous case. It can be seen that all rotational methods are generally faster than the L1-BF algorithm. In addition their execution times increase linearly with the number of samples. This is serious improvement compared to the L1-BF algorithm, where the execution time increases quadratically with n. Our experiments clearly show that the precision of the

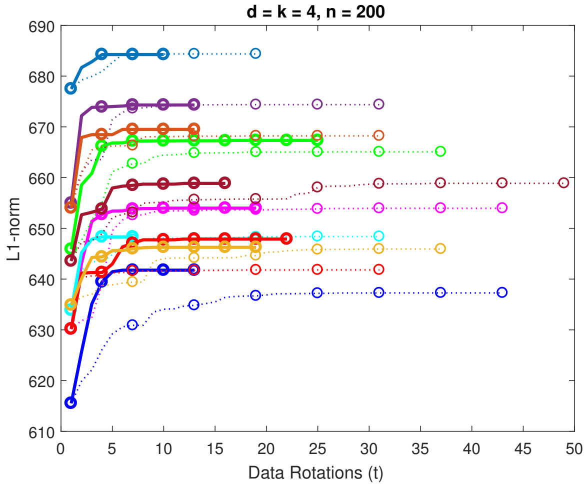


Fig. 3: Convergence curves obtained during 10 Monte Carlo runs for the L1-JQ method with $m = 20$ (solid lines) and the L1-JEX algorithm (dotted lines). Each Monte Carlo run is presented using different colour. The first data rotation in each sweep is depicted by circle.

L1-JQ method is paid for by increased computational burden. Nonetheless, its computational complexity can be still smaller than or at least comparable to that of the L1-BF algorithm. Also note that the rotational algorithms, including the proposed one, may not be the best choice for large d. For instance, when $d > n$, the L1-BF algorithm may offer better performance. However, a such scenario is rarely encountered in practice and thus less interesting.

In Fig. 3, we also show the convergence curves obtained for L1-JQ ($m = 20$) and L1-JEX methods. The L1-norm was measured after each data rotation for 10 independent Monte Carlo runs. The maximum number of sweeps was limited to 100, but none of the methods reached this limit. As we see, both methods converge in a relatively small number of sweeps (from 2 to 8), but the proposed method offers higher convergence rates. It also achieves higher values of the L1-norm more frequently, which is consistent with our previous findings. On the other hand, the computational cost of the single rotation of the proposed method is higher than that of the L1-JEX method. Thus, in overall, the L1-JEX method remains computationally more efficient.

V. Conclusion

In this paper, we proposed a novel version of the exhaustive Jacobi-based algorithm for maximization of the L1-norm. It was shown that the Jacobi rotations can be replaced by the quaternion-based rotations in SO(3). In this way, it is possible to increase the accuracy of the estimation at the expense of additional computational cost. Indeed, the simulation results show that the proposed method gives the best solution more frequently than other approximate methods. Although the algorithm was implemented using exhaustive search, the improvement in the accuracy was obtained for relatively small dictionary size. The results suggest that precision of angular representation of the rotations in higher-dimensions can be substantially lower than that of the two-dimensional rotations. Thus, a solution can be found exhaustively at a reasonable computational cost. Furthermore, for large datasets, the execution time of the proposed method is still smaller than that of the bit-flipping technique.

Future works include implementation optimizations, practical applications and more rigorous estimation error analysis. It could be especially interesting to establish the theoretical bounds of estimation error with respect to the precision of angular representation of the rotations.

References

[1] A. Borowicz, "Maximization of L1-norm using Jacobi rotations," in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022. doi: 10.23919/EUSIPCO55093.2022.9909924 pp. 1951–1955.

[2] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer Verlag, 2002.

[3] G. Golub and C. Van Loan, *Matrix Computations*. USA: Johns Hopkins University Press, 2013.

[4] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008. doi: 10.1109/TPAMI.2008.114

[5] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L_1-subspace signal processing," *IEEE Transactions on Signal Processing*, vol. 62, no. 19, pp. 5046–5058, 2014. doi: 10.1109/TSP.2014.2338077

[6] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4252–4264, 2017. doi: 10.1109/TSP.2017.2708023

[7] M. Dhanaraj and P. P. Markopoulos, "Novel algorithm for incremental L1-norm principal-component analysis," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018. doi: 10.23919/EUSIPCO.2018.8553239 pp. 2020–2024.

[8] R. Martın-Clemente and V. Zarzoso, "On the link between L1-PCA and ICA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 515–528, 2017. doi: 10.1109/TPAMI.2016.2557797

[9] A. Borowicz, "Independent component analysis based on Jacobi iterative framework and L1-norm criterion," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022. doi: 10.15439/2022F157 pp. 305–312.

[10] J. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 6, pp. 362–370, 1993. doi: 10.1049/ip-f-2.1993.0054

[11] J. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999. doi: 10.1162/089976699300016863

[12] W. Ouedraogo, A. Souloumiac, and C. Jutten, "Non-negative independent component analysis algorithm based on 2D Givens rotations and a Newton optimization," in *Latent Variable Analysis and Signal Separation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-15995-4 pp. 522–529.

[13] A. Borowicz, "Orthogonal approach to independent component analysis using quaternionic factorization," *EURASIP Journal on Advances in Signal Processing*, vol. 2020, no. 39, p. 23, September 2020. doi: 10.1186/s13634-020-00697-0

[14] J. B. Kuipers, *Quaternions and Rotation Sequences: a Primer with Applications to Orbits, Aerospace, and Virtual Reality*. Princeton, N.J.: Princeton University Press, 2002.

[15] M. Parfieniuk, "A parallel factorization for generating orthogonal matrices," in *International Conference on Parallel Processing and Applied Mathematics (PPAM) 2019*. Bialystok, Poland: Springer, 2019. doi: 10.1007/978-3-030-43229 pp. 567–578.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 889–894

DOI: 10.15439/2023F4794
ISSN 2300-5963 ACSIS, Vol. 35

Efficient Deep Learning Approach for Olive Disease Classification

Antonio Bruno
Institute of Information Science
and Technologies
National Research Council
Via Moruzzi 1, Pisa, Italy
Email: antonio.bruno@isti.cnr.it

Davide Moroni
Institute of Information Science
and Technologies
National Research Council
Via Moruzzi 1, Pisa, Italy
Email: davide.moroni@isti.cnr.it

Massimo Martinelli
Institute of Information Science
and Technologies
National Research Council
Via Moruzzi 1, Pisa, Italy
Email: massimo.martinelli@isti.cnr.it

***Abstract*—From ancient times olive tree cultivation has been one of the most crucial agricultural activities for Mediterranean countries. In recent years, the role of Artificial Intelligence in agriculture is increasing: its use ranges from monitoring of cultivated soil, to irrigation management, to yield prediction, to autonomous agricultural robots, to weed and pest classification and management, for example, by taking pictures using a standard smartphone or an unmanned aerial vehicle , and all this eases human work and makes it even more accessible.**

In this work, a method is proposed for olive disease classification, based on an adaptive ensemble of two EfficientNet-b0 models, that improves the state-of-the-art accuracy on a publicly available dataset by 1.6-2.6%. Both in terms of the number of parameters and the number of operations, our method reduces complexity roughly by 50% and 80%, respectively, that is a level not seen in at least a decade. Due to its efficiency, this method is also embeddable into a smartphone application for real-time processing.

I. Introduction

OLIVE tree cultivation represents one of the most important activities of agriculture for the civilizations of the Mediterranean area. Indeed the countries of this area produced roughly 65% of the world's olive oils in the last years [1]. Olive-derived products have shown health benefits due to their compounds [2]. In addition, olive trees are known to adapt to environmental stresses such as salinity, drought, heat and high levels of ultraviolet B rays [3], [4], [5] generating, during the millennia, 600 species within 25 genera [6]. However, even olive trees are affected by diseases: some of them are visible on their fruits and can happen only during specific periods of the year, while others have visible signs on the leaves [7]. The signs of a disease can be different in different hosts and can evolve over time.

Although olive cultivation techniques have been perfected over the centuries, artificial intelligence has only recently entered the olive industry, bringing a series of significant innovations and improving the management of many issues, like as predicting crop yields, plant health monitoring, disease prevention, identification and classification, irrigation management, monitoring and management of agricultural activities [8], [9], [10] (e.g. sowing, harvesting, pruning,...), even for olive disease [11], [12]. We propose here a highly efficient solution that allows to classify olive diseases affecting leaves directly from images taken by standard smartphone cameras. This paper is organized as follows: in Sec. II, the dataset used for experiments is described; in Sec. III, the solutions and the experimental setup are described, while results are shown in Sec. IV. The paper ends with a discussion and conclusion in Sec. VI.

II. Dataset description

To test our solution, the largest publicly available dataset [13] has been used: it is composed of 3400 images representing olive leaves affected by *Alucus olearius* or *Olive peacock spot* or *healthy*. Tab. I shows the distribution of the classes, while a sample of images for each class is shown in Fig. 1.

TABLE I: Data distribution of the dataset used.

Class	Size
Aculus olearius	890
Healthy	1050
Olive peacock Spot	1460

III. Design description

A. *EfficientNet*

We selected EfficientNet-b0 [14] as the core model because, according to its structure and the obtained results, it has the best accuracy/complexity trade-off. Two main factors give the efficiency of this architecture: the first is the compound scaling (Fig. 2) by which input scaling (i.e. input size), width scaling (i.e. convolutional kernel size) and depth scaling (i.e. the number of layers) are performed in conjunction since, by observation, they are dependent; the second is the use of the inverted bottleneck MBConv (first introduced in MobileNetV2, an efficient model designed to run on smartphones) as a main module, reducing the complexity of convolution by expanding and compressing the channels.

B. *Ensembling*

The most significant contribution to this work is given by ensembling: it is a technique of combining several models, called *weak* models, in order to provide produce a model

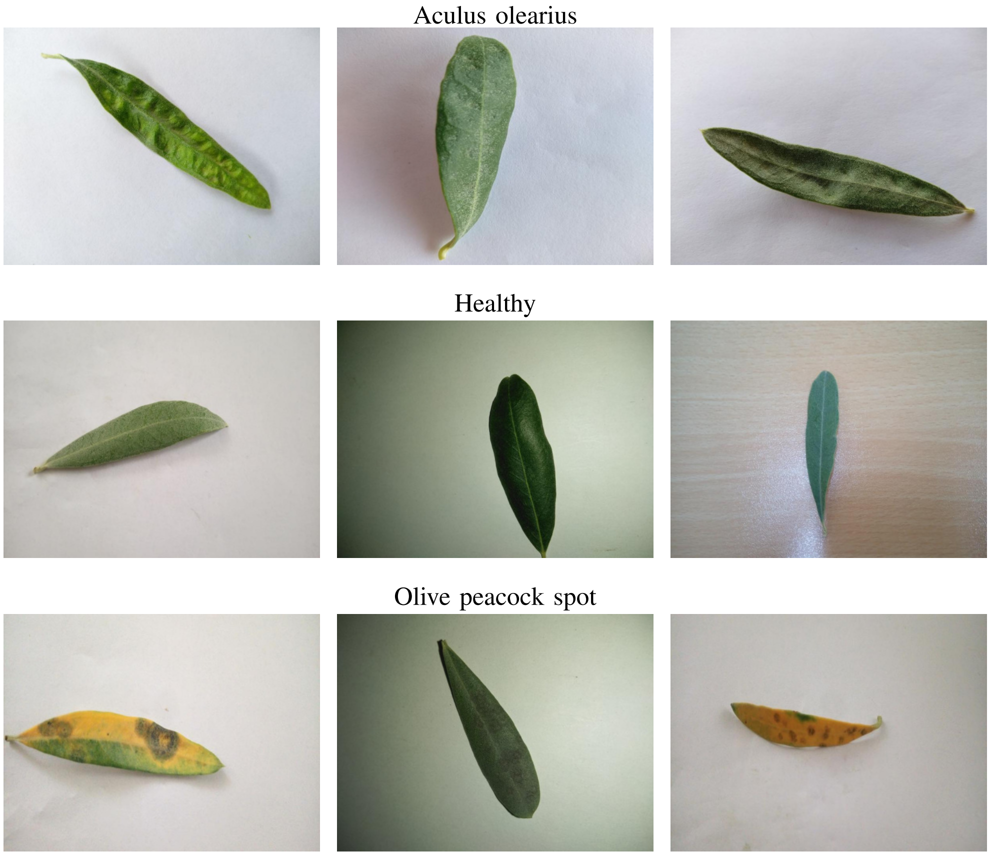


Fig. 1: Samples from the dataset for Aculus olearius (first row), Healthy (second row) and Olive peacock spot (third row)

having better results than a single one [15]. Ensembling is also known to reduce errors and improve the model's generalization capabilities. Due to its resource-consuming nature and the exponential growth of model complexity, however, ensembling is scarcely used in computer vision. By contrast, our method allows performing ensembling in an adaptive and efficient way (Fig. 3):

- we use only two weak models (achieving minimality and efficiency);
- the ensemble is not a typical aggregation function, but it is performed using a linear combination layer, trainable by gradient descent (obtaining adaptivity);
- the ensemble is performed using the deep features instead of the output, excluding redundant operations (for efficiency).

C. *Validation pipeline*

The validation pipeline can be split into two main phases:

1) 5-fold cross-validation with end-to-end EfficientNet-b0 training, using transfer learning [16] from ImageNet pretrained models [17], because transfer learning provides faster convergence;
2) 5-fold cross-validation with fine-tuning of the ensemble, using the two best models from the previous phase.

The design choices used during the validation are:

Input size: set to 512×512 because, after a preliminary investigation, it gives the best trade-off between image quality and computational costs.

Batch size: set to the maximum available using our GPU (32GB RAM), which is 50 for the end-to-end and 200 for the fine-tuning.

Regularization: early-stopping with patience of 10 epochs is used, helping to prevent overfitting.

Optimizer: AdaBelief [18] with learning rate $5 \cdot 10^{-4}$, betas (0.9, 0.999), eps 10^{-16}, using weight decoupling without rectifying, in order to have both fast convergence and generalization.

Validation metric: Weighted F1-score which better takes into account both errors and data imbalance.

Dataset split: training and test subsets are preset, in every run of the 5-fold cross-validation, the training set is split 80/20 in train/valid.

Standardization: data are processed in order to belong to a distribution with values around the average and the unit standard deviation, improving stability and convergence of the training.

Obviously, each run of the cross-validation of both phases is associated with a different initialization of the random model

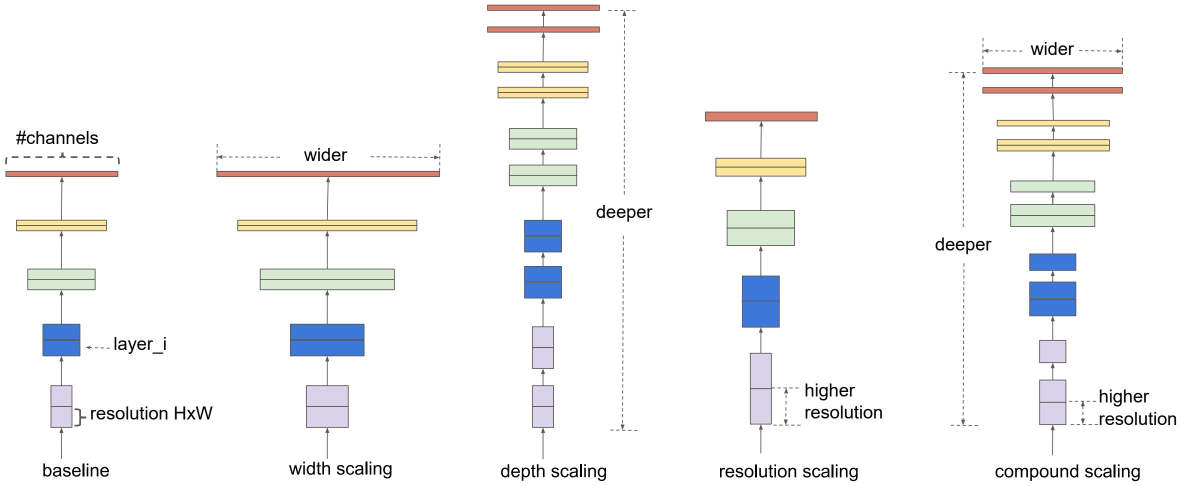


Fig. 2: Example of scaling types, from left to right: a baseline network example, conventional scaling methods that only increase one network dimension (width, depth, resolution) and, at the end, the EfficientNet compound scaling method. Image taken from the original paper [14].

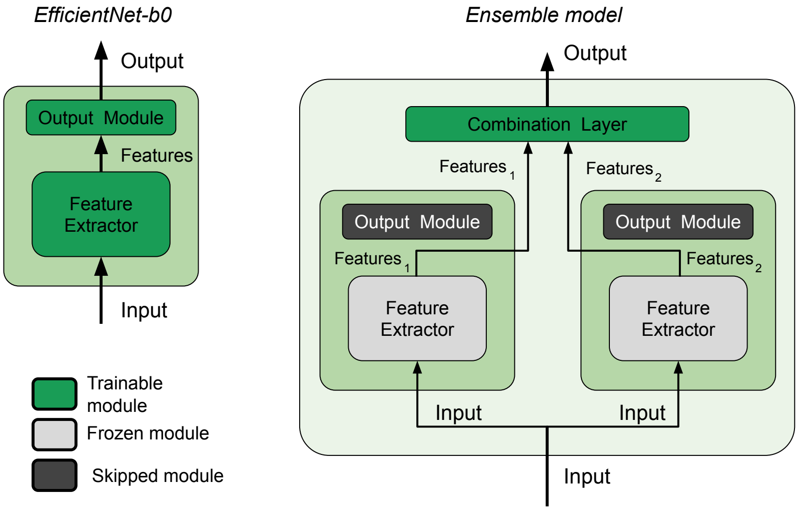


Fig. 3: Graphical scheme of the models used in this work: on the left, an end-to-end trainable EfficientNet-b0; on the right, the fine-tunable adaptive ensemble.

parameters.

IV. Experimental results

According to Tab. II, the EfficientNet-b0 with the selected design choices already provides a good starting point with an average F1-score of 0.969, 0.983 and 0.999 for test, valid and train set, respectively, with high robustness (i.e. low variance). The ensemble further reduces the variance and improves the generalization power (i.e. performance on valid and test) by an average of +1.5% and +1.4% on test and valid, respectively. The final errors are 12 (test: 9; valid: 2; train: 1) and in Fig. 4 the confusion matrix for the test set is shown.

The strength of the proposed solution is even more significant when compared with the State of the Art (SOTA) Tabs. III-IV, indeed the EfficientNet-b0 has the values of the same metric as the best performing SOTA model, and it uses only 52% of parameters and 21% of FLOPs, while considering the Ensemble the complexity (both parameters and FLOPs) is roughly doubled, but it is still lower than the SOTA, for a +1.6% on all the metrics.

V. Discussion

In order to stress our method, we tested an ensemble of five weak models: while using other datasets, generally this improves the results a little at the expense of complexity, as Tab. V shows, in this case, the results don't improve, the errors remain exactly on the same 12 images even if distributed among the different splits.

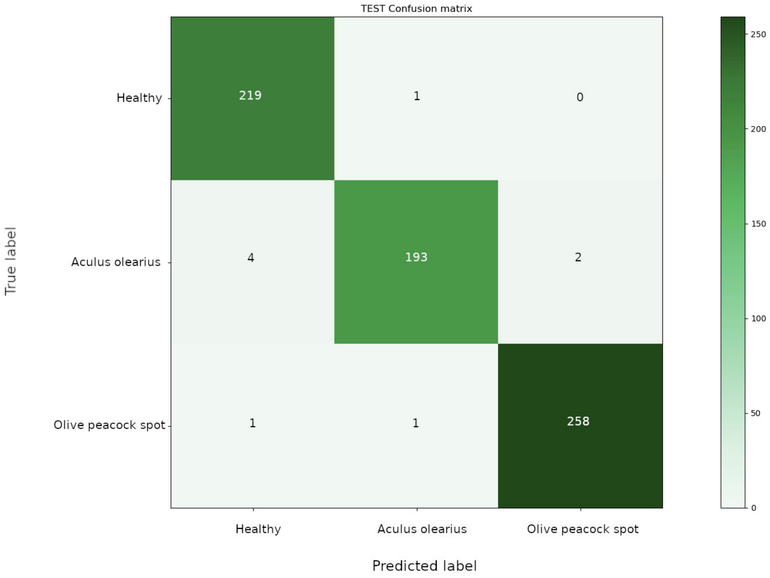


Fig. 4: The confusion matrix on the test split of the best ensemble model.

TABLE II: Metrics (F1-score) on the subset of 5-fold cross-validation runs of both end-to-end weak (left) and fine-tuning ensemble models (right). The ensemble has a twofold contribution: improving generalization performances (+1.5% on test, +1.4% on valid, on average) and robustness (halving the deviation). Data is organized best-to-worst fold (top-to-bottom), and then the models corresponding to the first two rows in the left table are used as weak models for the ensemble.

	Weak		
	Test	**Valid**	**Train**
	0.97206	0.98713	1.00000
	0.97203	0.98534	0.99724
	0.96925	0.97973	0.99862
	0.96777	0.98159	1.00000
	0.96620	0.98529	1.00000
Mean	*0.96946*	*0.98382*	*0.99917*
Std	*0.00231*	*0.00273*	*0.00110*

	Ensemble		
	Test	**Valid**	**Train**
	0.98676	0.99632	0.99954
	0.98382	0.99816	0.99862
	0.98382	0.99816	0.99862
	0.98382	0.99632	1.00000
	0.98382	0.99632	0.99954
Mean	*0.98441*	*0.99706*	*0.99926*
Std	*0.00118*	*0.00090*	*0.00055*

TABLE III: Comparing metrics of the SOTA models. Since, in their papers, the authors did not mention if the values refer either as mean/best or on test only/whole dataset, we reported the mean values (best in brackets) on both test only and whole dataset.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
VGG-19[19]	0.82	0.75	0.94	0.84
AlexNet[19]	0.84	0.86	0.87	0.86
VGG-16[19]	0.85	0.87	0.86	0.87
AlexNet (genetic)[20]	0.87	0.87	0.87	0.87
ViT Transformer[19]	0.95	0.94	0.98	0.96
DenseNet (genetic)[20]	0.96	0.97	0.96	0.96
ViT+VGG-16[19]	0.96	0.97	0.96	0.96
ResNet (genetic)[20]	0.97	0.97	0.97	0.97
EfficientNet-b0 (test)	0.970 (0.972)	0.970 (0.972)	0.969 (0.972)	0.969 (0.972)
EfficientNet-b0 (whole)	0.990 (0.992)	0.990 (0.992)	0.990 (0.992)	0.990 (0.992)
Ensemble-b0 (test)	0.986 (0.987)	0.985 (0.987)	0.986 (0.987)	0.985 (0.987)
Ensemble-b0 (whole)	0.996 (0.996)	0.996 (0.996)	0.996 (0.996)	0.996 (0.996)

TABLE IV: Comparing complexity (expressed by the number of parameters and FLOPs) of the SOTA models.

Model	#params	FLOPs
VGG-19	≈143.6M	≈19.63G
AlexNet	≈61.1M	≈0.71G
VGG-16	≈138.3M	≈15.47G
ViT Transformer[1]	≈88.2M	≈4.41G
DenseNet[2]	≈7.9M	≈2.83G
ViT+VGG-16	≈226.5M	≈19.88G
ResNet[3]	≈11.6M	≈1.81G
EfficientNet-b0	≈5.2M	≈0.39G
Ensemble-b0	≈10M[4]	≈0.78G[5]

[1] authors did not specify the version they used, metrics are about the lightest one (ViT-B-32).

[2] authors did not specify the version they used, metrics are about the lightest one (DenseNet-121).

[3] authors did not specify the version they used, metrics are about the lightest one (ResNet-18).

[4] the actual trainable parameters are 0.1M (the parameters of the combination layer) and the gradient backward propagation stops at this layer.

[5] the forward pass can be parallelized, having the same execution time of a weak model.

TABLE V: Metrics (F1-score) related to the best ensembles of five weak models.

Test	Valid	Train
0.98529	1.00000	0.99908
0.98529	1.00000	0.99908
0.98235	1.00000	0.99908
0.98235	1.00000	0.99908
0.98235	1.00000	0.99908

This approach to ensembling has been recently introduced and discussed in [21], [22]; it has already proved excellent applicability to AI-based methods for agriculture [23].

Specifically in [21], we tested our method on seven benchmarking datasets, that are: CIFAR-10 [24], CIFAR-100 [24], Stanford Cars [25], Food-101 [26], Oxford 102 Flower [27], CINIC-10 [28] and Oxford-IIIT Pet [26]. The results demonstrated that our novelties improve the SOTA for each dataset by an average of 0.5%, using different kinds of images, reducing complexity in terms of the number of parameters up to sixty times and of FLOPs up to one hundred times. This results in a considerable saving of time and costs compared to most recent models (i.e. Vision Transformers [29]).

In [30], our method was also tested on images of plants taken on the field, in different environments, backgrounds, light conditions and at different stages of growth of the weeds. This defined the baseline for an in-progress work, in which, with the help of farmers taking pictures directly on the field using a mobile app [31], a set of models trained and being continuously extended, are contributing to significantly improving the classification of about a hundred of the main stressors that can interfere with wheat cultivation, such as weeds, pests, diseases and damages.

Another real-world application using this solution on a different domain was presented and discussed in [22]: using a public database of lung ultrasound, the SOTA was reached with 100% of accuracy in classifying healthy from Covid-19 from pneumonia cases.

VI. Conclusions

In this paper, we presented an efficient adaptive ensemble method to classify olive leaf diseases using two EfficientNet-b0 as weak models. The ensemble is performed by a linear layer that combines the features of the weak models. Our method increased the generalization strenghtby about 1.5% and reduced the variance. Moreover, by parallelizing the independent weak models, the complexity is comparable to a single weak model, having 52% of parameters and 21% of FLOPs of the best SOTA solution.

Due to its efficiency, given a significantly smaller architecture in terms of the number of tunable parameters and floating point operations comparable to those of a decade ago, this solution can also be embedded into a smartphone application for real-time classifications.

Further studies will be performed to investigate the use of the efficient adaptive ensemble method with a greater number of weak models.

References

[1] J. Blázquez, "The origin and expansion of olive cultivation," *World Olive Encyclopaedia, Madrid*, pp. 19–20, 1996.

[2] S. Valente, B. Machado, D. C. Pinto, C. Santos, A. M. Silva, and M. C. Dias, "Modulation of phenolic and lipophilic compounds of olive fruits in response to combined drought and heat," *Food chemistry*, vol. 329, p. 127191, 2020.

[3] C. Brito, L.-T. Dinis, J. Moutinho-Pereira, and C. M. Correia, "Drought stress effects and olive tree acclimation under a changing climate," *Plants*, vol. 8, no. 7, p. 232, 2019.

[4] S. Silva, C. Santos, J. Serodio, A. M. Silva, and M. C. Dias, "Physiological performance of drought-stressed olive plants when exposed to a combined heat–uv-b shock and after stress relief," *Functional Plant Biology*, vol. 45, no. 12, pp. 1233–1240, 2018.

[5] L. Regni, A. M. Del Pino, S. Mousavi, C. A. Palmerini, L. Baldoni, R. Mariotti, H. Mairech, T. Gardi, R. D'Amato, and P. Proietti, "Behavior of four olive cultivars during salt stress," *Frontiers in plant science*, vol. 10, p. 867, 2019.

[6] H. K. Obied, P. D. Prenzler, D. Ryan, M. Servili, A. Taticchi, S. Esposto, and K. Robards, "Biosynthesis and biotransformations of phenol-conjugated oleosidic secoiridoids from olea europaea l." *Natural product reports*, vol. 25, no. 6, pp. 1167–1179, 2008.

[7] A. Graniti, R. Faedda, S. O. Cacciola, and G. M. di San Lio, "19. olive diseases in a changing ecosystem," 2011.

[8] L. Benos, A. C. Tagarakis, G. Dolias, R. Berruto, D. Kateris, and D. Bochtis, "Machine learning in agriculture: A comprehensive updated review," *Sensors*, vol. 21, no. 11, 2021. doi: 10.3390/s21113758. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3758

[9] P. Lameski, E. Zdravevski, V. Trajkovik, and A. Kulakov, "Weed detection dataset with rgb images taken under variable light conditions," in *ICT Innovations 2017*, D. Trajanov and V. Bakeva, Eds. Cham: Springer International Publishing, 2017. ISBN 978-3-319-67597-8 pp. 112–119.

[10] P. Lameski, E. Zdravevski, and A. Kulakov, "Weed segmentation from grayscale tobacco seedling images," in *Advances in Robot Design and Intelligent Control*, A. Rodić and T. Borangiu, Eds. Cham: Springer International Publishing, 2017. ISBN 978-3-319-49058-8 pp. 252–258.

[11] A. Sinha and R. S. Shekhawat, "Olive spot disease detection and classification using analysis of leaf image textures," *Procedia Computer Science*, vol. 167, pp. 2328–2336, 2020. doi: https://doi.org/10.1016/j.procs.2020.03.285 International Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920307511

[12] S. Uğuz and N. Uysal, "Classification of olive leaf diseases using deep convolutional neural networks," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4133–4149, May 2021. doi: 10.1007/s00521-020-05235-5. [Online]. Available: https://doi.org/10.1007/s00521-020-05235-5

[13] "Olive dataset kernel description," https://github.com/sinanuguz/CNN_olive_dataset, accessed: 2023-05-20.

[14] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, Jun. 2019, pp. 6105–6114.

[15] D. W. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, 1999. doi: 10.1613/jair.614. [Online]. Available: https://doi.org/10.1613/jair.614

[16] K. Weiss, T. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, 05 2016. doi: 10.1186/s40537-016-0043-6

[17] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. doi: 10.1109/CVPR.2009.5206848 pp. 248–255.

[18] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients," *Conference on Neural Information Processing Systems*, 2020.

[19] H. Alshammari, G. Karim, I. Ben Ltaifa, M. Krichen, L. Ben Ammar, and M. Mahmood, "Olive disease classification based on vision transformer and cnn models," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–10, 07 2022. doi: 10.1155/2022/3998193

[20] H. Alshammari, G. Karim, M. Krichen, L. Ben Ammar, M. Eltaib, A. Boukrara, and M. Mahmood, "Optimal deep learning model for olive disease diagnosis based on an adaptive genetic algorithm," *Wireless Communications and Mobile Computing*, vol. 2022, 03 2022. doi: 10.1155/2022/8531213

[21] A. Bruno, D. Moroni, and M. Martinelli, "Efficient adaptive ensembling for image classification," *accepted for publication by Expert Systems - Wiley on 31st July 2023, arXiv preprint arXiv:2206.07394*, 2023.

[22] A. Bruno, G. Ignesti, O. Salvetti, D. Moroni, and M. Martinelli, "Efficient lung ultrasound classification," *Bioengineering*, vol. 10, no. 5, 2023. doi: 10.3390/bioengineering10050555. [Online]. Available: https://www.mdpi.com/2306-5354/10/5/555

[23] A. Bruno, D. Moroni, R. Dainelli, L. Rocchi, S. Morelli, E. Ferrari, P. Toscano, and M. Martinelli, "Improving plant disease classification by adaptive minimal ensembling," *Frontiers in Artificial Intelligence*, vol. 5, p. 868926, 2022.

[24] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR 2021: The Ninth International Conference on Learning Representations*, 2021.

[25] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," 2021.

[26] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[27] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," 2021.

[28] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V. N. Boddeti, "Neural architecture transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, p. 2971–2989, Sep 2021. doi: 10.1109/tpami.2021.3052758

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[30] R. Dainelli, M. Martinelli, A. Bruno, D. Moroni, S. Morelli, M. Silvestri, E. Ferrari, L. Rocchi, and P. Toscano, *Recognition of weeds in cereals using AI architecture*, ch. 49, pp. 401–407. [Online]. Available: https://www.wageningenacademic.com/doi/abs/10.3920/978-90-8686-947-3_49

[31] M. Massimo, "Agrosat+ project," 2023, http://si.isti.cnr.it/index.php/hid-notcategorized-category-list/228-barilla [Accessed: 31st July 2023].

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 895–900
DOI: 10.15439/2023F3506
ISSN 2300-5963 ACSIS, Vol. 35

The scalability in terms of the time and the energy for several matrix factorizations on a multicore machine

Beata Bylina
0000-0002-1327-9747
Maria Curie-Skłodowska University
in Lublin
Pl. M. Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: beata.bylina@mail.umcs.pl

Monika Piekarz
0000-0002-3457-9335
Maria Curie-Skłodowska University
in Lublin
Pl. M. Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: monika.piekarz@mail.umcs.pl

***Abstract*—Scalability is an important aspect related to time and energy savings on modern multicore architectures. In this paper, we investigate and analyze scalability in terms of time and energy. We compare the execution time and consumption energy of the LU factorization (without pivoting) and Cholesky, both with Math Kernel Library (MKL) on a multicore machine. In order to save the energy of these multithreaded factorizations, the dynamic voltage and frequency scaling (DVFS) technique was used. This technique allows the clock frequency to be scaled without changing the implementation. An experimental scalability evaluation was performed on an Intel Xeon Gold multicore machine, depending on the number of threads and the clock frequency. Our test results show that scalability in terms of the execution time expressed by the Speedup metric has values close to a linear function with an increase in the number of threads. In contrast, scalability in terms of the energy consumed expressed by the Greenup metric has values close to a logarithmic function with an increase in the number of threads. Both kinds of scalability depend on the clock frequency settings and the number of threads.**

I. Introduction

SCALABILITY is one of the main requirements to be taken into account when implementing parallel software on multicore machines, in particular for numerical algorithms involving many matrix calculations. The scalability feature allows an increasing number of threads to be used on a multicore machine in the hope that both time and energy efficiency will increase rather than degrade. The classical approach to scalability in parallel processing focuses on performance in terms of runtime. In this work, we want to study scalability in terms of two criteria, both the running time of the numerical algorithm and the energy consumption. The importance and need to consider multiple criteria in relation to scalability in parallel processing is shown in the work [8].

A distinction is made between two basic concepts related to scalability: scalability in the strong sense and scalability in the weak sense. In this paper we will only study scalability in the strong sense, that is, for a given problem size we will increase the number of threads. We focus on strong scalability because the parallelism available on modern machines will continue to increase.

An in-depth understanding of scalability in terms of execution time and energy consumption and the correlation between the two can allow the design of specific optimizations to reduce runtime and energy consumption for applications in different domains. In particular, it is important to study applications that make deliberate use of cache. Such applications usually come from the field of numerical linear algebra and involve matrix computations. Linear algebra is an important component of many numerical algorithms for various scientific and engineering problems. Over the years, BLAS (Basic Linear Algebra Subroutines) [6] has become the standard interface for linear algebra operations. One of the most popular BLAS packages is Math Kernel Library (MKL) [1]. The MKL library also contains implementations of matrix factorizations such as the LU factorization and the Cholesky factorization. The implementations of all factorizations are based on the BLAS library. The classical approach implemented in the MKL library for parallel matrix factorizations in cache-based systems uses fixed-size blocks that fit in the cache to evenly distribute the workload between threads. Currently, the MKL library tends to optimize runtime and does not take into account energy consumption savings. It is a well-known fact that reducing computation time usually implies energy savings, and is not the only reason for energy saving. Therefore, in order to improve the saving of energy of the algorithms from the MKL library without changing their implementation on multicore architectures, this work uses the dynamic voltage and frequency scaling technique DVFS [9].

The main contributions of this paper:

- a thorough empirical study of the runtime and energy consumption of multithreaded matrix factorizations (LU and Cholesky) concerning changing clock frequency and a selected number of threads;
- a scalability study using Speedup and Greenup metrics for varying numbers of threads for different clock frequencies;

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

The remainder of this article is organized as follows. In Section II, we discuss metrics such as Speedup and Greenup used to measure scalability in terms of the time and energy of parallel applications on multicore machines. In Section III we briefly review the LU and Cholesky algorithms. In Section IV, we present the test methodologies and experimental evaluation. Finally, in Section V, we conclude and make suggestions for future work.

II. Metrics

Energy consumption is the product of runtime and power consumed. Energy can be saved in various ways, e.g. by shortening runtime, reducing power consumption, or both, extending time but reducing power consumption more or vice versa.

The Speedup metric is known in the literature and used to analyze performance in parallel programming between different code implementations. It is assumed that we have two implementations of the algorithm, one non-optimized (basic) code running in time T_B and the other optimized code running in time T_O. Speedup is defined as follows:

$$Speedup = \frac{T_B}{T_O}$$

In [2] Greenup is defined analogously to Speedup only in terms of energy consumption:

$$Greenup = \frac{E_B}{E_O}$$

where E_B is the total energy consumption of the non-optimized code and E_O is the total energy consumption of the optimized code.

III. Algorithms

We will briefly introduce the LU and Cholesky algorithms used to solve systems of linear equations. The LU factorization transform square nonsingular matrix A into a product of two matrices:

$$A = LU$$

where L and U are lower and upper triangular matrices respectively.

The Cholesky factorization is defined only for A being Hermitian and positive-definite and has a form:

$$A = LL^T$$

where L is a lower triangular matrix. In this article, we investigate the LAPACK [3] implementation of the LU factorization from MKL library, namely `dgetrfnpi` (LU) [5], `dpotrf` (Cholesky) routines. These implementations are based on BLAS and arise from the use of a multithreaded BLAS.

The total number of floating-point operations (add, multiply, divide) for the LU factorizations is equal approximately $\frac{2}{3}n^3$. The number of floating point comparisons for the LU factorization is equals 0. The number of floating-point operations in the Cholesky factorization is $\frac{1}{3}n^3$. The number n is the size of the factoring matrix A.

IV. Numerical experiment – methodology and results analysis

A. Methodology

We tested two versions of matrix factorization: LU and Cholesky. We tested all algorithms without parallelization (1 thread) and in parallelized versions for 10, 20, 30, and 40 threads.

TABLE I: LU factorization at 1.7GHz

Threads/ Frequency	Time[s]	Energy[J]	Performance	Efficiency
1/1.7	889.07	57216.78	26.426	0.411
10/1.7	90.72	10560.03	258.989	2.225
20/1.7	46.35	7724.50	506.902	3.042
30/1.7	32.65	6716.32	719.517	3.498
40/1.7	26.53	6169.10	885.598	**3.808**

TABLE II: Cholesky factorization at 2.0GHz

Threads/ Frequency	Time[s]	Energy[J]	Performance	Efficiency
1/2.0	403.73	26795.84	29.098	0.438
10/2.0	40.35	4570.90	291.150	2.570
20/2.0	21.24	3653.45	553.140	3.215
30/2.0	14.67	3375.74	801.009	**3.480**
40/2.0	13.26	3412.81	886.171	3.442

Our test dataset consists of a square matrix filled with double-precision values. The matrix has dimensions of nxn, where $n = 32786$. In other words, our test dataset comprises 1073741824 cells, amounting to a total data size of 8 GB. For all algorithm versions, we have adhered to a row-wise data arrangement. These algorithms have been implemented in C++, incorporating vectorization and parallel processing techniques.

In our experimental configuration, we utilized a computing platform featuring a contemporary multicore Intel(R) Xeon(R) Gold 5218R processor, boasting 40 cores and a clock speed of 2.1 GHz. Our system was powered by the Linux 4.18.0 kernel and ran on the AlmaLinux 8.4 operating system, with the Intel ICC version 2021.5.0 compiler.

The Linux kernel facilitates CPU performance scaling through the `CPUFreq` subsystem, comprising three layers: core, scaling drivers, and governors. The core of `CPUFreq` offers a universal code infrastructure and user interfaces for all platforms supporting CPU performance scaling. It establishes the foundational framework for the other components. Scaling drivers communicate with hardware, supplying scale managers with data on available P-states (or P-state ranges in some cases) and accessing platform-specific hardware interfaces to modify processor P-states as directed by scale masters. Governors execute algorithms for estimating the necessary CPU capacity, typically each manager employing a single, optionally customized scaling algorithm.

The default scaling driver and governor are automatically chosen, but advanced configurations can still utilize userspace tools like `cpupower`, `acpid`, laptop mode tools, or desktop GUI tools.

To modify clock frequencies, we employed `CPUfreq` with the `acpi_cpufreq` driver. By default, this driver follows the

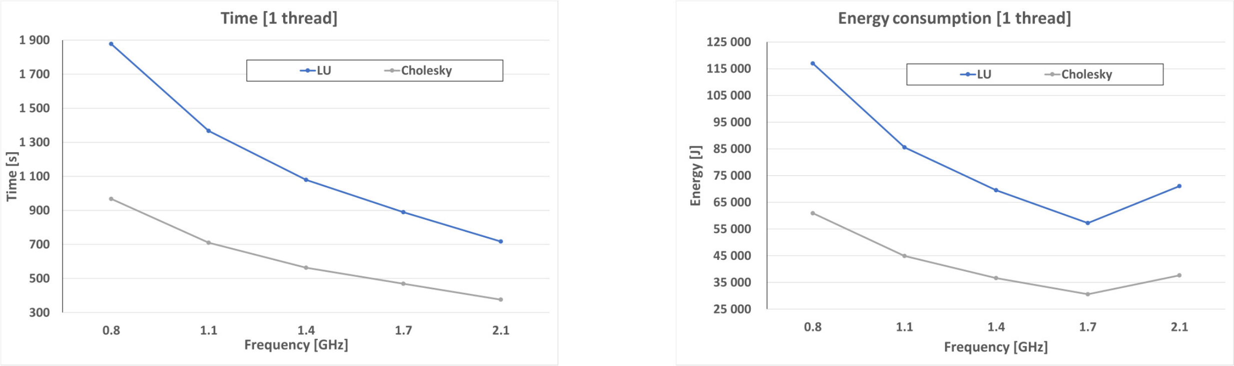


Fig. 1: Time execution for LU and Cholesky for 1 thread – left; Energy consumption of LU and Cholesky for 1 thread – right.

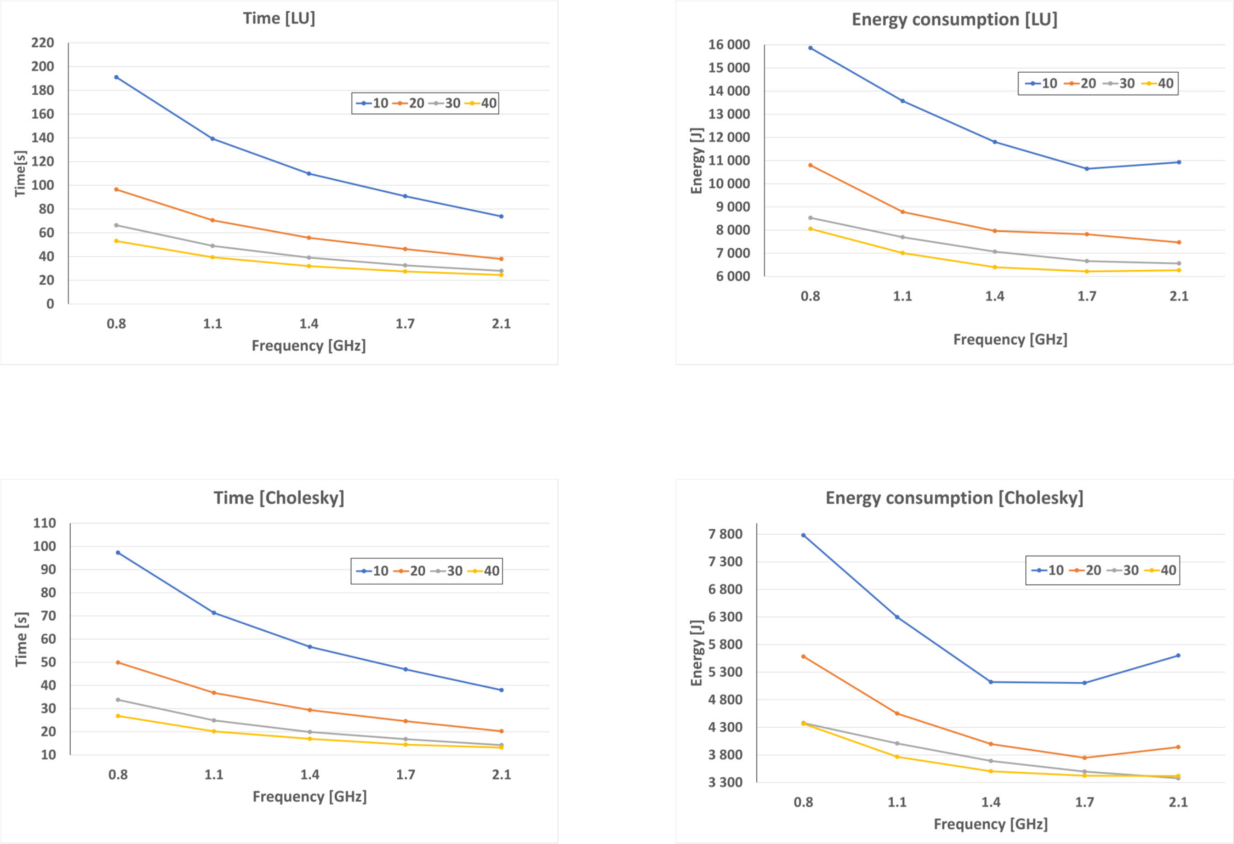


Fig. 2: Time execution for LU and Cholesky – left; Energy consumption of LU and Cholesky – right.

TABLE III: The most energy saving versions of algorithms

version of algorithm	energy consumption [J]	time [s]	waste of time [%]	energy saving [%]
LU ǀ 40 threads ǀ 1,7 GHz	6169.10	26.53	7.9	1.6
Cholesky ǀ 30 threads ǀ 2.0 GHz	3375.74	14.67	10.1	1.3

"conservative" governor, adjusting clock frequencies based on core load, selecting from available frequencies ranging from the minimum to the maximum supported by the processor.

We utilize the `cpupower` program to adjust the processor frequency limit's minimum and maximum values at a specific level, using the following commands:

```
cpupower frequency-set -d 1400000
cpupower frequency-set -u 1400000
```

for setting the minimum and maximum frequency limit values to 1.4 GHz. Executing these commands automatically switches the governor to `userspace`, enabling the configuration of a specific frequency. This frequency adjustment applies uniformly to all cores.

We do tests for the frequencies (P-states) available on our platform from 0.8 GHz to 2.1 GHz with step 0.1 GHz. We test first the following frequencies: 2.1 GHz, 1.7 GHz, 1.4 GHz, 1.1 GHz, and 0.8 GHz.

To assess the impact of algorithm optimizations on energy usage, we relied on data collected via the RAPL (Running Average Power Limit) interface, specifically designed for Intel processors. RAPL utilizes machine-specific records to continually monitor and regulate real-time energy consumption. In multi-socket systems, RAPL provides individual results for each socket or package, while also offering separate measurements for the memory modules (DRAM) linked to each socket. Starting with Haswell processors featuring fully integrated voltage regulators, RAPL's measurement accuracy has notably improved and meets acceptable standards [7]. Throughout our tests, we conducted measurements at 1-second intervals, considering the combined energy consumption of all sockets and their associated memory modules for analysis.

B. Time and energy consumption

In Fig. 1, we present the runtime and energy usage of individual algorithms when using a single thread. In Fig. 2, we display the same for parallel versions, i.e., for 10, 20, 30, and 40 threads. Notably, in the case of a single thread, the Cholesky factorization outperforms the LU factorization in terms of both time and energy consumption. This disproportion in performance is evident in Fig. 2, where we have different the y-axis scales to accommodate the dissimilarities.

In Fig. 2, we observe that reducing the clock frequency leads to an increase in runtime across all scenarios, while increasing the number of threads consistently reduces runtime. Thus, for our architecture, utilizing 40 threads at a frequency of 2.1 GHz proves to be the optimal choice in terms of time efficiency.

However, when considering energy consumption, a lower clock frequency, such as 1.7 GHz, can be advantageous in certain instances. This reduction in energy usage is evident for non-parallelized algorithm versions (approximately 19%) and for parallelized versions across all cases with 10 threads (3% for LU and 9% for Cholesky). Additionally, for 20 threads, a decrease in energy consumption is noticeable when employing 1.7 GHz with the Cholesky factorization (5%) and even for 40 threads with the LU factorization (1.6%). Lowering the clock frequency beyond 1.4 GHz does not yield any significant energy benefits.

Furthermore, we observe that energy consumption decreases as the number of threads used for calculations increases, with one exception: the Cholesky algorithm at 2.1 GHz. In this particular case, the algorithm is 1.2% more energy-efficient at 30 threads compared to 40 threads. A similar situation is observed at 2.0 GHz and 1.9 GHz (Fig. 3).

In response to the observed energy reduction when transitioning to a clock frequency of 1.7 GHz, we conducted additional experiments to explore the behavior of other frequencies within the range of 1.4 GHz to 2.1 GHz. The outcomes are depicted in Fig. 3. Subsequent tests indeed validated the presence of a localized energy consumption minimum at the 1.7 GHz frequency even for the LU algorithm executed with 40 threads.

The Table I and Table II show the test results for the frequencies at which we observe decreases in energy consumption for LU and Cholesky factorizations, respectively. The highest efficiency is achieved with LU at 1.7 GHz (Table I) running on 40 threads and in the case of Cholesky at 2.0 GHz (Table II) on 30 threads. Table III displays a compilation of algorithm versions and clock frequencies that resulted in the lowest energy consumption across both factorizations. In the table, the first column outlines the algorithm and the chosen configurations, the second column presents energy consumption in Joules, and the third column indicates the algorithm's runtime. The fourth and fifth columns reveal the percentage increase in runtime and the percentage reduction in energy consumption, respectively, relative to the configuration that achieved the shortest runtime — which, for both factorizations, was 40 threads and a 2.1 GHz clock frequency. Traditionally, optimizing for both time and energy efficiency involves increasing the number of threads and elevating the clock frequency. However, the two factorizations examined here demonstrate exceptions to this rule. If prioritizing energy savings over runtime, alternative thread and clock settings can be considered. Our tests have identified that the most energy-efficient configuration is achieved with 40 threads and a reduced clock frequency of 1,7 GHz for LU, while for Cholesky, it is with 30 threads and a clock frequency reduced to 2.0 GHz.

C. Speedup and Greenup

The figures in Fig. 4 illustrate the Speedup (left column) and Greenup (right column) values for different clock frequencies,

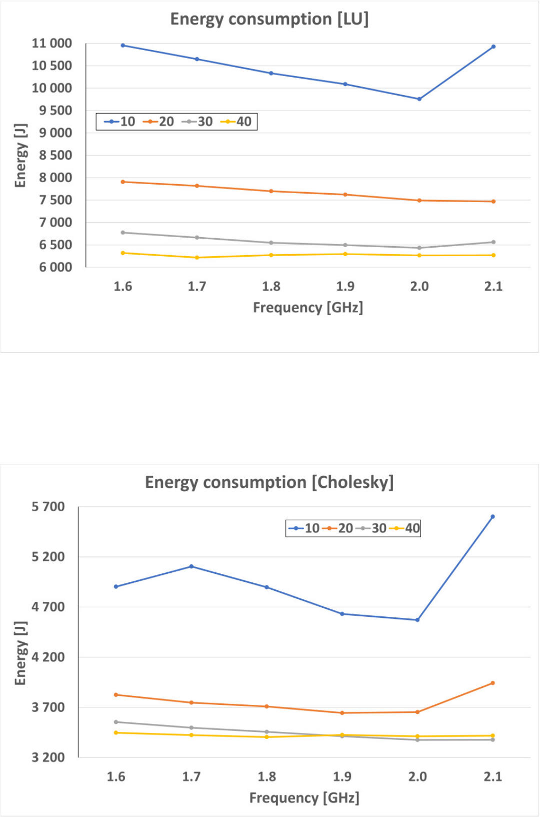


Fig. 3: Energy consumption of LU and Cholesky for frequencies from 1.6 GHz to 2.1 GHz.

derived from our experimental measurements. Each chart corresponds to one algorithm, either LU or Cholesky. The x-axis represents the number of active threads, while the y-axis displays the Speedup or Greenup values. To enhance reference, we've indicated the maximum expected Speedup (linear with the number of threads p) and Greenup (logarithmic with the number of threads p) with a dashed black line in the charts.

Across all two algorithms, it's evident that as the number of threads increases, irrespective of the clock frequency, the runtime improvement outpaces the reduction in energy consumption (time decreases more rapidly than energy consumption). For LU, Speedup approaches linearity for all frequencies, with deviations from the maximum expected value increasing as the thread count rises. Regarding Speedup, the 0.8 GHz frequency yields the most favorable results, while 2.1 GHz performs the poorest.

In general, the Greenup plot deviates further from the maximum expected value compared to Speedup, confirming our observations. For the algorithms we tested, similar to Speedup, the highest Greenup values are achieved at 0.8 GHz, while the lowest values, differing from the Speedup scenario, occur at 2.0 GHz. Notably, the 2.1 GHz frequency, which yielded the lowest Speedup values, still results in relatively high Greenup values (as seen in the purple line in the chart).

In our architecture, we observed the relationship:

$$Greenup \leq \alpha \log_2(Speedup)$$

where $\alpha > \beta$, and in our specific case, β falls within the interval $(2.84; 2.85)$. This leads to a research question: What is the value of α for other architectures?

V. Conclusion

This study explores scalability in relation to execution time and energy consumption for two matrix factorizations (LU and Cholesky) derived from the MKL library. To minimize energy consumption in these factorizations, we employed the DVFS technique. This approach allowed us to adjust clock frequency settings at the operating system level without modifying the implementation code.

We examined the impact of two parameters, clock frequency, and the number of threads, on execution time and energy consumption on a multicore machine. Execution time consistently decreases when using the highest clock frequency and the maximum number of threads for both factorizations. However, the same cannot be said for energy savings, as it varies based on the number of threads and clock frequency less regularly (see Table III).

Speedup and Greenup values increase with an expanding number of threads and typically decrease with a lower clock frequency. Experimental results reveal that Speedup values consistently surpass Greenup values, sometimes reaching up to 74% higher for specific combinations of clock frequency and thread count.

A deeper analysis of the research results, extended by tests of the LU factorization algorithm with pivoting and a study of the correlation between operation time and energy consumption using the Powerup and EDP metrics, is presented in [4]. Future research will address poor scalability by examining its impact on execution time and energy consumption in various multi-core machines and applications. Poor scalability entails keeping the problem size per processor constant while adding more computational units. Additionally, a key aspect to investigate is the correlation between strong and weak scalability regarding energy consumption.

References

[1] Intel Math Kernel Library, 2014. http://software.intel.com/en-us/articles/intel-mkl/.

[2] S. Abdulsalam, Z. Zong, Q. Gu, and Q. Meikang. Using the greenup, powerup, and speedup metrics to evaluate software energy efficiency. In *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, pages 1–8, 2015. 10.1109/IGCC.2015.7393699.

[3] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK users' Guide. Society for Industrial and Applied Mathematics*. SIAM, 1999. 10.1137/1.9780898719604.

[4] B. Bylina and M. Piekarz. Time–energy correlation for multithreaded matrix factorizations. *Energies*, 16, 08 2023. 10.3390/en16176290.

[5] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997. 10.1137/1.9781611971446.

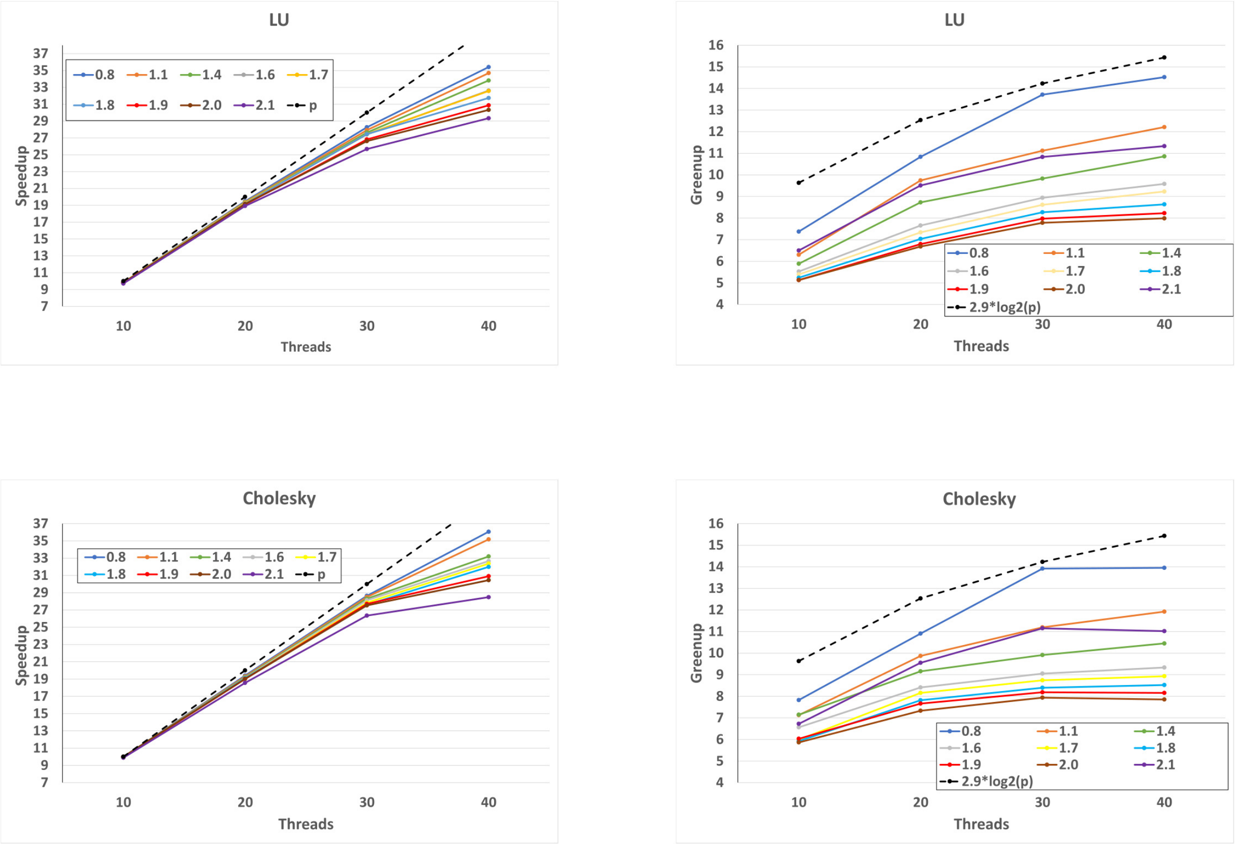


Fig. 4: Speedup for LU and Cholesky – left; Greenup of LU and Cholesky – right (p denotes number of threads).

[6] J. Dongarra, J. DuCroz, I. S. Duff, and S. Hammarling. A set of level-3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 16:1–28, 1990. 10.1145/77626.79170.

[7] K. Khan, M. Hirki, T. Niemi, J. Nurminen, and Z. Ou. RAPL in action: Experiences in using RAPL for power measurements. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, 3, 01 2018. 10.1145/3177754.

[8] Y. Ngoko and D. Trystram. Scalability in parallel processing. In S. K. Prasad, A. Gupta, A. L. Rosenberg, A. Sussman, and C. C. Weems, editors, *Topics in Parallel and Distributed Computing, Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms*, pages 79–109. Springer, 2018. 10.1007/978-3-319-93109-8_4.

[9] M. Weiser, B. Welch, A.J. Demers, and S. Shenker. Scheduling for reduced cpu energy. *1st OSDI*, pages 13–23, 11 1994.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 901–905
DOI: 10.15439/2023F492
ISSN 2300-5963 ACSIS, Vol. 35

Selection of floating photovoltaic system considering strong sustainability paradigm using SSP-COPRAS method

Aleksandra Bączkiewicz
0000-0003-4249-8364
Institute of Management, University of Szczecin
ul. Cukrowa 8, 71-004 Szczecin, Poland
Email: aleksandra.baczkiewicz@usz.edu.pl

Jarosław Wątróbski
0000-0002-4415-9414
Institute of Management, University of Szczecin
ul. Cukrowa 8, 71-004 Szczecin, Poland;
National Institute of Telecommunications
ul. Szachowa 1, 04-894 Warsaw, Poland
Email: jaroslaw.watrobski@usz.edu.pl

Abstract—**This paper presents research involving the selection of floating photovoltaics (FPV) system constructions under Polish conditions using a multi-criteria method incorporating criteria compensation reduction following the strong sustainability paradigm. The applied method is called SSP-COPRAS (Strong Sustainability Paradigm based Complex Proportional Assessment). The selection was carried out among four FPV designs and one reference conventional ground-mounted PV (GMPV) system. Data were obtained from the reference research paper. The results proved that the FPV system has a noticeable potential for making it competitive with GMPV, especially when technical criteria and criteria compensation reduction play an important role. However, GMPV's higher ratings, especially in terms of economics, show that FPV would have to reach a higher product maturity to become realistically competitive.**

I. Introduction

THE DEVELOPMENT of renewable energy sources (RES) has been an important element of energy and climate policy in European countries for many years. The objectives of the adopted policy oblige European Union member countries to increase the share of energy obtained from RES both in total energy consumption and in individual branches of the economy [1]. Poland's energy system is mainly based on coal [2]. However, the coal-based energy economy is one of the most important causes of climate change caused by carbon dioxide emissions into the atmosphere [3]. It implies that Poland is facing an urgent transition to energy systems using renewable energy sources [4]. Floating photovoltaics (FPV) can contribute to fulfilling this challenge [5]. Due to forecasts of rapid development of FPV in Europe [6], [7], it was decided to focus on applying this technology in Poland. FPV is currently a new and as yet immature technology [8]. However, factors such as the lack of available space for conventional photovoltaic systems, the increase in the number of producers, and financial encouragement in the form of fixed prices for FPV installations will stimulate the intensive development of this technology [9], [10].

This paper presents the assessment results concerning the technical and economic criteria of four different constructions for a designed FPV system. The data for the alternatives considered were derived from the reference paper, in which the analysis was carried out based on simulations performed on the PVsyst system [10]. The main objective of the analysis carried out in this article, which serves as a reference for this research work, was to investigate whether the application of FPV could be profitable in Polish conditions. The FPV under consideration has a capacity of 1 MWp. Such installed capacity was chosen because the auction mechanism provides the most cost-effective prices for PV systems under 1 MWp. The artificially created upper reservoir of the Porąbka-Żar pumped storage power plant was adopted as the target site for the considered structures. This reservoir has a limited usable area due to its rounded walls. In this article, the considered constructions were evaluated separately for each criterion with the performance values of each criterion. Simulations at PVsyst showed that FPV systems showed a slight advantage over ground-mounted PV (GMPV) for specific constructions.

Since FPV in Poland are new, this work provides a comprehensive source of knowledge on how such systems can work in Polish conditions, highlighting the novel character of the investigated topic. However, the manner of evaluation in the discussed article is complicated because it forces the analyst to consider the following criteria without considering them simultaneously. The present method also does not allow to assign of relevance to the evaluation criteria, which is essential from the decision-makers point of view. Finally, such a way of evaluation does not provide an opportunity to take into account the strong sustainability paradigm, which is important in terms of sustainable development of FPV systems [11]. Its consideration is justified by the fact that one system may have an extremely good value within one criterion that will compensate for less favorable values for other criteria. Preventing the phenomenon of criteria compensation is, therefore, one of the elements of the strong sustainability paradigm that should be considered in the field of RES. The limitations mentioned above in the discussed research became the motivation for presenting in this paper results of research using the new SSP-

COPRAS (Strong Sustainability Paradigm based Complex Proportional Assessment) multi-criteria method [12], [13], [14] for selecting the best construction of a floating solar farm from among four FPV variants and one reference system installed on the ground. MCDM methods have proven useful in FPV-related selection problems involving site selection [15], [16] and construction assessment [17].

The paper adopts nine evaluation criteria from a reference research paper: five are technical, and four are economic. The use of the MCDA method is justified by the fact that the MCDA results allow considering multiple criteria simultaneously and analyzing various scenarios, which is important from the decision-makers point of view [18], [19]. In addition, SSP-COPRAS makes it possible to reduce the compensation of criteria according to a strong sustainability paradigm [20], [21].

II. Methodology

This section presents the following steps of the SSP-COPRAS method, including basic assumptions and mathematical formulas. SSP-COPRAS implemented in Python is available at GitHub repository, along with a dataset of FPV constructions under consideration at link https://github.com/energyinpython/SSP-COPRAS-FPV.

Step 1. Create the decision matrix $X = [x_{ij}]_{m \times n}$ as Equation (1) shows. This matrix includes performance values x_{ij} collected for m alternatives, where $i = 1, 2, \ldots, m$ regarding n evaluation criteria, where $j = 1, 2, \ldots, n$.

$$X = [x_{ij}]_{m \times n} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{1}$$

Step 2. Calculate the Mean Deviation MD for each performance value x_{ij} by subtracting the mean value of each alternative's performance $\overline{x_j}$ for each criterion C_j. Multiply the resulting value by the sustainability coefficient s_j defined for each criterion as a real number in the range between 0 and 1. Equation (2) presents the complete procedure performed in this step.

$$MD_{ij} = (x_{ij} - \overline{x_j})s_j \tag{2}$$

Step 3. Assign 0 value to these MD values that for profit criteria C_j are lower than 0 (when x_{ij} is less than $\overline{x_j}$) and to these MD values that for cost criteria C_j are higher than 0 (when x_{ij} is higher than $\overline{x_j}$), as Equation (3) shows,

$$MD_{ij} = 0 \, \forall \, MD_{+ij} < 0 \, \vee \, MD_{-ij} > 0 \tag{3}$$

where MD_{+ij} represent MD values for profit criteria and MD_{-ij} define MD values for cost criteria. This step prevents unintended enhancement of performance values that are outliers from the average toward the worse.

The rest of the steps are the same as the classic COPRAS method.

Step 4. Normalize the decision matrix X using sum normalization method presented in Equation (4)

$$R = [r_{ij}]_{m \times n} = \frac{x_{ij} - MD_{ij}}{\sum_{i=1}^{m}(x_{ij} - MD_{ij})} \tag{4}$$

where $i = 1, 2, \ldots, m$ denotes ith alternative and $j = 1, 2, \ldots, n$ represents jth criterion

Step 5. This step involves calculating the weighted normalized decision matrix by multiplying values r_{ij} in normalized decision matrix R by the weights w_j determined for particular criteria, as Equation (5) demonstrates.

$$V = v_{ij} = r_{ij} w_j \tag{5}$$

Step 6. Calculate the sums of weighted normalized outcomes individual for profit criteria which have to be maximized (S_{+i}) and for cost criteria which have to be minimized (S_{-i}) as Equation (6) demonstrates,

$$S_{+i} = \sum_{j=1}^{n} v_{+ij}, \; S_{-i} = \sum_{j=1}^{n} v_{-ij} \tag{6}$$

where v_{+ij} are related to profit criteria which have to be maximized, and v_{-ij} are related to cost criteria which have to be minimized.

Step 7. Calculate the relative priority Q_i of evaluated options using Equation (7),

$$Q_i = S_{+i} + \frac{\sum_{i=1}^{m} S_{-i}}{S_{-i} \sum_{i=1}^{m} \frac{1}{S_{-i}}} \tag{7}$$

where an alternative with the highest value of Q_i is considered as the best option.

Step 8. Calculate the quantitative utility value U_i for each alternative,

$$U_i = \frac{Q_i}{Q_{max}} \tag{8}$$

where Q_{max} defines the highest relative importance score. The alternative with the highest U_i value is the best scored option.

III. Results

In this paper, a multi-criteria evaluation was performed using the SSP-COPRAS method considering the criteria compensation reduction for the four variants of FPV constructions and one corresponding ground-mounted PV system (GMPV) equivalent considered as a reference point for the FPV project assessment. Four FPV variants include two systems produced by Ciel&Terre: C&T S12 and C&T EW12 and two by Solaris Synergy: SolSyn S12 and SolSyn S25. Five technical parameters and four economic indexes serving as evaluation criteria are provided in Table I, together with units and objectives. Cost type represents criteria with the aim of minimizing performance values. On the other hand, Profit type defines criteria with the aim of maximizing performance values. The performance values of each FPV and reference GMPV construction collected for evaluation criteria are provided in Table II.

The investigation was conducted in two stages. Stage one involves an evaluation using SSP-COPRAS for the different relevance of the two criteria groups considered: technical parameters and economic indexes. When the significance of the technical criteria group was incremented from 0.25 to 0.75 with a step of 0.05, the significance of the economic group was reduced accordingly. Obtained values were then divided by 5 for the technical criteria and 4 for the economic criteria, and the resulting values were assigned to each criterion. Thus, an equal distribution of weights within the two criteria groups was applied. Sustainability coefficient s values were set as standard deviation values calculated from the normalized decision matrix for each criterion.

TABLE I
TECHNICAL PARAMETERS AND ECONOMIC INDICATORS OF DIFFERENT FPV SCENARIOS AND REFERENCE GMPV SYSTEM.

Criteria		Unit	Type
Technical parameters			
C_1	Area	[m^2]	Cost
C_2	Y$_f$ (Final PV system yield)	[kWh/kWp]	Profit
C_3	PR (Performance ratio)	[%]	Profit
C_4	AED (Annual Energy Density)	[kWh/m^2]	Profit
C_5	T$_{IWA}$ (Irradiance-weighted average temperature)	[$^\circ C$]	Cost
Economic indicators			
C_6	NPV (Net Present Value)	[€]	Profit
C_7	IRR (Internal Rate of Return)	[%]	Profit
C_8	LCOE (Levelized Cost of Energy)	[€/MWh]	Cost
C_9	Minimum Auction price for which NPV = 0	[€/MWh]	Cost

TABLE II
DECISION MATRIX WITH PERFORMANCE VALUES OF TECHNICAL PARAMETERS AND ECONOMIC INDICATORS OF DIFFERENT FPV SCENARIOS AND REFERENCE GMPV SYSTEM.

Technology	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
GMPV S25	10982	1079	86.92	96.5	25.5	82645	13.3	73.2	46.7
SolSyn S25	15220	1104	89.52	71.3	19.1	82426	12.9	74.5	46.9
SolSyn S12	9901	1046	89.26	103.8	19	44062	10.63	79	51.1
C&T S12	9901	1027	87.67	102	22.2	-26787	8	88.8	57.2
C&T EW12	8514	936	87.99	108	21.4	-83161	3.3	96	63.5

The SSP-COPRAS evaluation was then conducted sequentially for different scenarios of criteria relevance. SSP-COPRAS preference values obtained for this part of the study are contained in Table III.

In turn, rankings of the evaluated systems were built by sorting the preference values in descending order, as in the SSP-COPRAS evaluation, the alternative that received the highest preference value is considered the best scored. Rankings obtained for different weighting of technical and economic criteria groups are visualized in Figure 1. It can be observed that when economic criteria are more important and account for up to 60% of relevance, the leader of the ranking of evaluated systems is the reference system, namely the GMPV S25. This result coincides with the analysis of the authors of the reference article. However, when the relevance of the technical criteria group begins to dominate (from 65%), the ranking leader becomes SolSyn S12. This system receives the most significant promotion of all the alternatives when increasing the relevance of the technical criteria group.

SolSyn S12 has favorable performances in Annual Energy Density, T$_{IWA}$, performance ratio, and area. The C&T S12 and C&T EW 12 systems remain at the bottom of the ranking regardless of the change in the significance of the criteria groups. It is worth noting the two FPV systems, which are SolSyn S12 and SolSyn S25. SolSyn S25 has an advantage over SolSyn S12 when economic criteria are more relevant. When their relevance is aligned, SolSyn S12 gains an advantage over SolSyn S25. It is justified by the fact that SolSyn S25 has superiority over SolSyn S12 in terms of all economic criteria: NPV, IRR, LCOE, and minimum auction price. However, considering technical criteria, SolSyn S12 has an advantage over SolSyn S25 in terms of area, Annual Energy Density (significant advantage), and T$_{IWA}$.

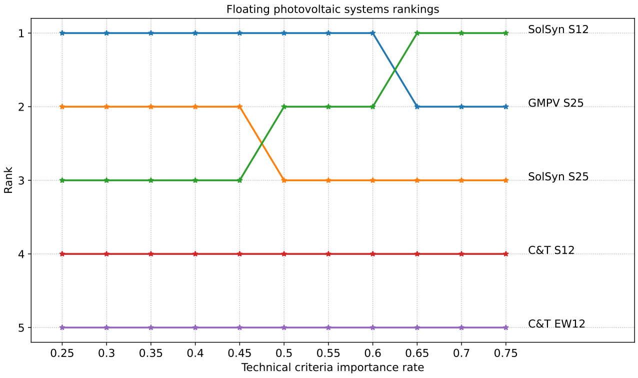


Fig. 1. SSP-COPRAS ranks of evaluated FPV and GMPV systems for different criteria weights.

In the case of FPV design, the desire for the smallest possible area is justified because the area of the power plant tank for which the study was conducted is limited, reducing the usable area for the floating system. In addition, a sufficient distance from the edge of the reservoir is required. Besides, a larger surface area requires more photovoltaic modules, which increases the cost of purchasing, installing, and maintaining the system. Annual Energy Density is an important profit criterion, as its high value increases the amount of electricity produced by the system during the year, which raises profits from system performance. Low irradiance-weighted average temperature (T$_{IWA}$) values increase the water cooling effect. The advantage of FPV systems over GMPV is partly due to the water-cooling effect, which enhances the efficiency of the solar farm.

In the following research stage, an analogous analysis was performed for modified values of the sustainability coefficient. Criteria weights were set as equal. Table IV provides performance values obtained for this analysis. Rankings of evaluated systems are displayed in Figure 2. In the case of sustainability coefficient modification, which was the subject of the second stage of the study, it turned out that all studied alternatives are stable in terms of the phenomenon of criteria compensation.

TABLE III
SSP-COPRAS PREFERENCE VALUES OF EVALUATED FPV AND GMPV SYSTEMS FOR DIFFERENT CRITERIA WEIGHTS.

Technical criteria group total weight Economic criteria group total weight	0.25 0.75	0.3 0.7	0.35 0.65	0.4 0.6	0.45 0.55	0.5 0.5	0.55 0.45	0.6 0.4	0.65 0.35	0.7 0.3	0.75 0.25
Technology											
GMPV S25	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9964	0.9867	0.9770
SolSyn S25	0.9782	0.9754	0.9726	0.9698	0.9670	0.9641	0.9612	0.9583	0.9519	0.9396	0.9274
SolSyn S12	0.9312	0.9397	0.9483	0.9570	0.9660	0.9751	0.9844	0.9939	1.0000	1.0000	1.0000
C&T S12	0.7478	0.7644	0.7815	0.7989	0.8168	0.8351	0.8538	0.8729	0.8893	0.9005	0.9117
C&T EW12	0.5433	0.5717	0.6008	0.6306	0.6613	0.6928	0.7251	0.7584	0.7897	0.8167	0.8441

TABLE IV
SSP-COPRAS PREFERENCE VALUES OF EVALUATED FPV AND GMPV SYSTEMS FOR DIFFERENT SUSTAINABILITY COEFFICIENTS.

Technology / Sustainability coeff.	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
GMPV S25	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9983	0.9921	0.9854	0.9783
SolSyn S25	0.9641	0.9632	0.9624	0.9615	0.9606	0.9597	0.9589	0.9564	0.9496	0.9424	0.9348
SolSyn S12	0.9690	0.9727	0.9767	0.9810	0.9856	0.9906	0.9959	1.0000	1.0000	1.0000	1.0000
C&T S12	0.8278	0.8346	0.8417	0.8492	0.8569	0.8651	0.8737	0.8813	0.8854	0.8895	0.8938
C&T EW12	0.7187	0.7209	0.7232	0.7257	0.7284	0.7312	0.7342	0.7362	0.7350	0.7337	0.7323

In the case of a significant degree of compensation reduction, we observe an advancement to the leading position of SolSyn S12, which, with a sustainability coefficient value of 0.7, outperforms the reference system GMPV S25.

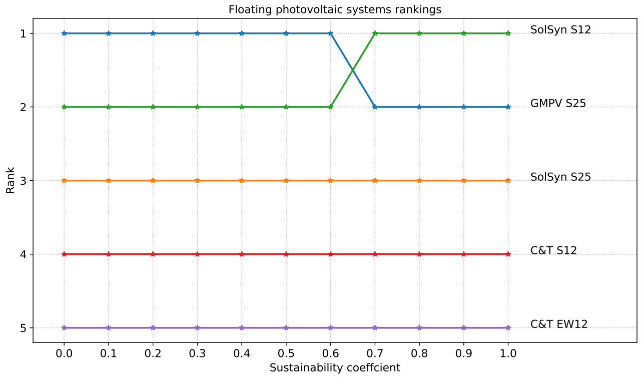


Fig. 2. SSP-COPRAS ranks of evaluated FPV and GMPV systems for different sustainability coefficients.

IV. Conclusion

Research results proved that the alternative that is robust in terms of technical criteria and sustainability and has the most potential to be a real competitor to the reference GMPV S25 design is the SolSyn S12. The potential of FPV systems in technical terms was also noted in the background paper [10] referenced in this research work, where a slight advantage of FPV systems over GMPV in terms of power generation capability was found. On the other hand, if economic factors play the most important role, the conventional reference PV design called GMPV S25 is the unquestionable leader. This result confirms the conclusions of the analysis carried out by the authors of the article [10], who found that FPV systems are currently less favorable from an economic point of view, especially in the auction system. It is because of the need for high capital expenditures, which currently cannot be compensated for even by a floating system with the best performance. FPV would have to reach a higher product maturity to become realistically competitive, especially from an economic point of view. In contrast, the results show promising potential in technical terms.

Acknowledgment

Publication funded by the state budget under the program of the Minister of Education and Science named Perły Nauki, Poland, project number: PN/01/0022/2022, total project value: PLN 165 000,00.

Ministry of Education and Science Republic of Poland
Minister of Education and Science Republic of Poland
PERŁY NAUKI

References

[1] M. Tutak and J. Brodny, "Renewable energy consumption in economic sectors in the EU-27. The impact on economics, environment and conventional energy sources. A 20-year perspective," *Journal of Cleaner Production*, vol. 345, p. 131076, 2022. doi: https://doi.org/10.1016/j.jclepro.2022.131076

[2] A. Wyrwa, W. Suwała, M. Pluta, M. Raczyński, J. Zyśk, and S. Tokarski, "A new approach for coupling the short-and long-term planning models to design a pathway to carbon neutrality in a coal-based power system," *Energy*, vol. 239, p. 122438, 2022. doi: https://doi.org/10.1016/j.energy.2021.122438

[3] M. Amin, H. H. Shah, A. G. Fareed, W. U. Khan, E. Chung, A. Zia, Z. U. R. Farooqi, and C. Lee, "Hydrogen production through renewable and non-renewable energy processes and their impact on climate change," *International journal of hydrogen energy*, vol. 47, no. 77, pp. 33 112–33 134, 2022. doi: https://doi.org/10.1016/j.ijhydene.2022.07.172

[4] H. Kryszk, K. Kurowska, R. Marks-Bielska, S. Bielski, and B. Eźlakowski, "Barriers and Prospects for the Development of Renewable Energy Sources in Poland during the Energy Crisis," *Energies*, vol. 16, no. 4, p. 1724, 2023. doi: https://doi.org/10.3390/en16041724

[5] M. Deveci, D. Pamucar, and E. Oguz, "Floating photovoltaic site selection using fuzzy rough numbers based LAAW and RAFSI model," *Applied Energy*, vol. 324, p. 119597, 2022. doi: https://doi.org/10.1016/j.apenergy.2022.119597

[6] R. Cazzaniga and M. Rosa-Clot, “The booming of floating PV,” *Solar Energy*, vol. 219, pp. 3–10, 2021. doi: https://doi.org/10.1016/j.solener.2020.09.057

[7] M. Kumar, H. M. Niyaz, and R. Gupta, “Challenges and opportunities towards the development of floating photovoltaic systems,” *Solar Energy Materials and Solar Cells*, vol. 233, p. 111408, 2021. doi: https://doi.org/10.1016/j.solmat.2021.111408

[8] T. T. E. Vo, H. Ko, J. Huh, and N. Park, “Overview of possibilities of solar floating photovoltaic systems in the offshore industry,” *Energies*, vol. 14, no. 21, p. 6988, 2021. doi: https://doi.org/10.3390/en14216988

[9] S. R. K. Soltani, A. Mostafaeipour, K. Almutairi, S. J. H. Dehshiri, S. S. H. Dehshiri, and K. Techato, “Predicting effect of floating photovoltaic power plant on water loss through surface evaporation for wastewater pond using artificial intelligence: A case study,” *Sustainable Energy Technologies and Assessments*, vol. 50, p. 101849, 2022. doi: https://doi.org/10.1016/j.seta.2021.101849

[10] A. Boduch, K. Mik, R. Castro, and P. Zawadzki, “Technical and economic assessment of a 1 mwp floating photovoltaic system in Polish conditions,” *Renewable Energy*, vol. 196, pp. 983–994, 2022. doi: https://doi.org/10.1016/j.renene.2022.07.032

[11] Q. Cao, M. O. Esangbedo, S. Bai, and C. O. Esangbedo, “Grey SWARA-FUCOM weighting method for contractor selection MCDM problem: A case study of floating solar panel energy system installation,” *Energies*, vol. 12, no. 13, p. 2481, 2019. doi: https://doi.org/10.3390/en12132481

[12] J. Wątróbski, A. Bączkiewicz, and I. Rudawska, “SSP COPRAS Based Approach Towards Sustainability Assessment in Healthcare,” in *AMCIS 2022 Proceedings*. AMCIS 2022, 2022, pp. 1–10.

[13] I. M. Hezam, A. R. Mishra, P. Rani, A. Saha, F. Smarandache, and D. Pamucar, “An integrated decision support framework using single-valued neutrosophic-MASWIP-COPRAS for sustainability assessment of bioenergy production technologies,” *Expert Systems with Applications*, vol. 211, p. 118674, 2023. doi: https://doi.org/10.1016/j.eswa.2022.118674

[14] I. M. Hezam, A. R. Mishra, R. Krishankumar, K. Ravichandran, S. Kar, and D. S. Pamucar, “A single-valued neutrosophic decision framework for the assessment of sustainable transport investment projects based on discrimination measure,” *Management Decision*, vol. 61, no. 2, pp. 443–471, 2023. doi: https://doi.org/10.1108/MD-11-2021-1520

[15] F. Guo, J. Gao, H. Liu, and P. He, “Locations appraisal framework for floating photovoltaic power plants based on relative-entropy measure and improved hesitant fuzzy linguistic DEMATEL-PROMETHEE method,” *Ocean & Coastal Management*, vol. 215, p. 105948, 2021. doi: https://doi.org/10.1016/j.ocecoaman.2021.105948

[16] S. Di Grazia and G. M. Tina, “Optimal site selection for floating photovoltaic systems based on Geographic Information Systems (GIS) and Multi-Criteria Decision Analysis (MCDA): a case study,” *International Journal of Sustainable Energy*, pp. 1–23, 2023. doi: https://doi.org/10.1080/14786451.2023.2167999

[17] J. L. Schaefer, J. C. M. Siluk, and P. S. de Carvalho, “An MCDM-based approach to evaluate the performance objectives for strategic management and development of Energy Cloud,” *Journal of Cleaner Production*, vol. 320, p. 128853, 2021. doi: https://doi.org/10.1016/j.jclepro.2021.128853

[18] A. Bączkiewicz, J. Wątróbski, B. Kizielewicz, and W. Sałabun, “Towards objectification of multi-criteria assessments: a comparative study on MCDA methods,” in *2021 16th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2021. doi: https://doi.org/10.15439/2021F61 pp. 417–425.

[19] J. Wątróbski and A. Bączkiewicz, “Towards Sustainable Transport Assessment Considering Alternative Fuels Based on MCDA Methods,” in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022. doi: https://doi.org/10.15439/2022F144 pp. 799–808.

[20] J. Wątróbski, A. Bączkiewicz, and I. Rudawska, “A Strong Sustainability Paradigm based Analytical Hierarchy Process (SSP-AHP) method to evaluate sustainable healthcare systems,” *Ecological Indicators*, vol. 154, p. 110493, 2023. doi: https://doi.org/10.1016/j.ecolind.2023.110493

[21] J. Wątróbski, A. Karczmarczyk, and A. Bączkiewicz, “Using the TOSS method in semi-autonomous passenger car selection,” *Sustainable Energy Technologies and Assessments*, vol. 58, p. 103367, 2023. doi: https://doi.org/10.1016/j.seta.2023.103367

DOI: 10.15439/2023F3686
ISSN 2300-5963 ACSIS, Vol. 35

Urban scene semantic segmentation using the U-Net model

Marcin Ciecholewski
Department of Geoinformatics
Faculty of Electronics, Telecommunication and Informatics
Gdańsk University of Technology
Gdańsk, Poland
Email: marcin.ciecholewski@pg.edu.pl

***Abstract*—Vision-based semantic segmentation of complex urban street scenes is a very important function during autonomous driving (AD), which will become an important technology in industrialized countries in the near future. Today, advanced driver assistance systems (ADAS) improve traffic safety thanks to the application of solutions that enable detecting objects, recognising road signs, segmenting the road, etc. The basis for these functionalities is the adoption of various classifiers. This publication presents solutions utilising convolutional neural networks, such as MobileNet and ResNet50, which were used as encoders in the U-Net model to semantically segment images of complex urban scenes taken from the publicly available Cityscapes dataset. Some modifications of the encoder/decoder architecture of the U-Net model were also proposed and the result was named the MU-Net. During tests carried out on 500 images, the MU-Net model produced slightly better segmentation results than the universal MobileNet and ResNet networks, as measured by the Jaccard index, which amounted to 88.85%. The experiments showed that the MobileNet network had the best ratio of accuracy to the number of parameters used and at the same time was the least sensitive to unusual phenomena occurring in images.**

I. Introduction

SEMANTIC segmentation of images is a very important topic in computer vision, and its purpose is to divide the image into regions of different semantic categories. This division is connected with the classification of the image in the sense that it produces per-pixel category prediction instead of image-level prediction [1]. This means that semantic segmentation can be seen as extending image classification from the image level to the pixel level. However, the training data intended for semantic segmentation requires manual labelling at the pixel level, which is much harder and more time-consuming than other vision tasks, such as image classification or object detection.

Much effort has gone into research on image segmentation in recent years and great progress has been made [2], [3], [4], [5], [6]. Despite this, segmentation still remains a difficult problem because of rich intra-class variation, context variation and ambiguities resulting from the low resolution of images.

State-of-the-art approaches used in semantic segmentation adopt a fully convolutional network (FCN) with an encoder/decoder architecture [7], [8]. The encoder generates low-resolution image features and then the decoder upsamples features to segmentation maps and is used for pixel-level classification of the feature representations.

Semantic segmentation has many different applications, notably including: augmented reality, autonomous driving, image editing, medical imaging, robotics, smart cities, and many others [9], [10].

The visual understanding of complex urban street scenes is crucial for problems concerning the smart city, in which autonomous vehicles can drive and certain infrastructure elements can communicate to ensure the greatest comfort of people and reduce the time lost. The use of various large-scale datasets contributed to a great development of research on object detection and a popularisation of methods using deep learning techniques [11], [12]. To use artificial neural networks (ANN) for the semantic segmentation of complex urban scenes, researchers can utilise Cityscapes [13], a benchmark suite and a large-scale dataset to train and test approaches for pixel-level and instance-level semantic labelling. Figure 1 shows example images available in the training subset of the Cityscapes dataset [13]. Images from the training set which can be semantically segmented using specific colours contain 30 different classes describing defined objects found in the city.

This paper presents research on the semantic segmentation of urban scenes using several different convolutional neural networks with an encoder/decoder architecture. For this purpose, MobileNet [14], [15] and ResNet50 [16], [17] were used as encoders in the U-Net model [18]. During the studies, some modifications to the U-Net model were also proposed based on the experiments carried out. The research work was done using the Cityscapes dataset [13]. The purpose of this research was to obtain improved segmentation results, and to assess the proposed solutions in detail, including their advantages and disadvantages.

II. Materials and methods

A. Data

Research work was carried out using the Cityscapes dataset [13]. This is a collection of 3,475 images from cities in Germany that were recorded during vehicle driving. They are saved in the *png* format and have a resolution of 2048×1024 pixels. This set was divided into a subset designed for training,

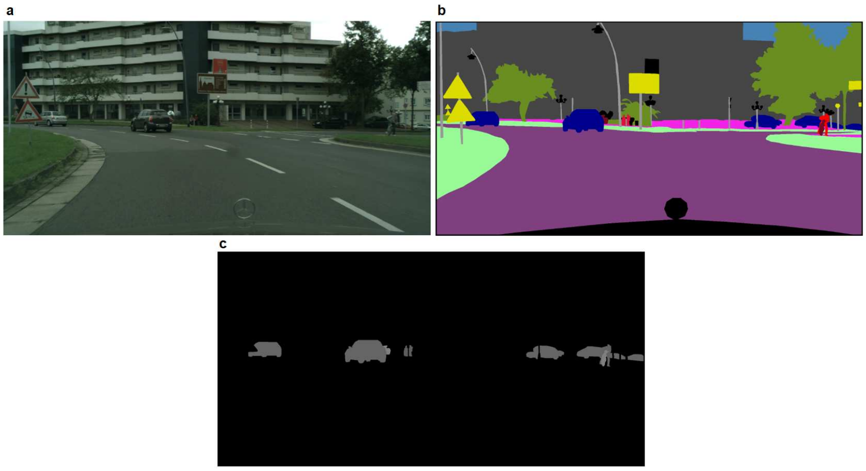


Fig. 1. Sample images from the training subset from the city of Stuttgart. (a) Source image. (b) Semantic segmentation in colour. (c) Segmentation with only vehicles and people marked.

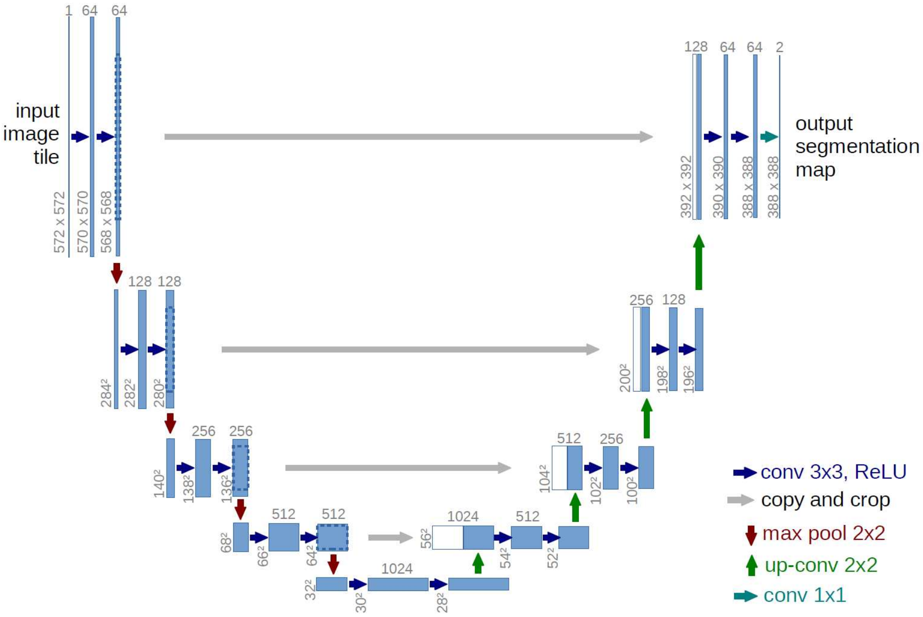


Fig. 2. Diagram of the U-Net model, in which the characteristic letter "U" is visible. Blue rectangles represent multi-channel feature maps. The current size of the maps is written on the left. The current number of channels is written above each rectangle. White rectangles are maps transferred to the decoding part of the model. Blue arrows are convolutional layers, red ones are pooling layers, and green arrows are layers that increase the resolution. Gray arrows connect feature maps obtained during encoding to their counterparts during decoding [18].

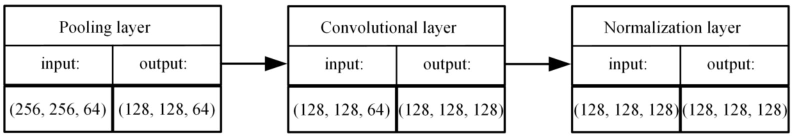


Fig. 3. Proposed encoder block. The first two numerical values are the height and the width of feature maps, and the third is the number of channels. The pooling layer reduces the resolution of feature maps. Then, the convolution layer uses filters to increase the number of channels. At the end, normalization is performed.

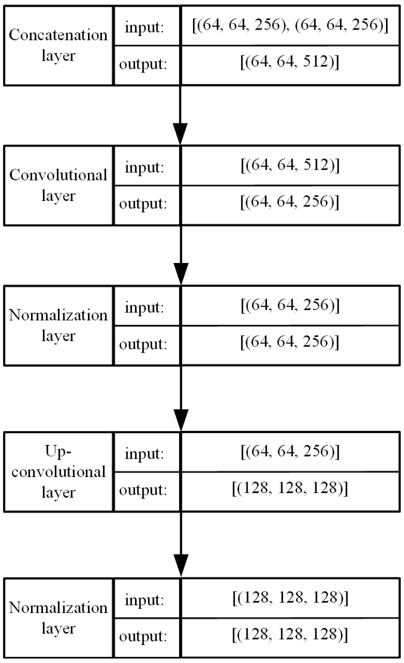


Fig. 4. Proposed decoder block. The first two numerical values are the height and width, the third is the number of channels. The concatenation layer connects the feature maps of the encoder and the decoder that have the same resolution, and then the convolutional layer reduces the number of channels. The next steps are: data normalization and the use of an up-convolutional layer which increases the resolution of feature maps. The last element of the block is the normalization layer whose output forms one of the inputs of the next concatenation layer that begins the next decoder block.

comprising 2975 images, and a subset for testing machine learning models, containing 500 images.

B. Preprocessing

Pre-processing is to shorten the network training time and to properly prepare the images so that the learning process is efficient and the highest possible results of semantic segmentation are obtained on the test set. For this purpose, image resolution change, random cropping and normalization were applied.

1) Resolution change: To improve the training time of ANNs, the original resolution of source images was reduced from 2048 × 1024 pixels to 600 × 300 pixels using the nearest neighbour method [19]. Apart from RGB channels, the rescaled images also contained a channel representing the segmentation of individual images.

2) Random cropping: In the next step, random cropping [20] was used to obtain images with the size of 256× 256 pixels. In addition, every image was mirrored with a probability of 1/2. This produces more diverse input data and reduces the risk that the network will analyse the general features of all images.

3) Normalization: The next step is data normalization. This means changing the value range of image RGB channels to the interval of [0, 1]. The last channel, which contains values representing the semantic segmentation, remains unchanged.

C. Convolutional network models used

During the study, an attempt was made to evaluate two convolutional networks, i.e. MobileNet [14], [15] and ResNet50 [16], [17], used as the encoder in the U-Net [18] model to perform semantic segmentation. Some modifications to the U-Net model were also proposed based on experiments carried out on the training set.

1) U-Net model: U-Net is a neural network model whose original purpose was the semantic segmentation of medical images [18]. The U-Net model consists of two paths which make the model diagram resemble the letter "U", namely the contraction and expansion paths representing the encoder and the decoder, respectively. Both paths are shown in Figure 2.

2) Modified U-Net model: A modified network model based on the standard U-Net model with added normalization layers, abbreviated as MU-Net, was proposed for performing the semantic segmentation. The Rectified Linear Unit (ReLU) [21] was used as the activation function. Example network encoder and decoder blocks are shown in Figures 3 and 4. Convolutional layers use 3×3 filters. A 1×1 filter is used for concatenation layers and at the resolution of 256×256 pixels, when transition to pixel classification occurs. During the convolution, there is a descent to feature maps with the size of 8×8 pixels. This network is configured only for performing the semantic segmentation. This is why the appropriate parameters were selected during many trials to train the network and the possible decrease of the accuracy during the classification of the entire image or the detection of individual objects was not taken into account.

Therefore, during many attempts to teach the network on the training set, appropriate parameters were selected

3) MobileNet: The Mobilenet [14] is a convolutional network that can be used on mobile devices. It is characterized by fewer parameters and a shorter training time than other models of convolutional networks. It has a high ratio of accuracy to the parameter number. The MobileNetV2 network [15] is an extension of the Mobilenet network. The authors mention semantic segmentation as one of the applications of this network. The main changes compared to the previous version are the use of the ReLU6 activation function instead of ReLU, and of the so-called bottleneck [22]. According to the authors' calculations, this network is more accurate than the original version, while the number of parameters is significantly reduced.

4) ResNet50: ResNet50 is a network belonging to the group of so-called residual neural networks [16] introduced in 2015, where the 50 in the name represents the number of network layers. They are characterized by the possibility of skipping some layers during the analysis. The ResNet network has a block-skipping mechanism which transfers to the next layer the parameter value processed only by the activation function. Network blocks use the bottleneck method just like in MobileNetV2. The ResNet50V2 network has a small block consisting of a normalization layer followed by a ReLU activation function. Pre-activation, i.e. the use of blocks before fully convolutional layers, speeds up the training of the network and improves its accuracy.

5) Decoders used: A decoding part was added to each neural network used to encode image features so that the numbers of encoder and decoder parameters are similar. In addition, in the case of Mobilenet and ResNet50, the same decoder was used for v1 and v2. Because of the similar number of parameters in the MU-Net and ResNet50 encoders, the MU-Net model uses the same decoder as the ResNet50 model.

Data from Table I shows that regardless of using one decoder for the MobileNet network and another for the remaining networks, every U-Net model has a different number of decoder parameters. This is due to the different number of channels in specific encoder layers. The consequence of this is that concatenation layers that follow these layers and have these layers as input also have a different number of channels, resulting in a different number of parameters.

TABLE I
THE NUMBER OF PARAMETERS IN THE ENCODING AND DECODING PARTS OF NETWORKS BASED ON THE U-NET MODEL.

	Number of parameters		
Network	Encoder	Decoder	Entire U-Net model
MobileNet	3 228 864	2 788 834	6 017 698
MobileNetV2	2 257 984	2 288 610	4 546 594
ResNet50	23 587 712	19 483 426	43 071 138
ResNet50V2	23 564 800	15 698 722	39 263 522
MU-Net	25 163 136	19 554 850	44 717 986

III. EXPERIMENTS COMPLETED AND THEIR RESULTS

The accuracy of segmentation performed with CNNs was measured using the Jaccard index. This is the most widespread method of evaluating semantic segmentation. It allows calculating the similarity of the obtained segmentation to the manually labelled by experts. After the process of training on a set of 2,975 images, the results obtained were evaluated on a set of 500 images 256×256 pixels in size, produced by the random cropping of the original test set. It can be said that all ANNs achieved very similar results, as shown in Table II and in Figure 5.

TABLE II
TABLE SHOWING THE ACCURACY OF THE U-NET MODEL NETWORK USING SPECIFIC ENCODERS. THE RESULTS TURNED OUT TO BE VERY SIMILAR DESPITE VERY LARGE DIFFERENCES IN THE NUMBER OF PARAMETERS.

Encoding network	Jaccard index values
MobileNet	86.19%
MobileNetV2	86.20%
ResNet50	86.23%
ResNet50V2	86.27%
MU-Net	88.85%

It is worth noting that the improvements in the new versions of both MobileNet and ResNet50 led to a slight increase in the Jaccard index values of the semantic segmentation, while the number of parameters was reduced by, respectively: 24.4% and 8.9%. For this reason, only the newer versions of both networks were used in subsequent experiments that checked the accuracy of segmentation using the Jaccard index. It can be concluded that increase of performance is not caused by reducing the number of parameters, it is the result of improving the network architecture. The difference in Jaccard index values between the most and least accurate ANNs amounts to 2.7%.

A. Noise in images

The impact of noise on the accuracy of the segmentations performed was checked for 59 images from the test set from the city of Lindau. Noise was introduced in the images using the Hue, Saturation, Value (HSV) colour space and an additional Holdness parameter. The channel values of Hue vary from 0 to 180, and of Saturation and Value from 0 to 255. The Holdness parameter has values from the interval [1, 8] and is inversely proportional to the hue variation. Table III shows the segmentation results measured with the Jaccard index. Figure 6 shows an example source image before and after noise was added, and Table III shows the segmentation results measured with the Jaccard index. Noise with the values of (Hue, Saturation, Value, Holdness) = (10, 22, 22, 1) was added to all images from the test set from the city of Lindau. Even though the noise had been selected so that it would not hinder humans from recognizing any image elements, ANNs encountered a problem and Jaccard index values fell by about 20%.

TABLE III
DIFFERENCE IN ACCURACY OF U-NET MODELS BEFORE AND AFTER NOISE WAS ADDED TO IMAGES.

Encoding network	Original set	Noisy set
MobileNetV2	75.81%	51.90%
ResNet50V2	74.46%	56.52%
MU-Net	79.36%	59.48%

B. Non-standard lighting – shaded images

The 7 most shaded examples were selected from the test image set to test the impact of low light on segmentation accuracy. The results are presented in Table IV.

TABLE IV
A TABLE SHOWING THE ACCURACY OF THE U-NET NETWORK MODEL CHECKED ON IMAGES WITH POOR LIGHTING CAUSED BY SHADE.

Encoding network	Jaccard index value
MobileNetV2	84.22%
ResNet50V2	83.52%
MU-Net	86.39%

The results show that strong image shading does not hinder obtaining positive segmentation results. The approximately 2% drop in accuracy may be due to other features of the selected images.

C. Class imbalance

Class imbalance is a phenomenon in which the analysed classes are not equally represented. A dominant number of pixels belonging to one or several classes may occur in the

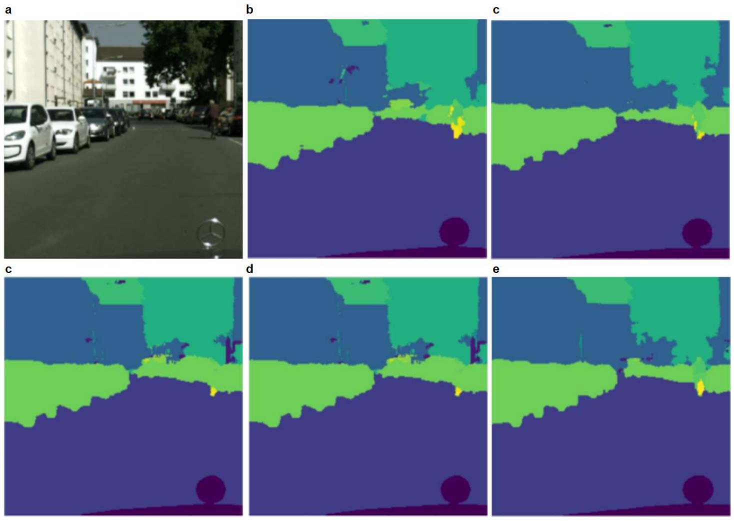


Fig. 5. Example results of a semantic segmentation on a sample image from the test set. (a) Original image (b) MobileNet (c) MobileNetV2 (d) ResNet50 (e) ResNet50V2 (f) MU-Net.

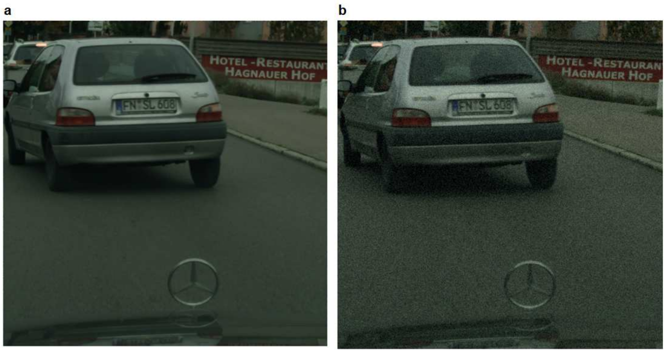


Fig. 6. Example test images from the city of Lindau (a) Original source image (b) Image with added noise with values of (Hue, Saturation, Value, Holdness) =(10, 22, 22, 1).

semantic segmentation. An example is shown in Figure 7, in which the road and vegetation are darker, and bright sunlight penetrates only to a small extent. As a result, two dominant classes are visible, namely the road and vegetation. ANNs frequently do not receive images with strongly dominant classes during training, or receive too few such images to later produce correct results when the classifier is tested. To check the segmentation results, 20 images with strongly dominating classes were selected from the test set, and the results obtained are presented in Table V. The results from Table V demonstrate a certain advantage of the ResNetV2 network in this test. The MobileNetV2 network also achieved a better result than the proposed MU-Net model, which may indicate some overtraining of this network, which produced the worst result this time.

TABLE V
A TABLE SHOWING THE ACCURACY OF THE U-NET MODEL NETWORK USING SPECIFIC ENCODERS, CHECKED ON 20 IMAGES WITH DOMINANT CLASSES.

Encoding network	Jaccard index value
MobileNetV2	76.67%
ResNet50V2	79.36%
MU-Net	74.99%

IV. Conclusions

This paper describes the practical properties of neural network models, namely MobileNet, ResNet, U-Net, and the MU-Net model, used for the semantic segmentation of images from the Cityscapes dataset [13]. The U-Net model is a very interesting approach to the problem of semantic segmentation, which is an extremely difficult area of digital image analysis. However, this model has some accuracy limitations and the constant increase of the number of parameters will not ensure satisfactory results, which is one of the conclusions. During the research, the author was able to propose an MU-Net model, i.e. an ANN dedicated to semantic segmentation, which produced results slightly better than universal networks like MobileNet or ResNet. However, the MobileNetV2 network turned out to be the most interesting and promising ANN used. It has a

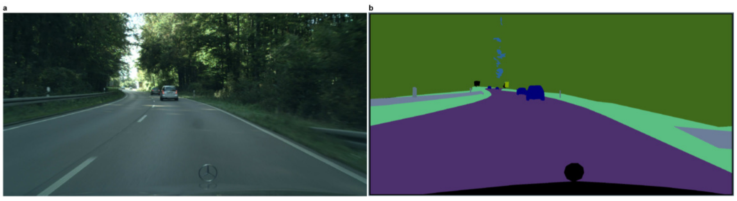


Fig. 7. Example image with semantic segmentation showing non-standard lighting and class imbalance. Most of the image is covered by the road and vegetation, while bright sunlight and moving cars occupy a small fragment of the image. (a) Original image. (b) Semantic segmentation containing mainly two classes.

very good ratio of accuracy to the number of parameters and, at the same time, is less affected by non-standard phenomena in images. Due to the constantly increasing computing power of mobile devices, neural networks designed for analysing images on mobile devices with even better parameters can be expected in the near future. In future research, it is definitely worth investigating improving the accuracy of the semantic segmentation of noisy images and the issue of class imbalance. There are also other interesting directions of research, e.g. performing a semantic segmentation that simulates autonomous vehicle driving using recorded videos, and carrying out a three-dimensional semantic segmentation of urban scenes. It is also worth trying to supplement training sets using various augmentation methods, but keeping in mind the need to prevent learning the wrong patterns.

References

[1] J. Long, E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation,” *In Proceedings of the IEEE conference on computer vision and pattern recognition* 2015, pp. 3431-3440, https://doi.org/10.1109/CVPR.2015.7298965.

[2] L.C. Chen, Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei and W. Liu, “Encoder-decoder with atrous separable convolution for semantic image segmentation," *In Proceedings of the European conference on computer vision (ECCV)* 2018, pp. 801-818, https://doi.org/10.1007/978-3-030-01234-2_49.

[3] J. Fu, J. Liu, J. Jiang, Y. Li, Y. Bao and H. Lu, “Scene segmentation with dual relation-aware attention network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32(6), 2020, pp. 2547-2560, https://doi.org/10.1109/TNNLS.2020.3006524.

[4] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz and D. Terzopoulos, “Image segmentation using deep learning: A survey, " *IEEE transactions on pattern analysis and machine intelligence*, vol. 44(7), 2021, pp. 3523-3542, 10.1109/TPAMI.2021.3059968.

[5] P. Malík, Š. Krištofík K. Knapová, “Instance segmentation model created from three semantic segmentations of mask, boundary and centroid Pixels verified on GlaS dataset, " *In 2020 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 569-576*, http://dx.doi.org/10.15439/2020F175.

[6] L. Ming, Y. Qingbo, L. Mingyu, “Retinal blood vessel segmentation based on multi-scale deep learning, " *In: 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 1-7*, http://dx.doi.org/10.15439/2018F127

[7] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A.I. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, " *IEEE transactions on pattern analysis and machine intelligence*, vol. 40(4), 2017, pp. 834-848, https://doi.org/10.1109/TPAMI.2017.2699184.

[8] V. Badrinarayanan, A. Kendall and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation, " *IEEE transactions on pattern analysis and machine intelligence*, vol. 39(12), 2017, pp. 2481-2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[9] M. Siam, S. Elkerdawy, M. Jagersand and S. Yogamani, “Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges, " *In 2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pp. 1-8, https://doi.org/10.1109/ITSC.2017.8317714.

[10] Z. W. Hong, C. Yu-Ming, S. Y. Su, T. Y. Shann, Y. H. Chang, H. K. Yang, *ldots* & C. Y. Lee, “Virtual-to-real: Learning to control in visual semantic segmentation, " *arXiv preprint*, 2018, 1802.00285, https://doi.org/10.48550/arXiv.1802.00285.

[11] A. Krizhevsky, I. Sutskever, G.E. Hinton, “Imagenet classification with deep convolutional neural networks, " *Communications of the ACM*, 2017, vol. 60(6), pp. 84-90, https://doi.org/10.1145/3065386.

[12] J. Long, E. Shelhamer, T. Darrell, “Fully convolutional networks for semantic segmentation, " *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440, https://doi.org/10.1109/CVPR.2015.7298965.

[13] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, B. Schiele, “The cityscapes dataset for semantic urban scene understanding, " *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213-3223, https://doi.org/10.1109/CVPR.2016.350.

[14] A. G. Howard, Z. Menglong, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications, " *CoRR*, 2017, abs/1704.04861, https://doi.org/10.48550/arXiv.1704.04861.

[15] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, “Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation, " *CoRR*, 2018, abs/1801.04381.

[16] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition, " *CoRR*, 2015, abs/1512.03385, https://doi.org/10.1109/CVPR.2016.90.

[17] K. He, X. Zhang, S. Ren, J. Sun, “Identity mappings in deep residual networks, " *CoRR*, 2016, abs/1603.05027, https://doi.org/10.1007/978-3-319-46493-0_38.

[18] O. Ronneberger, P. Fischer, T. Brox, “U-net: Convolutional networks for biomedical image segmentation, " *In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany*, Part III 18, pp. 234-241, https://doi.org/10.1007/978-3-319-24574-4_28.

[19] O. Rukundo, H. Cao, “Nearest neighbor value interpolation, " *arXiv preprint*, 2012, 3:25:30, https://doi.org/10.14569/IJACSA.2012.030405.

[20] R. Takahashi, T. Matsubara, K. Uehara, “Data augmentation using random image cropping and patching for deep CNNs, " *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, vol. 30(9), pp. 2917-2931, https://doi.org/10.1109/TCSVT.2019.2935128.

[21] V. Nair, G. E. Hinton, “Rectified linear units improve restricted boltzmann machines, " *In Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.

[22] E. R. De Rezende, G. C. Ruppert, A. Theophilo, E. K. Tokuda, T. Carvalho, “Exposing computer generated images by using deep convolutional neural networks. Signal Processing, " *Image Communication*, 2018, vol. 66, pp. 113-126, https://doi.org/10.1016/j.image.2018.04.006.

DOI: 10.15439/2023F9537
ISSN 2300-5963 ACSIS, Vol. 35

The Effects of Native Language on Requirements Quality

Fayona Cowperthwaite
0000-0002-7501-7048
University of Gothenburg
Sweden
facowperthwaite@gmail.com

Jennifer Horkoff
0000-0002-2019-5277
University of Gothenburg
Chalmers University of Technology
Sweden
jennifer.horkoff@gu.se

Sylwia Kopczyńska
0000-0002-9550-3334
Poznan University of Technology
Poland
sylwia.kopczynska@cs.put.poznan.pl

***Abstract*—[Context and motivation] More and more often software development projects involve participants of diverse nationalities and languages. Thus, software companies tend to use English as their business language. Moreover, to better prepare for future jobs, students consciously choose university courses in English. [Question/problem] As a result there is an increasing number of software engineers who are working or studying in a language which is not their native language. The question arises whether native language has an effect on the quality of natural language requirements. [Principal ideas/results] From the analysis of the requirements formulated by 44 participants of our empirical study, it follows that native language may have a negative effect on requirements quality, e.g., ambiguity, variability, and grammar issues. Furthermore, different native languages might drive to different quality issues. [Contribution] In order to prevent quality issues, our findings might be used by educators to adjust their materials to cater to different language groups, while practitioners might use them to improve their requirements review process.**

I. Introduction

SOFTWARE engineering is a diverse field, both in terms of research areas and worker backgrounds. This diversity is present in the industry, and companies are increasingly using English as their business language, no matter what country they are based in. University students are also globally mobile, with many who have the means often choosing to study all or part of their higher education abroad in English. This means that there is an increasing number of software engineers who are working or studying in a language that is not their native language.

Software engineers often use requirements specifications, either writing or developing systems from them, where the quality of the specification could determine the quality of the end product. The success of a software development project is said to depend on the quality of its requirements specification [1], [2]. Requirements are often written in natural language and, thus, the language used in that requirement could also have an effect on the quality of the specification.

The purpose of this study is to analyze natural language requirements written in English to determine (1) whether a author's native language has an effect on the quality of these requirements, and (2) which qualities are affected. In this paper, the term "native language" is defined as being the language of the country in which a person is born, raised, and receives early years of education. In an agile context, natural language requirements can either be written in the Software Requirements Specification (SRS) style or as user stories.

The findings from this study could support industry practitioners, research, and requirements engineering education. Targeted teaching and training could be developed to improve not only the overall quality of requirements but also to focus on the qualities that native speakers frequently have problems with. The study outcome could also help companies with requirements review processes, and quality checklists definition to identify or avoid requirements issues early on in development.

II. Background and Related Work

The IEEE Recommended Practice for Software Requirements Specifications [3] presents guidelines on how to produce "good" natural language SRS-style requirements. The guidelines detail eight characteristics that individual requirements should possess and five characteristics that a set of requirements should have. The recommended practice states that individual requirements should be: necessary; appropriate; unambiguous; complete; singular; feasible; verifiable; correct; and conforming (when applicable). A set of requirements should be: complete; consistent; feasible; comprehensible; and able to be validated. If an individual requirement or set of requirements violates one or more of these qualities, then it is not considered to be "good".

The INVEST criteria, originally discussed by Wake in 2003 [4], are specifically for evaluating the quality of user stories, rather than SRS-style requirements. According to the criteria, a user story should be: independent, negotiable, valuable, estimable, small, and testable [5]. If the story does not meet one or more of these criteria, then it is not of good quality.

There is a large body of work on requirements quality, with some focusing on specific qualities of a requirements specification and others giving a broader overview of what quality might be. Kiyavitskaya et al. [1] and Fabbrini et al. [6] take a detailed linguistic approach to identify ambiguity in requirements specifications. Antinyan et al. [7] focus on different requirements quality and developed a metric to measure the complexity of a requirement. With a broader look at all of the potential qualities of a requirements specification, Knauss et al. [8] developed a GQM approach to improving requirements quality. Genova et al. [2] also had a wider view of which requirements qualities to consider when creating the framework and tool for improving the quality of a requirements specification. However, while these studies were conducted in English, none of them looked at the linguistic background of the participants.

III. Research Methodology

Research Questions. Our study aims to answer the following research questions:
RQ1: Does the native language have an effect on the quality of natural language requirements?

- RQ1.1: Which requirements qualities are affected?
- RQ1.2: Do any particular languages have greater effects on requirements quality?

Participants and Data Collection. We aimed to find participants who had a software engineering background, and who could potentially be asked to write requirements. The participants were selected on the basis of convenience sampling. Survey participants were reached via the REFSQ 2022 conference, LinkedIn, Facebook, Twitter, Discord, and email, and via sharing the survey link with the students studying software engineering at the Universities the authors work for. Thus, the participants were a mix of students, researchers, and industry practitioners within software engineering. We created an online survey hosted on sosci.de. The survey was piloted by two representatives of the study participants. We decided to ask two students (those who might have the lowest experience with requirements) who gave feedback which was used to refine the survey questions. The first five questions in the survey were demographic questions. The sixth question was a simple domain description after which the participant was asked to write five natural language requirements (either SRS style or user stories) for the example domain. The survey questions and study material are available online [9].

Data Analysis. The qualitative data was analyzed using thematic coding as per Saldana [10] with two coding iterations. The thematic coding process used a coding dictionary that we created, which covered violations of any of a selected subset of the IEEE characteristics of individual requirements [3] or four of the INVEST criteria for user stories [4], [5]. When analyzing SRS-style requirements, we used the 2018 IEEE guidelines [3] that detail what good individual requirements should possess: correct; ambiguous; verifiable; necessary; appropriate; complete; singular; and feasible. The characteristic of "conforming" was not included in the analysis as the participants in the case study were not given a set template or writing style to follow. We chose to exclude the five characteristics for a set of requirements as we only asked participants to provide a sample of requirements rather than a complete requirements specification, and we did the analysis on each individual requirement. We also looked at whether a requirement is vague because we felt that being imprecise might not necessarily mean the requirement is ambiguous or unverifiable – it may just need more details or explanation.

For user story analysis, we used the INVEST criteria [4], [5]. "Independent" was excluded as it would require evaluation of the user stories as a set, while analysis was conducted on individual user stories. We also made note of whether the user story was correctly formed according to the Agile Alliance user story template [5]. As SRS-style requirements and user stories have different purposes and quality criteria, we did not use the SRS-style characteristics to analyze user stories, and the INVEST criteria were not applied to SRS-style requirements. The requirements in this study are in written form, and so we also considered language quality as a contributor to the overall requirements quality. Therefore, we applied codes for typos and grammar issues.

After the first author completed the first analysis pass, a sample of 10 randomly-chosen responses (a total of 50 requirements) was analyzed by the second author. Then, we came together to discuss any differences and how to improve the coding book. Coding was redone by the first author based on these discussions. Tab. I shows three examples of requirements received in the survey and the final codes that were applied. The final coding book with examples is available online [9]. Fig. 1 gives an overview of the thematic codes.

IV. Results

47 people answered the survey. However, three respondents did not complete the requirements writing task sufficiently; therefore, 44 survey responses were considered for the analysis with 220 requirements in total. For simplicity, and to aid comparison, we report percentages over all collected requirements (user stories and SRS-style ones), even though not all errors are applicable to all requirements.

Respondent Demographics. Fig 2 shows the native languages of our respondents. The majority of respondents had Polish as a native language, due to the third author sharing the survey link with the Master students of software engineering specialty. Swedish, Chinese, and English were the next most common native languages of respondents. Although there are many dialects and languages, the participants are known as students of Beijing University of Technology where the language of instruction is Beijing Mandarin.

In terms of roles within software engineering, 22/44 respondents were students of master-level studies who might be treated as novice requirements engineers. Industry practitioners were the next largest group with 8 participants, and there were also 5 Researchers. 9/44 respondents had multiple roles within software engineering: 6 were both a student and an industry practitioner; 2 were both a student and researcher; one person was an industry practitioner and a researcher.

Among the 14 respondents who selected the industry practitioner role as either their only role or as one of their multiple job roles, 4 stated their roles as "Developer" and 3 "Software Developer". There was one answer each for the following roles: "Senior Software Engineer"; "software engineer"; "System Architect"; "Technical project manager";

the 44 participants) given in the online survey, 233 codes were applied. This means that multiple codes were applied to some requirements. The four codes that were applied the most were: unverifiable (25.91% of all codes); ambiguous (21.82%); grammar issue (18.64%); and incorrect format (11.82%). The codes with the fewest applications (but more than 0) were: incorrect (0.45% of codes); unfeasible (also 0.45%); and inappropriate (0.91%).

Looking at Table II, the native Chinese speakers had by far the highest percentage of occurrence of unverifiable codes (46.67%). The native Arabic speakers had the second highest percentage (30%), and the native Polish speakers had the third highest percentage of unverifiable requirements with 28.24%. Native Arabic speakers had the highest percentage of ambiguity occurrences with 50% of the requirements given being coded as ambiguous. The Polish native speakers had the second highest percentage of ambiguous code occurrences with 28.24%.

Observation 2: There are four requirements qualities that were affected the most that are: verifiability, unambiguity, grammar correctness, and correct format.

Observation 3: Native speakers of Polish, Arabic, and Chinese introduced the highest number of errors.

Other Factors. In our survey, we collected data on other factors such as level of education, number of languages spoken, and mother tongue. We found that holding a Bachelor's degree as the highest level of education and speaking four or more languages had a negative effect on requirements quality. This data is omitted for space reasons, but results are available online [9].

V. Discussion

All participants in the study did make requirements quality errors, regardless of their native language. However, being a native speaker of Chinese, Arabic or Polish may have a negative influence on the quality of requirements that are written by those speakers. Two of these three languages have a writing system that is entirely different from English, which uses the Roman alphabet.

Unverifiability was the most common error made by the study participants and is a quality that often concerns Non-Functional Requirements (NFRs). The second most common error was Ambiguity. Althouth, as mentioned in Section II, ambiguity is a widely-researched topic within software engineering [11], [12], [1], [13], [14], the results from the study in the present paper suggest that continuing research and education in this area seems still needed.

The third most common error—grammar issues—could also be considered to be connected to ambiguity in some cases. Introducing grammar-checking tools and proofreading into the requirements writing process might help in preventing these errors. Then, there was the incorrect format error type as the survey participants did not use what is considered to be the standard user story format [5], [4]. Thus, using such frameworks and tools for improving user story quality [15], [16] might be valuable.

Chinese, Arabic and Polish appeared to have a greater negative effect on requirements quality than the rest of the languages in our studies. However, we cannot claim what is the root cause of this observation. It is necessary to investigate whether requirements quality is affected by the native language itself (linguistic differences), the level of English education, education within software engineering, or other factors. Future studies that discover the root causes might deliver guidelines for requirements for engineers and educators.

VI. Threats to Validity

Internal: Thematic coding brings threats to validity due to being subjective in its nature and subject to the bias and experience of the person doing the analysis. In order to mitigate this and minimize the threat, the second author received the coding dictionary that we created and independently coded a sample of 20% of the requirements obtained in the study. The English level of participants was not taken as the variable in the study, but we had an inclusion criterion– the participants need to have enough knowledge and skills so that they are able to either study or work in English.

External: The study may not have a large scope of generalisability as even though the survey was shared with non-students, a large portion of the data collection was reliant on students. However, it could be argued that the results from student data could be indicative of the software engineering industry as they frequently work and might be treated as novice employees.

VII. Conclusion

This study investigates whether native language has an effect on the quality of requirements. The results from the analysis of the online survey data suggest that native language may indeed have an effect on requirements quality as well as on the type of error introduced by the requirements writer. It follows from our study that more work and education need to be carried out on improving verifiability and ambiguity within requirements. Moreover, more training is needed also on how to write user stories so that they are well-formed. Grammar issues were also quite prevalent across all requirements. Our results might be used by practitioners to include quality checks of the errors in their review process and by educators to draw the attention of students to errors they might introduce and teach them how to prevent making those errors. Moreover, researchers might use our results to investigate the root causes of why native speakers of some languages make more errors than native speakers of other languages.

References

[1] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications," *Requir. Eng.*, vol. 13, no. 3, pp. 207–239, 2008.

[2] G. Génova, J. M. Fuentes, J. Llorens, O. Hurtado, and V. Moreno, "A framework to measure and improve the quality of textual requirements," *Requir. Eng.*, vol. 18, no. 1, pp. 25–41, 2013.

[3] "Iso/iec/ieee international standard - systems and software engineering – life cycle processes – requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1–104, 2018.

TABLE II
RAW DATA OF THE NUMBER OF ERRORS FOUND FOR EACH REQUIREMENTS QUALITY PER NATIVE LANGUAGE OF THE PARTICIPANTS

Native language [no. of participants]	Requirements quality code counts (% of requirements with quality code per native language)											
	Incorrect	Ambiguous	Unverifiable	Unnecessary	Inappropriate	Not singular	Unfeasible	Unnegotiable	Untestable	Incorrect format	Typo	Grammar issue
Polish [17]		24 (28.24%)	24 (28.24%)	2 (2.35%)	1 (1.18%)	7 (8.24%)			1 (1.18%)	5 (5.88%)	13 (15.29%)	23 (27.06%)
Swedish [7]		5 (14.29%)	5 (14.29%)					3 (8.67%)	5 (14.29%)	4 (11.43%)	1 (2.86%)	3 (8.57%)
Chinese [6]	1 (3.33%)	6 (20%)	14 (46.67%)			5 (16.67%)	1 (3.33%)			5 (16.67%)		4 (13.33%)
English [5]		4 (16%)	6 (24%)			1 (4%)		1 (4%)	4 (16%)	5 (20%)	3 (12%)	
Arabic [2]		5 (50%)	3 (30%)	1 (10%)							1 (10%)	2 (20%)
Cantonese [1]		3 (60%)	2 (40%)								1 (20%)	2 (40%)
German [1]										2 (40%)		
Greek [1]									1 (20%)	3 (60%)	1 (20%)	2 (40%)
Persian [1]				1 (20%)				1 (20%)		2 (40%)		2 (40%)
Brazilian Portuguese [1]			2 (40%)	1 (20%)		1 (20%)						
Amharic [1]		1 (20%)	1 (20%)								1 (20%)	1 (20%)
Korean [1]					1 (20%)	1 (20%)						2 (40%)

TABLE III
REQUIREMENTS QUALITY CODE OCCURRENCES OVER ALL 220 REQUIREMENTS RECEIVED IN THE ONLINE SURVEY

Requirements quality code	**Number (% of occurrence)**
Incorrect	1 (0.45%)
Ambiguous	48 (21.82%)
Unverifiable	57 (25.91%)
Unnecessary	5 (2.27%)
Inappropriate	2 (0.91%)
Not singular	15 (6.82%)
Unfeasible	1 (0.45%)
Unnegotiable	5 (2.27%)
Untestable	11 (5%)
Incorrect format	26 (11.82%)
Typo	21 (9.55%)
Grammar issue	41 (18.64%)
Total number of codes	233

[4] B. Wake, "INVEST in good stories, and SMART tasks - XP123," https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/, Aug. 2003, accessed: 2022-3-4.

[5] "What does INVEST stand for?" https://www.agilealliance.org/glossary/invest/, Dec. 2015, accessed: 2022-3-4.

[6] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool," in *Proceedings 26th Annual NASA Goddard Software Engineering Workshop*. IEEE Comput. Soc, 2002.

[7] V. Antinyan, M. Staron, A. Sandberg, and J. Hansson, "A complexity measure for textual requirements," in *2016 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA)*. IEEE, 2016.

[8] E. Knauss and C. E. Boustani, "Assessing the quality of software requirements specifications," in *2008 16th IEEE International Requirements Engineering Conference*. IEEE, 2008.

[9] F. Cowperthwaite, J. Horkoff, and S. Kopczyńska, "The Effects of Native Language on Requirements Quality - Additional Material," Feb. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7649140

[10] J. M. Saldana, *The coding manual for qualitative researchers*, 2nd ed. London: SAGE Publications, 2013.

[11] M. Bano, "Addressing the challenges of requirements ambiguity: A review of empirical literature," in *2015 IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE)*, 2015, pp. 21–24.

[12] V. Gervasi, A. Ferrari, D. Zowghi, and P. Spoletini, "Ambiguity in requirements engineering: Towards a unifying framework," in *From Software Engineering to Formal Methods and Tools, and Back*. Cham: Springer International Publishing, 2019, pp. 191–210.

[13] B. Gleich, O. Creighton, and L. Kof, "Ambiguity detection: Towards a tool explaining ambiguity sources," in *Requirements Engineering: Foundation for Software Quality*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 218–232.

[14] A. Bajceta, M. L. Ortiz, W. Afzal, P. Lindberg, and M. Bohlin, "Using nlp tools to detect ambiguities in system requirements - a comparison study," in *5th Workshop on Natural Language Processing for Requirements Engineering @ REFSQ*, March 2022. [Online]. Available: http://www.ipr.mdh.se/publications/6390-

[15] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Improving agile requirements: the quality user story framework and tool," *Requir. Eng.*, vol. 21, no. 3, pp. 383–403, 2016.

[16] ——, "Forging high-quality user stories: Towards a discipline for agile requirements," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE, 2015.

One-shot federated learning with self-adversarial data

Anastasiya Danilenka, Karolina Bogacka
0000-0002-3080-0303
0000-0002-7109-891X
Faculty of Mathematics and Information Science
Warsaw University of Technology
ul. Koszykowa 75, 00-662 Warsaw, Poland
Email: {anastasiya.danilenka.dokt, karolina.bogacka.dokt}@pw.edu.pl

Katarzyna Wasielewska-Michniewska
0000-0002-3763-2373
Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland
Email: Katarzyna.Wasielewska@ibspan.waw.pl

***Abstract*—Federated learning (FL) is a decentralized approach that aims at training a global model with the help of multiple devices, without collecting or revealing individual clients' data. The training of a federated model is conducted in communication rounds. Still, in certain scenarios, numerous communication rounds are impossible to perform. In such cases, a one-shot FL is utilized, where the number of communication rounds is limited to one. In this article, the idea of one-shot FL is enhanced with the usage of adversarial data, exploring and illustrating the possibilities to improve the performance of resulting global models, including scenarios with non-IID data, for image classification datasets: MNIST and CIFAR-10.**

I. Introduction

FEDERATED learning [1] is a popular research field that attracts thousands of researchers due to its simple, yet, open for improvements idea that is inline with current trends in distributed computing infrastructures. The core of federated learning lies in its collaborative nature, which allows multiple devices (clients) to use their own private data to jointly train one global model, managed by the centralized server. In general, the federated learning workflow can be summarized as follows: (1) a global model is initialized (during the first round) or aggregated (for subsequent rounds) on a server and sent to the set of client devices, (2) client devices receive the current version of the global model and use their private data to train the model for a set number of epochs, (3) each client returns resulting updates/weights/whole model back to the server, (4) server receives updates from clients and aggregates them into the new version of the global model. The client-based training round happens multiple times and is referred to as a communication round. At the end of each communication round the updated models are aggregated into the new version of the global model utilizing the federated averaging (FedAvg [1]) algorithm, which averages the updated models' weights. The federated averaging can also be easily combined with a weighting technique, for instance, its first version [1] weighted individual client's update based on the size of the local dataset this client possessed and used during training.

Communication between the centralized server and client devices is a well-known bottleneck for the federated learning pipelines [2], therefore, techniques for improving the convergence time [3], minimizing energy consumption [4] or improving network resource management scheme [5] were studied. One of the ways to mitigate the communication burden between the server and the client is to utilize the concept of few-shot learning.

Few-shot learning [6] is usually referred to as a learning technique where during training a model only sees a small portion of data (for instance, a few examples of each class in a classification task instead of a full dataset) and then is considered ready for performing testing/inference. In the case of federated learning, the few-shot learning idea restricts the number of communication rounds that happen between clients and centralized servers, e.g. one-shot federated learning implies only one communication round [7].

Despite few-shot learning techniques being able to drastically reduce the number of communication rounds needed to train the model, new questions arise, concerning the performance of the resulting models, since machine learning models usually require numerous epochs to reach the best possible accuracy. The problem of few-shot FL is further complicated by the privacy-preserving nature of the FL, which does not allow revealing any information about the local data that clients used during local training. In some cases, the problem of non-IID data can materialize, which can further damage the performance of the resulting global model [8], [9], [10]. Non-IID data can manifest itself in many ways. One of the possible classifications can be described as follows [9]: (1) quantity skew (different sizes of the local datasets that clients possess), (2) attribute skew (local datasets have unique distinct features for the same event/object they are describing, e.g. writing style), (3) label skew (only a subset of labels is present in local data, leaving some labels with no samples), (4) temporal skew (local datasets have time-dependent nature, e.g. were collected in different moments of time), (5) preference skew (same

The work of Anastasiya Danilenka and Karolina Bogacka was supported by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

event/object in different local datasets has a different target value due to subjective preference, e.g. ratings). Combinations of the different skews can also be present inside one scenario.

In this article, one-shot federated learning is researched in the context of aggregation of the clients' updated models and the possibility of using adversarial images as a source of client-picking guidance and performance improvement in the presence of label skew non-IID data.

II. Related works

As was stated in section I, the decision to drastically limit the number of communication rounds increases the importance of the aggregation algorithm, which can significantly influence the best possible performance on the test dataset. One of the first approaches to one-shot FL was to utilize ensemble learning, where each updated client model was treated as a part of the ensemble [7]. It was acknowledged, that in FL scenarios, the size of the ensemble of models depends on the number of participating clients, which can reach millions of devices. Moreover, not all clients are equally "useful" in terms of the data they have. Therefore, the ensemble of the models was restricted to a subset of models depending on the selection criteria. For example, models for the ensemble could be chosen randomly, or based on some indicator. One of the possible indicators for best candidates is the local test performance of the model, which requires the model to save a portion of its local data as a test set to measure the performance of the updated global model after local training on the remaining local data. Another similar technique uses a local cross-validation performance as a performance indicator for ensemble model picking.

Distillation technique was also researched with respect to one-shot FL. Data distillation was studied as an alternative to communicating whole models/model updates from clients back to the server [11]. Instead, each client, after receiving the global model used it to distill its own local data and sent the resulting set of distilled data and targets back to the server. Although communicating clients' data even in a distilled form that cannot be directly interpreted by the human eyes may be considered a violation of the clients' privacy, the authors state, that acquiring distilled data will not let the adversary replicate the resulting global model. After receiving the resulting datasets from all clients, the server uses them to train its own global model. Some enhancements were also presented in the process of data distillation, for instance, soft labels. Another distillation technique is proposed to treat clients' models as teachers and use them for training a student model on the server side. For instance, client devices can use their data to train conditional variational autoencoder (CVAE [12]). Moreover, the ensemble of these decoders is further distilled on the server into one decoder that can further be used as a data generator for training a global model on the server side [13].

The presented approaches to one-shot FL are capable of reaching a good final accuracy while preserving the benefits of the reduced number of communication rounds. Nevertheless, they may still require training additional models (e.g. encoders) or be prone to suffering from non-IID data (cross-validation ensembles). Therefore, a new way of performing one-shot FL can be of interest. This article presents an algorithm that can acknowledge the presence of label skew non-IID data and mitigate its effect on the final model, without requesting any additional data from the clients or imposing any additional computation on the client devices by using adversarial data.

III. Adversarial attack

Neural networks are susceptible to various kinds of adversarial attacks [14]. The attacks aim at misleading the trained models into incorrect predictions, by altering the perfectly correct source data sample in a way that is unrecognizable by the human eye. This changed source sample is referred to as an adversarial sample.

There are several methods for generating adversarial samples. The method that the adversary prefers can depend on how much information about the source model the adversary has. The attack methods that require full access to the target model gradients are called 'white-box" attacks, while attacks that can operate on a limited set of information from the source model, e.g. the predictions from the target model, are called "black-box" attacks. Among the most popular attack methods, one can name: the fast gradient sign method (FGSM) [15], its iterative version I-FGSM [16], momentum-enhanced MI-FGSM [17], Carlini and Wagner (C&W) [18] attacks, and more.

Moreover, with respect to image classifiers, attacks can also be divided into two categories based on the precise intention of the attack. The attack which aims solely to mislead the trained model into misclassifying an image into any class that is not the right one is called a non-target adversarial attack, while the attack that aims at making the classifier make a mistake by predicting a certain class, set by the adversary, is called a targeted attack.

One of the fascinating properties of the adversarial samples is their transferability [19] – adversarial samples generated for one target model can also mislead models which were trained to solve similar tasks. In other words, models with similar architectures that were trained on non-intersecting subsets of the dataset will most likely be successfully attacked by the same adversarial sample. The reason behind this phenomenon is that models that have similar tasks tend to come up with similar decision boundaries. This behavior can be valuable in terms of federated learning scenarios, where clients with similar datasets are training a set of individual models with respect to the same task objective.

To sum up, during the targeted adversarial attack, the adversary creates an adversarial sample by modifying the source sample based on the selected attack method (e.g. by using gradient-based algorithms like FGSM-family methods). These changes applied to the source sample force it to cross the decision boundary estimated by the target model [20], resulting in misclassification.

IV. Proposed approach

As discussed in section II, one of the possible approaches to one-shot FL is an estimation of the most successful client models and using them for making an ensemble of models instead of combining all clients' models into one global model version. The proposed approach described in this paper uses the knowledge retrieved from the adversarial samples to find the most promising clients to be included in the ensemble of models. A more complex algorithm based on the idea of adversarial data was described in a previous article [21] and is referred to as AdFL (Adversarial FL). Although the AdFL algorithm implies training in epochs, in this article the algorithm is adapted to a one-shot FL scenario and further extended to be used in an ensemble of models. The description of the algorithm is given in Algorithm 1.

Algorithm 1 Server-side of proposed one-shot FL, where n – positive size of the model ensemble, w_0 – initialized global model

```
Ensure: $\mathrm{w}_0$; clients are ready;
Require: n > 0;
for client in Clients do
    $\mathrm{w}_0^{client}$ ← run training on client($\mathrm{w}_0$)
end for
adv data ← create adversarial data($\mathrm{w}^{[0,...,Clients]}$)
$\mathrm{CS}^{[0,...,Clients]}$ ← calculate CS(adv data, $\mathrm{w}_0^{[0,...,Clients]}$])
$\mathrm{w}_0^{[0,...,Clients]}$ ← sort desc($\mathrm{w}_0^{[0,...,Clients]}$, $\mathrm{CS}^{[0,...,Clients]}$)
model ensemble ← $\mathrm{w}_0^{[0,...,n-1]}$
```

1) The server initializes the global model and sends it to all clients, participating in training.
2) Clients perform local training with the whole data they possess for the specified number of epochs.
3) Clients send the resulting updated models back to the server.
4) After collecting all updated models, the server uses them to generate $C \times N$ adversarial samples, where C – is the number of classes in the classification task and N – is the number of updated models returned from the clients as described in section IV-A.
5) Based on the generated adversarial samples, for each updated client model, a *coherence score* (*CS*) is calculated as described in section IV-B.
6) *CS* is further used as a performance indicator to identify the top-performing models and use them as an ensemble.

A. Adversarial samples generation

In order to identify clients that can potentially be useful for the ensemble, the proposed algorithm exploits the transferability property of adversarial samples that was described in section III. Still, per definition, adversarial samples are created on the basis of existing samples drawn from the source data. This condition is generally unacceptable for strictly privacy-preserving FL. Therefore, considering the absence of the source data on the server side where the creation of the adversarial samples happens, a random noise image is considered the starting point for adversarial sample generation.

The targeted MI-FGSM method is used in this article to generate adversarial samples [17]. The algorithm can be summarised in a few steps as described in formulas listing (1), where t is the current iteration of the method, x states for the input noise image, y – target class, to which the resulting image should eventually be attributed, g_t – accumulated gradients through previous t iterations, θ – source model (in case of presented algorithm, one of the updated clients' models), and J – loss function (e.g. cross-entropy).

$$g_{t+1} = \mu * g_t + \frac{\nabla_x J(\theta, x_t^*, y))}{||\nabla_x J(\theta, x_t^*, y))||_1} \quad (1)$$

$$x_{t+1}^* = x_t^* + \alpha * sign(g_{t+1}) \quad (2)$$

The algorithm is parameterized by a number of parameters, namely, μ – decay factor, α – step size. Moreover, the resulting x_{t+1}^* image is further clipped at the end of each iteration in the clipping range e. The parameters used in this paper are different from those presented in the referenced paper, due to the different intentions for the resulting adversarial samples. Initially, adversarial samples are created to perform an attack on trained classifiers during inference time, but in the case of the one-shot FL algorithm described in this paper, the target is the transferability measure of the samples. Therefore, the constraints on the amount of the changes applied to the source image were loosened to allow more gradient information to be added to the adversarial sample. For instance, the number of MI-FGSM iterations was increased to 30, and step size α was increased to 1.

B. Coherence score

After local training, clients return the resulting updated models back to the server, where each model is used to generate one adversarial sample per class in the classification task. These samples are further used to estimate their transferability across all models. The idea behind this action is to find models that can generate transferable samples and are susceptible to samples, generated by other models. This two-side transferability might come from the similar decision boundaries that were learned during the local training step as noted in section III. Therefore, it can be assumed that such models learned somehow in a similar way.

The two-sided transferability measure is called a coherence score (CS) and is calculated out of two separate measures, each of which summarizes either the ability to create or the ability to identify adversarial samples.

The ability of a certain model to produce adversarial samples that are being recognized by other models participating in training is measured according to equation (3), where k stands for the model that was predicting adversarial samples generated by the source model, c stands for a target class that the adversarial sample was made to represent. The measure is calculated across all models from the training set with respect to the adversarial data, generated by the source model for which the measure is computed. The predictions of this source model for its own data are omitted.

$$\text{was predicted} = \sum_{k=1}^{K} \sum_{c=0}^{C-1} \text{is correct}_{k,c} \cdot \text{returned prob.}_{k,c} \quad (3)$$

This equation uses a binary feature to identify if the prediction of the adversarial sample was correct or not. This binary flag is then multiplied by the confidence of the prediction. Therefore, no punishment is done due to the wrong prediction and less certain predictions will accumulate less significance.

The ability of the model to correctly predict classes of adversarial data generated by other models is measured according to equation (4), where k stands for the model that generated the adversarial data and c stands for the target class of the adversarial sample.

$$\text{predicted others} = \sum_{k=1}^{K} \sum_{c=0}^{C-1} \text{is correct}_{k,c} \times \\ \times \text{returned prob.}_{k,c} \quad (4)$$

Again, the results of the model predicting its own adversarial data are omitted.

The resulting coherence score is a simple summation of the two previously described measures (equation (5)).

$$\text{coherence score} = \text{predicted others} + \text{was predicted} \quad (5)$$

Each of the models returned by the clients, participating in training, acquires its own coherence score and the list of models can then be easily sorted based on the resulting measure. The models that scored higher in CS are treated as those that provide more value to the model ensemble and are picked first. On the other hand, the coherence score can also be used for weighted federated averaging to create one global model instead of an ensemble.

V. Experimental section

To illustrate the efficiency of the presented adversarial-based method in one-shot federated scenarios, a series of experiments were performed on two datasets for image classification tasks: MNIST and CIFAR-10.

The results of the presented approach are compared to three algorithms: classic federated learning (FedAvg), and two ensembles – an ensemble of random models and a cross-validation ensemble.

In all experiments, only 10% of models were included in the ensemble, making the number of models per ensemble equal to 5. For cross-validation, during the local training part, 5 validation folds were performed, with train/validation data division being 80/20 respectively. The validation performance during cross-validation was collected. After that, the model was trained once again with the whole local data and sent back to the server together with validation metrics.

A. Data partition

Each of the experiments featured 50 federated clients that had approximately 400 images from the training dataset as their local data. The local datasets were created in a non-intersecting manner. Moreover, all classes inside the local dataset had the same number of instances (locally balanced dataset). The local training round consisted of 15 epochs, with the optimizer set to Adam, with starting learning rate of 0.001. For the test performance estimation, test sets provided by the datasets were used.

As was mentioned in section I, non-IID data can create additional challenges to federated pipelines. Therefore, during the experiments, both IID and non-IID data partition scenarios were examined. In the case of IID data, all classes were equally presented in the local datasets of clients. The label-skewed non-IID data partition was emulated by constructing each client's local dataset only from the limited number of classes. In both MNIST and CIFAR-10 experiments, the number of unique classes presented in the local dataset was limited to 4. So, before starting the FL pipeline, each client's class set was constructed individually by sampling 4 classes from the set of all classes. For each class, a probability of its occurrence in the local dataset is drawn from a normal distribution. The example of probability distribution used for CIFAR-10 experiments is showed in Figure 1.

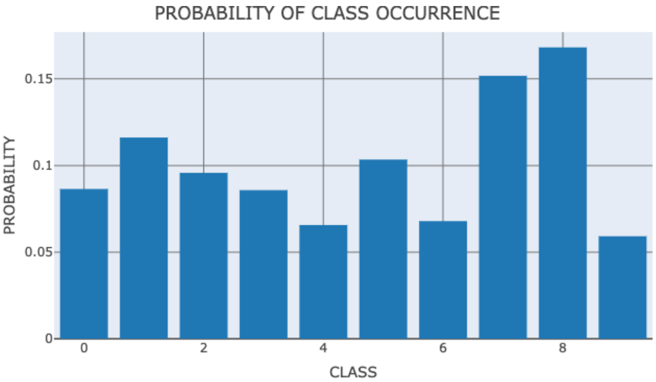


Fig. 1. Class occurrence probability for CIFAR-10 label skew experiments

Due to the custom distribution of the probabilities of class occurrence, some classes were less represented globally across client devices, while others – more.

B. Model configuration

For the MNIST classification task, a simple LeNet5 [22] configuration was used for all the experiments. As for the CIFAR-10, a custom configuration of a Convolution Neural Network was implemented, featuring 6 convolution layers, each pair followed by a maximum pooling layer, at the end followed by three dense layers.

C. Results

All experiments were performed at least 15 times to better capture the statistical significance of the results. As the training process consists only of one epoch, the final performance of the models was summarised across multiple runs. For the MNIST dataset and IID data partition, the result is showed in Figure 2.

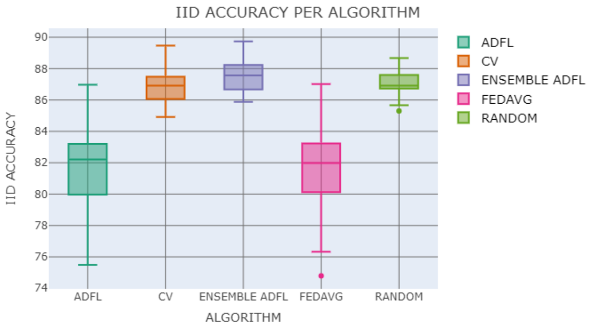


Fig. 2. Test accuracy comparison for IID MNIST experiment

It is seen, that, in general, all algorithms managed to get more than 80% of accuracy, while centralized models on the MNIST dataset reach up to 98% of accuracy. Still, some algorithms scored higher than others: ensembles of models performed better than aggregated global models – with median accuracy for AdFL, cross-validation and random ensembles being 87.6%, 86.9%, and 86.9% respectively, while aggregated FedAvg and AdFL achieved 82% and 82.2%, respectively.

In contrast, the non-IID scenario shows a different behavior, as showed in Figure 3 in addition to a way smaller resulting accuracy across all algorithms.

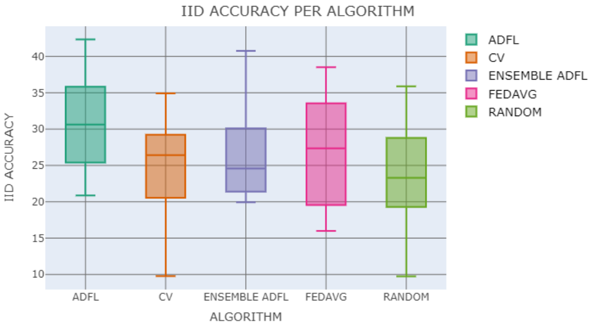


Fig. 3. Test accuracy comparison for non-IID MNIST experiment

Here, aggregated versions of the models perform better than the ensembles, with AdFL and FedAvg reaching 30.6% and 27.4%, and AdFL, cross-validation, and random ensembles reaching 24.6%, 26.4%, 23.3%, respectively.

This difference in behavior on varying datasets depending on either presence or absence of non-IIDness, may come from the fact that individual model evaluation cannot spot the non-IID clients. Therefore, it is not guaranteed that models which were exposed to heterogeneous data during training will appear in the ensemble.

To identify if this behavior can be replicated, the same experiment was performed on the CIFAR-10 dataset. The results for the IID data partition are showed in Figure 4.

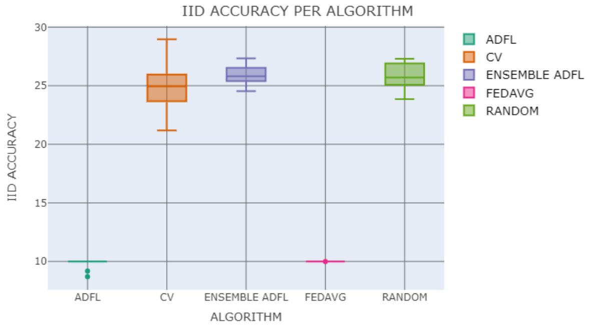


Fig. 4. Test accuracy comparison for IID CIFAR-10 experiment

In this case, again, ensemble versions perform better than aggregated models in the presence of IID data. AdFL, cross-validation, and random ensembles reached 25.8%, 24.9%, and 25.6%, respectively, while aggregated versions could not manage to achieve any meaningful results in the provided scenarios.

However, when examining the results for the CIFAR-10 non-IID scenario (Figure 5), the results differ from those observed on the MNIST dataset with non-IID data.

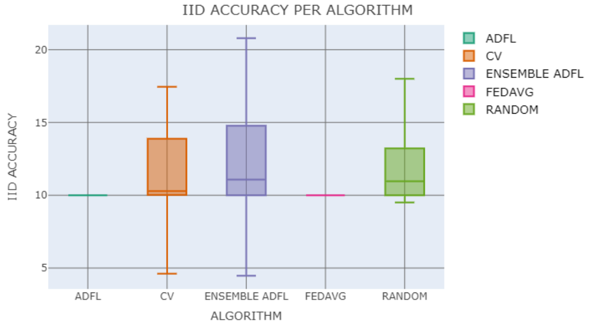


Fig. 5. Test accuracy comparison for non-IID CIFAR-10 experiment

In the CIFAR-10 non-IID scenario, contrary to MNIST, aggregated models (AdFL and FedAvg) still could not get any meaningful performance, while ensemble methods showed low, but, somehow diverse across experiments, accuracy with AdFL, cross-validation, and random ensembles achieving 11.1%, 10.3%, and 10.9% median accuracy, respectively. Although median accuracy is low, maximum accuracy for AdFL, cross-validation and random ensemble reached 20.8%, 17.5%, and 18%. This low performance may be a sign that the proposed task was overly complex and, therefore, may need more experiments with bigger local datasets or require some knowledge transfer techniques.

VI. Conclusion

In this work, a new approach to building a model ensemble for one-shot federated learning was introduced and compared with other ensembling techniques for both IID and non-IID scenarios. It was observed, that, for the MNIST dataset, in the presence of IID data, presented ensembling techniques achieve better performance than the aggregated models, but for non-IID data the situation is opposite. For the CIFAR-10 dataset, the studied non-IID scenario presented a complicated scenario and did not replicate the results of the MNIST dataset. Still, the described technique utilizing adversarial data shows similar or better performance when compared to other algorithms with respect to test accuracy for both MNIST and CIFAR-10 image classification tasks. Further research may inspect other non-IID data scenarios, use more sophisticated model architectures and datasets, and improve the ensemble construction technique.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[2] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao, "Communication efficiency in federated learning: Achievements and challenges," 2021.

[3] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2021. doi: 10.1109/TWC.2020.3042530

[4] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2021. doi: 10.1109/TWC.2020.3037554

[5] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, p. e2024789118, 2021. doi: 10.1073/pnas.2024789118. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2024789118

[6] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, p. 594–611, apr 2006. doi: 10.1109/TPAMI.2006.79. [Online]. Available: https://doi.org/10.1109/TPAMI.2006.79

[7] N. Guha, A. Talwalkar, and V. Smith, "One-shot federated learning," 2019.

[8] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *CoRR*, vol. abs/1806.00582, 2018. [Online]. Available: http://arxiv.org/abs/1806.00582

[9] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *CoRR*, vol. abs/2106.06843, 2021. [Online]. Available: https://arxiv.org/abs/2106.06843

[10] C. Xiao and S. Wang, "An experimental study of class imbalance in federated learning," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, dec 2021. doi: 10.1109/ssci50451.2021.9660072. [Online]. Available: https://doi.org/10.1109\%2Fssci50451.2021.9660072

[11] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu, "Distilled one-shot federated learning," 2021.

[12] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros da Silva, and S. Lima Netto, *Variational Autoencoder*. Cham: Springer International Publishing, 2021, pp. 111–149. ISBN 978-3-030-70679-1. [Online]. Available: https://doi.org/10.1007/978-3-030-70679-1_5

[13] C. E. Heinbaugh, E. Luz-Ricca, and H. Shao, "Data-free one-shot federated learning under very high statistical heterogeneity," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=_hb4vM3jspB

[14] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020. doi: https://doi.org/10.1016/j.eng.2019.12.012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S209580991930503X

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014. [Online]. Available: https://arxiv.org/abs/1412.6572

[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017.

[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," 2018.

[18] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. doi: 10.1109/SP.2017.49 pp. 39–57.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[20] O. Suciu, R. Marginean, Y. Kaya, H. D. III, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018. ISBN 978-1-939133-04-5 pp. 1299–1316. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/suciu

[21] A. Danilenka, "Mitigating the effects of non-iid data in federated learning with a self-adversarial balancing method," 2023, submitted to publication.

[22] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791

DOI: 10.15439/2023F6549
ISSN 2300-5963 ACSIS, Vol. 35

Mitigating the effects of non-IID data in federated learning with a self-adversarial balancing method

Anastasiya Danilenka
0000-0002-3080-0303
Faculty of Mathematics and Information Science
Warsaw University of Technology, Warsaw, Poland
Email: anastasiya.danilenka.dokt@pw.edu.pl

***Abstract*—Federated learning (FL) allows multiple devices to jointly train a global model without sharing local data. One of its problems is dealing with unbalanced data. Hence, a novel technique, designed to deal with label-skewed non-IID data, using adversarial inputs is proposed. Application of the proposed algorithm results in faster, and more stable, global model performance at the beginning of the training. It also delivers better final accuracy and decreases the discrepancy between the performance of individual classes. Experimental results, obtained for MNIST, EMNIST, and CIFAR-10 datasets, are reported and analyzed.**

I. Introduction

FEDERATED learning (FL) was introduced in [1]. It aims at creating a shared model, combining information from multiple sources, without sharing local data. In standard FL, training proceeds in rounds. Here: (1) global model is initialized, (2) current version of the global model is sent to selected clients, (3) they complete training, using their local data, (4) model updates are gathered on the server and used to generate a new version of the global model [1]. In this work, complete models are communicated (in (2) and (4)).

How to aggregate updates is a subject of intensive research. The basic approach is to average updates (using the FedAvg algorithm). A natural extension to FedAvg is weighting, i.e. assigning individual importance to each client, based on additional knowledge. For example, clients with more local data receive larger weights for their updates [1].

In FL, the complete dataset is never "known". Hence, the statistical properties of local and global datasets are unknown. Therefore, it cannot be established if data is identically independently distributed (IID). However, it has been established that non-IID datasets negatively affect the quality of the FL-trained model [2]. Non-IID data can be classified on the basis of the source of heterogeneity [2], i.e.: (1) data quantity skew (local datasets differ in size), (2) label distribution skew (different devices have different subsets of labels inside local dataset), (3) attribute skew (local data has unique characteristic features, noise, perturbations, etc.), (4) temporal skew (local data distributions differ over time, or data was collected in different time periods). Obviously, combinations of skews can materialize. This work concentrates on non-IID datasets with the label distribution skew. Depending on how many labels are represented in the local dataset, the skew can be extreme, with only 1 label in each dataset, to a $(C-1)$-label skew, where C is the number of labels in the set. Here, each local dataset lacks data for exactly one label. In research experiments, the data partition strategy determines, which label(s) is(are) missing.

In this context, a novel method, to overcome the problems caused by the label distribution skew, called *Adversarial Federated Learning* (AdFL), is proposed. It is inspired by adversarial attacks and is applicable, primarily, to neural networks applied to image data. To the best of our knowledge, it is the first attempt to recover local data distribution information, from the clients, using adversarial inputs and to use adversarial images to coordinate the training process.

In what follows, in Section II, related research is outlined. Section III, introduces adversarial attacks followed, in Section IV, with the description of the AdFL algorithm. Section V, presents the experimental setup and results, obtained with MNIST [3], EMNIST [4] and CIFAR-10 [5]. Conclusions and directions for future research complete this work.

II. Related work

When the problem of data heterogeneity was acknowledged, the quantity skew was mitigated by weighting the updates, based on the number of data samples in local datasets. In other approaches, to deal with the problem of non-IID data, FedProx [6] and SCAFFOLD [7], tackled the problem by restricting local model updates using proxy terms and control variates. Moreover, use of gradient correction [8], utilization of knowledge distillation [9], applied client picking [10], data sharing [11] and adapted loss functions in presence of data imbalance [12] have been explored.

While many methods reported improvements on standard datasets (MNIST, CIFAR-10, CIFAR-100), still, (i) some required additional information from clients (e.g. distribution information, averaged data), (ii) others rely on external datasets (representative of the local data), or (iii) involve training of additional, data generator, models. This may not be feasible in the real world or may introduce new privacy risks. Hence, the proposal is articulated in what follows.

III. Adversarial federated learning

In FL, in the past, adversarial techniques were used to generate data (a) to improve resistance to adversarial attacks [13], or (b) to increase the amount of locally available data [14].

Here, a different way of integrating adversarial data into FL is proposed.

A. Adversarial attack

Adversarial attacks make it possible to alter a sample from the training data, in a way that is undetectable to humans, so that the network will misclassify such (previously classified) sample [15]. In other words, when the attack is performed on a trained model then, by modifying the target sample, the adversary tries to make a valid target sample cross the decision boundary of the classifier [16], and be misclassified.

Depending on the applied changes, one can distinguish *non-targeted* and *targeted* attacks. A non-target attack aims at making the model deliver incorrect predictions. Targeted attacks set the class, to which the misclassification should be attributed. Separately, *white-box* attacks can access the model's architecture and parameters, while *black-box* can access only the model's output. Among the most famous attacks are: one-step Fast Gradient Sign Method (FGSM) [17], its iterative version I-FGSM [18], and its version enhanced with momentum MI-FGSM [19].

B. Transferability of adversarial inputs

An important property of adversarial inputs is their transferability, i.e. adversarial inputs generated for one model will mislead also other model(s) trained on similar datasets, to solve similar tasks. Note that in FL clients share the same task, model architecture, and data space. Hence, adversarial samples, generated by all clients, should be transferrable. To establish this, a series of experiments, to measure the attack success rate (ASR) – a common metric for qualifying the performance for adversarial attacks [19] – was performed. In these experiments, and in what follows, MI-FGSM was used to create targeted adversarial attacks [19]. Overall, data was IID distributed among 40 clients (all clients had samples of all classes in local datasets). During each server-side epoch, 10 clients were selected, according to the strategy described in Section IV, and performed local training. Next, clients' models (returned to the server) were used to generate one adversarial sample per class (see, Section IV-A). Hence, $10{*}C$ adversarial images were generated, where C is the number of classes. Next, updated clients' models classified the adversarial samples, and if the predicted class was equal to the target, the attack was qualified as successful. Table I shows the transferability metric for standard datasets (MNIST, EMNIST, and CIFAR-10) during the 5th, 25th, and 50th epochs.

TABLE I
ATTACK SUCCESS RATE (ASR) STUDY INSIDE A FEDERATED SCENARIO

	Epochs		
Dataset	5	25	50
MNIST	1	1	1
EMNIST	0.98	1	1
CIFAR-10	0.5	0.99	1

As can be seen, ASR is high, therefore, due to the high transferability of adversarial data and its connection with the decision boundaries, it is possible to derive insights about the local data, by asking clients' models to produce adversarial images, while "keeping private information private". This observation became the foundation of the AdFL algorithm.

IV. AdFL ALGORITHM

The AdFL algorithm uses adversarial images as a source of additional knowledge about FL training. To do so, the adversarial data is generated first. Here, there are two places where adversarial images can be generated. (1) Clients can produce them, as they have access to all necessary data. (2) Server can use the updated models to generate them. In AdFL, adversarial data is generated on the server, using a random noise image as a starting point. The server-side part of the AdFL algorithm is summarized in Algorithm 1.

Algorithm 1 AdFL algorithm (Server); Cl states for client; Cl_e – subset of clients picked for training on epoch e; *global distribution* tracks distribution of classes during FL training; $distr_{all}$ – estimated classes presence in clients' local datasets

```
Ensure: global model $w_0$, global distribution, clients ready
  for $e$ in $epochs$ do
    if $e == 0$ then
      $Cl_e \leftarrow$ all clients
    else
      $Cl_e$, global distribution $\leftarrow$ pick clients($distr_{all}$, global distribution)
    end if
    for $Cl$ in $Cl_e$ do
      $w_e^{Cl} \leftarrow$ run training($w_e$)
    end for
    adv data $\leftarrow$ create adversarial data([$w_e^0$, ..., $w_e^{Cl_e}$])
    if $e == 0$ then
      $distr_{all} \leftarrow$ estimate distribution(adv data)
    end if
    $CS_{[0-Cl_e]} \leftarrow$ calculate coherence(adv data, $w_e^{[0,..,Cl_e]}$)
    $w_e \leftarrow$ FedAvg([$w_e^{[0,..,Cl_e]}$], $CS_{[0-Cl_e]}$)
  end for
```

Overall, there are 6 steps that define the AdFL algorithm. Note that all AdFL-specific steps take part on the server, with no additional Client-side computations.

1) During "warm-up", round **all** clients perform local training, and return models to the server. In subsequent rounds, local training is completed by a subset of clients on the received version of the global model.
2) On the server, updated clients' models generate adversarial samples, as described in Section IV-A
3) Generated adversarial samples are used to estimate the distribution of classes across clients (see, Section IV-B).
4) Coherence scores (CS) are calculated, based on updated models and adversarial samples (see, Section IV-D).
5) CS are used as weights during aggregation, resulting in the next global model, sent to the clients.

6) From there on, the subset of clients that participate in the training is defined by the client-picking strategy (Section IV-C) and the process repeats.

A. Adversarial inputs creation

To create adversarial data, the data-free approach has been selected, to protect clients' data. Hence, the initial data source is a random noise image that, by default, is not classifiable. Starting from this image, a targeted adversarial attack is performed, using the MI-FGSM. Here, the adversarial image creation lacks malicious intent, becoming less sensitive to the amount of allowed changes. Hence, the MI-FGSM parameters have been aligned with the pursued goal. Overall, each updated client model generates C images representing classes that are present in the task. The steps of adversarial input generation are presented in Algorithm 2, with the default federated steps omitted.

Algorithm 2 Adversarial data generation

```
Ensure: $targets \leftarrow [0, ..., C-1]$
Ensure: $w_e^{[0,..,Cl_e]}$ {Clients' updated models during epoch $e$}
  for $target$ in $targets$ do
    for $w_e^i$ in $w_e^{[0,..,Cl_e]}$ do
      $adv\ img \leftarrow$ rand noise[$Ch$, $H$, $W$]{Random sample}
      for $step$ in $num\ steps$ do
        $adv\ img^i_{target} \leftarrow$ step($w_e^i$, $adv\ img^i_{target}$, $target$)
      end for
    end for
  end for
```

B. Local distribution estimation

Based on generated adversarial data, it is possible to estimate the class distribution among the clients. Here, it was established that predictions of adversarial samples, returned by the clients' updated models, are illustrative of the presence of certain classes in the local dataset (of this client). Therefore, uncovering class presence within clients' data can be used for balancing the label-skew. It is also the reason why adversarial sample generation runs for a set number of steps (30). If the target class is missing from the client's local dataset, the model will fail to generate an adversarial sample of that class. Moreover, using this knowledge, AdFL performs a warm-up epoch, by initiating training on all clients. This allows capturing a meta-level picture of class presence across clients, gathering the data needed for the client-picking routine that will occur in each training round.

After the updated models return to the server, adversarial data generation occurs, and the results of all models' predictions are used to estimate class distributions. Here, all updated models generate adversarial samples and predict the resulting samples. Next, for each model, its predictions are summarised and classes that appeared in the predictions are treated as signs of these classes being present in the client's dataset.

A set of experiments was performed to establish how precise the estimation of classes' presence in local datasets is. The experiments were performed for two extreme cases: IID setup (all classes are present), and $\leq$20% of classes setup, i.e. two classes per client for MNIST and CIFAR-10 datasets, and 12 classes for the EMNIST dataset. The effectiveness was measured as the percent of classes detected from the adversarial data in total, across all clients during the warm-up FL training round, against the average number of classes detected per client. The results are listed in Table II.

TABLE II
DETECTED CLASSES (DC) METRICS FROM ADVERSARIAL DATA

	MNIST		CIFAR-10		EMNIST	
Metric	IID	2	IID	2	IID	12
DC (%)	100	100	100	100	100	97.1
Avg. DC	10	2	10	2	62	11.4

Although the experiments show that class detection can be performed quite accurately, it remains only an estimation of the actual classes' presence. Moreover, the quality of the adversarial samples depends on the number of local training epochs. Thus, tuning this parameter is important for obtaining a proper class distribution prediction. Hence, for the reported experiments, the number of local epochs is set to 10 for MNIST and EMNIST and 2 for CIFAR-10.

C. Client-picking strategy

After predicting classes that are present in the local datasets, a client-picking strategy that will mitigate the presence of local label-skew can be proposed. Based on the estimation of classes' presence in local datasets obtained during the "warm-up" round, a simple approach to balance the training process was designed. Specifically, the balance of classes during the training process is maintained by the global label frequency vector of size C, where C is the number of unique classes in the classification task. This vector is updated in each FL training epoch after a client is selected to be involved in the current training epoch. It reflects the frequency of a particular class being picked for training. To ensure equal exposure for all classes, the clients for each FL round are selected to make the values in C close to a uniform distribution. Here, a Kullback–Leibler (KL) divergence is used (for details, see [20]). This allows each FL round to include clients with rare classes in their data, by computing the KL-divergence of the global classes frequency, with respect to the uniform distribution, if a client (its data classes) is to be added to the training round. In each round, clients with the smallest KL divergence are picked. Here, note that several clients may have the same KL divergence (possibly, a minimum for the current set of clients). In this case, random client selection is applied.

D. Clients coherence measurement

Another outcome of the transferability of adversarial samples is an ability to identify "problematic clients", i.e. clients, whose models are not able to produce or identify transferable adversarial samples. In order to measure the "two-side transferability", the coherence score (CS) measure was defined.

CS calculation can be described as follows: (1) at the end of the training round, all updated models create adversarial samples for each target class, (2) each updated model predicts all adversarial samples produced by models participating in the round, (3) for each updated model the CS consists of two parts: (i) describing how good this model recognized adversarial samples from other models, and (ii) how successfully other models recognized this models samples. Therefore, models with high CS excel in both creating transferable samples and correctly classifying those created by others.

Here, the ability of the updated model to predict adversarial images produced by other models is measured according to Equation 1, where each multiplication consists of a binary flag indicating whether or not the prediction for class c generated by model k was correct, and the probability returned by the model. Predicting own inputs is omitted.

$$\text{predicted others} = \sum_{k=1}^{K} \sum_{c=0}^{C-1} \text{is correct}_{k,c} * \text{returned prob.}_{k,c} \tag{1}$$

A similar measure was applied to evaluate the ability, of the updated model, to produce adversarial images that are recognized by the other models. The individual CS is a result of a simple summation of the two scores. After the coherence scores, for all clients, are calculated, they are normalized and used as weights for the FedAvg aggregation.

V. Experimental setup, results, and their analysis

A. Data partitioning

During experiments, label-skew was simulated using three parameters: (1) number of unique classes in dataset, (2) total number of data samples in dataset, and (3) probability of class appearing in data. The number of unique classes inside the local dataset is fixed for an experiment, and all clients have the same number of unique classes. However, the set of local labels differs between clients. Total number of samples is also fixed, and all clients have the same amount of data. Moreover, local data is equally divided among classes, e.g. for 2 classes, 50% of data will be of class 1, and 50% of class 2. The probability of all classes appearing in a local dataset is defined, by drawing a sample from the normal distribution for each of the classes. Next, the resulting values are normalized to represent the probability vector. To determine the classes of a specific client, a random subset of the set size is drawn. Since the normal distribution was used, borderline cases may materialize, when few classes have a high probability of appearing. Therefore, they will be overrepresented, while a few other classes may be left out, because of their extremely low probability of occurrence. Here, random seeds were used to ensure robustness. An example of the probability of occurrence for 10 classes is presented in Figure 1.

Note that this data partitioning scheme introduces a challenging label-skew scenario, as it allows some classes to be "common" in the clients' population, while others are rare, therefore, producing a global class imbalance.

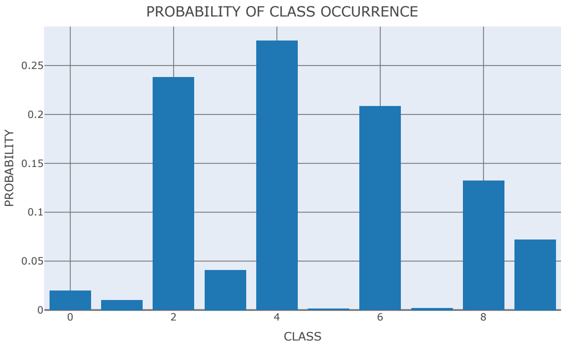


Fig. 1. Probability of occurrence for 10 classes

B. Models and hyperparameters

In the experiments, Convolutional Neural Networks (CNN) were used. For MNIST and EMNIST, a LeNet-5 architecture was used [21]. For CIFAR-10, the pre-trained version of the mobilenetv2 [22] model was used, provided by the torchvision package, with weights coming from Imagenet dataset [23]. The pre-trained version of the mobilenetv2 model was chosen to verify the applicability of the algorithm to more complex datasets and architectures that are not trained from scratch. If not stated otherwise, the cross-entropy loss function was used. The Stochastic Gradient Descent (SGD) was used as the optimizer. Model and algorithm-specific hyperparameters are presented in Table III.

TABLE III
Hyperparameters used in the experiments

Parameter	MNIST	EMNIST	CIFAR-10
Num. classes per client	2	12	2
Total classes	10	62	10
Clients per training round	10	10	10
Total clients	40	40	40
Data samples per client	400	1200	400
Learning rate	0.001	0.001	0.001
Batch size	10	10	10
Num fed epochs	10	10	2
FedProx μ	0.1	0.1	0.1
FedMix M	400	1200	400
FedMix λ	0.2	0.2	0.05
Num of adv. steps	30	30	30

C. Experimental setup

The performance of AdFL was tested on three datasets: MNIST, EMNIST, and CIFAR-10. The project, including both AdFL and other algorithms, was implemented in Python (3.7.9), using PyTorch (1.10.0). Moreover, for ready model architectures and datasets, torchvision (0.11.1) was used.

D. Experimental results and their analysis

AdFL performance is compared to FedAvg, FedProx, and FedMix algorithms (Section II). The parameters for FedProx and FedMix were as in [11]. The experiments were performed 10 times for MNIST and 6 times for EMNIST and CIFAR-10, with different random seeds, while preserving data partition. For MNIST, the resulting median test accuracy, over a set of experiments, is presented in Figure 2.

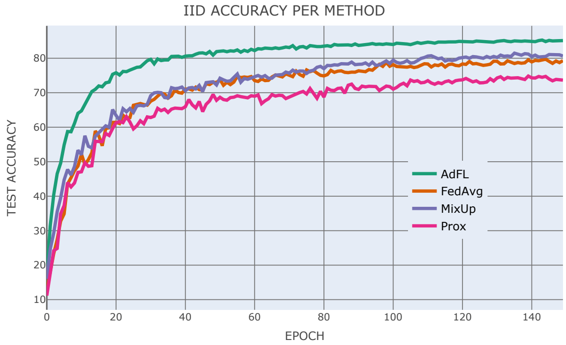


Fig. 2. Test accuracy for MNIST

AdFL improves model performance at the beginning of the training, and ensures stable performance, with slight accuracy improvement, later. This can be valuable in the case of the limitation of client-server communication rounds. The minimum accuracy improvement is about 3.3%.

For EMNIST, the median test accuracy of different algorithms is presented in Figure 3. Again, AdFL is more stable at the beginning and shows gradual improvement without significant peaks. It also results in test accuracy improvement of around 2% (compared to FedAvg).

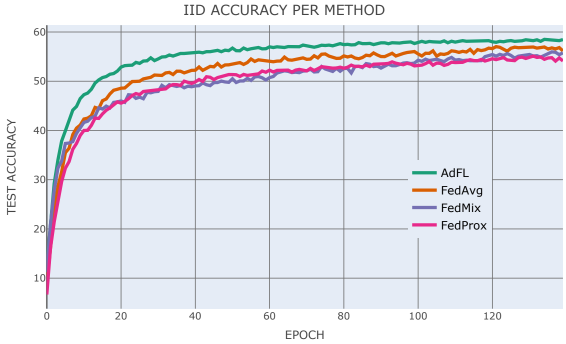


Fig. 3. Test accuracy for EMNIST

Note that, with a very challenging label-skew scenario, the performance of the models depends directly on the probabilities of class occurrence and how they are distributed among clients. Here, an additional set of 7 EMNIST experiments was performed, where the random seeds were changed each time before generating probabilities of classes occurrence and classes distribution, while preserving other parameters listed in Table III. The EMNIST dataset was chosen for testing, as it has 62 classes (as opposed to 10 classes in the remaining datasets). The average accuracy improvement and the Wilcoxon signed-rank test [24] results are presented in Table IV. As can be seen, the accuracy improvement remains for varying data partitions, with respect to FedAvg, FedProx, and FedMix algorithms.

TABLE IV
AdFL PERFORMANCE STUDY ON VARYING EMNIST DATA PARTITIONS

	FedAvg	FedProx	FedMix
Acc. impr.	1.70%	3.87%	1.88%
Std	1.11	1.61	2.91
p-value	0.0469	0.0156	0.0313

For CIFAR-10, the median accuracy is shown in Figure 4. It can be seen that, compared to previous datasets, training was less stable and the results show accuracy fluctuations even for a median of results. Still, on average, AdFL shows better accuracy from the very start of the training, resulting in a final accuracy improvement of around 2%.

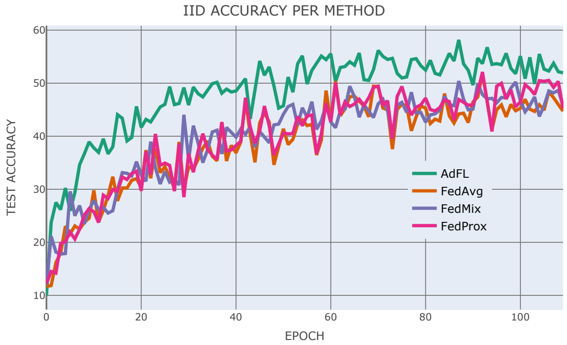


Fig. 4. Test accuracy for CIFAR-10

The summary of all experiments is presented in Table V together with the standard deviation of the final accuracy. The statistical accuracy improvement, related to AdFL, was measured with the Wilcoxon signed-rank test, on results presented in Table V and is depicted in Table VI.

TABLE V
EXPERIMENTS SUMMARY

	MNIST		CIFAR-10		EMNIST	
	ACC	STD	ACC	STD	ACC	STD
AdFL	56.44	0.51	51.77	2.36	56.67	0.41
FedAvg	53.71	1.75	47.70	2.53	53.39	1.38
FedProx	52.28	1.39	48.42	3.21	51.99	1.09
FedMix	51.03	1.53	49.13	2.30	50.81	1.47

TABLE VI
WILCOXON SIGNED-RANK TEST P-VALUE

	MNIST	EMNIST	CIFAR-10
FedAvg	0.0039	0.0313	0.0313
FedProx	0.0039	0.0313	0.0313
FedMix	0.0039	0.0313	0.0313

The results of the Wilcoxon signed-rank test show that the difference in accuracy achieved by AdFL is statistically significant. To better encapsulate the unstable performance on the CIFAR-10, the median over the last 10 epochs was taken as the final accuracy. AdFL improves the accuracy of the global model and, moreover, reduces the gap in performance between individual classes, despite their uneven distribution across local datasets. It can be measured as a standard deviation between accuracy among all classes (see, Table VII).

TABLE VII
TEST ACCURACY DEVIATION AMONG INDIVIDUAL CLASSES PER METHOD

Dataset	FedAvg	FedProx	FedMix	AdFL
MNIST	0.095	0.179	0.112	**0.025**
EMNIST	0.34	0.33	0.39	**0.26**
CIFAR-10	0.35	0.34	0.37	**0.23**

For all datasets, the standard deviation within the classes is significantly lower for the AdFL algorithm, therefore, illustrating the benefits of balanced training. Finally, the Wilcoxon signed-rank test was applied and it was found that the obtained results are statistically significant.

VI. CONCLUDING REMARKS

In this work, it was shown that utilizing adversarial data on the server side, during FL training, can reveal data distribution information. Use of this information results in more balanced performance in all classes, in the case of label-skewed data. Future research can concentrate on (1) exploring properties of adversarial samples, and (2) applicability of AdFL to more complex datasets, models, and label-skew scenarios. Improvements can also be made to the client-picking strategy and the adversarial data generation process. Additional research can also explore AdFL's potential to battle other non-IID scenarios, e.g., by locating clients with corrupted data.

ACKNOWLEDGMENT

This work was supported by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme and by the Laboratory of Bioinformatics and Computational Genomics and the High-Performance Computing Center of the Faculty of Mathematics and Information Science Warsaw University of Technology.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282.

[2] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021. doi: https://doi.org/10.1016/j.neucom.2021.07.098

[3] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[4] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: Extending mnist to handwritten letters," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017. doi: 10.1109/IJCNN.2017.7966217 pp. 2921–2926.

[5] A. Krizhevsky, "Learning multiple layers of features from tiny images," https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2009.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds. mlsys.org, 2020.

[7] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5132–5143.

[8] E. Ozfatura, K. Ozfatura, and D. Gündüz, "Fedadc: Accelerated federated learning with drift control," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE Press, 2021. doi: 10.1109/ISIT45174.2021.9517850 p. 467–472.

[9] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.1109/CVPR52688.2022.00993 pp. 10 164–10 173.

[10] M. Tang, X. Ning, Y. Wang, Y. Wang, and Y. Chen, "Fedgp: Correlation-based active client selection for heterogeneous federated learning," 03 2021.

[11] T. Yoon, S. Shin, S. J. Hwang, and E. Yang, "Fedmix: Approximation of mixup under mean augmented federated learning," in *International Conference on Learning Representations*, 2021.

[12] L. Wang, S. Xu, X. Wang, and Q. Zhu, "Addressing class imbalance in federated learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, pp. 10 165–10 173, May 2021. doi: 10.1609/aaai.v35i11.17219

[13] C. Chen, Y. Liu, X. Ma, and L. Lyu, "Calfat: Calibrated federated adversarial training with label skewness," 2023.

[14] Y. Lu, P. Qian, G. Huang, and H. Wang, "Personalized federated learning on long-tailed data via adversarial feature augmentation," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. doi: 10.1109/ICASSP49357.2023.10097083 pp. 1–5.

[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013.

[16] O. Suciu, R. Marginean, Y. Kaya, H. D. III, and T. Dumitras, "When does machine learning FAIL? generalized transferability for evasion and poisoning attacks," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018. ISBN 978-1-939133-04-5 pp. 1299–1316.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[18] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017.

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. doi: 10.1109/CVPR.2018.00957 pp. 9185–9193.

[20] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951. doi: 10.1214/aoms/1177729694

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. doi: 10.1109/5.726791

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. doi: 10.1109/CVPR.2018.00474 pp. 4510–4520.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. doi: 10.1109/CVPR.2009.5206848 pp. 248–255.

[24] D. Rey and M. Neuhäuser, *Wilcoxon-Signed-Rank Test*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1658–1659. ISBN 978-3-642-04898-2

DOI: 10.15439/2023F8513
ISSN 2300-5963 ACSIS, Vol. 35

Real-time Communication Model for IoT Systems

Stanisław Deniziak
0000-0002-6812-5227
Kielce University of Technology
Faculty of Electrical Engineering, Automatic Control and Computer Science, Kielce, Poland
s.deniziak@tu.kielce.pl

Mirosław Płaza
0000-0001-9728-3630
Kielce University of Technology
Faculty of Electrical Engineering, Automatic Control and Computer Science, Kielce, Poland
m.plaza@tu.kielce.pl

Łukasz Arcab
0000-0003-4726-732X
Kielce University of Technology
Faculty of Electrical Engineering, Automatic Control and Computer Science, Kielce, Poland
lukasz.arcab@protonmail.com

***Abstract*—Internet of Things solutions typically involve interaction between sensors, actuators, the cloud, embedded systems and user applications. Often in such cases, there are time constraints specifying the maximum response time to a request. This time depends on the calculation time and transmission time. Existing Internet communication solutions do not ensure the implementation of transmissions in a way that guarantees meeting the set time constraints. This paper proposes a new model of Internet communication dedicated to real-time Internet of Things systems, which includes a communication protocol, as well as a transmission scheduling and routing method. The protocol takes into account information about transmission time constraints, which is used for packet scheduling by routers, allowing to increase quality of service. In addition, the proposed static routing mechanism makes it possible to parallelize transmissions if time constraints are still exceeded. Also presented are preliminary results of experiments showing to what extent the proposed methods allow improving the quality of service in real-time Internet of Things systems.**

***Index Terms*— real time routing, tasks scheduling, IoT, communication protocols.**

I. Introduction

The rapid development of the Internet of things (IoT) concept has led to a very large increase in interest in these solutions in almost all areas of our lives [1-3]. In the age of accelerating solutions in this area, there is a steadily increasing demand for IoT systems that, using various communication technologies, will also meet real-time requirements [4]. An important challenge of this research direction is to ensure that time requirements can be met optimally. The need to design real-time IoT (RTIoT) systems was recognized more than 10 years ago [5], when the first technologies and standards to support these solutions began to emerge (e.g.: Time Coordinated Computing (TCC) [6, 7], or the IEEE 802.1 standard – Time Sensitive Network (TSN) [8]). Usually, however, the solutions known today do not guarantee a satisfactory level of Quality of Service (QoS), which in most cases is crucial for the correct operation of the designed system. Therefore, it is necessary to undertake research work to develop methods and technologies to build IoT applications that meet real-time requirements. The first stage of this work was the development of the RTIoT system design methodology by the authors [9].

The key elements of the aforementioned methodology are to propose efficient scheduling and routing methods dedicated directly to RTIoT systems. The primary task of implementing such solutions will be to obtain better QoS performance values relative to standard scheduling methods, especially routing. In this case, the QoS value should be calculated for the worst case, i.e. for conditions that determine the maximum expected load on the system.

Thus, the research problem can be formulated as follows: given is a set of *N* endpoint devices and computing nodes of an IoT network that can send and/or receive transmitted data. The individual devices are interconnected using the network infrastructure that includes, among other things, va.rious communication links of a certain bandwidth and active devices, including routers. Given are also *M* different types of transmission between endpoint devices. In addition, there are strict time requirements associated with selected transmissions. Thus, it is necessary to find a solution for organizing the transmission in order to achieve the best QoS parameters (such as the average QoS of all real-time transmissions), that is, to minimize the average violation of time constraints. For this purpose, transmission scheduling algorithms (determining the order in which data is transmitted) as well as routing algorithms (optimizing routing for individual transmissions) can be used. This work assumes that the above problem will be optimized using static routing. This will ensure the predictability of the developed solutions, enabling the design of RTIoT systems based on small and medium-sized networks, such as a metropolitan area network.

The subsequent part of the article is organized as follows. Section II analyses the current state of the art of communication protocols currently used in IoT systems. Section III describes the assumptions that define the specification of the proposed system and the QoS optimization assumptions. Section IV proposes a transmission scheduling algorithm and an algorithm for selecting optimal routes. Section V contains the results of the experiments conducted, while Section VI presents conclusions and directions for further research.

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

II. Related work

In real-time IoT systems, the use of appropriate types of transmission media and the implementation of proper communication protocols, with particular emphasis on routing methods, plays an important role. Known communication protocols that are worth considering when designing RTIoT solutions include: RTSP (Real Time Streaming Protocol) [10], WebRTC (Web Real Time Communication), XMPP (Extensible Messaging and Presence Protocol) [11], MQTT (Message Queue Telemetry Transport) [12], CoAP (Constrained Application Protocol) [13], WebSocket [14], 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) [15].

The most well-known routing protocols used in IoT systems include:

- RPL (Routing Protocol for Low Power and Lossy Network – LLNs). LLNs are devices characterized by low power consumption, memory, and reduced resource engagement for processes. This protocol belongs to the family of distance vector protocols, which has been designed to work on multiple links [16].
- CTP (Collection Tree Protocol), which is a distance vector routing protocol and was developed for packet routing in WSNs (Wireless Sensor Networks). This protocol assumes the construction of a network topology tree taking into account routes for potential data packets [17].
- LOADng (Lightweight On-Demand Ad hoc Distance Vector Routing Protocol – next generation), is a lighter version of the AODV (Ad hoc On-Demand Distance Vector) protocol for LLNs. It was designed on the premise that LLNs are unoccupied for most of their time. This protocol follows an approach in which routes are determined in the direction of the packet's destination only when there is data to be sent [18].
- CORPL (Cognitive Radio RPL routing protocol), which is based on the RPL routing protocol and, with the modifications made, enables its use in Cognitive Radio environments [19].
- CARP (Channel-Aware Routing Protocol) – a protocol that uses a multi-hop approach to deliver data packets for WSNs. CARP has the advantage of taking link quality into account in the node selection process for next-hop [20].
- E-CARP (Enhanced CARP) – it is characterized by energy efficiency in the process of transmitting packets from the transmitter to the destination. In addition, this protocol does not differentiate the priority of attributes [21].

When it comes to designing real-time IoT networks, it is important to consider the issues of routing protocols and aim to achieve the best QoS transmission parameters. Based on their ability to deliver packets at specific/set deadline values, these protocols can be divided into two major groups: hard real-time and soft real-time [22]. Real-time routing protocols include:

- QoSR (Quality-of-Service Routing) – its greatest asset is its low energy consumption in determining the path for delivering data packets from source to destination. Unfortunately, the protocol exhibits poor support for scalable networks [23].
- QoSAM (QoS Aware Multi-Hop) – similarly to the QoSR protocol, it solves the problem of excessive energy consumption in determining the path for a packet from source to destination. Unfortunately, this protocol has much room for improvement in terms of reliability [24].
- MIMO (Multiple Inputs and Multiple Output) – a protocol that is dedicated to widely scaled WSNs. The implementation of this routing protocol offers the benefits of better energy utilization, lower transmission delays, packet loss and better bandwidth utilization of the transmission link [25].
- PRTR (Potential-based Real-Time Routing) – similarly to the MIMO routing protocol, it is characterized by scalability and a reduced probability of packet loss during transmission – resulting in the protocol requiring additional power, energy [26].
- QEMPAR (QoS and Energy Aware Multi-Path Routing Algorithm) – increases the lifetime of the network, unfortunately at the cost of increased delays [27].
- PT (Pheromone Termite) – features very good packet transmission performance by using a termite-based approach for routing. The protocol specifically focuses on finding the shortest route while maintaining QoS requirements. The PT protocol provides two new properties: pheromone sensitivity, which helps determine the link throughput, and packet generation rate, which helps update nodes in relation to the number of generated packets. One of the disadvantages of this protocol is that it is dedicated to large-scale networks [28].

A lack of an approach that takes into account proper packet scheduling (using appropriate scheduling methods) with deadline values, which can translate into improved QoS performance values, can be noticed in all of the above-mentioned communication protocols.

III. Assumptions

IoT systems usually have predefined functions, i.e. they can be specified in the form of a set of communicating tasks. In most cases, it is possible to estimate task execution times (e.g. for the worst case) and transmission volumes. Thus, for RTIoT systems, a design methodology analogous to that used for distributed embedded systems can be proposed [29].

According to our RTIoT system design methodology, the system specification is represented by a set of annotated task graphs (ATGs) [30]. Each ATG can be activated at a certain maximum frequency. The maximum number of instances of a given graph is also given. Designing an RTIoT system involves mapping the specifications to a target architecture consisting of 4 layers (Fig. 1): the Sensor Layer (SL), the Edge Layer (EL), the Cloud Layer (CL) and the User Layer (UL)

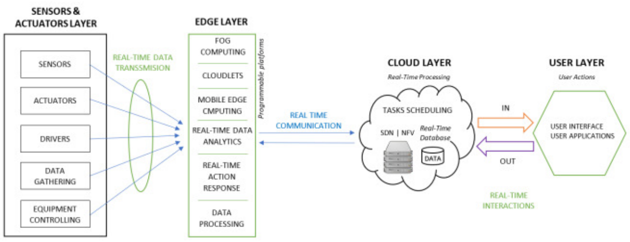


Fig. 1. General architecture of a real-time IoT system

A. System specification

A task graph is a directed acyclic graph $G=\{V,E\}$ in which nodes $v_i \in V$ represent tasks and edges $e_{i,j} \in E$ describe relationships between tasks, usually related to communication. The attributes of the graph describe the assignment of tasks to the layers of the RTIoT architecture and transmission volumes between tasks. Sample annotated task graph is shown in Fig. 2. Attributes that define layers are represented by colours. Attributes describing transmissions are represented by edge labels. From the perspective of network communication, only the transmissions between layers are relevant.

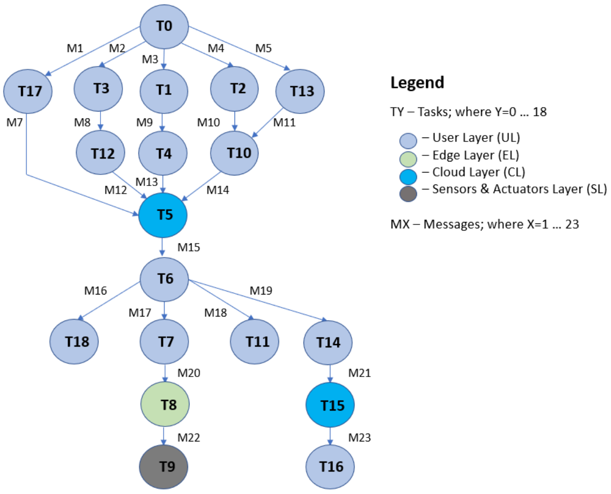


Fig. 2. Sample annotated task graph

Graph given in Fig. 2 describes the main function of a smart city system for managing parking spaces [9]. The user activated task graph specifies the following system functionalities: searching for the parking space closest to the user's current location, the function of finding the user's car in the parking lot based on their license plate number, the function of reserving any free parking space based on the entered search criteria, the function of charging a parking fee for the used parking lot, and the function of retrieving information on weather conditions.

In the example presented in Fig. 2, one task graph activation can cause the following transmissions: M7, M12, M13, M14, M15, M20, M21, M22, M23, where successive numbers indicate individual transmission types. Multiple instances of a given task graph can be activated at any given time, caused by the occurrence of multiple simultaneous events activating given function. For example, multiple users can run an application that sends requests to an IoT system. Thus, there may be a large number of simultaneous transmissions causing a significant load on communication links, leading to the violation of the time constraints.

B. QoS optimisation

A d_{max} time constraint can be associated with any v_i task. This constraint determines the time in which the task should be completed. Soft real-time systems are considered in the paper. In such systems, two types of restrictions are defined: d^h_{max} and d^s_{max}, The d^s_{max} constraint can be exceeded but then the quality of service (QoS) is lower. The d^h_{max} constraint specifies the maximum time that can no longer be exceeded. Violation of the d^h_{max} constraint means packet lost.

The goal of optimizing RTIoT systems is to achieve the highest possible QoS. QoS for a single constraint can be defined as:

$$QoS_i = \begin{cases} 0 \text{ when } t_i > d^h_{max} \\ 1 \text{ when } t_i < d^s_{max} \\ 1 - \frac{t_i - d^s_{max}}{d^h_{max} - d^s_{max}} \textit{ in other cases} \end{cases} \tag{1}$$

where: t_i – is the current finish time of task covered by the i-th constraint.

Then the total QoS for the system can be determined as the average value of all QoS_i:

$$QoS = \sum_{i=0}^{n} \frac{QoS_i}{n} \tag{2}$$

where: n – is the number of all constraints in all instances of task graphs.

If for task v_i constraints d^h_{max} and d^s_{max} are specified then time constraints for all transmissions represented by edges $e_{x,i}$ entering node v_i can also be specified as follows:

$$\text{ex}d^h_{max} = d^h_{max} \text{ - } te_i \tag{3}$$

$$\text{ex}d^s_{max} = d^s_{max} \text{- } te_i \tag{4}$$

where: te_i – is the expected execution time of task v_i, usually determined by WCET (Worst Case Execution Time) estimation.

Time constraints for all transmissions can be determined in an analogous way. Thus, the goal of transmission optimization will be to organize the transmission of messages in such a way that each transmission ends before $\text{ex}d^s_{max}$ or exceeds this time as little as possible while not exceeding $\text{ex}d^h_{max}$. This can be achieved by appropriate transmissions scheduling and/or the use of routing that minimizes collisions of simultaneous transmissions.

IV. Real-time Routing

Existing methods of Internet communication are mainly based on ensuring the most efficient transmission. They do not take into account time constraints or the issue of predictability of transmission time. For these reasons, these methods are not suitable for RTIoT systems.

In order to take into account time constraints, the routing method should use transmission scheduling mechanisms in such a way as to minimize delays and not use overly time-consuming route determination algorithms.

A. Transmission scheduling

It is assumed that individual packets contain information identifying real-time transmissions. Real-time transmissions are processed in the first step, in the order determined by the scheduling algorithm. The remaining transmissions are processed in FIFO order when the list of real-time transmissions is empty. The real-time transmission scheduling algorithm is based on the Least Laxity First (LLF) algorithm [31], which provides optimal task scheduling in real-time systems. Associated with each such packet is information specifying the deadline exd^{s}_{max} and the expected transmission time tt_i estimated based on the length of transmission and the average bandwidth of communication links. Draft scheduling algorithm is shown in Fig. 3.

```
Schedule ($e_i$, RTList){
if RTList[0]=Φ
    TList[0]=$e_i$
else {
    pos=0;
    while RTlist[pos]!= Φ
        if (Laxity(Tlist[pos])<Laxity($e_i$)) pos++;
        else {
            Insert($e_i$, RTList, pos);
            break;
        }
 }
return RTList;
}
```

Fig. 3. Draft transmission scheduling algorithm

The *Laxity(e$_i$)* function computes the transmission time reserve as follows:

$$L_i=exd^{s}_{max}-tt_i \tag{5}$$

The *Insert(ei, RTList, pos)* function inserts the transmission e_i into the *RTList* at the *pos* position. Thus, the scheduling algorithm creates a list of transmissions ordered from the smallest value of L_i.

B. Choice of routes

The choice of transmission routes for transmitting individual packets affects both transmission time and collision-related delays. Thus, the main goals of optimization should be to find the shortest routes and avoid collisions for simultaneous transmissions. Collisions cannot be avoided for transmissions using the same transceivers or receivers.

In the case of real-time systems, it is crucial to implement the transmission such that the violation of time constraints is eliminated or minimized. In addition, the real-time system should be predictable, only then can adequate QoS be guaranteed under a set system load. Predictability can only be achieved with static routing. Then, assuming that the network topology is fixed and the network load is known, the assumed minimum QoS level will be guaranteed. We also assume that all non-colliding paths between given nodes may be found using existing methods e.g. as in the NoC systems [30].

The problem of optimizing real-time transmission for a given network topology can be defined as the problem of allocating communication routes for worst-case scenarios. Suppose that at any given time, m transmissions of data M_1, ..., M_m need to be made between S_i and D_i nodes. Then, if after scheduling the transmissions according to the algorithm from Fig. 3, the transmission delay, for any transmission, resulting the position in the list will cause the deadline to be exceeded, it means that it is necessary to send packets through different routes in order to parallelize the transmissions. Otherwise, all packets can be sent via a single route.

The algorithm for allocating transmissions to routes is shown in Fig. 4. The input to the algorithm is a list of non-colliding *PList* routes and an ordered list of *RTList* transmissions. The algorithm then sequentially schedules transmissions for the next paths in a loop. Transmissions allocated to routes are removed from the *RTList*. The *Time* counter adds up the times of consecutive transmissions allocated to a given path. If the allocation of the next transmission to a particular path results in exceeding the deadline for that transmission then the transmission remains in the list and the algorithm will try to allocate it to the next path in the next loop run. The algorithm returns the number of routes required to complete all transmissions, or an ERROR value if it fails to ensure that all transmissions complete within the required time. In that case, either the network topology needs to be modified to create more routes, or the remaining ones need to be allocated with a minimal violation of the time constraint in order to achieve the lowest QoS drop.

```
AssignPath(Plist, RTList) {
   PathNo=0;
   do {
        Path=Plist[PathNo];
        Time=0;
        For (Pos=0; Length(RTList); Pos++)
          if (Laxity[RTList[Pos]-Time >=0) {
             Assign(RTList[Pos], Path);
             Time+=RTList[Pos].tt;
             Remove(RTList[Pos]);
          }
         if (RTList==Φ) return PathNo;
        PathNo++;
     }
    while PList[PathNo]<>Φ;
     return ERROR;
}
```

Fig. 4. Draft route assignment algorithm

When transmissions involve different destination nodes, routes and ordered *RTLists* should be determined for each node and the algorithm shown in Fig. 4 should be performed independently for each pair of lists *(RTList*, *PList*).

V. Experimental results

The work performed included four experiments. Each of them was performed with given initial conditions such as: equal bandwidth of transmission links; no other type of data packets in the network; 100 different packet transmissions were assumed in the same period, with transmission time for a single packet not exceeding 50ms. Soft deadline (ranging from 1000ms to 5000ms) and hard deadline (ranging from

2000ms to 7000ms) values were also set randomly. The experiments may correspond to any transmission from Fig. 2, between 2 layers (e.g. M20), assuming simultaneous activation of this transmission by 100 users.

The first experiment was conducted for a network in which communication over a single transmission link is assumed. Packet handling by a router with an implemented static routing mechanism is done according to a random packet queue. With this type of approach, it is observed that QoS requirements are not met for a lot of transmitted packets. For 40% of all transmissions the QoS were lower than 1 and 13% of transmissions failed i.e. the hard deadlines were not fulfilled.

The second experiment is an extension of the first approach, which was extended to include the implementation of the LLF-based packet scheduling algorithm (described in Section IV.A of this paper). The results of this experiment clearly show the benefits of using packet scheduling. With appropriate transmission scheduling using the LLF-based algorithm, transmission quality improvement is achieved by obtaining better QoS parameter values with respect to the original values. Only 9% of transmissions exceeded the soft deadline. Thus, this approach is closer to meeting the conditions for real-time transmission.

The third experiment assumed the existence of two independent routes through which packets can be sent using routing mechanisms. In addition, for the purposes of the experiment, it was assumed that the distribution of packets between the previously mentioned routes is even, i.e. half of the previously assumed 100 packets are routed through one link and the remainder through the other link. The results of this experiment showed that, despite the existence of a second, alternative communication link, not all individual transmissions were able to achieve satisfactory QoS results – not all packets (only 87%) were delivered while maintaining the QoS parameter at the level specified by the soft deadline.

The last experiment is an extension of the approach tested in the third experiment. In this case, as in the second experiment, the LLF algorithm that schedules data packets was used. The results of this experiment showed that the existence of two routes in combination with the implementation of a packet data scheduling algorithm allows for the best results in terms of QoS parameters. In this case, all individual transmissions achieved the highest value of the QoS parameter equal to 1. Table I presents a summary of the results obtained for all four experiments conducted. The first column (PAR) defines the parameter name. The following rows contains values of: Average transmission time (ATT), soft deadlines (SD), hard deadlines (HD), number of messages (NM), the number of transmissions that exceeded the soft deadline (NM<SD), the number of transmissions that exceeded the hard deadline (NM<HD), Quality of Service (QoS) obtained for each experiment.

Fig. 5-8 illustrate the dependence of subsequent data transmissions on QoS parameters. Transmissions that did not meet any QoS requirements in the experiments were marked in red, transmissions that only met the requirements of the soft deadline were marked in blue, and those that met all requirements were marked in green.

Analysing the results of the research, it can be seen that for the first experiment, 27 different transmissions did not meet the requirements of the soft deadline, while 13 did not meet the requirements of the hard deadline. In the second and third experiments 9 and 12 different transmissions, respectively, did not meet the requirements of the soft deadline. In the last experiment, all QoS requirements for all types of transmissions were met.

TABLE I. SUMMARY OF EKSPERIMENTAL RESULTS

PAR	Random Singe route	LLF Singe route	Random Double route	LLF Double route
ATT	3434ms			
SD	1000ms, 3000ms, 5000ms			
HD	2000ms, 5000ms, 7000ms			
NM	100			
NM < SD	27	9	13	0
NM < HD	13	0	0	0
QoS	**0,81**	**0,97**	**0,95**	**1**

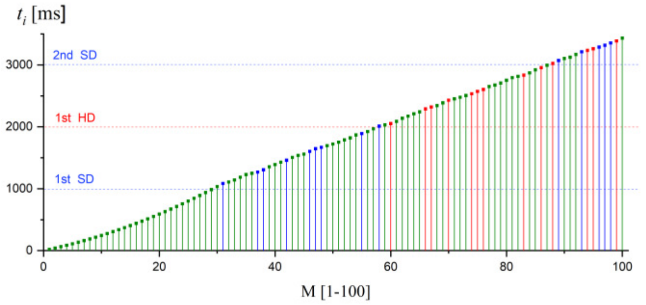


Fig. 5. Dependence of QoS parameters on subsequent data transmissions for first experiment

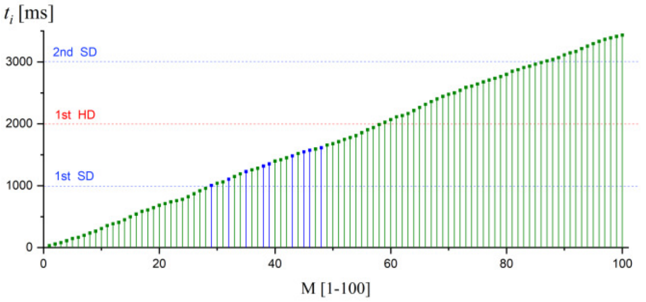


Fig. 6. Dependence of QoS parameters on subsequent data transmissions for second experiment

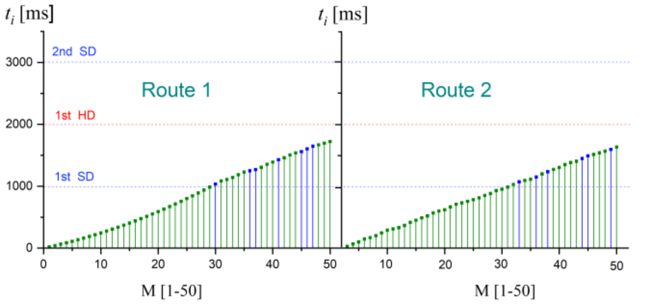


Fig. 7. Dependence of QoS parameters on subsequent data transmissions for third experiment a) first route, b) second route

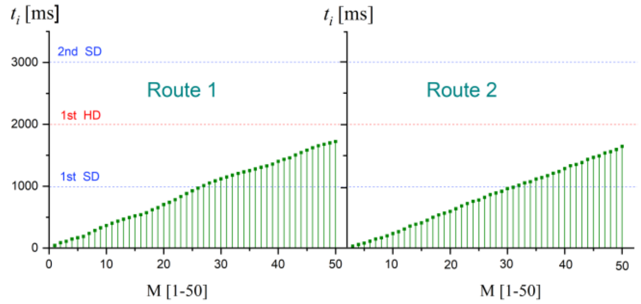


Fig. 8. Dependence of QoS parameters on subsequent data transmissions for fourth experiment a) first route, b) second route

VI. CONCLUSIONS

The article presents the model of Internet communication, dedicated directly to the needs of IoT systems, where real-time requirements are particularly important. The proposed solution is based on data transmission scheduling algorithms and the use of routing methods. The model takes into account information about time constraints at both the soft deadline level and the hard deadline level. Both proper data scheduling and routing mechanisms improve QoS parameters in the system under consideration, as demonstrated by the experiments presented in the paper.

The experiments, conclusions and observations that follow indicate the justification of the approach in which both packet data scheduling methods and appropriate routing methods are applied in RTIoT networks. Based on the simulations and calculations, it should also be noted that the number of routes used for packet transmission also plays an important role in improving QoS parameters for both individual data transmissions and the entire designed system.

The future work on the presented topic will focus on further improvements and extensions to the discussed model. In particular, we will address the implementation capabilities of dynamic routing protocols, as well as other known scheduling methods. The result will be a complete RTIoT system design and implementation environment, ensuring the development of systems with a high level of QoS.

REFERENCES

[1] M. Płaza, R. Belka, Z. Szcześniak, "Towards a different world – on the potential of the Internet of everything", IAPWGIOS, vol. 9(2), pp. 8-11, June 2019, https://doi.org/10.5604/01.3001.0013.2539"

[2] P. Pięta, S. Deniziak, R. Belka, M. Płaza, and M. Płaza, "Multi-domain model for simulating smart IoT-based theme parks", Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 108082T, October 2018.

[3] R. Belka, S. Deniziak, M. Płaza, M. Hejduk, P. Pięta, M. Płaza, P. Czekaj, P. Wołowiec, K. Ludwinek, "Integrated visitor support system for tourism industry based on IoT technologies", Proc. SPIE 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 108081J, October 2018.

[4] M. Płaza, R. Belka, M. Płaza, S. Deniziak, P. Pięta, Sz. Doszczeczko, "Analysis of feasibility and capabilities of RTLS systems in tourism industry", Proc. SPIE, 10808, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018, 108080C, October 2018.

[5] S. Bąk, R. Czarnecki, S. Deniziak, "Synthesis of real-time cloud applications for Internet of Things", Turk. J. Elec. Eng. & Comp. Sci., vol 23, pp. 913-929, 2015, https://doi.org/10.3906/elk-1302-178

[6] Intel. Intel Time Coordinated Computing Tools. 2022. Available online: https://www.intel.com.

[7] Intel. Real-Time at the Edge: Overview. https://www.intel.com, 2022.

[8] J. Lee. S. Park, "Time-sensitive network (TSN) experiment in sensor-based integrated environment for autonomous driving", Sensors, vol. 19(5), pp. 1111, March 2019, https://doi.org/10.3390/s19051111

[9] S. Deniziak, M. Płaza, Ł. Arcab, "Approach for designing real-time IoT systems. Electronics, vol. 11(24), pp. 1-21, December 2022.

[10] J. Lee, J. Kim, S. Kim, Ch. Lim, J. Jung, "Enhanced distributed streaming system based on RTP/RTSP in resurgent ability", Proc. Fourth Annual ACIS ICIS'05, Jeju Island, South Korea, 14-16 July 2005.

[11] M. Kirsche, R. Klauck, "Unify to bridge gaps: Bringing XMPP into the Internet of Things", 2012 IEEE Int. Conf. on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19-23 March 2012

[12] MQTT. MQTT: The Standard for IoT Messaging. 2022. Available online: https://mqtt.org (accessed on 6 January 2023)

[13] C. Bormann, A.P. Castellani, Z. Shelby, "CoAP: An application protocol for billions of tiny internet nodes", IEEE Internet Computing, vol. 16, pp. 62 - 67, 2012.

[14] G. L. Muller, HTML5 WebSocket protocol and its application to distributed computing. https://arxiv.org/abs/1409.3367.

[15] M. Ha, D. Kim, S. H. Kim, S. Hong, "Inter-MARIO: A fast and seamless mobility protocol to support inter-pan handover in 6LoWPAN" 2010 IEEE GLOBECOM, 06-10 December 2010, pp. 1-6

[16] J.V.V. Sobral, J.J.P.C. Rodrigues, R.A.L. Rabêlo, J. Al-Muhtadi, V. Korotaev, "Routing Protocols for Low Power and Lossy Networks in Internet of Things applications", Sensors, vol. 19, pp. 2144, 2019.

[17] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, "The collection tree protocol (CTP)", Proc.(SenSys, November 2009

[18] T. Clausen, J. Yi, U. Herberg, "Lightweight on-demand ad hoc distance-vector routing - next generation (LOADng): Protocol, extension, and applicability", Computer Networks, vo. 126, pp. 125-140, 2017.

[19] Z. Yang, S. Ping, H. Sun, A. H. Aghvami, "CRB-RPL: A receiver-based routing protocol for communications in cognitive radio enabled smart grid", IEEE Trans. on Veh. Tech., vol.66(7), pp.5985-5994, 2017.

[20] S. Basagni, C. Petrioli, R. Petroccia, D. Spaccini, "Channel-aware routing for underwater wireless networks", Proc. Oceans-Yeosu, 2012.

[21] Z. Zhou, B. Yao, R. Xing, L. Shu, S. Bu, "E-CARP: An energy efficient routing protocol for UWSNs in the Internet of underwater things", IEEE Sensors Journal, vol. 16(11), pp. 4072-4082, June 2016.

[22] S. Malik, S. Ahmad, I. Ullah, D. H. Park, D. H. Kim, "An adaptive emergency first intelligent scheduling algorithm for efficient task management and scheduling in hybrid of hard real-time and soft real-time embedded IoT systems", Sustainability, vol. 11(8), pp. 2192, 2019

[23] A. M. Alkahtani, M. E.Woodward, K. Al-Begain, "An overview of Quality of Service (QoS) and QoS Routing in communication networks", Computer Science, 2003.

[24] A. Alanazi, K. Elleithy, "Real-time QoS routing protocols in wireless multimedia sensor networks: Study and analysis", Sensors, vol. 15, pp. 22209-22233, August 2015, https://doi.org/10.3390/s150922209

[25] N. Kumar R. Khanna, "A compact multi-band multi-input multi-output antenna for 4G/5G and IoT devices using theory of characteristic modes", Int. J. of RF and Microwave Comp.-Aided Eng., vol. 30(6), Jan. 2020.

[26] Y. Xu, F. Ren, T. He, C. Lin, C. Chen, S. K. Das, "Real-time routing in wireless sensor networks: A potential field approach", ACM Transactions on Sensor Networks, vol. 9(3), May 2013.

[27] S. R. Heikalabad, H. Rasouli, F. Nematy, N. Rahmani, „QEMPAR: QoS and Energy Aware Multi-Path Routing Algorithm for Real-Time Applications in Wireless Sensor Networks", International Journal of Computer Science Issues, vol. 8(1), pp. 466-471, January 2011.

[28] A. Razaque, K. Elleithy, "Pheromone termite (PT) model to provide robust routing over Wireless Sensor Networks", Proc. of the 2014 ASEE Zone 1, pp. 1-6, 2014.

[29] S. Deniziak, R. Tomaszewski, "Codesign of energy and resource efficient contention-free Network-on Chip for real-time embedded systems, 2018 11th NoCArc, Fukuoka, Japan, pp. 1-6, 2018.

[30] S. Deniziak, R.Tomaszewski, "Co-synthesis of contention-free energy-efficient NOC-based real time embedded systems", Journal of Systems Architecture, vol. 98, pp. 92-101, 2019.

[31] S. Teng, W. Zhang, H. Zhu, X. Fu, J. Su, B. Cui, "A Least-Laxity-First scheduling algorithm of variable time slice for periodic tasks", in Y. Wang (Ed.), Breakthroughs in Software Science and Computational Intelligence, IGI Global, pp. 316-333.

DOI: 10.15439/2023F7277
ISSN 2300-5963 ACSIS, Vol. 35

Review of Automated Code Refactoring of C# Programs

Anna Derezińska
0000-0001-8792-203X
Warsaw University of Technology
Institute of Computer Science
Nowowiejska 15/19,
00-665 Warsaw, Poland
Email: A.Derezinska@ii.pw.edu.pl

Dawid Sygocki
Warsaw University of Technology
Institute of Computer Science
Nowowiejska 15/19,
00-665 Warsaw, Poland,
Email:
dawid.sygocki.stud@pw.edu.pl.

***Abstract*—Code refactoring is supported by many Integrated Development Environments. This paper is focused on the automated code refactoring of C# programs. We have analyzed more than sixty refactorings available in three popular IDEs. We cataloged different restrictions, defects, and other quality concerns associated with the implementation of the refactorings, taking into account both modification of the production code and of the corresponding test cases. An extension to automate selected refactoring improvements has been developed for the ReSharper platform and experimentally verified.**

***Index Terms*—code refactoring, unit testing, code quality, code and test maintenance, C#.**

I. Introduction

REFACTORING techniques have been proposed to improve code quality [1][2]. Their effective application can be assisted with automated tools incorporated into Integrated Development Environments (IDEs). We have examined automated refactoring that can be used in programs written in the C# programming language. We have reviewed three environments commonly used for C# program development, namely: Microsoft Visual Studio, JetBrains Rider, and Visual Studio Code. The following research questions were addressed:

1. Can we rely on automated refactoring, i.e. is the final code always correct?
2. Are the unit tests corresponding to the refactored area adjusted together with the refactoring completed?
3. Can we, in an automated way and transparently to a developer, fix malfunctions detected in refactoring or add any improvements?

To perform the study, a benchmark program was developed covering code variants consistent with these environments. We have found that some refactorings can be correctly applied only under certain restrictions, while others can produce invalid code. Moreover, the test cases related to the refactored area could be improved.

The main contributions of the paper are the following:

- A review of over 60 automated refactorings of C# programs supported by three popular environments.
- Identification and classification of different restrictions, extensions, defects, and quality concerns associated with the refactoring implementation.
- As a proof of concept, development of Refix, a prototype that fixes defects of selected refactorings of the ReSharper tool-(used in JetBrains Rider 2022 and also as a plugin applied to the Microsoft Visual Studio [3]).

The paper is structured as follows: In the next Section we discuss the related studies. In Section 3, we give an overview of the refactoring consequences for code and tests in three popular environments. A developed tool (Refix) is briefly presented in Section 4. In Section 5, we conclude the paper.

II. Related Work

There exist many tools that help in automated refactoring in software development and maintenance. Though, their usage still causes difficulties for developers, as reported in [4].

Deficiencies in the refactoring tools were mainly studied for Java and C programs. Testing of refactoring engines [5] found 1.4% of refactoring tasks failing for Java and 7.5% for C. In [6], the problems of name binding and accessibility rules in refactoring are discussed. Some problems could be similar, but there are no studies of C#.

Another problem of refactoring implementation is its impact on the test cases [7]. Many experiments related to the test maintenance of refactored programs were performed in Java [8], [9], [10]. In [8], it was shown that tests often require additional handling when the production code is refactored. To handle this, a prototype developed in Eclipse was discussed in [9]. In [10], refactoring of Java programs with JUnit tests was examined. Various flaws in updating tests were identified. RefactorPlugin was developed to correct selected defects in tests and create additional tests in accordance with the refactoring performed.

Refactoring in C# programs was studied as one of the aspects to explore similarities and differences between test and production classes [11]. It was found that while production classes underwent more changes, the maintenance of tests caused major problems. In the thesis [12], an extension to Visual Studio was developed, but the author focused on a new kind of refactoring, not on fixing the existing ones. To the best of our knowledge, improvements in the C# code and tests after refactoring were not considered.

The impact of using the ReSharper tool on the results of test runs, builds, and version control commands was examined in [13]. Experiments on Enriched Event Stream Dataset led to a higher rate of failure builds and higher percentage of commits. ReSharper was also studied in some research and was claimed to be popular among developers [14].

III. Review of Refactoring

In the refactoring review, we have checked whether an automated refactoring provides a valid transformation of the code and whether the corresponding unit tests are correctly modified. We focused on three different IDEs that support refactoring of C# programs:

1) Refactoring embedded in Microsoft Visual Studio 2022, i.e. MVS without additional extensions.

2) JetBrains Rider 2022, and the same refactoring engine used in the ReSharper plugin applied to MVS [3][15].

3) Visual Studio Code with an addition to support the C# language (ms-dotnettools.csharp).

A. Benchmark for Refactoring Review

Analysis of the effects of refactoring was assisted by a benchmark program [16]. We have developed it to cover a set of refactoring examples and the corresponding unit tests. The production code was developed in three versions related to IDEs mentioned above.

The benchmark also encompasses three variants of a set of unit tests corresponding to the popular unit test frameworks that support C#: MSTest, NUnit, and xUnit.net.

B. Comparison of the Refactoring Capabilities

We have reviewed the way of implementation of all refactorings that were supported in the mentioned environments. As a result, we made recommendations on many refactorings. They were classified into the following four categories:

1. **Extension (E):** the refactoring implementation is extended in comparison to its basic meaning [1].
2. **Restriction (R)**: the implementation of the refactoring is restricted compared to its basic meaning.
3. **Defect (D)**: a fault was detected in the refactoring implementation that should be fixed, as it causes the project not to compile. This situation is often associated with a “conflict”, i.e. a warning reported by an IDE after an attempt of a refactoring.
4. **Quality concern (Q)**: a shortcoming occurs that does not cause a compilation error or another improvement to the code and tests could be suggested.

In Tables I and II, we summarize all refactorings implemented in three environments. The last three columns correspond to Microsoft Visual Studio - MVS, JetBrains Rider (also ReSharper) – R/R, and Visual Studio Code – VSC, accordingly. The sign ‘-’ denotes that the refactoring is not supported in the environment. If the refactoring was implemented, the character ‘+’ shows that no recommendations were related to it. Otherwise, a combination of letters (E, R, D or Q) signifies the recommendation categories associated with the refactoring. The assigned recommendations are discussed in the subsequent subsections.

C. Review of Refactoring in Visual Studio

Here, we deal with the refactorings embedded directly in MVS 2022. Those supported by the Resharper extension, often applied to MVS, are discussed in the next subsection.

Restrictions (R):

- In #43 and #44. The move method and move field refactoring is limited to static members.

Defects (D):

- In #2. In synchronization of a namespace and a folder name, the `using` directive in tests could be not updated. The situation was rare and this defect can be treated as a minor one.
- In #14. In the conversion between a property and `get` method, the refactoring does not consider references to the transformed property present in the object initialization. In this case, a warning is shown.
- In #43 and #44. It refers to moving a field or a method. A static class can be indicated as an `internal` one. Therefore, its members are not accessible in the test project. One of the popular conventions is the application of an attribute to make these internal types accessible and structural testing possible. Hence, the improvement of this defect is of low priority.

Quality improvements (Q):

- In #39 and #40. When a method or a field is pulled up to the base class, other descendant classes could be checked in terms of these member occurrences.
- In #39, #40, and #54. While a method or a field is pulled up, or when a superclass is extracted, we could consider using the base class where possible. The refactoring can be followed by another refactoring #38.

D. Review of Refactoring in ReSharper

Among the solutions discussed, the ReSharper engine, used in JetBrains Rider and as an extension in MVS, provides the largest number (64) of automated refactorings. Below, we list the recommendations passed on to these refactorings.

Extensions (E):

- In #31. The change of signature refactoring can be applied not only to a method but also properties, indexers and constructors. Moreover, two alternative forms of the transformation are available: *change signature* and *transform parameters*.

Restrictions (R):

- In #8. The refactoring to make a member static can be completed only if the method takes an instance as a parameter.

TABLE II.
COMPARISON OF AUTOMATED REFACTORING FOR C# PROGRAMS

No	Refactoring	Environment		
		MVS	R/R	VSC
1	Safe delete	-	Q	-
2	Sync namespace and folder name	D	+	+
3	Sync a type and filename	+	+	+
4	Convert an abstract class to interface/vice versa	-	+	-
5	Convert anonymous type to class	+	+	+
6	Convert anonymous type to tuple	+	-	D
7	Convert extension method to plain static/vice versa	-	+	-
8	Make member static	+	R	-
9	Use expression body or block body for lambda expression	+	+	-
10	Convert anonymous function to local one	-	+	-
11	Convert local function to method	+	+	+
12	Make local function static	+	-	-
13	Replace constructor with factory method	-	+	-
14	Convert `get` method to property/ and vice versa.	D	D	D
15	Convert method to indexer/vv.	-	+	-
16	Convert between auto property and full property	+	+	R
17	Encapsulate field	+	+	Q
18	Replace loop with pipeline	+	+	+
19	Convert between `for` loop and `foreach` statement	+	+	+
20	Simplify LINQ expression	+	+	-
21	Convert between regular string and verbatim string literals	+	+	+
22	Simplify string interpolation	+	+	-
23	Use pattern matching	+	+	-
24	Convert `if` statement to `switch` statement or expression	+	+	R
25	Convert `switch` statement to `switch` expression	+	+	-
26	Split or merge `if` statements	+	+	+
27	Simplify conditional expression	+	+	-
28	Use explicit type	+	+	-
29	Use `new()`	+	+	-
30	Copy type	-	Q	-
31	Change signature/Transform parameters	+	ED Q	-
32	Add `null` checks of parameters	+	+	E
33	Introduce parameter	+	+	+
34	Introduce parameter object	-	+	-
35	Invert conditional expressions and AND/OR operators	+	+	+
36	Invert `if` statement	+	+	+

TABLE I.
COMPARISON OF AUTOMATED REFACTORING (CONTINUATION)

No	Refactoring	Environment		
		MVS	R/R	VSC
37	Invert Boolean	-	Q	-
38	Use base type where possible	-	+	-
39	Pull up method	Q	Q	-
40	Pull up field	Q	Q	-
41	Push down method	-	D	-
42	Push down field	-	D	-
43	Move method	RD	R	-
44	Move field	RD	+	-
45	Move a type to a matching file	+	+	+
46	Move to folder	-	+	-
47	Move type to another namespace	-	+	-
48	Remove dead code	+	+	-
49	Remove unused references	+	-	-
50	Extract method	+	Q	+
51	Inline method	+	Q	-
52	Extract class	+	R	-
53	Inline class	+	R	-
54	Extract superclass	Q	+	DQ
55	Extract interface	+	+	+
56	Extract members to partial class	-	+	-
57	Introduce field	-	+	-
58	Wrap, indent and align	+	+	+
59	Sort `using` declarations	+	+	-
60	Introduce local variable	+	+	+
61	Move declaration near reference	+	+	+
62	Rename	+	+	+
63	Change member or internal type visibility to public/ internal/ protected/ private protected/ private	-	D	-
64	Change type visibility to public/ internal	-	+	-
65	Change to virtual/ non-virtual	-	D	-
66	Change to abstract/ non-abstract	-	D	-
67	Make method override/Add `new` keyword	-	+	-

- In #43. The refactoring of move method is restricted, because the target class has to be a parameter of the moved method. It does not pertain to static methods.
- In #52. The extract class refactoring includes only a variant in which the reference to the new class remains in the old one. The test cases remain unchanged, although they could have been modified by changing the subject of the tests to the new class.
- In #53 In the inline class refactoring, the absorbed class is required to include reference to the target class.

In the following cases, a refactoring could provoke erroneous behavior.

Defects (D):

- In #14. When converting between a property and `get` method, references to the transformed property that are present in the object initialization are invalid. To signal this problem, the refactoring tool creates a warning.
- In #31. In the refactoring of transform parameters, the transformation of a method with an `out` parameter to an expression is incorrect. In the method body, an assignment of the already missing parameter exists, and therefore the program cannot compile.
- In #41 and #42. After the push down refactoring, the tests of the modified base class do not compile any more. They should also be refactored and moved to the appropriate descendant class to which a method or a field was pushed down. The analogous problem is in the pull up refactoring.
- In #63. In the refactoring that changes a member or internal type visibility, the corresponding tests are not modified accordingly. In dependence of the qualifier used, `private` in particular, the final project could not compile.
- In #65 and #66. After the addition or deletion of the `virtual` or `abstract` modifiers, the corresponding tests are not updated. They either need to be deleted or adjusted by changing their subject.

Other improvements in refactoring (Q) could be suggested:

- In #1. In the refactoring of safe delete, several lines with references to a deleted element are also removed. Therefore, it could be beneficial to remove empty or pointless corresponding tests.
- In #30. Refactoring the type copy could also require modification of the corresponding tests. They could either be duplicated or enhanced by applying parametrized tests according to the refactored type.
- In #31. Depending on the details of the change signature refactoring, we could update the tests. For example, if a list of parameters was shortened, some variables could be deleted from a test case; if a parameter was added, a variable with a default value could be introduced in a test case, etc.
- In #37. If a Boolean value is inverted, an Assert could be changed to do tests more legible, e.g., substitute `Assert.True(!value)` with `Assert.False (value)`. However, the exact behavior could be different in dependence on the test library used.
- In #39 and #40. This refers to tests after a method or a field pull up. For example, the corresponding tests could be moved to the base class if they have no dependencies on the original class and if the base class is not an abstract one.
- In #50. After an extract method refactoring, a new method appears. The creation of new test cases, e.g., automated test generation, could be considered.
- In #51. After applying an inline method, the tests of the method remain and their code is merged with the tested method. Consequently, a code duplication encounter. The useless tests could be deleted.

E. Review of Refactoring in Visual Studio Code

The number of refactorings supported by Visual Studio Code (VSC) was the smallest among the three environments. As refactoring variants, two restrictions and one extension were identified.

Extension (E):

- In #32. When parameters are of the `string` type, additional checks could be applied using the `IsNullOrEmpty` method.

Restrictions (R):

- In #16. In conversion between an auto and full property, only one direction of the conversion is supported. An auto property can be converted to a full property, but the vice versa transformation is not possible.
- In #24. Conversion from the `if` instruction to the `switch` instruction or expression is restricted. It can only be applied in cases where relations in consecutive `if` statements refer to the same variable.

The following defects (D) were recognized:

- In #6. In conversion from an anonymous type to a tuple, if an object table that has a transformed anonymous type is created using the shortened inscription `new[]` then a compilation error occurs. This could be avoided by using the full description in the form `new object[]` or by casting to the object type for one of the initialization elements.
- In #14. When converting between a property and a `get` method, references to a refactored property that are used in an object initialization are not updated.
- In #54. The extract superclass refactoring does not take into account dependencies on other fields or methods. If in an extracted method, a reference to a member of the original class exists, a compilation error can arise.

Two quality improvements (Q) were recommended:

- In #17. When a field is encapsulated, visibility of the resultant field is always set to `private` and of a property set to `public`. The transformation does not take into account the initial modifiers.
- In #54. After extracting a superclass, a new base type is used instead of its descendant, where possible.

F. Summary of the Review

In general, the level of refactoring correctness in the environments is similar. Calculating the ratio of the number of refactorings classified as a defect (D) to the number of all refactorings supported by the IDE we obtained 9% for the pure MVS, 11% for JetBrains Rider (and the same for ReSharper), and finally 12% for VSC.

We observed that some defects are specific to selected environments, while others are common to different IDEs. For example, the same problem of converting a property and a `get()` method in #14, occurs in all three tools. Furthermore, similar quality issues refer to pull up refactoring (#39, #40) in the environments in which they are implemented.

The recommendations relate to the modification of the production code but also to the corresponding unit tests. Considering the defects identified in Rider/ReSharper, 5 out of 7 defects refer to test cases. Moreover, all 8 quality issues considered in this environment suggest improvements in the tests.

In summary, the first and second research questions addressed in the Introduction, have negative answers. We cannot always rely on automated refactoring, and tests associated with the refactored area are often inadequately handled.

IV. Automated Fixing of Refactoring Defects (Refix)

Based on the analysis provided, we propose an approach to extend automated refactoring. The tool should correct selected defects and improve the quality of code and tests.

We have selected the ReSharper tool to be enhanced. This platform supports a large number of refactorings, can be used in at least two popular environments, and can be extended with plugins. As a proof of concept, an extension Refix was designed and implemented [16].

The Refix extension integrates with the ReSharper platform and can react when a refactoring is executed. If required, an additional "fixing" activity is undertaken. Currently, it supports improvements of defects related to refactorings #14, #31, #41, and #42, as described in Sect.III.D.

Refix has been tested on JetBrains Rider 2022.1.2 and MVS 2022 Community Edition with ReSharper 2022.1.2.

The benchmark developed to assess refactoring in different environments was also used in the evaluation of the Refix tool (Sect.III.A). It was run with Refix in both environments mentioned above. Unit tests from all three test libraries were used in experiments. Refactoring with Refix was completed correctly, according to expectations.

Furthermore, the evaluation of Refix was based on three real programs derived from the GitHub platform [16]. The experiments were carried out using JetBrains Rider. Unit tests of the projects were run with the xUnit.net framework.

The detailed description of the prototype, and its experimental evaluation are beyond the scope of the paper.

V.Conclusion

After analyzing automated refactorings of C# supported in three popular IDEs, we have made a set of recommendations. Several malfunctions were recognized that influenced the resulting code and tests. In particular, some refactorings deliver code that does not compile. The environments significantly differ in the number of supported transformations, but their general realization quality is at a similar level. Moreover, we have recognized the same or similar problems that referred to the same refactorings in different environments.

Three frameworks for unit tests (MSTest, NUnit, and xUnit.net) were applied in all considered environments. We have not noticed any differences in using the frameworks, as far as the problems of refactored programs are concerned.

A prototype tool has been developed to automate code repair after refactoring [16]. The Refix plugin can be used in JetBrains Rider or MVS with the ReSharper extension. The tool was evaluated using the benchmark and some real programs from GitHub. Due to the prototype developed and its preliminary evaluation, we could positively answer the third research question. In the future, the tool could be extended to cover the remaining defects and other quality concerns.

References

[1] M. Fowler, *Refactoring: improving the design of existing code*. 2nd ed. Addison-Wesley, 2018.

[2] A. A. B. Baqais and M. Alshayeb, "Automatic software refactoring: a systematic literature review," *Software Quality Journal*, vol. 28, 2020, pp. 459-502, http://dx.doi.org/10.1007/s11219-019-09477-y

[3] "ReSharper: The Visual Studio extension for .NET developers by JetBrains," 2023, https://www.jetbrains.com/resharper/, [Online, Accessed 20 Jan 2023]

[4] A. M. Eilertsen and G. C. Murphy, "The usability (or not) of refactoring tools," in *Proc. IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, 2021, pp. 237-248. http://dx.doi.org/10.1109/SANER50967.2021.00030

[5] M. Gligoric, F. Behrang, Y. Li, J. Overbey, M. Hafiz, and D. Marinov, "Systematic Testing of Refactoring Engines on Real Software Projects," in *Proc. ECOOP 2013 – Object-Oriented Programming*, LNCS vol 7920. Springer, Berlin, Heidelberg. 2013, pp. 629–654, https://doi.org/10.1007/978-3-642-39038-8_26

[6] M. Schäfer, A. Thies, F. Steimann, and F. Tip, "A Comprehensive Approach to Naming and Accessibility in Refactoring Java Programs," *IEEE Transactions on Software Engineering*, vol. 38, no. 6, 2012, pp. 1233-1257, http://dx.doi.org/10.1109/TSE.2012.13

[7] Y. Kashiwa, K. Shimizu, B. Lin, G. Bavota, M. Lanza, Y. Kamei, and N. Ubayashi, "Does refactoring break tests and to what extent?" in *Proc. IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2021, pp.171-182, http://dx.doi.org/10.1109/ICSME52107.2021.00022

[8] Y. Gao, H. Liu, X. Fan, Z. Niu, and B. Nyirongo, "Analyzing refactoring' impact on regression test cases," in *Proc. IEEE 39th Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2, 2015, pp. 222-231, http://dx.doi.org/10.1109/COMPSAC.2015.16

[9] H. Passier, L. Bijlsma, C. Bockisch,, "Maintaining unit tests during refactoring," in *Proc. 13th International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*, no. 18, 2016, pp.1-6. http://dx.doi.org/10.1145/2972206.2972223

[10] A. Derezinska, O. Sobieraj, "Enhancing unit tests in refactored Java programs," in *Proc. 18th Inter. Conf. on Evaluation of Novel Approaches to Software Engineering - ENASE*, Scitepress, 2023, pp. 734-741, http://dx.doi.org/10.5220/0011997800003464

[11] M. Gatrell, S. Counsell,S. Swift, R. M. Hierons, and X. Liu, "Test and production classes of an industrial C# system: a refactoring and fault perspective," in *Proc. 41st Euromicro Conference on Software Engineering and Advanced Applications*, 2015, http://dx.doi.org/10.1109/SEAA.2015.40

[12] M. Linka, "Visual Studio refactoring and code style management toolset," M.S. thesis, Charles University in Prague, 2015.

[13] E. Firouzi and A. Sami, "Visual Studio automated refactoring tool should improve development time, but ReSharper led to more solution-build failures," in *IEEE Workshop on Mining and Analyzing Interaction Histories (MAINT)*, Hangzhou, China, 2019, pp. 2-6, http://dx.doi.org/10.1109/MAINT.2019.8666936

[14] S. Amann, S. Proksch, S. Nadi, and M. Mezini, "A study of Visual Studio usage in practice," in *Proc. IEEE 23rd International Conference on Software Analysis Evolution and Reengineering (SANER*), vol. 1, pp. 124-134, 2016.

[15] "Rider: The Cross-Platform .NET IDE from JetBrains," 2023, https://www.jetbrains.com/rider/, [Online, Accessed 20 Jan 2023]

[16] Refix, https://galera.ii.pw.edu.pl/~adr/Refix/ [Online , Accessed 30 July 2023]

DOI: 10.15439/2023F2907
ISSN 2300-5963 ACSIS, Vol. 35

Employee Technostress in South Africa's Hybrid Workplaces: Causes and Coping Mechanisms

Shelley Dowrie
0009-0007-7391-1998
Dept. of Information Systems,
University of Cape Town, Rondebosch,
7701, South Africa
Email: dwrshe001@myuct.ac.za

Jean-Paul Van Belle
0000-0002-9140-0143
Dept. of Information Systems,
University of Cape Town,
Rondebosch, 7701, South Africa
Email: jean-paul.vanbelle@uct.ac.za

Marita Turpin
0000-0002-4425-2010
Department of Informatics,
University of Pretoria, Lynnwood
Road, Pretoria, 0001, South Africa
Email: marita.turpin@up.ac.za

Abstract— **During the COVID-19 pandemic, South African organisations were forced to provide suitable working conditions for its employees. The increased reliance on technology while working from home resulted in technostress. This paper considers how technostress experiences have evolved under the newly adopted hybrid working model. It investigates the underlying causes of technostress experiences and how employees are currently coping with technostress under the hybrid model. Semi-structured interviews were conducted and supplemented with secondary data provided by respondents who are currently working under a hybrid model and who use ICTs for work purposes. The findings reveal several hybrid working specific causes of technostress, including instances of stressful workstation setups, office disruptions and power outage issues as a result of loadshedding (rolling power blackouts). Stresses related to loadshedding appear to be a specific South African issue. To deal with technostress, employees adopted reactive and proactive coping behaviours driven by problem-focused and emotion-focused coping strategies respectively.**

Index Terms—**technostress, hybrid workplaces, South African organisations.**

I. Introduction

Technostress escalated once the COVID-19 pandemic forced organisations to impose response plans to resume work as smoothly as possible [3]. Many resorted to work-from-home (WFH) styles where employees had to adapt their current work dynamic with the incorporation of ICTs. This caused an obligated reliance on technology by the employees and the organisations [4]. An increased use of ICTs leads to higher workload demands on employees [5]. This induces the inability to manage these demands therefore stifling the capability to process further information often leading to burnout and technostress [5]. In other words, technostress occurs when there are changes in working conditions that stem from the adoption and use of ICTs. This forces employees to adapt and adjust almost instantaneously [5].

In hybrid working environments, employees are required to find a working dynamic that compliments varying reliance on ICTs between the alternating working locations i.e., at home and at the office. This unanticipated shift in working modes could either cause more technostress amongst employees or alleviate some of the technostress experienced during pure remote working. Due to the adoption of a hybrid working model being relatively new for most organisations, there remains a gap in literature pertaining to the experiences of technostress within this new working environment.

The purpose of this study is to explain the shift of employee technostress experiences along with the underlying causes and coping mechanisms. With this research purpose in mind, the research aims to address the following research questions:

Primary Research Question:

- How has the experiences of employee technostress changed when hybrid workplaces were implemented in South Africa?

Secondary Questions:

- Why are South African employees experiencing technostress in these hybrid workplaces?
- How are South African employees currently coping with such instances of technostress?

II. Literature Review (Abbreviated)

The literature review focussses only on the causes or determinants of technostress due to space limitations. It looks at the standard technostressors: system performance issues, technology demands and lack of digital literacy.

The Big Five Technostressors

Technostress literature has established the five standard technostressors: **techno-overload, techno-invasion, techno-complexity, techno-insecurity, and techno-uncertainty. Techno-overload** is triggered when ICT users are required to work for longer and at a faster pace when using ICTs [8]. This also deals with the handling of excess features and information when using ICTs for work [2]. **Techno-invasion** requires the employee to be constantly connected and available to respond timeously even outside work hours, leading to an invasion of their personal environment [2]. **Techno-complexity** refers to feelings of incompetency amongst employee when using the ICTs [9]. This is because of the inherent quality of Information Technology (IT) and ICTs [10]. Due to these feelings of appearing inadequate with IT skills, ICT users invest in spending more time and effort to fully understand the particulars of the technology [8]. **Techno-insecurity** refers to ICT users fearing the loss of their jobs in terms of having some sort of technology eventually take over their role or that their fellow colleagues possess a better understanding of the usage of the ICT. Finally, **techno-uncertainty** refers to the constant ICT upgrades that unsettle ICT users forcing them to continually learn and familiarise themselves with the new technology [8].

System Performance Issues

Technostressors can also extend to system performance issues such as problems with security, usability, and system breakdown [8]. **Security issues** originate from insecure system infrastructure that allow threats which compromise the information involved in that system. Technostress emerges when users are forced to comply with security policies implemented by the organisation which require them to remember passwords and multiple usernames [8]. **Usability** issues stem from poorly designed systems such as bad interfaces, challenges in intuitively navigating around the application/system and in general lack of effectiveness, efficiency and learnability [1]. This causes the users of these system to experience higher cognitive overloads. Finally, **system breakdown** refers to the malfunctioning of ICTs such as error messages [8].

ICT Use in the Workplace

The "technology demands" predictor signifies the costs employees incur as a result of the effort needed for ICT use. These costs are of psychological and physiological natures [11]. These types of demands involve role ambiguity, ergonomic stress, monotonous ICT activities and general work overload [11]. Role ambiguity occurs when ICT tasks are ill defined [11] which is further claimed to restrain the user's abilities and development [12]. Demands can also originate from a societal sense whereby employees experience social isolation, role conflict and emotional overload when trying to form human relationships around the usage of ICTs [11]. Technostress has a positive relationship with how often ICTs are used for work purposes. Technostress can also be instantiated by the usage of multiple ICTs at once. This is derived from higher demands or greater pressure on workers to learn and embrace multiple ICTs [5].

III. Propositions

Table I below indicates the research propositions adapted from literature that drove the data collection process:

TABLE I.
RESEARCH PROPOSITIONS

Research Area	Relationship/Themes	Reference
ICT use in the workplace	Technostress can be induced by having a high dependency on an evolving ICT/Information System in the workplace.	[8]
	If a user possesses high skill levels of digital literacy/skills, he/she will find it relatively easier to learn and adapt to new technologies.	[13]
Techno-stressors	Techno-complexity, techno-overload, techno-invasion, techno-insecurity and techno-uncertainty lead to technostress.	[2]
	Unreliability of ICTs cause technostress.	[8]
Coping behaviours	Reduce ICT-related stress: Distancing; venting.	[14]
	Establish ICT use demarcations: Time-related use, Separation of use, autonomy.	[14]

IV. Research methodology

An interpretive philosophy was adopted for this research. Interpretive research assumes that human experiences shape the social realities and invites subjective interpretations of the respondents in the social context, in this case the virtual and physical workplaces in organisations [15]. As this study adopted an interpretivist paradigm, the most suitable strategy to be adopted was a qualitative one. Qualitative research intends to derive meaning-based forms of data analysis. It enforces the notion of contextual understanding through providing in-depth descriptions of insights that cannot be shown through quantitative measures [16]. This research was conducted over eight months in 2022.

Data was collected through semi-structured interviews [16]. The target audience for this study were employees within organisations that were making use of a hybrid working model and respondents could be conveniently accessed. Twelve respondents were interviewed with ten respondents obtained from the insurance company and two respondents from a university. Table II shows respondent's job role, IT skill level and their estimate technostress level. *Data saturation was achieved after about 9 interviews.*

TABLE II.
DESCRIPTIVE SUMMARY OF RESPONDENTS

Role	Skill level	Techno stress level
Client Relationship Manager,	Medium	Medium
Finance Team Leader	Medium	Medium
Section 14 Technical Team Leader	Medium	Low
Business Specialist, MIS	High	Low-Medium
Client Relationship Manager	Medium	Dependent
Corporate Client Services Team Leader	Medium	Medium
Servicing Team Leader	High	Low- Medium
Client Relationship Manager,	Medium	Medium
Social Media Complaints	Medium	Medium-Intense
Complaints Handler	Medium	Medium-Intense
ICTs: Senior Business Analyst	High	Low-Medium
HR Analytics	High	Medium

The analysis took on the form of categorizing themes and patterns that derived at a set of concepts (codes), constructs (categories) and relationships [15]. NVivo was used to perform the thematic analysis. Ethical clearance for the study was obtained from the ethics committee of the university.

V. Research Findings and Analysis

The themes from the data analysis are discussed below.

A. Hybrid Working Specific Causes of Technostress

a) Non-assigned desk setup ("hot desks") creates unnecessary technostress

On the days designated to go work at the office, three respondents indicated device configuration issues between the provided monitors and their own laptops. These experiences were also discovered in China [7]. Five respondents complained about insufficient technical support at the office should they encounter tech-related difficulties. Two respondents alluded to occurrences of missing equipment at the office which meant search for appropriate equipment to set up their workstation, delaying the start of their working day. These stresses were more prominent at the start of the hybrid model.

b) Office distractions related to ICT use causes stress

Eight respondents expressed that being in office meant having to endure office-based interruptions where otherwise they wouldn't have experienced while working from home. Three out of the eight claimed that there are challenges with having partial teams present at the office on designated days. Those in-office would have to dial in the rest of the team who are working from home. This meant holding virtual meetings via MS Teams which would create echoes in the office, often distracting other teams and employees who are also in that day. This was claimed to add to stress as it would disturb concentration levels and productivity.

c) Power issues creating stress

Seven respondents from the interviews and three instances from the secondary data referred to the disturbances that loadshedding (rolling blackouts) has had on their workstation setup at home and in office resulting in downtime. While most confirmed that their organisation provided adequate support for power outages at home such as an UPS, some referred to the stress of worrying whether there would be enough stored power to last the loadshedding slots, especially under the higher levels of loadshedding. *"My blood pressure almost went up one day because I was panicking. I thought, where am I gonna work?" - OM05*

B. The Change in Traditional Technostressors under Hybrid Working

a) Techno-uncertainty with the adoption of a new ICT/technology when remote working was introduced

Six respondents experienced initial stress familiarising themselves with new systems implemented by their organisation. OM02 experienced communication inconsistencies when liaising with her IT department about the introduction of a new system. She expressed how stressful it was to interpret the technical jargon which created misunderstandings in the requirements for the new system.

Five respondents alluded to the minimal/insufficient technical support and training for new system rollouts. This created stress in a sense that employees were now forced to learn the new systems by themselves. This resulted in them having to factor in time to learn the new systems which meant neglecting work duties for a time period, hence creating more stress. *"There's no training on it. It's like we just learning on the job, like on top of each other, on multiple applications besides all our existing applications."- OM10*

b) Techno-overload creating technostress while WFH

Nine respondents indicated feelings of hyperconnectivity while working from home on the designated days which made it easier to be interrupted thorough application notifications and alerts. This also stems from organisational expectations on employees to be able to respond quickly and often outside of work hours. Three respondents mentioned how using multiple applications at the same time can become overwhelming and stressful especially since sometimes the systems don't easily speak to each other. This is consistent with the belief that using multiple ICTs can instantiate technostress [5].

c) Techno-insecurity creating technostress

Experiences of techno-insecurity weren't that significant amongst the sample, with only two respondents referring to feeling insecure about their IT skills. OM10 mentioned the impact of the imbalance of IT skills within the team that creates stress. This was to do with new employees in the team possessing IT skills that they have gained from experience in other teams/departments which instantiated feelings of insecurity within the old employees who didn't possess such experience. This also made them feel as if it was burdensome when bothering these new employees for IT-related help.

d) Techno-invasion distorts work/life balance when WFH

This theme relates to the distortion of work life balance as a result of ICT use for work. Six instances within the secondary data referred to challenges with work life balance under the hybrid working model. In line with [6], nine respondents from the interviews recalled feelings of techno-invasion where they often found themselves logging in after work hours, on weekends and late in the evenings. This was mainly due to having a convenient setup at home which enabled them with the ability to connect or simply the use of laptops which made it easy to resume working when coming back from the office. This meant putting in additional work hours without even realising it. One respondent mentioned the negative impact techno-invasion had on her personal relationships at home. *"Technology has invaded our private space and the lines between your work day and your domestic day have become blurred. That has led to stress in my life."- UCT01*

e) Techno-complexity causing technostress

Ten respondents indicated that some of the systems are quite complex to understand at first, sometimes even after the system/application has been used for a while. Four of these respondents referred to instances of system upgrade inconsistencies. Some expressed that the new systems that were introduced were often counterintuitive. *"These programs have to be complex. And as much as they're trying to be user intuitive, they don't often succeed there because they're trying to be different from their competitors."- OM04*

C. Coping Mechanisms Reduce Technostress

a) Proactive Coping Behaviour

Separating personal and work life by using different devices for the different ICT related tasks limits the exposure to work-related ICT tasks outside of work settings. Other forms of ICT use demarcation found in the responses were related to structuring ICT use according to time periods. UCT01 mentioned blocking out a period of time to sort out an IT-related issue or to limit usage of technology by blocking out time in his calendar to avoid using technology. Four respondents alluded to sticking to a routine therefore proactively coping with the technostress. This was described as attempts to come in early to the office to factor in time to deal with a stressful incident should it arise. OM10 stated that: *"You tend to go early so you can get that desk cause other person's gonna take it."* This also linked with trying to find a suitable, adequately equipped desk with the correct devices and cables.

Six respondents mentioned preferring to work longer hours in hopes of reducing future instances of stress, therefore displaying proactive coping behaviours. This shows that by leveraging what once were technostressors (techno-overload and techno-invasion), stress can actually be reduced.

b) Reactive Coping Behaviour

Three responses pertained to reactive coping behaviours such as walking away from the stressful situation, regrouping, and closing all applications. This demonstrates the distancing coping mechanism where employees can temporarily separate themselves from the IT-related task and focus on something else [14]. Five respondents expressed that they would usually resort to venting tactics, a reactive coping behaviour, should they encounter a stressful incident related to using an ICT/ICTs. This was believed to help employees not feel isolated and to see if others are going through similar situations.

VI. Discussion of Findings

The descriptive findings uncovered relationships between different characteristics of employees and their corresponding technostress levels.

A. Findings around the Standard Technostressors

While respondents experienced the five standard technostressors, it was clear that they had more issues with them during purely remote working than under the hybrid model.

a) Techno-uncertainty as a technostressor

There was reference to techno-uncertainty triggers that occurred when new systems were introduced when remote working was initially adopted where the reliance on ICTs spiked. This caused stress related to the initial familiarisation of the new system and the minimal technical support to accompany the new system rollout. This meant employees had to dedicate additional time to learn the new system, potentially outside of work hours therefore increasing stress levels.

b) Techno-overload as a technostressor

Respondents mentioned experiencing techno-overload with regards to technology-related interruptions while working from home which triggered stress levels. Some also referred to feeling overwhelmed with the use of multiple applications that they deal with daily. This feeling of being overwhelmed could extend to experiences at the office as respondents confirmed that they used the same number of devices and applications at the office as they did at home.

c) Techno-insecurity as a technostressor

This particular factor was not very suggestive as a trigger of technostress. Only two respondents expressed feeling stressed that fellow employees may cope better with technological demands than themselves.

d) Techno-invasion as a technostressor

The data showed instances of techno-invasion in the form of extending work hours into an employee's personal time therefore distorting boundaries between work and personal spheres when working from home specifically. This was seen to also diminish work life balance. Some respondents expressed instances of lack of separation of work-related ICTs on different devices which generated stress.

Techno-complexity as a technostressor

The data suggested employee challenges with system complexity that instantiated feelings of stress. This pointed to IT skills levels struggling to match with systems' expectations which mainly stemmed from the perceived system complexity that existed, especially in new versions of systems. This can be seen to be a result of the counter intuitiveness expressed by some of the respondents which made the systems seem unfamiliar therefore triggering feelings of stress. Either employees need to be upskilled through training programmes to have their skills match with system expectations or systems need to be more simply designed.

B. Findings around the Hybrid-specific Technostressors

The data suggested that there are also distinctive causes of technostress.

a) Using hot desks creates unnecessary technostress

Since some respondents (which was found to exclude most managers) had to secure a desk each time they came into the office on their designated days, the desks often varied in equipment availability. As a result, they had to ensure coming into the office early enough to secure an appropriate desk or to hide some equipment in cupboards to ensure they would be sorted the next time when coming into the office. This was mainly due to the instances of missing equipment such as monitors, adapters, keyboards, cables etc. which meant that the equipment floated around from desk to desk. This implies that should organisations opt to keep hybrid working models, adequate resources in terms of equipment and amenities should be provided for employees on their days in-office so as to mirror their home working stations. In addition, some respondents mentioned how stressful it was to configure and synchronise various devices when coming into the office. There was reference to compatibility issues which can be seen as a direct cause of the lack of equipment availability mentioned already.

b) Office distractions related to ICT use causes stress

The comparison between working at home and at the office surfaced consistencies across some respondents regarding the disruptive atmosphere of the office. In addition to the usual office disruptions that existed before the COVID-19 lockdown, some respondents found having partial teams present at the office raised some unusual disruptions. This broke their concentration and added to their stress levels.

c) Power issues impacting access to ICTs creates stress

Loadshedding and electricity/power issues presented challenges for employees to conduct their work using ICTs that demanded sufficient power. While the organisations provided infrastructure support, often the loadshedding schedules were unpredictable and left employees in crisis situations. This meant employees experienced anxiety and stress as they were now unable to complete any work and had to make drastic arrangements to resume work. This is a macro issue specific to South Africa that can't be solved by the organisation itself but it remains crucial to implement as much support as possible to counter the unpredictable instances of loadshedding.

C. Findings around Coping Mechanisms

a) Proactive coping behaviours

The main coping behaviour was of a proactive nature utilising a problem-focused coping strategy. Most respondents resorted to proactive tactics like demarcating ICT use according to time and separation of use. Others expressed enforcing a routine to maintain a structure that could mitigate the impacts of a stressful encounter should it occur. Most respondents confirmed that working longer hours actually helped them reduce further anticipated stress. This meant logging onto systems to perform work activities on the weekend in order to reduce work backlog and hence further stress.

b) Reactive Coping Behaviours

The data showed how the standard reactive coping behaviours remain prominent when dealing with stress. This referred to emotion-focused coping strategies such as distancing oneself from the stressful situation and venting to others in hopes of reducing feelings of isolation and anxiety.

VII. Conclusion

The purpose of this research was to understand how employee technostress experiences have changed since hybrid models were adopted in South African organisations. The research wanted to find the underlying causes of these new experiences and how employees are currently coping with these new instances of technostress.

The findings suggest that the **standard** technostressors weren't as prevalent in the hybrid working model. These seemed to be more prevalent at the start of implementing the purely WFH approach and decreased somewhat under hybrid. South African employees appear to have adapted under hybrid, pointing to new emerging experiences. **Hybrid-working-specific** causes of technostress include stressful workstation setups upon return to the office on designated days which involved configuration, compatibility, and synchronisation issues along with the lack of equipment on hand. Office distractions caused unwarranted stress for employees in-office. This specifically pertained to virtual meeting noise and related echoes. Lastly, power shortages as a result of loadshedding became a hinderance and contributed to feelings of stress. *The stresses related to loadshedding and power outages appear to be a distinctive South African issue.*

To deal with technostress, employees adopted reactive and proactive coping behaviours driven by problem-focused and emotion-focused coping strategies respectively. **Reactive** behaviours involved distancing from the stressful situation and venting to others. **Proactive** behaviours involved demarcating ICT use through structured time use and separation of use. This included implementing a structured routine and working longer hours to reduce future stressful encounters.

This research had **limitations** with regard to the sample size and representation. A bigger sample size will allow better representativeness. Since the research was a cross-sectional study, technostress experiences under the hybrid model could only be recorded within the early stages of its inception; if the research was conducted over a longer period of time, a more holistic understanding could have been extracted.

Future research could compare the three modes of working (purely remote, hybrid and purely on-site) over a longer period of time to uncover more accurate understandings of the differences in the experiences of technostress. Another recommendation is to expand on the sample size.

References

[1] C. Sellberg and T. Susi, "technostress in the office: a distributed cognition perspective on human–technology interaction," *Cognition, Technology & Work,* vol. 16, no. 2, pp. 187-201, 2014/05/01 2014, doi: 10.1007/s10111-013-0256-9.

[2] M. Tarafdar, C. L. Cooper, and J.-F. Stich, "The technostress trifecta - techno eustress, techno distress and design: Theoretical directions and an agenda for research," *Information Systems Journal,* vol. 29, no. 1, pp. 6-42, 2019, doi: https://doi.org/10.1111/isj.12169.

[3] M. H. R. Bussin and C. Swart-Opperman, "COVID-19: Considering impacts to employees and the workplace," *2021,* COVID-19; employee impact; workplace impact; pandemic; performance vol. 19, 2021-08-20 2021, doi: 10.4102/sajhrm.v19i0.1384.

[4] K. Khuzaini and Z. Zamrudi, "technostress among marketing employee during the COVID-19 pandemic: Exploring the role of technology usability and presenteeism," *JEMA: Jurnal Ilmiah Bidang Akuntansi dan Manajemen,* vol. 18, no. 1, pp. 36-60, 2021, doi: https://doi.org/10.31106/jema.v18i1.10050.

[5] L. Camarena and F. Fusi, "Always Connected: Technology Use Increases technostress Among Public Managers," *The American Review of Public Administration,* vol. 52, no. 2, pp. 154-168, 2022/02/01 2021, doi: 10.1177/02750740211050387.

[6] I. Savolainen, R. Oksa, N. Savela, M. Celuch, and A. Oksanen, "COVID-19 Anxiety—A Longitudinal Survey Study of Psychological and Situational Risks among Finnish Workers," *International Journal of Environmental Research and Public Health,* vol. 18, no. 2, p. 794, 2021. [Online]. Available: https://www.mdpi.com/1660-4601/18/2/794.

[7] Y. Wang *et al.*, "Returning to the Office During the COVID-19 Pandemic Recovery: Early Indicators from China," presented at the Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 2021. [Online]. Available: https://doi.org/10.1145/3411763.3451685.

[8] A. S. Nisafani, G. Kiely, and C. Mahony, "Workers' technostress: a review of its causes, strains, inhibitors, and impacts," *Journal of Decision Systems,* vol. 29, no. sup1, pp. 243-258, 2020/08/18 2020, doi: 10.1080/12460125.2020.1796286.

[9] P. Spagnoli, M. Molino, D. Molinaro, M. L. Giancaspro, A. Manuti, and C. Ghislieri, "Workaholism and technostress During the COVID-19 Emergency: The Crucial Role of the Leaders on Remote Working," (in English), *Frontiers in Psychology,* Brief Research Report vol. 11, 2020-December-23 2020, doi: 10.3389/fpsyg.2020.620310.

[10] I. Hwang and O. Cha, "Examining technostress creators and role stress as potential threats to employees' information security compliance," *Computers in Human Behavior,* vol. 81, pp. 282-293, 2018/04/01/ 2018, doi: https://doi.org/10.1016/j.chb.2017.12.022.

[11] L. Atanasoff and M. A. Venable, "technostress: Implications for Adults in the Workforce," *The Career Development Quarterly,* vol. 65, no. 4, pp. 326-338, 2017, doi: https://doi.org/10.1002/cdq.12111.

[12] X. Zhao, Q. Xia, and W. Huang, "Impact of technostress on productivity from the theoretical perspective of appraisal and coping processes," *Information & Management,* vol. 57, no. 8, p. 103265, 2020/12/01/ 2020, doi: https://doi.org/10.1016/j.im.2020.103265.

[13] R. Berger, M. Romeo, G. Gidion, and L. Poyato, "Media use and technostress," in *INTED2016 Proceedings*, 2016: IATED, pp. 390-400, doi: https://doi.org/10.21125/inted.2016.1092.

[14] M. Tarafdar, H. Pirkkalainen, M. Salo, and M. Makkonen, "Taking on the "dark side"—Coping with technostress," *IT professional,* vol. 22, no. 6, pp. 82-89, 2020, doi: 10.1109/MITP.2020.2977343.

[15] A. Bhattacherjee, *Social science research: Principles, methods, and practices*. 2012.

[16] T. Azungah, "Qualitative research: deductive and inductive approaches to data analysis," *Qualitative Research Journal,* vol. 18, no. 4, pp. 383-400, 2018, doi: 10.1108/QRJ-D-18-00035.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 949–953
DOI: 10.15439/2023F7569
ISSN 2300-5963 ACSIS, Vol. 35

Improving Domain-Specific Retrieval by NLI Fine-Tuning

Roman Dušek
Allegro sp. z o.o.
Wierzbięcice 1B, 61-569 Poznań, Poland
Email: roman.a.dusek@allegro.com

Christopher Galias, Lidia Wojciechowska
Allegro sp. z o.o.
Wierzbięcice 1B, 61-569 Poznań, Poland
Email: {krzysztof.galias,lidia.wojciechowska}@allegro.com

Aleksander Wawer
0000-0002-7081-9797
* Allegro sp. z o.o.
Wierzbięcice 1B, 61-569 Poznań, Poland
** Institue of Compter Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warszawa
Email: ** axw@ipipan.waw.pl, * aleksander.wawer@allegro.com

***Abstract*—The aim of this article is to investigate the fine-tuning potential of natural language inference (NLI) data to improve information retrieval and ranking. We demonstrate this for both English and Polish languages, using data from one of the largest Polish e-commerce sites and selected open-domain datasets. We employ both monolingual and multilingual sentence encoders fine-tuned by a supervised method utilizing contrastive loss and NLI data. Our results point to the fact that NLI fine-tuning increases the performance of the models in both tasks and both languages, with the potential to improve mono- and multilingual models. Finally, we investigate uniformity and alignment of the embeddings to explain the effect of NLI-based fine-tuning for an out-of-domain use-case.**

I. Introduction

QUERY and sentence embedding vectors are used in information retrieval to match the searched query to results, for example in ranking of the results returned by lexical search engines [1] or in vector-based similarity search [2].

The standard approach to training text encoders is to use large-scale corpora such as Wikipedia or CommonCrawl and the Masked Language Modeling (MLM) objective. A setup like this was used to train HerBERT [3], the state-of-the-art monolingual BERT for the Polish language, which utilized Polish-specific datasets and the Sentence Structural Objective in addition to MLM. CommonCrawl, Wikipedia, and MLM were also used to train XLM-RoBERTa [4], a transformer supporting 100 languages.

In past years there have been numerous applications of natural language inference (NLI) data in training large language models such as sentence encoders. One example supporting the Polish language is the multilingual Universal Sentence Encoder (USE) [5]. For the 16 covered languages, training data included question-answer pairs, translation pairs, and the SNLI [6] corpus, translated using Google Translate into target languages. The model was trained in a dual encoder setup and comes in two variants: a lightweight convolutional neural network and a transformer.

Recently, NLI data were applied in a combination with contrastive loss in a method called SimCSE [7]. It demonstrated superior performance on STS (Semantic Textual Similarity) tasks. Contrastive fine-tuning was also reported to improve ranking quality when applied to multilingual encoders [8].

Unfortunately, large NLI datasets suitable for model training are usually not available in languages other than English. For this reason, in this work we test the feasibility of using machine translated NLI data and demonstrate this approach for Polish. We will use both monolingual (Polish and English) and multilingual models and evaluate them on data in both languages.

In this paper, we focus on two information retrieval tasks: the retrieval task, which aims to find a set of documents that match the query, and the ranking task, which sorts the results by relevance to the query. To demonstrate the proposed approach, our experiments will be performed on out-of-domain models, by which we mean generic, pre-trained neural language models that have not been tuned to real-world search data such as user clicks. We explore the impact of using translated NLI data for contrastive fine-tuning. We consider how does the fine-tuning affect information retrieval and ranking tasks. Furthermore, we investigate whether the uniformity and alignment of embeddings are linked to out-of-domain information retrieval performance.

The paper is organized as follows: in Section II we introduce datasets and experimental setup, Section III discloses results and Section IV concludes the paper by drawing conclusions.

II. Experiments

A. Datasets

We examined the performance of the models on three types of benchmarks.

The first one is not directly related to information retrieval. This is a generic approach to evaluate pre-trained large neural language models. The first part is based on a GLUE-like collection for testing the selected model on a number of downstream benchmarks. We use it in Polish, where such a benchmark is the KLEJ framework [9]. In our paper we report averaged model performance on KLEJ datasets. The second part consists of semantic textual similarity (STS) tasks:

- translated SICK-R [10] available from the Polish version of SentEval[1],
- CDS-R [11], a Polish dataset based on SICK-R,
- translated STSB[2].

These datasets contain pairs of sentences human labelled based on the relatedness.

The second benchmark is ranking using a random sample consisting of 86K search listings from one of the largest e-commerce platforms in Poland. The listings consist of a search phrase and the first page of results (on average 50 offers) from the lexical search engine along with information about the clicked items. We sorted the listings according to the cosine similarity between the embedding of the search phrase and the embedding of each offer title. We assessed the performance of the models by calculating click-based NDCG and averaging the results.

The third benchmark consists of two retrieval tasks. Here we applied Polish monolingual and multilingual models used in previous benchmarks, but also English monolingual models to extend our research to other languages. To evaluate Polish models in the retrieval task, we used an internal dataset from one of the largest e-commerce Polish platforms, which consists of search results. It is a sample of 30K user queries and 1M product titles, containing at least one clicked product for each of the user queries. English language models were tested on two datasets. The first one is WANDS [12], a similar dataset from the e-commerce domain. Its test subset contains 379 queries and 43K candidate products with human-labelled query-product pairs. The main purpose is evaluation of semantic search in e-commerce. To broaden our evaluation, we further tested English models on the second English dataset, outside of e-commerce, namely SciFact [13]. It is included in BEIR [14], an information retrieval benchmark. SciFact's test subset contains 300 scientific claims (queries) verified against a corpus of 5K abstracts.

B. NLI translation

We evaluated the translations using COMET (Crosslingual Optimized Metric for Evaluation of Translation) [15] scores, an automated method of assessing translation quality. COMET is a new neural framework for evaluating multilingual machine translation models. COMET is designed to predict human judgments of machine translation quality. We used the older model, namely wmt20-comet-qe-da[3] to compare the translation results. The newer COMET release has a better correlation with human evaluation and a less skewed distribution of scores, but the calculated values were more difficult to interpret and establish a threshold value that indicates good vs bad translation quality.

The mBart[4] model reached score a of 0.49 compared to 0.40 of m2m100[5], which is why we decided to translate the data using mBart. We also experimented with choosing the best of two translations for each sentence, which we comment on later in Section III-D.

C. Training details

We selected several models for fine-tuning with the supervised SimCSE framework[6]. In the case of Polish, we applied SimCSE to the Polish monolingual model HerBERT [3], which achieved top scores in the Polish KLEJ benchmark. In the case of English, we selected the English-only monolingual base variant of BERT (BERT-base-uncased) [16]. Finally, we applied SimCSE to the multilingual model XLM-RoBERTa [4], which also is the best multilingual model on the KLEJ leaderboard. We fine-tuned HerBERT and XLM-RoBERTa models using the SNLI dataset translated to Polish[7], and the English SNLI and MNLI data in the case of English BERT and XLM-RoBERTa (in the case of English fine-tuning).

D. SimCSE: Contrastive loss using NLI

SimCSE [7] is a contrastive learning method aimed at generating sentence embeddings. First, it utilizes an unsupervised approach, which takes an input sentence and predicts itself in contrastive objective, with dropout used as noise. Authors find that dropout acts as minimal data augmentation, and removing it leads to a representation collapse. Then, they propose a supervised approach, which incorporates annotated pairs from natural language inference (NLI) datasets into the contrastive learning framework by using "entailment" pairs as positives and "contradiction" pairs as hard negatives. The contrastive loss is formulated for paired examples $D = \left\{\left(x_i, x_i^{+}\right)\right\}_{i=1}^{m}$, where x_i and x_i^{+} are semantically related. Assuming that h_i and h_i^{+} are representations of x_i and x_i^{+}, the training objective is:

$$\ell_{contrastive} = -\log \frac{\mathrm{e}^{\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_i^{+}\right)/\tau}}{\sum_{j=1}^{N} \mathrm{e}^{\operatorname{sim}\left(\mathbf{h}_i, \mathbf{h}_j^{+}\right)/\tau}}$$

where τ is a temperature hyperparameter and sim(h_i, h_i^{+}) is the cosine similarity.

Following the SimCSE [7] we used their supervised training framework to fine-tune selected models on SNLI dataset translated into Polish. This supervised task takes advantage of human-labelled pairs of sentences. As in the original work, we treated entailment pairs as positives and contradiction pairs as a hard negatives.

[1] https://github.com/sdadas/polish-sentence-evaluation

[2] https://huggingface.co/datasets/stsb_multi_mt/viewer/pl/train

[3] https://github.com/Unbabel/COMET

[4] https://huggingface.co/facebook/mbart-large-50-one-to-many-mmt

[5] https://huggingface.co/docs/transformers/model_doc/m2m_100

[6] https://github.com/princeton-nlp/SimCSE

[7] We also tested a combination with MNLI, but this resulted in worse performance in information retrieval tasks.

E. Uniformity and alignment

Wang et al. [17] identify two key properties of embeddings, *uniformity* and *alignment*, and propose to use them to measure embedding quality. Later work [18] in the recommender domain also suggests that better uniformity and alignment increases NDCG. Alignment is meant to measure whether similar samples have similar embeddings and is given by

$$\ell_{align} \triangleq \mathbb{E}_{(x,y)\sim p_{pos}} \| f(x) - f(y) \|_2^{\alpha}, \quad \alpha > 0,$$

where f is a function mapping an entity to its embedding and p_{pos} is a distribution of positive pairs. Uniformity measures whether maximal information is preserved between the input and embedding space, which leads to spreading out of the representations, and is given by

$$\ell_{uniform} \triangleq \log \mathbb{E}_{x,y\sim p_{data}} \left[e^{-t\|f(x)-f(y)\|_2^2} \right], \quad t > 0,$$

where p_{data} is the input distribution.

III. Results

A. Results of SimCSE with translated NLI

As we can see in Table I, the role of SimCSE is ambiguous: it greatly improves the STS performance, but in the case of the best Polish monolingual model Herbert, it degrades its performance on the KLEJ benchmark.

The results regarding STS and general benchmarks such as KLEJ agree with the observations of SimCSE authors in [7]. They are somewhat selective: the focus is on evaluating SimCSE on semantic textual similarity (STS), and indeed in this benchmark their method performs in a competitive manner. However, the performance on many other typical downstream tasks, such as for example GLUE benchmark's sentiment analysis, is not competitive and is mentioned only in the appendix of the SimCSE paper. Authors conclude that sentence-level objective of SimCSE may not directly benefit such transfer tasks.

B. Results of information retrieval benchmark

Table II presents the results of the English benchmark. To get the best possible performance from used models we use both mean-pooling (average representation of tokens in sequence) and the CLS token representations. This doesn't discriminate against models which are not fine-tuned for utilisation of the CLS token (e.g. BERT). Tables III and IV show results of the Polish language tasks. Generally, SimCSE fine-tuning improves both NDCG and recall. For both languages the best results in terms of retrieval, as reflected in Recall@100 scores, were obtained by monolingual BERTs with SimCSE fine-tuning. Except for the case of the English WANDS benchmark, USE was second in terms of performance, ahead of XLM-RoBERTa fine-tuned by SimCSE. In the ranking task HerBERT, SimCSE-HerBERT, and USE shared first place when using the mean of the last hidden layer to represent the utterance. In the CLS+pooler representation, SimCSE-HerBERT was the best one.

C. Uniformity and alignment

We calculated uniformity and alignment using the search phrase and title with a click, utilizing a batch size of 1024 over 300K of pairs, with the default $\alpha = 2$ and $t = 2$. Contrastive fine-tuning improved the performance of both HerBERT and XLM-RoBERTa. However, only uniformity improved as the alignment metric increased (see Figure 1).

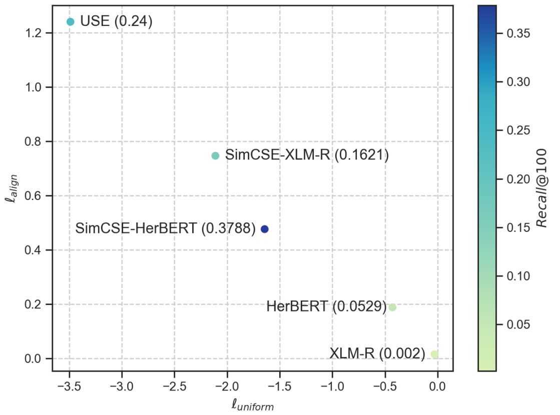


Fig. 1. Recall@100 on the plot of ℓ_{align} versus $\ell_{uniform}$ on vector-search dataset. For both axes lower is better. Colors and numbers in parentheses indicate Recall@100.

D. Influence of translation quality

In order to examine the influence of poorly translated sentences we conducted experiments where we filtered translated sentences based on the COMET score. Using both translation from mBart and m2m100 models, we selected the highest COMET score translation to pick one example from each of the translated datasets. The average COMET score on SNLI rose by 7 percentage points after filtering. After inspecting the cleaned datasets many examples with scores close to zero were still found. Removing examples with scores lower than 0.05 resulted in reducing the dataset size by 1/3. Fine-tuning the model on the cleaned dataset resulted in worse performance than baseline.

IV. Discussion

Using the translated SNLI dataset had a comparable effect to the results reported in [7]. This confirms the role of translated NLI for improving the model performance, even despite possible translation errors.

[8] We used the transformer variant available at https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3

[9] KLEJ value cannot be computed for USE in a manner directly comparable to other solutions, because it supports only one input and does not support the '[SEP]' special tokens as the other transformer models do. Some of the KLEJ subsets are paired, as for example question-answer or paraphrase data.

[10] We computed statistical significance of averaged NDCGs using the paired T-test, p-value$<$ 0.05. Non-significant pairs where we could not confirm the differences were USE vs SimCSE-HerBERT and XLM-RoBERTa vs HerBERT. In other cases the differences are statistically significant.

TABLE I
RESULTS OF THE STS AND KLEJ EVALUATION TASKS AND NUMBER OF SUPPORTED LANGUAGES (#LANGS).

	STSB-PL	SICK-R	CDS-R	Avg STSB-PL	Avg KLEJ	#langs
HerBERT	0.302	0.369	0.605	0.425	**86.3**	1
SimCSE-HerBERT	0.742	**0.781**	0.905	**0.809**	84.5	1
XLM-RoBERTa	0.584	0.561	0.821	0.655	81.5	100
SimCSE-XLM-RoBERTa	0.727	0.766	0.888	0.793	81.7	100
USE[8]	**0.749**	0.691	**0.909**	0.783	-[9]	16

TABLE II
RESULTS OF EVALUATION ON RETRIEVAL TASK USING ENGLISH DATASETS. NUMBERS REPORTED REPRESENT RECALL@100.

	WANDS		BEIR-SciFact	
Model / Inference Pooling	mean	CLS+pooler	mean	CLS+pooler
BERT-base-uncased	0.2543	0.0632	0.5134	0.0200
SimCSE-BERT-base-uncased	0.4933	**0.4991**	**0.7832**	0.6306
XLM-RoBERTa	0.1648	0.1458	0.1506	0.2368
SimCSE-XLM-RoBERTa	0.3986	0.4338	0.5701	0.6878
USE	0.3964	-	0.7665	-

TABLE III
RESULTS OF EVALUATION ON RANKING TASK IN POLISH. NUMBERS REPORTED REPRESENT NDCG[10].

	Ranking test set	
Model / Pooling	mean	CLS+pooler
HerBERT	**0.312**	0.307
SimCSE-HerBERT	**0.312**	**0.312**
XLM-RoBERTa	0.306	0.305
SimCSE-XLM-RoBERTa	0.309	0.309
USE	**0.312**	-

TABLE IV
RESULTS OF EVALUATION ON RETRIEVAL TASK IN POLISH. NUMBERS REPORTED REPRESENT RECALL@100.

	Retrieval test set	
Model / Pooling	mean	CLS+pooler
HerBERT	0.0230	0.0222
SimCSE-HerBERT	0.2476	**0.2562**
XLM-RoBERTa	0.0020	7.48e-5
SimCSE-XLM-RoBERTa	0.1487	0.1621
USE	0.2407	-

The USE model competes with monolingual models when it comes to STS benchmarks. Contrastive loss, as applied in SimCSE, is not used in the USE model. Moreover, the USE model is multilingual, as it supports 16 languages, and it contains only 80 mln parameters in the large variant, compared to 110 mln of the HerBERT and XLM-RoBERTa base versions. The only element that is common to both the USE and HerBERT with SimCSE fine-tuning is the usage of NLI data for model training. Therefore, we conclude that it is the NLI fine-tuning that plays the key role in information retrieval and STS performance.

Another interesting observation is that the averaged KLEJ score is not related to information retrieval capability. However, better performance on the semantic textual similarity tasks (STSB-PL, SICK-R and CDS-R) is. Our results demonstrate that SimCSE fine-tuning degrades monolingual model performance on the KLEJ benchmark, therefore it should not be considered as a one-size-fits-all method for tuning language models. We believe that using NLI data for model pre-training and/or fine-tuning has a positive effect in representing text for information retrieval problems.

We observed a link between information retrieval and uniformity dimension only. We did not observe a relationship between alignment and information retrieval as is reported in [7] or in the context of recommender systems [18]. Previous work assessed alignment and uniformity using an in-domain setting, compared to our case of an out-of-domain scenario — but the impact of this setting concerning alignment remains an open research question.

All multilingual models scored higher on uniformity compared to monolingual models. We believe this is because multilinguality makes the model use more of the embedding space. Moreover, the alignment of all multilingual models was worse compared to monolingual models. This shows that alignment and uniformity do not directly translate to capabilities of sentence encoders.

V. CONCLUSIONS AND FUTURE WORK

Our results show that state-of-the-art performance in out-of-domain retrieval and ranking tasks can be achieved with a method based on contrastive loss and NLI data, such as SimCSE, applied to a pre-trained language model. We confirm the positive effect of contrastive loss using both monolingual and multilingual models, pointing to the conclusion that the key to superior performance in out-of-domain information retrieval is fine-tuning sentence encoders using NLI data.

In this paper we did not train the model on clicks. This could be done using contrastive loss. In the future we plan to optimize sentence encoders on click data using alignment and uniformity in the loss function, as in [18].

REFERENCES

[1] W. Guo, X. Liu, S. Wang, H. Gao, A. Sankar, Z. Yang, Q. Guo, L. Zhang, B. Long, B.-C. Chen, and D. Agarwal, "DeText: A deep text ranking framework with BERT," in *Proceedings of the 29th ACM International Conference*

on Information & Knowledge Management, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 2509–2516. [Online]. Available: https://doi.org/10.1145/3340531.3412699

[2] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

[3] R. Mroczkowski, P. Rybak, A. Wróblewska, and I. Gawlik, "HerBERT: Efficiently pretrained transformer-based language model for Polish," in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*. Kiyv, Ukraine: Association for Computational Linguistics, Apr. 2021, pp. 1–10. [Online]. Available: https://www.aclweb.org/anthology/2021.bsnlp-1.1

[4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[5] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," 2019. [Online]. Available: https://aclanthology.org/2020.acl-demos.12.pdf

[6] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[7] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552

[8] R. Litschko, I. Vulić, S. P. Ponzetto, and G. Glavaš, "On cross-lingual retrieval with multilingual text encoders," *Information Retrieval Journal*, vol. 25, no. 2, pp. 149–183, Jun. 2022. [Online]. Available: https://doi.org/10.1007/s10791-022-09406-x

[9] P. Rybak, R. Mroczkowski, J. Tracz, and I. Gawlik, "KLEJ: Comprehensive benchmark for Polish language understanding," Online, pp. 1191–1201, Jul. 2020. [Online]. Available: https://aclanthology.org/2020.acl-main.111

[10] S. Dadas, M. Perełkiewicz, and R. Poświata, "Evaluation of sentence representations in Polish," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, 2020, pp. 1674–1680. [Online]. Available: https://aclanthology.org/2020.lrec-1.207

[11] A. Wróblewska and K. Krasnowska-Kieraś, "Polish evaluation dataset for compositional distributional semantics models," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 784–792. [Online]. Available: https://aclanthology.org/P17-1073

[12] Y. Chen, S. Liu, Z. Liu, W. Sun, L. Baltrunas, and B. Schroeder, "WANDS: Dataset for product search relevance assessment," in *Proceedings of the 44th European Conference on Information Retrieval*, 2022.

[13] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or Fiction: Verifying Scientific Claims," in *EMNLP*, 2020.

[14] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=wCu6T5xFjeJ

[15] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2685–2702. [Online]. Available: https://aclanthology.org/2020.emnlp-main.213

[16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[17] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 9929–9939. [Online]. Available: https://proceedings.mlr.press/v119/wang20k.html

[18] C. Wang, Y. Yu, W. Ma, M. Zhang, C. Chen, Y. Liu, and S. Ma, "Towards representation alignment and uniformity in collaborative filtering," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1816–1825. [Online]. Available: http://arxiv.org/abs/2206.12811

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 955–959

DOI: 10.15439/2023F9586
ISSN 2300-5963 ACSIS, Vol. 35

Ant Colony Optimization for Workforce Planning with Hybridization

Stefka Fidanova
Institute of Information and Communication Technology
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., bl. 25A,
Sofia, Bulgaria
E-mail: stefka.fidanova@iict.bas.bg

Maria Ganzha
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: maria.ganzha@ibspan.waw.pl

Abstract—**Production organization plays a key role in the success of any enterprise. Optimizing workforce planning can improve the overall organization of production. The main goal is to minimize the assignment cost of the workers who will perform the planned work. The problem is known to be NP-hard, therefore we will apply methods from the field of artificial intelligence. For this reason, most of the existing methods hardly find feasible solutions. We propose Ant Colony Optimization Algorithm with hybridization, combination with local search procedures. We compare and analyze their performance.**

Index Terms—**Workforce Planning, Ant Colony Optimization, Metaheuristics, Hybrid Method, Local Search**

I. Introduction

PROPER management of human resources plays an important role in the organization of production. It is common problem for all industrial sectors. It is NP-hard optimization problem, which includes a lot of level of complexity. Workforce planning is the process of determining the skills and human resources needed to perform a given task. The problem consists of two parts: selection and assignment. First the employers are selected from the set of available workers. After they are assigned to jobs, which they will perform. The aim is minimization of assignment cost, while staying within the framework of work requirements. Human resource management includes workforce planning. Exact methods as well as traditional numerical methods are unable to solve this problem for instances with realistic size. These types of methods can be applied only on special simplified variants of the problem.

It exist various metaheuristic algorithms applied on workforce planning problem. They include genetic algorithm [1], memetic algorithm [11], scatter search [1] etc.

Ant Colony Optimization (ACO) algorithm is proven to be very effective solving various complex optimization problems [5], [10]. In our previous work [6], [7] we propose ACO algorithm for workforce planning. We have considered the variant of the workforce planning problem proposed in [1].

Current paper is the continuation of [7] and [8]. Other variant of hybridization is proposed. They are compared and discussed. The aim is to improve algorithm efficiency .

The rest of the paper is organized as follows. The mathematical description of the problem is presented in Section 2. ACO algorithm for workforce planing problem is presented in Section 3. Computational results, comparisons of different hybridization and discussion are done in Section 4 . A conclusion and directions for future work are proposed in Section 5.

II. Workforce Planning Problem

In this section we will give definition and description of the variant of Workforce Planing Problem (WPP) we solve. We intend the variant of the problem considered by Alba [1] and Glover [9].

There is a fixed period of time and a set of jobs $J = \{1, \dots, m\}$. All jobs need to be finished during this period. For every job j is known that it requires d_j hours to be completed. There are workers, which are candidates for assignment to perform the jobs, the set $I = \{1, \dots, n\}$. In terms of work quality and efficiency, each worker must work on each of their assigned jobs for a minimum of h_{min} hours. We know the availability of every worker, worker i is available for s_i hours. Workers may have different qualifications and may not be qualified for all the tasks to be performed. The set A_i contains the jobs, for which worker i is qualified. There is a limit t to the maximum number of workers that can be assigned during this period. This means that at most t workers can be selected from a set I of workers, and this must be done in such a way that they are able to perform and complete the planned work. The worker i is assigned to perform job j at c_{ij}. The purpose is to find feasible solution, that minimize assignment price, which is the objective function of this problem.

The following is the description of the mathematical model of the workforce planing problem:

$$x_{ij} = \begin{cases} 1 & \text{if the worker } i \text{ is assigned to job } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if worker } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ij} = \text{number of hours that worker } i$$

$$\text{is assigned to perform job } j$$

Q_j = set of workers qualified to perform job j

$$\text{Minimize} \sum_{i \in I} \sum_{j \in A_i} c_{ij}.x_{ij} \tag{1}$$

Subject to

$$\sum_{j \in A_i} z_{ij} \le s_i.y_i \qquad i \in I \tag{2}$$

$$\sum_{i \in Q_j} z_{ij} \ge d_j \qquad j \in J \tag{3}$$

$$\sum_{j \in A_i} x_{ij} \le j_{max}.y_j \qquad i \in I \tag{4}$$

$$h_{min}.x_{ij} \le z_{ij} \le s_i.x_{ij} \qquad i \in I, j \in A_i \tag{5}$$

$$\sum_{i \in I} y_i \le t \tag{6}$$

$$\begin{array}{ll} x_{ij} \in \{0,1\} & i \in I, j \in A_i \\ y_i \in \{0,1\} & i \in I \\ z_{ij} \ge 0 & i \in I, j \in A_i \end{array}$$

Every manufacturer strives to reduce the cost of production. This can be achieved with good organization and optimization of production process. One of the biggest costs is the cost of hiring workers. Therefore, workforce planning and optimization is a fundamental issue for every enterprise. The goal of the problem of workforce planning is the minimization of the total assignment cost, respecting the constraints. Inequality 2 represents the limitation of the number of hours the selected worker can be assigned. Inequality 3 show the completion time for hall jobs. The limitation of the number of jobs, that every worker can perform is done by the inequality 4. If a worker works too short on a job, his work will be inefficient and often of poor quality. Therefore, a minimum amount of time is required for each worker to work on each of their assigned jobs. This requirement is represented by inequality 5. There is always some reason to limit the number of workers working at the same time. This may be the available space; the amount of tools; number of machines or something else. The limitation of the number of the assigned workers is represented by inequality 6.

This mathematical model of the workforce planning problem can be used with a variety of objective functions, depending on what our goal is and what we want to optimize. Regarding the goal, there are various variants of the problem. The focus of this paper is minimization of total assignment cost. Let's $\tilde{c}_{ij}$ is the cost the worker i to performs the job j for one hour. The cost of assigning workers to complete all assigned jobs is represented by function 7. Minimizing this function is the objective function used in this paper.

$$f(x) = \text{Min} \sum_{i \in I} \sum_{j \in A_i} \tilde{c}_{ij}.x_{ij} \tag{7}$$

Some of the workers may have preferences for some of the activities for which they are qualified. In this case, the objective function would be the maximum satisfaction of their desires. Another option is for the task to have two objective functions. Simultaneous minimization of the total cost of appointment and maximum satisfaction of preferences.

Workforce planning problems fall into two broad groups: structured and unstructured. The problem is structured when the time to complete a job is proportional to the minimal time the worker need to work on separate job, or parameter d_j is proportional to the parameter h_{min}. When d_j is not proportional to the parameter h_{min} the problem is unstructured. The algorithms find more frequently feasible solutions for structured problems, then for unstructured.

III. Hybrid Ant Colony Optimization Algorithm

One of the most successful methods for solving combinatorial optimization problems is Ant Colony Optimization (ACO). It is a metaheuristics, following the real ants behavior when looking for a food. Normally ants use chemical substance, called pheromone, to mark their path ant to can return back.

A. Main ACO Algorithm

NP-hard problems and in particular combinatorial optimization problems require exponential number of calculations and memory use. So large problems can not be solved for reasonable time by exact algorithms or traditional numerical methods [3].

First realization of the idea to use ant behavior is applied by Marco Dorigo [2] for solving Traveling Salesman Problem. Later some modifications and improvements are proposed, mainly in pheromone updating rules [3] and the method was applied on big variety of combinatorial optimization problems. The ACO methodology is based on the ants behavior simulation. One of the main things in the algorithm is the representation of the problem by graph, called graph of the problem. This allows solutions to be represented as paths in the graph. The problem boils down to finding a shortest path in a graph subject to given constraints.

The transition probability $P_{i,j}$ leads the ants how to choose the next node j to be added to the partial solution, when the last node selected is i. It is a product of the heuristic information $\eta_{i,j}$ and the pheromone trail quantity $\tau_{i,j}$ corresponding to the move from node i to the node j, where $i, j = 1, \dots, n$. The transition probability formula is as follows:

$$P_{i,j} = \frac{\tau_{i,j}^a \eta_{i,j}^b}{\sum\limits_{k \in Unused} \tau_{i,k}^a \eta_{i,k}^b}, \tag{8}$$

where $Unused$ is the set of unused nodes of the problem graph, a and b are the influence of the pheromone and the heuristics information, respectively.

Equality 8 shows that the attractiveness of a node increases, when the heuristic information and/or the quantity of the pheromone related to it increases, because the probability the node to be selected increases and it becomes more advantageous.

The level of the initial pheromone is the same for all graph elements and is set to a small positive constant value τ_0, $0 < \tau_0 < 1$. The algorithm is iterative and the goal is on the next iteration the ant to try to construct new better solutions, taking in to account the information from previous iterations. At the end of every iteration the ants update the pheromone values of graph elements according the quality of the achieved solution during the iteration. Different ACO algorithms utilize different procedures for updating pheromone values [3]. A node, from the problem graph, becomes more desirable if it accumulates more pheromone, but the accumulation of too much pheromone can lead to stagnation and repetition of the same solutions without their improvement.

The main update rule for the pheromone trail level is:

$$\tau_{i,j} \leftarrow \rho\tau_{i,j} + \Delta\tau_{i,j}, \tag{9}$$

where ρ is the evaporation parameter. It decreases the value of the old pheromone, because the old information is not so current. On the other hand, we do not lose it, but only reduce its influence. Thus we mimics evaporation in a nature and try to prevent early stagnation and help the ants to avoid local minima. $\Delta\tau_{i,j}$ is a new added pheromone, and it is proportional to the quality of the newly constructed solution. The quality of the newly constructed solutions is measured by the values of the objective function, corresponding to these solutions.

B. Workforce Planing Problem ACO Algorithm

In this section we describe ACO algorithm for workforce planning without local search procedure from our previous paper [6]. Proper graph representation of the problem play important role in ACO algorithm application. The problem is described by 3 dimensional graph. Essential is which elements of the problem are represented by the nodes and what is the meaning of the arcs. In our problem the node (i, j, z) represents the worker i assigned to the job j for time z. The maximal value of z is dependent of the completion time of job j. Completion time is different for different jobs, so the graph of the problem is asymmetric.

As we mentioned in the subsection above, an ant starts solution construction from a random node of the graph of the problem. Thus at the beginning of every iteration we generate three random numbers for every ant. The first random number belongs to the interval $[0, \dots, n]$ and shows to the worker who is chosen to be assigned. The second random number belong to the interval $[0, \dots, m]$. It is related with the job, the worker is assigned to do. In case the worker is not qualified to do this job, a new job is chosen in a random way. The third random number belong to the interval $[h_{min}, \min\{d_j, s_i\}]$ and is related with number of hours worker i is assigned to performs job j.

By traditional ACO algorithm next nodes are included applying transition probability rule. These steps are repeated till all ants construct their solutions. The termination condition of the solution construction process is the impossibility of adding new nodes without violating any of the constraints of the problem.

We propose the following heuristic information to be applied, where worker i, performs job j for time l, formula 10:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & l = z_{ij} \\ 0 & otherwise \end{cases} \tag{10}$$

This heuristic information incentives the assignment of the cheaper workers for as long as possible, thereby reducing the overall cost of assigning the workers. Following the rules of ACO algorithm the next included node in the partial solution is the node with highest probability. If there happen to be several nodes with a probability equal to the maximum, then one of them is chosen at random as the next node in the partial solution. Each time a new node is included, it is checked whether the constraints of the problem are not violated, only then the new node is accepted.

If any of the constraints is not satisfied, then the value of the transition probability function corresponding to this node is set to be 0. If for all possible nodes the value of the probability function is 0 the solution construction stops, since it is impossible to include new node in the current partial solution. When the achieved solution is feasible, the value of the objective function is calculated as a sum of assignment cost of all assigned workers. The value of the objective function can not be negative. Therefore we set the value of the objective function to be -1 for infeasible solutions.

We deposit additional pheromone only on the elements of feasible solutions and it reflects the quality of the problem solution, which is measured by the value of the objective function. Workforce planning problem is a minimization problem, so the new added pheromone is proportional to the reciprocal value of the objective function:

$$\Delta\tau_{i,j} = \frac{\rho - 1}{f(x)} \tag{11}$$

So the elements of the graph of the problem, belonging to better solutions with less value of the objective function will accumulate more pheromone than others and will be more wanted in the next iteration. The global best so far solution is updated at the end of every iteration. We compare the iteration best solution with the current global best one and if the iteration best solution is better, with less value of the objective function, we accept it as a new global best solution. In our application as end condition we apply number of iterations. When the algorithm reaches the pre-fixed number of iterations, it stops further calculations.

IV. Local Search Procedures

A common practice is to combine a metaheuristic algorithm with some other algorithm. This can be another metaheuristic algorithm, a numerical method, an exact method, or a local search procedure. These are the so called hybrid approaches.

The purpose of combining algorithms can be in several directions. The combination can be aimed at avoiding local optima or falling into a region of infeasible solutions. In this case, a combination with a local search procedure is usually used. Another goal may be to prevent early stagnation of the algorithm and find better solutions. The combination of methods, especially if it is applied to each iteration, leads to an increase in the time to run the iteration. Combining methods can lead to finding good solutions at an earlier stage, with fewer iterations, and in turn reduce the time to solve the problem.

In this paper we propose several variants of local search procedures, which are specifically tailored to the workforce planning problem. Our aim is decrease the number of infeasible solutions and thus to increase the diversification.

Local search procedures generate one or more solutions to the problem based on a current solution. These solutions are called neighborhood solutions. If the neighboring solutions thus generated are feasible, then we compare the best among the feasible neighboring solutions with the current solution. If the neighboring solution is better than the current one, then we replace the current solution of the problem with the neighboring one.

As noted, the local search procedure increases the execution time of a single iteration. If it is not efficient enough, it could also increase the execution time of the algorithm, the time to find good solutions. We apply the local search procedure only on the infeasible solutions. Our goal is to increase the number of feasible solutions and thus increase the choice. This, in turn, could lead to finding good solutions at an early stage of algorithm execution, which would reduce the time to solve the problem.

The main thing in the local search procedures that we offer is the removal of some of the appointed workers and the appointment of new ones in their place. After removing part of the workers, we get a partial solution, which is supplemented by assigning new workers by the use of ACO algorithm. The algorithm is stochastic, thus with a high probability, the new solution will be different from the previous one. From our previous research [7], we have found that it is best to remove half of the assigned workers.

We have compared three variants of the local search procedure:

- The workers to be removed are randomly selected. The procedure is applied once, regardless of whether the new solution is feasible or not [7];
- The workers to be removed are randomly selected. The procedure is repeated until a valid solution is constructed [8];
- The most expensive workers are removed. The procedure is applied once, regardless of whether the new solution is feasible or not.

The workforce planning problem is very complex with tight constraints. Because of this, it happens that there are iterations in which no ant succeeds in finding a feasible solution. We observe that after applying any of the listed procedures for local search, the number of infeasible solutions in subsequent iterations is greatly reduced. So the local search procedure is mainly applied to the first iterations. In subsequent iterations, it is less and less necessary to apply it. Due to the application of the local search procedure only on the infeasible solutions and reducing the need to apply it on subsequent iterations, it does not significantly increase the execution time of the algorithm.

V. Computational Results and Discussion

In this section are shown and compared the test results of application of proposed hybridization. The proposed local search procedures, combined with the ACO algorithm are tested on 10 structured and 10 unstructured problems. In our previous work [7] we research on the impact of the number of the removed workers from the solution. We tested with removing a quarter of the assigned workers, removing half of the assigned workers and removing all of the assigned workers (full restart). We found that the best results are achieved when removing half of the assigned workers. So in this work, when we apply any of the proposed local search procedures , we remove half of the assigned workers and complete the solution applying ACO algorithm.

The software, which realizes the algorithm is written in C programming language and is run on Pentium desktop computer at 2.8 GHz with 4 GB of memory. The proposed hybridizations are tested on artificially generated problem instances from [1].

The set of test problems consist of 10 Structured problems, enumerated from S1 to S10 and 10 Unstructured problems, enumerated respectively from U1 to U10. A problem is structured, when parameter d_j is proportional to the parameter h_{min} and it is unstructured when d_j is not proportional to the parameter minimal working time h_{min}. In our previous work [6] is shown that our ACO algorithm without hybridization outperforms Genetic algorithm and Scatter search from [1]. The stopping criteria is achieving the best found solution for the same test instance from [7], [8]. We apply same parameter settings for all variants of hybridization of ACO algorithm and they are fixed after several experiments.

The process of searching and constructing solutions in solving the workforce planning problem is very complex because of the strict constraints. The aim of the application of local search procedure is as many infeasible solutions of the problem, from the current iteration, become feasible, as well as to reduce the number of infeasible solutions found by the traditional ACO algorithm in the next iterations. This increases the chance that the underlying algorithm will find better solutions, as well as reduces the number of iterations needed to find those solutions. The proposed local search procedures do not spend much computational time because they are applied only over the infeasible solutions. Moreover, there is a sharp reduction in the number of iterations required to find these solutions.

We perform 30 independent runs with every of the test problems, because the algorithm is stochastic and to guarantee the robustness of the average results. We apply ANOVA

TABLE I: Calculation time in seconds

test instance	random remove	many times remove	maximal remove
S1	4.01 s	**3.75 s**	**3.92 s**
S2	19.97 s	**4.48 s**	**4.52 s**
S3	32.96 s	17.57 s	**7.75 s**
S4	37.22 s	46.64 s	**14.50 s**
S5	3.78 s	3.79 s	**2.40 s**
S6	5.12 s	**4.26** s	7.07 s
S7	31.23 s	36.18 s	**10.50 s**
S8	31.35 s	28.98 s	**16.88 s**
S9	**19.17** s	22.28 s	21.56 s
S10	10.19 s	15.78 s	**4.23 s**
U1	**5.25 s**	13.296 s	7.98 s
U2	2.07 s	**1.76 s**	4.15 s
U3	**4.88 s**	**4.86 s**	7.89 s
U4	3.11 s	**2.53 s**	18.84 s
U5	7.98 s	**3.22 s**	4.53 s
U6	**6.74 s**	11.22 s	14.24 s
U7	20.30 s	22.29 s	**11.11 s**
U8	4.17 s	4.12 s	**3.48 s**
U9	18.68 s	**12.98 s**	17.37 s
U10	**5.64 s**	6.224 s	9.95 s

test for statistical analysis to guarantee the significance of the difference between the average results. We compare the calculation time to find the best solution for every of the 20 tests.

Table I shows the needed calculation time to find best solution. The first column is the name of the test. The second column shows the needed time to find best solution, when we remove from infeasible solutions randomly chosen half of the workers, no matter if the new solution is feasible. The third column shows the needed time to find best solution, when we apply random remove of the half of the workers till the solution become feasible. The fourth column shows the needed time to find best solution when we remove from infeasible solution half for the workers, which are most expensive, no matter if the new solution is feasible. With the bold is shortest time to find best solution. Comparing structured problems, we observe that the hybrid ACO algorithm with local search procedure removing half for the workers, which are most expensive, needs less time to achieve best solution, eight of the ten cases. Regarding unstructured problems result is different. Hybrid ACO algorithm with local search procedure removing half fo the workers in a random way and applied one time, achieves best solution for a least time four times, when the local search procedure is applied many times till achieving feasible solution the least time is five times and when the local search procedure removes the most expensive workers, algorithm achieves the least time only two times. We observe big difference in hybrid algorithms performance when they are applied on structured and on unstructured problems. We can confirm that for structured problems is better to apply hybrid ACO algorithm with local search removing most expensive workers and the local search procedure can be applied only ones, no matter if the new solution is feasible. For unstructured problems it seems better to apply many times local search procedure till the solution becomes feasible. The difference comes from the fact that in unstructured problems it is more difficult to reach feasible solutions.

VI. Conclusion

In this paper we apply hybrid ACO algorithms to solve workforce planning problem. The traditional ACO algorithm is combined with several local search procedures. The local search procedures remove half of the assigned workers. Two of the procedures chose the removed workers in a random way and the third removes the most expensive workers and try to assign more cheapest. All local search procedures are applied only on infeasible solutions. The proposed hybrid algorithms are tested on 10 structured and 10 unstructured test instances. We observe that for structured instances, best performance has the local search procedure, which removes most expensive workers.

Acknowledgment

The work is supported by National Scientific Fund of Bulgaria under the grant DFNI KP-06-N52/5. and the rant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program and co-financed by the European Union through the European structural and Investment funds, and by the Polish-Bulgarian collaborative grant “Practical aspects for scientific computing”.

References

[1] Alba E., Luque G., Luna F., *Parallel Metaheuristics for Workforce Planning*, J. Mathematical Modelling and Algorithms, Vol. 6(3), Springer, 2007, 509-528.

[2] Bonabeau E., Dorigo M. and Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York,Oxford University Press, 1999.

[3] Dorigo M, Stutzle T., *Ant Colony Optimization*, MIT Press, 2004.

[4] Easton F., *Service completion estimates for cross-trained workforce schedules under uncertain attendance and demand*, Production and Operational Management 23(4), 2014, 660–675.

[5] Fidanova S., Roeva O., Paprzycki M., Gepner P., *InterCriteria Analysis of ACO Start Startegies*, Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, 2016, 547-550.

[6] Fidanova S., Luquq G., Roeva O., Paprzycki M., Gepner P., *Ant Colony Optimization Algorithm for Workforce Planning*, FedCSIS’2017, IEEE Xplorer, IEEE catalog number CFP1585N-ART, 2017, 415-419.

[7] Roeva O., Fidanova S., Luque G., Paprzycki M., Gepner P., *Hybrid Ant Colony Optimization Algorithm for Workforce Planning*, Annals of Computer Science and Information Systems, Vol. 15, 2018, pp. 233-236. ISSN: 2300-5963, DOI: http://dx.doi.org/10.15439/2018F47.

[8] Fidanova S., Luque G., New Local Search Procedure for Workforce Planning Problem, CYBERNETICS AND INFORMATION TECHNOLOGIES, Vol. 206, 2020, 40-48, DOI: 10.2478/cait-2020-0059

[9] Glover F., Kochenberger G., Laguna M., Wubbena, T. *Selection and assignment of a skilled workforce to meet job requirements in a fixed planning period.* In:MAEB’04, 2004, 636–641.

[10] Grzybowska K., Kovács, G., *Sustainable Supply Chain - Supporting Tools*, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Vol. 2, 2014, 1321–1329.

[11] Soukour A., Devendeville L., Lucet C., Moukrim A., *A Memetic algorithm for staff scheduling problem in airport security service*, Expert Systems with Applications, Vol. 40(18), 2013, 7504–7512.

DOI: 10.15439/2023F2375
ISSN 2300-5963 ACSIS, Vol. 35

Future and Backward Exploration of XR Environments

Jakub Flotyński, Paweł Sobociński, Michał Śliwicki, Mikołaj Maik
Department of Information Technology, Poznań University of Economics and Business
Email: [jakub.flotynski, pawel.sobocinski, michal.sliwicki, mikolaj.maik]@ue.poznan.pl

***Abstract*—XR environments are successfully used in various domains, such as medicine, education, training, and industry. Such environments contain domain knowledge expressed through content and users' behavior. However, current approaches to XR creation lack possibility of exploration of the knowledge included in the environments and expressed by the users' behavior. In this paper, we propose a method of knowledge exploration in XR environments, enabling the analysis of past and potential behavior of users and objects, with queries and automated reasoning. This solution aims to enhance knowledge dissemination using XR.**

I. Introduction

XR SYSTEMS' rising popularity stems from their potential across domains and their ability to offer immersive experiences. However, existing systems often overlook event and interaction analysis within virtual environments, leading to a loss of valuable information. To enhance user experience, there's a crucial need to focus on exploring and analyzing XR system data for hidden insights.

Actions and interactions within XR environments can be explored through semantic queries and automated reasoning, particularly beneficial in employee training systems. This exploration offers valuable insights into users' behavior across different timeframes.

The knowledge gained from exploration is vital for monitoring, analyzing, and controlling XR environments, as well as understanding users' skills, experiences, interests, and preferences. Domain-specific terminology helps specialists, the primary users of XR environments, take full advantage of behavior exploration.

This paper introduces a new method for exploring XR environments, encompassing both forward and backward exploration through semantic queries on the XR environment's knowledge-based representation. The proposed methods were applied to represent the behavior of the virtual environment in an industrial XR training system developed for Amica S.A, Poland's leading house appliances manufacturer, to train employees on specialized industrial devices.

The paper's structure is as follows: Section II presents an overview of the current state of exploring virtual environments. Then, Section III explains the methods used to describe virtual content. Section IV elaborates on exploration methods with examples and an overview of exploration for the XR training system. Finally, Section V concludes the paper and suggests potential areas for future research.

II. Related work

Numerous approaches model 3D content behavior using ontologies and semantic web standards. One notable approach, as discussed in Pellens et al. [1], [2], [3], introduces temporal operators for expressing both primitive and complex behaviors. They also offer a graphical tool to model complex behavior using diagrams, encoding it within X3D scenes [4].

In another approach, De Troyer et al. [5] combine primitive actions like move, turn, and rotate to represent complex behavior in a user-friendly manner. This approach enables end users to specify complex behavior without extensive knowledge of 3D graphics and animation.

Krieg-Brückner et al. [6] introduce a tool using semantic concepts, services, and hybrid automata to describe 3D content behavior. The tool consists of a client component based on a 3D content presentation tool (e.g., XML3D browser) and a server component with various services for content selection and configuration. Additionally, an extra module manages intelligent avatars and their perception of the scene.

Chmiel et al. [7] proposed XSD-based semantic metadata schemes for 3D object interactivity, specifying events, conditions, and actions. CL ontologies [8], [9] represent multi-user virtual environments and avatars, defining geometry, space, animation, and behavior of 3D content. They include semantic counterparts to widely used formats like VRML and X3D. Environmental objects are described by attributes like translation, rotation, and scale, while avatars have names, statuses, user interfaces (UIs), and behavior defined through code bases

In recent research, there has been a notable focus on humans and their interactions within virtual reality, particularly in creating ontologies for experimental purposes. One such recent work is the Virtual Human-Building Interaction Experimentation Ontology (VHBIEO) proposed by Chokwitthaya et al. [10]. The primary objective of this solution is to establish a standardized approach for conducting experiments related to human-building interactions within virtual reality environments. In a similar vein, Heitmayer et al. [11] have introduced another ontology focusing on the human-centred analysis and design of virtual reality conferencing. The main goals of this ontology include enhancing user experience, facilitating research on VR conferencing (particularly in the realms of psychology and behavior), and enabling the sharing of research findings among the scientific community.

III. Representation of XR Environment

In order for the XR environment to be appropriate for exploration, it needs a proper representation of interaction and 3D content. This can be created using knowledge representation technologies such as the semantic web (the Resource Description Framework—RDF [12], the RDF Schema—RDFS [13], the Web Ontology Language—OWL [14] and the SPARQL query language [15]) and ontologies.

For the presented methods of exploration, we prepared behavioral semantic models for the representation of activities, called Activity Ontology, as well as the representation of the workflow.

A. Representation of Activities and Features

Activities and features act as domain-specific states that connect the behavior model and the implementations of XR components. Within the Activity Ontology, the Semantic Web approach defines the representation of these activities and properties. This ontology comprises a TBox and an RBox that contain axioms outlining how users' and objects' features and activities are implemented within XR components and environments. The activity ontology is well suited for both procedural and object-oriented XR implementations.

Fig. 1 shows how class in the XR system is mapped to ontology to activity ontology.

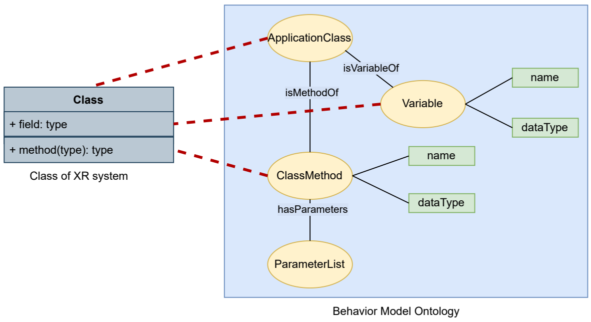


Fig. 1. Mapping XR system class to Activity Ontology

Since both procedural and object-oriented XR implementations rely on functions as the fundamental building blocks of workflow, the activity ontology supports structuring both types of implementations.

The ontology specifies the following classes and attributes associated with code elements:

1) ApplicationClass — is the class of all application classes specified in the code of XR components.
2) Variable — is the class of all variables.
3) ParameterList — is the class of all lists of method parameters.
4) ClassMethod — is the class of all methods specified in the code of XR components

The example of a knowledge base generated using activity ontology is presented in listing 1. The example shows described class *Battery*, which has variable *Charge* and method *ConnectRectifier*.

Listing 1. A fragment of knowledge base describing element of XR system

```
ao:Battery rdf:type owl:NamedIndividual ,
                    ao:ApplicationClass .

ao:ConnectRectifier rdf:type owl:NamedIndividual ,
                    ao:ClassMethod ;
           ao:isMethodOf ao:Battery ;
           ao:datatype "bool"^^rdfs:Datatype ;
           ao:name "connectRectifier"^^xsd:string .

ao:Charge rdf:type owl:NamedIndividual ,
                      ao:Variable ;
             ao:isVariableOf ao:Battery ;
             ao:datatype "integer"^^rdfs:Datatype ;
             ao:name "charge"^^xsd:string .
```

B. Workflow Representation

The workflow representation adapts the behavior model to describe states, events, and time related to the execution of class methods. Hence, it enables the specification of XR components' behavior upon their underlying imperative implementation.

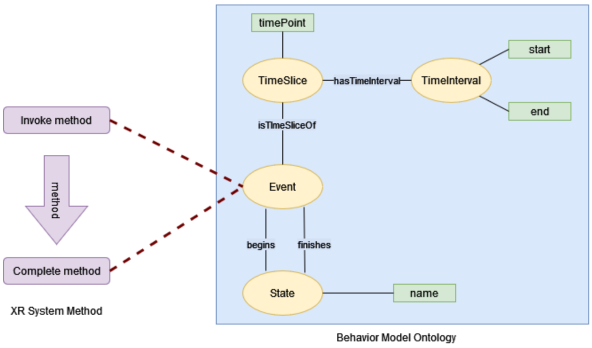


Fig. 2. Mapping the XR system method to fluent ontology

The example of the mapping method of the XR system to behavior model ontology is presented in Fig. 2. An event mapping states that the beginning of an activity is a method invocation or that a method completion is the finish of an activity. Hence, event mappings enclose states of method executions using domain events. Using such mapping, the system can generate a knowledge base, which can be treated as behavioral logs, therefore, can be explored by proper queries. An example of such a knowledge base is presented in Listing 2. The example describes two events that happened while using the system. The first event was ”The visual inspection of the battery”. That event began *BatteryState1* and finished *BatteryState2*. This event had assigned *TimeSlice*, which gives us information about when and how long the event lasted.

Listing 2. A fragment of knowledge base describing event

```
fo:Event1 rdf:type owl:NamedIndividual ,
                   fo:Event ;
          fo:begins fo:BatteryState1 ;
          fo:finishes fo:BatteryState2 ;
          fo:name "Visual␣inspection␣of␣battery"^^xsd:
               string .

fo:TimeSlice1 rdf:type owl:NamedIndividual ,
                       fo:TimeSlice ;
              fo:hasTimeInterval fo:TimeInterval1 ;
              fo:isTimeSliceOf fo:Event2 .
```

```
fo:TimeInterval1 rdf:type owl:NamedIndividual ,
                          fo:TimeInterval ;
                 fo:end "2023-05-20T10:03:12"^^xsd:dateTime
                     ;
                 fo:start "2023-05-20T10:04:22"^^xsd:
                     dateTime .
```

IV. Exploration

In this section, we propose different types of knowledge exploration and visualization within explorable XR environments, utilizing the models proposed in the previous section. We classify these exploration types based on the target periods, distinguishing between simulation with forward and backward exploration.

Simulation and forward exploration facilitate the process of reasoning and querying potential events and states in the XR environment. These events and states may remain undetermined, contingent upon the occurrence or absence of other events and states within the environment. Engaging in simulation and forward exploration necessitates a composed XR environment, which does not necessarily need to be compiled and run. Consequently, simulation and forward exploration can be initiated promptly after composing the environment.

Backward exploration enables the process of reasoning, querying, and visualizing past and present events and states that have been logged in behavior records. To engage in backward exploration, the XR environment must be executed, and behavior logs need to be generated.

Queries play a crucial role in acquiring knowledge about explorable XR environments. Nevertheless, despite their possible support in query languages such as SPARQL, our focus does not revolve around query result presentation operations like result limitation, sorting, or data aggregation. Instead, our primary emphasis lies in utilizing queries to gain deep insights into the properties and behaviors of the XR environment, facilitating exploration and understanding.

By applying these techniques for knowledge exploration and visualization, we elevate our comprehension of the dynamic nature of XR environments and empower users to interact effectively, reason, and query within these environments. These approaches serve as invaluable tools for researchers and developers, enabling them to delve into the potential of explorable XR environments and propel innovation in this field.

We used new exploration methods to query the behavior of users and 3D objects inside a virtual environment for the industrial worker training XR system, which allows trainees to learn how to act safely in an industrial setting. The training scenario implemented in the system focuses on safe work with a forklift. It was developed using resources from Amica S.A., a major producer of household equipment in Poland.

A. Simulation with forward exploration

Forward exploration facilitates reasoning and queries about the potential behaviors of users and objects within explorable XR environments. It encompasses various states and events associated with features and activities, encompassing autonomous actions and interactions among users, objects, and their interplay. Crucially, forward exploration is intimately intertwined with the simulation of environmental behavior, which aims to fulfil conditions necessary for events and states. As a result, simulation precedes forward exploration, encompassing both aspects within simulation queries. Importantly, since simulation and forward exploration revolve around potential events and states, they necessitate a workflow specification but do not mandate that the XR environment be actively running.

Simulation queries to an environment specification enable forward exploration of the environment without running it. Therefore, they must specify the conditions for which the exploration is accomplished. The illustrative simulation queries presented in this section assume that the delay between an event and another following event is equal to 0.001, and no exceptions are thrown during the execution of methods:

$Delay$ =0.001 $\bigwedge exception(executed(Method, ExecutionID), null)$.

To allow forward exploration, the following elements of the XR system were mapped to semantic representation using Activity Ontology:

Classes:

1) Trainee (class representing trainee),
2) Forklift (class representing forklift),
3) Battery (class representing battery),
4) Rectifier (class representing rectifier).

Methods:

1) startForkliftInspection (method of Trainee class),
2) finishForkliftInspection (method of Trainee class),
3) chargeBattery(method of Rectifier class),
4) insertBattery(method of Forklift class),
5) showChargingState(method of Rectifier class),
6) plugIn(method of Rectifier class)

Additional predicates were also used, allowing for the representation of temporal entities:

time(event,tp) — predicate that is true for a given event and a time point if and only if the event occurs at the time point.

holds(event,ti) — predicate that is true for a given event and a time interval if and only if the fluent is true within the time interval

Using such prepared knowledge representation, we are able to create the following queries:

1. How long will it take to check the visual state of a forklift?

$time(startForkliftInspection(Trainee, Forklift)), TP_{start}) \bigwedge time(finishForkLiftInspection, TP_{end}) \bigwedge Lenght = TP_{end} - TP_{start}$

It determines the time points of starting and finishing a forklift inspection and calculates the inspection length. (The Fig. 3 presents how a user inspects the forklift in XR system.) The query result is the following:

$TP_{start} = 10,\ TP_{end} = 22,\ Lenght = 12$

2. What will happen after a trainee finishes charging the battery?

$holds(chargeBattery(Trainee, Battery), TI_1) \bigwedge holds(Action, TI_2) \bigwedge after(T_1, T_2)$

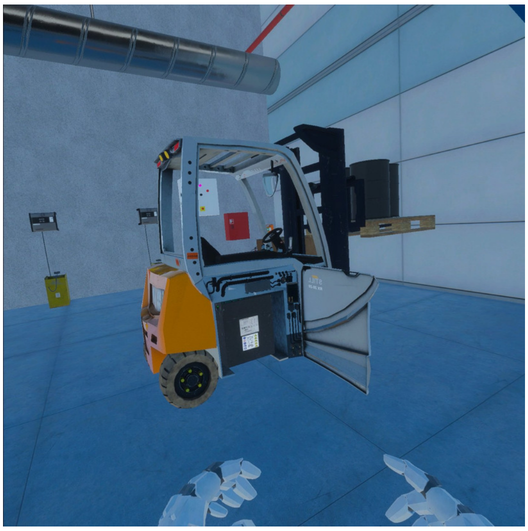


Fig. 3. A user inspects the forklift

It searches for the charging battery event and its time interval and time interval and action that happened later by using the after predicate, which compares time intervals. The query result is the following:

$Action = insertBattery(Trainee, Battery))$

3. What are the possible states (color of light) of rectifier after plugging in the battery?

$holds(showChargingState(lightColor), T_1) \bigwedge holds(plugIn(Battery), TI2) \bigwedge after(T_1, T_2)$

It searches for the event describing showing the charging state of the rectifier that happened after plugging in the battery. The possible answers are:

$lightColor = red,\ lightColor = yellow,\ lightColor = green$

B. Backward exploration

Backward exploration allows for the analysis and querying of activities that took place during the operation of an explorable XR environment.

Unlike forward exploration, which relies on simulating the environment's behavior, backward exploration uses logged activities. This eliminates the need for environmental behavior simulation. Furthermore, the inclusion of temporal statements with visual descriptors in behavior logs enables the visualization of past activities.

Queries specifically designed for backward exploration are referred to as exploration queries. The output of an exploration query is defined similarly to a simulation query. The behavior logs are structured based on RDF, enabling the utilization of the SPARQL language for conducting backward exploration. The example of behavior logs in the form of knowledge-base was presented in Listing 2.

a) Query 1: Which events had happened before the forklift was turned on?

```
SELECT  ?eventName
WHERE { ?event rdf:type fo:Event  .
        ?timeSlice fo:isTimeSliceOf ?event .
        ?event fo:name ?eventName .
        { ?timeSlice fo:timePoint ?time . }
        UNION{
        ?timeSlice fo:hasTimeInterval ?timeInterval .
        ?timeInterval fo:start ?time . }
        ?PushButtonEvent fo:name "Turning on the forklift
             "^^<http://www.w3.org/2001/XMLSchema#string> .
        ?PushButtonTimeSlice fo:isTimeSliceOf ?
             PushButtonEvent .
        ?PushButtonTimeSlice fo:timePoint ?pushButtonTime .
FILTER (  ?time < ?pushButtonTime)}
ORDER BY ?time
```

The first query provides information about what happened in the scene before the trainee pushed the press button, which activated the forklift. The query searches for events and their assigned time slices that happened before the event with the name "Turning on the forklift". The action of turning on the forklift presents Fig. 4.

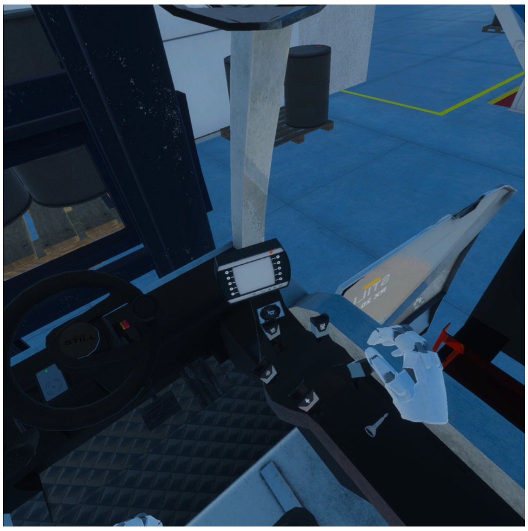


Fig. 4. A user turns on the forklift

b) Query 2: How long the battery was inspected?

```
SELECT ?start ?end
WHERE { ?event fo:name "visually controlling the battery
     "^^<http://www.w3.org/2001/XMLSchema#string> .
        ?timeSlice fo:isTimeSliceOf ?event .
        ?timeSlice fo:hasTimeInterval ?timeInterval .
        ?timeInterval fo:start ?start .
        ?timeInterval fo:end ?end .}
```

The second query gives information about the duration of the visual inspection of the battery. The query searches for the time slice of the event named "visually controlling the battery", then using the time interval object, access the information when the event started and ended. Fig. 5 presents, how user inspects the battery in virtual scene.

c) Query 3: When and What states did the battery transition into?

```
SELECT ?begins  ?stateName ?stateID
WHERE{ ?state rdf:type fo:InstantState .
        ?object fo:hasState ?state .
        ?object fo:name "battery"^^xsd:string .
        ?state fo:name ?stateName .
        ?state fo:id ?stateID .
        ?event fo:begins ?state .
```

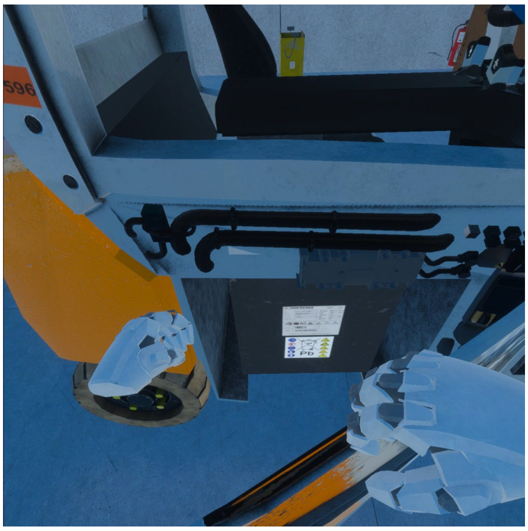


Fig. 5. The user inspects the battery

```
    ?timeSlice fo:isTimeSliceOf ?event .
    ?event fo:name ?eventName .
    { ?timeSlice fo:timePoint ?begins . }
    UNION {
    ?timeSlice fo:hasTimeInterval ?timeInterval.
    ?timeInterval fo:start ?begins . } }
ORDER BY ?begins
```

The last query provides information about what happened sequentially with the battery. The query searches for events that changed the states of the scene object named "battery". Then using proper time slices, the query determines the time by which the states are ordered. The results consist of the name and id of the states and the time when they have begun.

V. Conclusions and Future Work

The use of exploration of behavior and 3D content in XR systems can have multiple applications, like learning about users, their experience, preferences, and interests and measuring their skills. This can be beneficial, for example, in virtual training, where we want to maximize the training results.

In this paper, we have proposed methods for forward and backward exploration based on semantic queries. The presented approach uses the knowledge-based representation of the XR environment, which is represented by described ontologies. Moreover, we provided examples of explorations based on the developed XR system for employee training in an industrial environment.

The possible future research directions could focus on developing available inexperienced user plug-ins for simplifying and automating the process of creating semantic queries and visualization of the results.

Acknowledgment

The presented research was funded by the Polish National Center for Research and Development under grant number LIDER/55/0287/L-12/20/NCBR/2021.

References

[1] B. Pellens, O. De Troyer, W. Bille, and F. Kleinermann, "Conceptual modeling of object behavior in a virtual environment," in *Proceedings of Virtual Concept 2005*. Biarritz, France: Springer-Verlag, 2005, pp. 93–94.

[2] B. Pellens, O. De Troyer, W. Bille, F. Kleinermann, and R. Romero, "An ontology-driven approach for modeling behavior in virtual environments," in *Proceedings of On the Move to Meaningful Internet Systems 2005: Ontology Mining and Engineering and its Use for Virtual Reality (WOMEUVR 2005) Workshop*, R. Meersman, Z. Tari, and P. Herrero, Eds., no. 3762, Springer-Verlag. Agia Napa, Cyprus: Springer-Verlag, 2005, pp. 1215–1224.

[3] B. Pellens, F. Kleinermann, and O. De Troyer, "A development environment using behavior patterns to facilitate building 3d/vr applications," in *Proc. of the 6th Australasian Conf. on Int. Entertainment*, ser. IE '09. ACM, 2009, pp. 8:1–8:8.

[4] B. Pellens, O. De Troyer, and F. Kleinermann, "Codepa: a conceptual design pattern approach to model behavior for x3d worlds," in *Proceedings of the 13th International Symposium on 3D web technology*, Los Angeles, August 09-10, 2008, pp. 91–99.

[5] O. De Troyer, F. Kleinermann, B. Pellens, and W. Bille, "Conceptual modeling for virtual reality," in *Tutorials, posters, panels and industrial contributions at the 26th Int. Conference on Conceptual Modeling - ER 2007*, ser. CRPIT, J. Grundy, S. Hartmann, A. H. F. Laender, L. Maciaszek, and J. F. Roddick, Eds., vol. 83. Auckland, New Zealand: ACS, 2007, pp. 3–18.

[6] P. Kapahnke, P. Liedtke, S. Nesbigall, S. Warwas, and M. Klusch, "ISReal: An Open Platform for Semantic-Based 3D Simulations in the 3D Internet," in *International Semantic Web Conference (2)*, 2010, pp. 161–176.

[7] J. Chmielewski, "Describing interactivity of 3d content." in *Interactive 3D Multimedia Content*, W. Cellary and K. Walczak, Eds. Springer, 2012, pp. 195–221.

[8] Y. Chu and T. Li, "Using pluggable procedures and ontology to realize semantic virtual environments 2.0," in *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ser. VRCAI '08. New York, NY, USA: ACM, 2008, pp. 27:1–27:6.

[9] Y.-L. Chu and T.-Y. Li, "Realizing semantic virtual environments with ontology and pluggable procedures," in *Applications of Virtual Reality*, C. S. Lanyi, Ed. Rijeka: IntechOpen, 2012, ch. 9.

[10] C. Chokwitthaya, Y. Zhu, and W. Lu, "Ontology for experimentation of human-building interactions using virtual reality," *Advanced Engineering Informatics*, vol. 55, p. 101903, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474034623000319

[11] M. Heitmayer, M. G. Russell, S. Lahlou, and R. D. Pea, "An Ontology for Human-Centered Analysis and Design of Virtual Reality Conferencing," in *TMS Proceedings 2021*, nov 3 2021, https://tmb.apaopen.org/pub/3rbumwgw.

[12] W3C. (accessed March 24, 2015) Rdf. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[13] ——. (accessed March 24, 2015) Rdfs. [Online]. Available: http://www.w3.org/TR/2000/CR-rdf-schema-20000327/

[14] ——. (accessed March 24, 2015) Owl. [Online]. Available: http://www.w3.org/2001/sw/wiki/OWL

[15] ——, "Sparql query language for rdf," accessed March 24, 2015 2008. [Online]. Available: http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/

DOI: 10.15439/2023F7106
ISSN 2300-5963 ACSIS, Vol. 35

Improving the Performance of Multiscene Marketing Video Content through its Dynamics Adjustments

Kacper Fornalczyk, Kamil Bortko, Aneta Disterheft, Jarosław Jankowski
0000-0003-3752-3473
0000-0002-3658-3039
West Pomeranian University of Technology
Faculty of Computer Science and Information Technology
Żołnierska 49
71-210 Szczecin, Poland
Email: kacper_fornalczyk@wp.pl, anetadisterheft@gmail.com, {kbortko, jjankowski}@zut.edu.pl

***Abstract*—The use of online video content plays a vital role in marketing strategies and is a significant component of internet usage. The challenge lies in evaluating the impact of video content on user engagement and finding ways to enhance its performance without employing techniques that overwhelm users or prompt ad avoidance behavior. This study investigates the correlation between video dynamics metrics and eye-tracking patterns to determine if user engagement, as indicated by fixations, is influenced by these metrics. The findings demonstrate that dynamic metrics can accurately predict eye-tracking patterns for brief videos and can be applied to measure both inter and intra-scene dynamics in multiscene videos.**

I. Introduction

ONLINE video content is a popular medium that comprises a significant share of internet usage, with its consumption expected to rise to 82% of all internet traffic by 2022, up from 75% in 2017, according to Cisco's 2018 report [6]. Video content is widely used for marketing purposes, in the form of in-stream ads or integrated with editorial content on social platforms, games, or portals[17]. Content creators often use techniques that increase user engagement through emotional content, visual effects, and high dynamics, but these techniques can also increase cognitive load and distract users from their main goals within websites[14]. This decreased user experience may lead to users skipping advertising content, particularly when it fails to catch their attention at first glance. Hence, content producers face the challenge of creating video ads that are less likely to be skipped by consumers, which can be achieved by lowering intrusiveness and the dynamics of the video content. This paper investigates how the dynamics of video, as represented by dedicated metrics, relate to eye-tracking patterns, and whether the dynamics of video can predict user engagement, as represented by fixations. The secondary goal is to examine the impact of intra-scene differences on user attention within multi-scene videos. The primary aim is to explore methods of building videos with low dynamics and low cognitive load while maintaining enough differences between them to sustain user attention.

II. Literature review

Although there is already some research on television advertising, there is still an opportunity to delve deeper into the area of online video ads, as this format is more captivating and hence more widely used than static display or text ads [19]. Advertisers and advertising area providers need to know when video ads start to become too distracting, which can result in the use of ad blockers. This situation underscores the need to search for factors that affect video ad performance [20], especially those that can be used to attract users' attention and keep them engaged [5] [11]. High cognitive load may result from different ad characteristics, such as high video dynamics or the use of intense colors or sounds that are perceived as disturbing[15]. The following research focuses on measuring video ad dynamics as a factor affecting performance. An algorithm that automatically extracts several features, including video-level visual variance, scene-to-scene visual variance, and average scene cut frequency, was developed and published in [13] and further elaborated in [19] [2]. As the algorithm is publicly available, it was used in the following study.

One concept that plays a significant role in this notion is the level of involvement shown by consumers. Numerous writers have consistently reported that viewers who are more engaged in video content are less likely to skip it [10][21]. Other studies in this field have highlighted the correlation between ad avoidance and cognitive factors that are closely linked to engagement. For instance, in [1] [4], it was discovered that avoiding ad content may be connected to high and low arousal levels elicited by the content. A comparable finding was outlined in [18], which revealed that unstimulating, uninteresting content increases the desire to skip ads[9]. The primary factor influencing consumers' tendency to avoid ads is the engagement level of the ad content. Additionally, the length of the video ad is another critical factor that affects ad avoidance [18]. Numerous studies have discussed the relationship between ad length and the rate of ad avoidance, identifying a correlation between increased skipping behaviour and longer content [18] [12]. Other research has suggested that longer ads result in more disruption for goal-oriented search [9][3]. Recent studies have revealed that consumers' acceptance of longer videos has decreased, and nowadays, most people only accept very short video ads, such as fifteen or six seconds in length [18]. As a result, the current trend is to produce short video marketing content that better targets consumers' attention spans[7][8]. This trend is advantageous for ad providers because they can increase the rate of presenting content to consumers without increasing the total costs of a campaign. However, some studies have shown that longer ads may be more effective in enhancing brand recognition [12]. Ads that are shortened to fifteen seconds can achieve similar results in terms of awareness and brand recall as thirty-second spots[16].

In this current study, various factors were examined to evaluate the performance of videos. We aimed to investigate how the video's dynamics and differences between scenes can affect eye-tracking patterns and how user engagement relates to video dynamics, both at the level of single videos and intra-scene differences.

Conceptual framework

The conceptual drawing Figure 1 shows the stage 1 in which you can see the prepared video base divided into the length of the films and their dynamics. After preparation of the base, the test was carried out in laboratory conditions, with the intention of obtaining an increased number of fixations along with increasing dynamics.

Stage 2 presents the users' gaze pattern thanks to the eyetracker examination of interstage dynamics. You can see here an increase in dynamics in relation to the increasing number of fixations for short scenes versus long scenes, where such a relationship does not exist.

III. Algorithm description

To study video dynamics we implemented algorithm parented in Xi Li, Mengze Shi and Xin (Shane) Wang [13] and we extended it towards measuring inter and intra-scene dynamics. The input of the algorithm apart from file is the list of numbers being the numbers of consecutive frames that constitute the beginning of each new scene. Technically, a scene is a group shots which are successively taken together at a single location. A shot is a basic narrative element of the video which is composed of a number of frames that are presented from a continuous viewpoint. Automatically dividing a video into its shots is called the shot boundary detection problem in which the basic idea is identifying consecutive frames that form a transition from one shot to another. Currently, there are more or less effective solutions to this problem that can be used to obtain the above-mentioned list.

Additionally, one of the two possible parameters should be specified at the input of the algorithm. One of them takes numerical values ranging from [1, 100], and this value determines what percentage of all frames from each scene should be included in the calculation. For example, if the scene has 200 frames and the parameter value is 20, then 0.2 * 200 = 40 frames, possibly equally spaced from each other, will be extracted from the scene. The second parameter takes values greater than 0 and is an alternative to the previously described parameter. Its value determines the length of the time interval, which is the frequency with which the frame for the analysis will be extracted from the scene. If the scene is 10 seconds long and the parameter is set to 0.5 seconds, then 10 / 0.5 = 20 frames will be set in the scene.

In the first step, the algorithm loads the movie and run through its frames. Each frame is stored in the algorithm's memory as a three-dimensional matrix with dimensions equal to the resolution of the video. Each of the three layers of this three-dimensional matrix contains values that define one of the components of the RBG space for each pixel in the frame. Knowing the numbers of the first frames of all detected scenes, algorithm extracts the appropriate number of frames from each scene with a given interval or percentage, creating a new list for each scene containing the frames extracted for it.

The very measurement of dynamics of a video message is based on the measure provided by Xi Li, Mengze Shi and Xin (Shane) Wang [13]. In their work, the authors present the measure they named "visual variation" which is a normalized measure of the changes in visual information in a video. Determining the visual variation for two frames is carried out in the following few steps.

First, the frames are reduced from RGB to grayscale by averaging the color components of each pixel. The next step is to normalize the values from the range [0, 255] to the range [0, 1]. Authors of the measure mention that normalization serves the purpose of compensation for possible expousure difference. Further, the distance between the individual pixels of the two frames is calculated, where the distance is defined as the Manhattan norm

$$d(x'_i, y'_i) = |x'_i - y'_i| \tag{1}$$

After calculating the matrix with dimensions equal to the resolution of the compared frames, where each position in the matrix is the distance between individual pixels at the same position, the algorithm proceeds to the last step. Here our implementation of the measure differs from the one proposed by its authors. In the original implementation of this measure, in this step of the algorithm, all the determined absolute distances between each pair of pixels should be summed up. In our version, we chose to calculate the mean over the absolute distance of every pair of pixels. This change was aimed at obtaining the visual variation result in the range [0, 1]. This averaged value represents the size of visual variation between two frames.

In the next step determination of the internal dynamics of scenes takes place. This step consists in determining for each list containing extracted frames from individual scenes the average visual variation occurring between consecutive frames

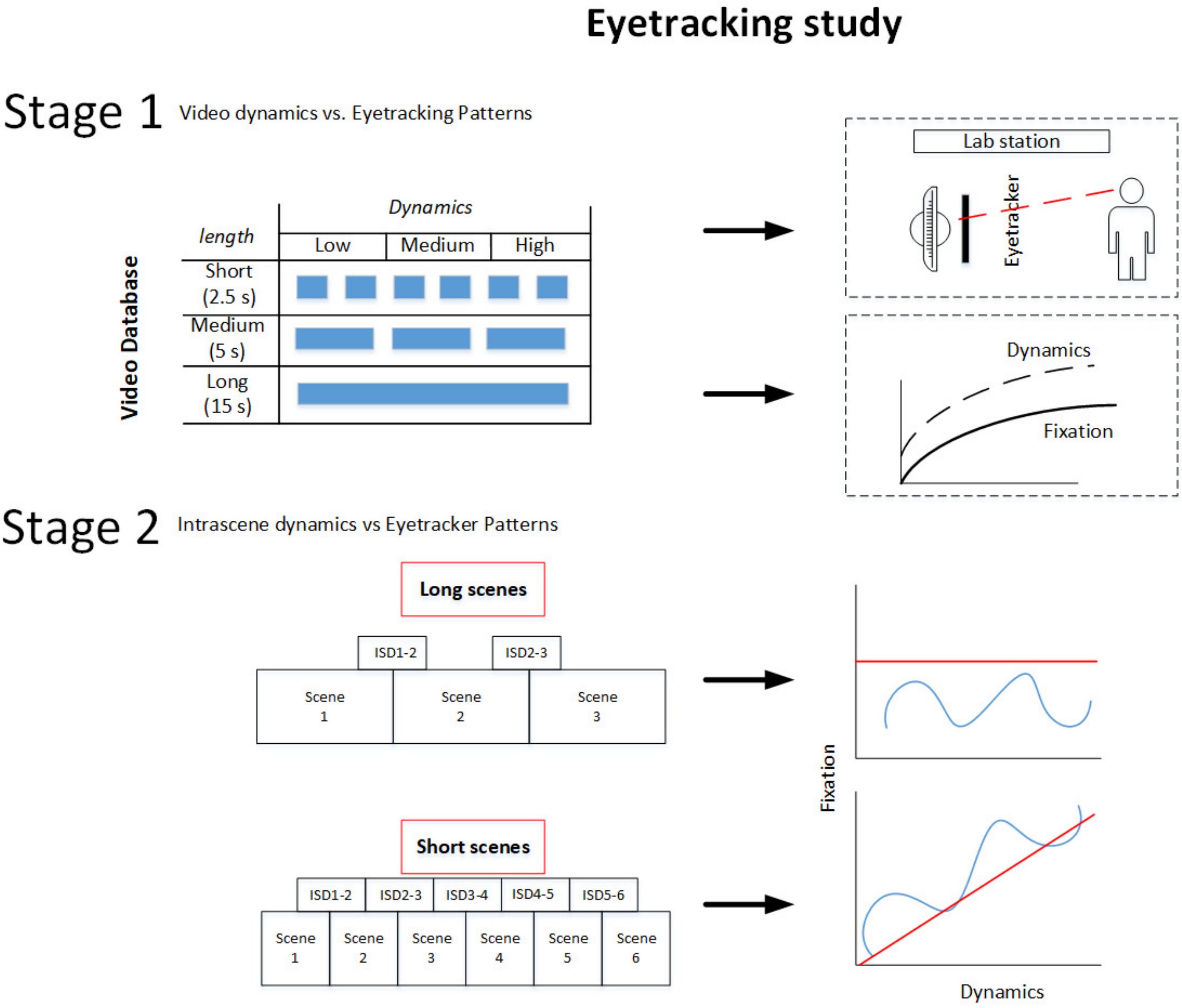


Fig. 1. Conceptual framework for study

in the list. For example, if only 3 frames have been extracted from the scene, then we calculate the value of visual variation between frames no. 1 and no. 2, and between frames no. 2 and no. 3. Then both values are averaged. The process could be written as follows

$$\mathrm{VVL} = \frac{1}{n}\sum_{i=1}^{n} d(F_i, F_{i+1}) \tag{2}$$

The equation of Visual Variation Level (VVL) where $\boldsymbol{n}$ is the number of frames, $\boldsymbol{F_i}$ is the frame number in the list and $\boldsymbol{d}$ is a function that returns the visual variation between two frames.

A. Determining the external dynamics between the scenes

External dynamics is the working name for the visual variation determined between successive scenes. The calculation of this measure takes place when the algorithm extracts appropriate frames from individual scenes into new lists. The algorithm knows the order of the lists, which corresponds to the order in which the scenes appear in the entire video transmission. Thanks to this, it can determine the visual variation between individual scenes in the same way as it was presented for a series of frames extracted from one scene.

The first step in determining external dynamics is to average the colors of all frames within each list. The averaged values should be rounded off as the values of the three-dimensional matrix should be integers in the range [0, 255]. This process can be represented by the following formula:

$$Af = [\frac{1}{n}\sum_{i=1}^{n} F_i] \tag{3}$$

Where $\boldsymbol{F_i}$ - a frame in the form of a three-dimensional matrix, $\boldsymbol{n}$ - the number of frames in a leaf, $\boldsymbol{Af}$ - the resulting averaged three-dimensional matrix.

The above step is performed for each list corresponding to a single scene. This way the algorithm comes to a point where it has an averaged three-dimensional matrix / frame for each scene. Now these averaged frames / matrices can be treated as ordinary frames for which the visual variation within one scene was calculated in the previous section. Here, however, it is done each time only between two averaged frames corresponding to two consecutive scenes (for example, the algorithm calculates the visual variation between the averaged frame from the first scene - s1 - and from the second scene - s2, then between s2 and s3, then s3 a s4… sn-1 a sn). Here, as before, each of the averaged frames is previously reduced to grayscale, normalized and the distances between individual pixels at corresponding positions are calculated, and finally the average is determined from these distances.

The algorithm returns the determined internal dynamics for each scene and external dynamics between successive scenes, as well as additional information about the total number of frames in the examined video, the length of the recording, the number of frames per second, the number of scenes, and the values of the set parameters.

Experiment and results

The experiment involved 30 people involved. The research group consists of 14 women and 16 men aged 20 to 40 years. The experiment used a 27-inch Dell monitor and the Tobii Pro X3 eyetracker with a sampling frequency of 120 Hz. A special stand with a tripod was prepared for the experiment, which made it possible to keep the head of each participant in a stationary position. The participant sat in front of the monitor at a distance of about 54 cm. Calibration was performed before each test.

The whole experiment was as follows: each person performed the task of clicking on points in accordance with the concept of Fitts' law. This task was a kind of a break between the screening of individual films prepared by us. These films were divided into 1, 3 or 6 scenes. Each of the films lasted 15 seconds. The important thing is the variety of internal and external dynamics of films. The internal dynamics have been divided into three levels: low, mid and high. The dynamics measures were determined thanks to the above-described algorithm. So the films were prepared in such a way that, depending on the number of scenes, each of them had the same dynamics.

The films have been divided into 21 combinations, as shown in the Table I The films of 3 and 6 scenes had dynamics measures individually for each of them. The division with respect to the internal dynamics was made additionally to the external dynamics between the scenes, hence so many combinations. Here we see diversity in terms of individual films and scenes. At first glance, the measures and the average number of fixations increase in line with the increase in the dynamics measure for individual scenes in specific movies. Movies and scenes with relatively the highest internal and external dynamics have the best average number of fixations. Here it should be mentioned that external dynamics can only be made for movies that have been divided into a plural number of scenes, ie more than one. Therefore, movies with one default scene were not taken into account in determining the external dynamics measures. As you can see in the Table I, the measure columns for external dynamics are defined for transitions between specific scenes. This allowed to define the dynamics just between them. Figure 2 (A) shows the influence of one intergroup factor, which is the number of fixations per scene, on the dependent variable which is the appropriate measure of dynamics. We can notice a clear upward trend in the number of fixations in relation to the increase in the internal dynamics of films. The number of fixations increases, respectively, from the average to the value of 4.4 for the low dynamics, 4.5 for the mid dynamics, up to the level of 5.0. Figure 2 (B) It shows the total dynamics of scenes for

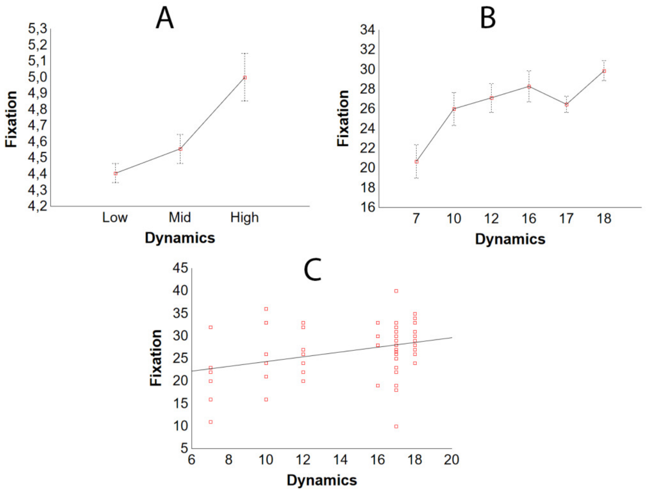


Fig. 2. Fig. A shows a curve prepared with the use of ANOVA statistics showing the trend of the increase in dynamics against the number of fixations for individual scenes. Fig. B shows the total dynamics of scenes showing the increase in dynamics in relation to the number of fixations for entire movies with 6 scenes. Fig. C shows a scatterplot for movie scenes with 6 scenes.

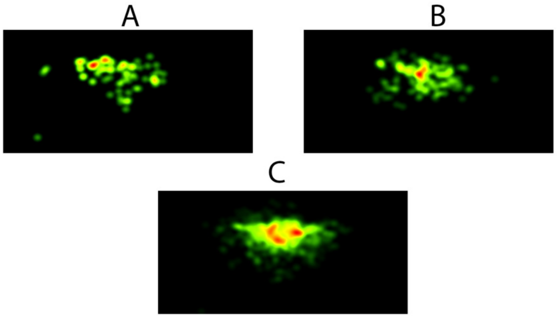


Fig. 3. Heatmaps with visible differences between the various dynamics of the film. Heatmaps A, B, C show three dynamics of movies, respectively: low, mid and high.

TABLE I
MEASURES OF DYNAMICS OF EACH SCENE OF INDIVIDUAL FILMS IN EXTERNAL AND INTERNAL DYNAMICS.

	internal dynamics						external dynamics				
film	s1	s2	s3	s4	s5	s6	s1-s2	s2-s3	s3-s4	s4-s5	s5-s6
1_l	0,031										
1_m	0,096										
1_h	0,155										
3_l_l	0,003	0,005	0,005				0,023	0,018			
3_l_m	0,050	0,048	0,057				0,106	0,130			
3_l_h	0,004	0,002	0,006				0,223	0,208			
3_m_l	0,061	0,067	0,060				0,068	0,121			
3_m_m	0,099	0,085	0,064				0,115	0,103			
3_m_h	0,107	0,059	0,088				0,175	0,412			
3_h_l	0,116	0,110	0,119				0,089	0,077			
3_h_m	0,117	0,151	0,136				0,102	0,107			
3_h_h	0,164	0,150	0,147				0,192	0,290			
6_l_l	0,016	0,008	0,017	0,014	0,009	0,021	0,049	0,047	0,039	0,040	0,052
6_l_m	0,046	0,054	0,069	0,037	0,063	0,033	0,174	0,146	0,169	0,131	0,138
6_l_h	0,027	0,013	0,046	0,003	0,028	0,005	0,234	0,233	0,223	0,230	0,230
6_m_l	0,088	0,060	0,078	0,076	0,091	0,078	0,093	0,100	0,091	0,067	0,078
6_m_m	0,074	0,074	0,083	0,079	0,078	0,068	0,163	0,184	0,185	0,137	0,106
6_m_h	0,076	0,077	0,059	0,092	0,079	0,039	0,391	0,376	0,333	0,374	0,189
6_h_l	0,112	0,162	0,130	0,095	0,063	0,089	0,124	0,080	0,079	0,095	0,122
6_h_m	0,105	0,096	0,159	0,144	0,147	0,119	0,171	0,155	0,100	0,091	0,101
6_h_h	0,117	0,110	0,143	0,105	0,114	0,110	0,172	0,171	0,170	0,223	0,184

TABLE II
MEASURES THE AVERAGE NUMBER OF FIXATIONS ON THE SCENE OF A GIVEN MOVIE.

	fixations per scene					
film	s1	s2	s3	s4	s5	s6
1_l	4,278					
1_m	4,313					
1_h	3,591					
3_l_l	3,286	3,333	3,286			
3_l_m	3,000	3,167	3,000			
3_l_h	3,148	3,111	3,222			
3_m_l	2,714	2,571	3,095			
3_m_m	2,556	3,333	3,000			
3_m_h	2,267	2,800	2,867			
3_h_l	2,833	2,667	2,708			
3_h_m	2,833	3,222	3,778			
3_h_h	2,909	3,152	3,152			
6_l_l	4,667	5,167	5,000	5,667	5,500	5,167
6_l_m	4,000	4,429	4,286	5,000	4,714	5,143
6_l_h	4,333	4,167	3,833	4,333	4,667	4,667
6_m_l	3,500	4,375	3,625	4,125	4,375	4,625
6_m_m	3,111	4,444	4,444	4,333	4,000	5,000
6_m_h	3,571	3,857	4,714	4,571	4,571	4,286
6_h_l	3,900	4,400	4,700	4,300	4,900	5,000
6_h_m	3,800	4,700	4,600	5,300	5,100	5,100
6_h_h	4,143	4,857	4,857	4,429	4,857	4,714

individual films. Here we see an increase in the number of fixations in line with the increase in dynamics in the movies. The trend is clearly increasing, i.e. the number of fixations increases proportionally to the total dynamics. We can see here the ranges for the weakest dynamics on the 7th level of the total number of almost 21 fixations, up to the range of 26 to 30 for the rest of the dynamics. Figure 2 (C) scatter regression plot shows the data in particular dynamics. The presented trend is clearly increasing. Figure 3 shows heatmaps with different dynamics variants. Heatmap A shows the dynamics of Low, the heatmap B shows the dynamics of mid, and the heatmap C the dynamics of high. You can see dense clusters for the high dynamics compared to the other two dynamics.

In this case, the analysis was based on the analysis of internal dynamics for individual films with high, medium and low dynamics. Also in this case, we see the differences between the individual films in each group of internal dynamics broken down by external dynamics.

Using Anova analysis, we see the significance for each dynamics in relation to the number of fixations for individual videos. The significance at the level of $p = 0.017$ indicates a significant dependence of the dynamics measures in relation to each film with the appropriate amount of fixations.

In order to standardize the measures, a division into three groups of dynamics was made using the cluster analysis, which made it possible to reliably and efficiently organize and standardize the groups of individual measures.

Mann - Whitney U statistical analysis, we can see that the intergroup comparison shows a statistical significance below $p < 0.05$ in a few cases. Performing an intergroup comparison here for 6 scene films only showed the significance of w between each group of dynamics. You can see strong differences between the low, mid and high dynamics.

Anova's analysis showed that the summary analysis of scenes for each type of movie shows a significance of $p < 0.05$, which is $p = 0.044$. This shows the strong influence of the amount of fixation on a given scene in the movie.

IV. CONCLUSIONS

Effective video content requires the integration of various elements that increase user engagement, such as emotional appeal, dynamic visuals, and attention-catching techniques. However, the extensive use of video content for marketing purposes has led to avoidance behaviors, such as video ad skipping or blocking. The acceptable length of videos for users has reduced to just a few seconds. Therefore, it is crucial to develop methods that allow the creation of effective content without sacrificing user experience. In our proposed approach, we demonstrated how metrics of video dynamics are correlated with eye-tracking patterns and can be used to create video content using scenes with different dynamics. By using a modified algorithm that determines the dynamics of individual films and scenes, we correlated these dynamics with the number of fixations while watching them. Our experiment's results showed that dynamic metrics as a predictor of eye-tracking patterns are effective for short videos and can be used for multi-scene films to measure dynamics between and within scenes. The statistics clearly showed an increase in the number of fixations in relation to the increase in dynamics, indicating a directly proportional relationship.

Moving forward, to address the complex challenge of combining these factors in a hybrid approach, in future we propose a framework that integrates both qualitative scene analysis and quantitative visual intensity measurements.

The hybrid approach will first involve a comprehensive scene analysis, where various elements such as objects, shapes, colors, and spatial relationships will be identified and categorized. This qualitative understanding of the scene will provide valuable context for the subsequent analysis. Nonetheless, we believe that this hybrid approach has the potential to enhance our understanding of visual perception and contribute to various fields, such as computer vision, human-computer interaction, and visual design. Through continued research and refinement, the proposed framework could open new avenues for investigating human visual perception and its applications.

Based on these findings, we recommend that advertisers create video content from short films to maximize user absorption. In the future, research will focus on identifying not only the characteristics of scenes based on color differences but also the characteristics of objects within scenes. This will allow for the evaluation of differences based on scene elements, not just visual intensities.

V. Acknowledgements

This work was supported by the National Science Centre of Poland, the decision no. 2017/27/B/HS4/01216 and by ZUT Highfliers School /Szkoła Orłów ZUT/ project coordinated by Dr. Piotr Sulikowski, within the framework of the program of the Minister of Education and Science /Grant No. MNiSW/2019/391/DIR/KH, POWR.03.01.00-00-P015/18/, co-financed by the European Social Fund, the amount of financing PLN 1,704,201,66.

References

[1] Belanche, D., Flavián, C., Pérez-Rueda, A.: Understanding interactive online advertising: Congruence and product involvement in highly and lowly arousing, skippable video ads. Journal of Interactive Marketing **37**, 75–88 (2017)

[2] Belanche, D., Flavián, C., Pérez-Rueda, A.: User adaptation to interactive advertising formats: The effect of previous exposure, habit and time urgency on ad skipping behaviors. Telematics and Informatics **34**(7), 961–972 (2017)

[3] Campbell, C., Mattison Thompson, F., Grimm, P.E., Robson, K.: Understanding why consumers don't skip pre-roll video ads. Journal of Advertising **46**(3), 411–423 (2017)

[4] Carretié, L., Hinojosa, J.A., Mercado, F.: Cerebral patterns of attentional habituation to emotional visual stimuli. Psychophysiology **40**(3), 381–388 (2003)

[5] Chaturvedi, I., Thapa, K., Cavallari, S., Cambria, E., Welsch, R.E.: Predicting video engagement using heterogeneous deepwalk. Neurocomputing **465**, 228–237 (2021)

[6] Cisco, V.: Cisco visual networking index: Forecast and trends, 2017–2022. White paper **1**(1) (2018)

[7] Codispoti, M., De Cesarei, A., Biondi, S., Ferrari, V.: The fate of unattended stimuli and emotional habituation: Behavioral interference and cortical changes. Cognitive, Affective, & Behavioral Neuroscience **16**(6), 1063–1073 (2016)

[8] Ferrari, V., Mastria, S., Codispoti, M.: The interplay between attention and long-term memory in affective habituation. Psychophysiology **57**(6), e13572 (2020)

[9] Hegner, S.M., Kusse, D.C., Pruyn, A.T.: Watch it! the influence of forced pre-roll video ads on consumer perceptions. In: Advances in Advertising Research (Vol. VI), pp. 63–73. Springer (2016)

[10] Jeon, Y.A.: Skip or not to skip: Impact of empathy and ad length on viewers' ad-skipping behaviors on the internet. In: International Conference on Human-Computer Interaction. pp. 261–265. Springer (2018)

[11] Jeon, Y.A., Son, H., Chung, A.D., Drumwright, M.E.: Temporal certainty and skippable in-stream commercials: Effects of ad length, timer, and skip-ad button on irritation and skipping behavior. Journal of Interactive Marketing **47**, 144–158 (2019)

[12] Li, H., Lo, H.Y.: Do you recognize its brand? the effectiveness of online in-stream video advertisements. Journal of advertising **44**(3), 208–218 (2015)

[13] Li, X., Shi, M., Wang, X.S.: Video mining: Measuring visual information using automatic methods. International Journal of Research in Marketing **36**(2), 216–231 (2019)

[14] Mukherjee, A., Dubé, L.: Mixing emotions: The use of humor in fear advertising. Journal of Consumer Behaviour **11**(2), 147–161 (2012)

[15] Myrick, J.G., Oliver, M.B.: Laughing and crying: Mixed emotions, compassion, and the effectiveness of a youtube psa about skin cancer. Health communication **30**(8), 820–829 (2015)

[16] Pace-Schott, E.F., Shepherd, E., Spencer, R.M., Marcello, M., Tucker, M., Propper, R.E., Stickgold, R.: Napping promotes inter-session habituation to emotional stimuli. Neurobiology of learning and memory **95**(1), 24–36 (2011)

[17] Quach, S., Septianto, F., Thaichon, P., Chiew, T.M.: Mixed emotional appeal enhances positive word-of-mouth: The moderating role of narrative person. Journal of Retailing and Consumer Services **62**, 102618 (2021)

[18] Raditya, D., Gunadi, W., Setiono, D., Rawung, J.: The effect of ad content and ad length on consumer response towards online video advertisement. The Winners **21**(2), 119–128 (2020)

[19] Schwenzow, J., Hartmann, J., Schikowsky, A., Heitmann, M.: Understanding videos at scale: How to extract insights for business research. Journal of Business Research **123**, 367–379 (2021)

[20] Semerádová, T., Weinlich, P.: The (in) effectiveness of in-stream video ads: Comparison of facebook and youtube. In: Research Anthology on Strategies for Using Social Media as a Service and Tool in Business, pp. 668–687. IGI Global (2021)

[21] Wright, C.I., Fischer, H., Whalen, P.J., McInerney, S.C., Shin, L.M., Rauch, S.L.: Differential prefrontal cortex and amygdala habituation to repeatedly presented emotional stimuli. Neuroreport **12**(2), 379–383 (2001)

DOI: 10.15439/2023F7822
ISSN 2300-5963 ACSIS, Vol. 35

Performance assessment of OpenMP constructs and benchmarks using modern compilers and multi-core CPUs

Bartłomiej Gawrych and Paweł Czarnul
Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology
Narutowicza 11/12, 80-233 Poland, email: pczarnul@eti.pg.edu.pl

***Abstract*—Considering ongoing developments of both modern CPUs, especially in the context of increasing numbers of cores, cache memory and architectures as well as compilers there is a constant need for benchmarking representative and frequently run workloads. The key metric is speed-up as the computational power of modern CPUs stems mainly from using multiple cores. In this paper, we show and discuss results from running codes such as: batch normalization, convolution, linear function, matrix multiplication, prime number test and wave equation; using compilers such as: GNU gcc, LLVM clang, icx, icc; run on four different 1 or 2-socket systems: 1 x Intel Core i7-5960X, 1 x Intel Core i9-9940X, 2 x Intel Xeon Platinum 8280L, 2 x Intel Xeon Gold 6130. Results can be regarded as suggestions concerning scaling on particular CPUs including recommended thread number configurations.**

I. Introduction

PARALLEL computing has become increasingly popular due to the widespread availability of multi- and many-core CPUs and accelerators such as GPUs, not only in cluster nodes, servers and workstations but also desktops and even mobile devices. In line with the hardware developments, many APIs are used for general purpose programming in such environments, including: OpenMP and OpenCL for shared memory systems with offloading to accelerators, OpenACC for directive based accelerator programming, CUDA for NVIDIA GPUs, Message Passing Interface (MPI) for internode communication among processes of a parallel application. OpenMP is very important due to its relatively easy to learn directive + library based multithreaded model allowing easy parallelization of sequential codes and support for offloading computations to accelerators such as GPUs [1].

The contribution of this paper over the state-of-the-art described in Section II, is assessment of OpenMP's implementation performance for a combination of: a variety of specific constructs and benchmarks, each of which benchmarked on various 1 and 2 socket systems with modern multi-core Intel CPUs and each tested using 4 compilers: GNU gcc, LLVM clang, icx, icc, run for various data sizes. Benchmarks include: batch normalization used in deep learning, convolution frequently used in signal processing, linear addition function benchmark, matrix multiplication, prime number test as well as wave equation simulation.

II. Related work

In [2] a set of microbenchmarks derived from EPCC and based on SKaMPI was run and analyzed on IBM SP3 and SunFire systems. Those included OpenMP's lock/unlock, critical section, barrier, single, parallel and parallel for directives. Times were measured for the two systems between 1 and 8 processors showing generally much better values for the Sun system especially showing sharp increases of times across the ranges for IBM SP3 vs Sun for barrier, for reduction, parallel and for single for >2 processors, critical for >4 and lock/unlock for >5 processors. In [3] performance of a Loongson-3A SMP quad-core system was assessed for EPCC microbenchmarks and NPB, using: gcc, OMPi with pthreads or psthreads. Testing parallel, for, parallel for, barrier and single for 1-4 threads, OMPi+pthreads tested best; for critical, unlock/lock, ordered and atomic gcc resulted in much larger overhead for 2-4 threads than the other very comparable solutions. For loop scheduling: static OMPi+pthreads and gcc were best while for dynamic and guided OMPi+pthread shall be preferred. The analyzed platform was also compared to Intel i5 with normalized (versus CPU clock) ratios for NPB (4 threads) between 1.3 (EP) and 5.1 (CG).

Authors of paper [4] benchmarked a 72-way Sun Fire 15K multiprocessor system with several EPCC microbenchmarks including measurements of overheads of OpenMP's frequently used construct implementations. OpenMP directives benchmarked included parallel, for, parallel for, barrier, single, critical, lock/unlock, atomic, along with scheduling modes such as static, dynamic and guided (1-128 chunk size). C and Fortran implementations were tested using 6, 12, 24, 48, 64 and 70 threads. Generally, overheads increase expectedly with the number of threads, in selected cases considerably starting with a given number of threads e.g. 48+ threads for critical, lock/unlock and atomic for the C implementation. Additionally, overhead of approximately 20% was measured for separate parallel+for in C and equivalent parallel+do in Fortran compared to combined versions. For NAS parallel benchmarks various maximum speed-ups were obtained: approximately 50 for BT and SP, 70 for LU, over 95 (superlinear) for CG, over 50 for MG and over 20 for FT.

In paper [5] the author investigated various OpenMP implementations of one of the most popular parallel programming paradigms – master-slave. Six versions were implemented, based on: OpenMP locks, the tasking construct, for loop dynamically partitioned, the latter two without and with overlapping merging results and data generation. Two concrete applications were implemented: one with irregular adaptive quadrature numerical integration and the second implementing finding a region of interest within an irregular image. gcc version 9.3.0 was used on two systems with: the first one with Intel i7-7700 3.60 GHz Kaby Lake CPU and 8 logical processors, the second one with two Intel Xeon E5-2620 v4 2.10 GHz Broadwell CPUs and 32 logical processors. All in all, for integration the best results were obtained for tasking and dynamic for with or without overlapping (for systems 2 and 1) while for image recognition for system 1 dynamic for and using locks while for system 2 dynamic for (both versions) and tasking with overlapping.

Scalability and overheads during execution of parallel code is studied in more detail in [6] where authors distinguished 4 overhead categories such as: need for synchronization among threads, imbalance, limited parallelism i.e. not (fully) parallelized code and thread management. For benchmarking the authors used OpenMP's version of NAS Parallel Benchmarks (class C) characterizing presence of particular OpenMP constructs in particular benchmarks – present mostly LOOP, PARALLEL and PARALLEL_LOOP in all tested as well (except PARALLEL in FT) as master in BT, LU, MG and SP; ATOMIC in BT, EP, LU and SP; BARRIER in IS and LU; CRITICAL in SP; SINGLE in LU. Codes were benchmarked using between 2 and 32 threads on an 32 CPU Itanium-2 based SGI Altix machine. All in all, imbalance appeared to be the largest overhead generally, as much as 20% for SP; synchronization turned out to be significant for IS (largest) and visible for LU. Thread management was noticed in IS, MG and CG although not large.

In paper [7] authors implemented and benchmarked 3 versions of OpenMP codes for an iterative Jacobi solver for 2D structured grids, representative of geometric SPMD codes such as for e.g. CFD applications. The code versions included: standard shared memory OpenMP host code with `parallel` and `do` directives, standard code augmented with `target` and `target data` directives and code with `target`, `target data`, `teams`, `distribute` directives. Codes were run on 2 systems: one with 2 Intel Xeon E5-2670 CPUs + 4 Intel 5110P Phis, the other with AMD Interlagos CPU + NVIDIA K20X GPU. For the first system (Intel compiler), offloading has been shown to be effective, even if offloading to self case, almost as good as the standard OpenMP code. For the second system (Cray compiler), only standard code on CPU and offload to GPU performed well.

In work [8] authors benchmarked OpenMP as a programming API through its language constructs on the IBM Cyclops64 system with 160 processing cores within a single chip. Specifically, EPCC microbenchmarks were used with their 3 elements testing: synchronization, scheduling as well as array directives and clauses. Overheads in terms of cycles versus numbers of threads within the 1-128 range were tested. Specifically, the overhead of `FOR` turned out to be only minimally higher than that of `BARRIER` and of `PARALLEL FOR` minimally larger than that of `PARALLEL`. The overhead of `SINGLE` is comparatively large. `DYNAMIC(1)` resulted in very large overhead, especially compared to `DYNAMIC` for chunk sizes 64-128 and `STATIC` for equivalent chunk sizes. Additionally, overheads of `PRIVATE` and `FIRSTPRIVATE` used in conjunction with `PARALLEL` add very small, minimally larger for the latter.

III. Methodology and benchmarks

Within the paper, we aim at comparative analysis of several orthogonal aspects in terms of OpenMP applications, including: many various benchmarks that differ in compute and memory intensity, as well as OpenMP directives used; tests for various input data sizes; several popular compilers: GNU gcc, LLVM clang, icx, icc; several CPUs representing various architectures and generations.

To evaluate performance, several programs were written using various OpenMP directives and their combinations. Problems benchmarked are as follows:

- **Batch-Normalization** – popular function used in deep learning, especially in computer vision problems [9].
- **Convolution** – method commonly used in signal processing, but also very popular in computer vision problems. This benchmark tests the parallelization of five nested loops and how collapse directive and its parameters are impacting performance.
- **Linear function** – performing multiplication and addition to each element of the array ($y = a * x + b$), testing if OpenMP's SIMD directive affects the execution time of the program.
- **Matrix multiplication** – we used implementation with O(n^3) complexity and parallelization with OpenMP's `schedule(static)` and `collapse` directives.
- **Prime number test** – implementation, which divides number by all numbers from 2 to $\sqrt{n}$, has been chosen in order to compare schedule clauses that are static, guided, and dynamic.
- **Wave equation** – benchmark testing the performance impact of using the `parallel` directive both within and outside of the time step loop (making the `parallel` directive called only once).

The experiments were carried out to test the performance of specific OpenMP implementations with an increasing number of threads for varying issue sizes, as well as the effect of various work-sharing directives. Multiple iterations of benchmarks were performed using sizes that were selected based on subjective criteria in order to conduct operations on various sizes, from small to large. The number of rounds was adjusted so the fastest execution of full benchmark measurement took longer than 1 second. Apart from measuring average time of a single run, our testing framework also calculated standard deviation which can be found on GitHub [10], along with full compilation configuration and compiler flags used for each platform.

IV. Experiments

A. Testbed environments

Table I details tested systems with configurations imposed by the production environments.

	CPU	S/C/T	Operating System	RAM
a)	Xeon Gold 6130	2/16/32	Ubuntu 18.04.3 LTS	256 GB
b)	Xeon Platinum 8280L	2/28/56	CentOS Linux 7 (Core)	192 GB
c)	Core i7-5960X	1/8/16	Ubuntu 18.04.5 LTS	16 GB
d)	Core i9-9940X	1/14/28	Ubuntu 20.04.2 LTS	128 GB

TABLE I
Configuration used to benchmark OpenMP implementations (S - Sockets, C - Cores, T - Threads)

Table II presents compilers and OpenMP versions which were used to evaluate the performance. Our intention was to compile with the latest stable OpenMP release available for each compiler at the time.

Compiler	Compiler Version	Name of OpenMP lib	OpenMP version in CMake
GNU GCC	10.2.0	libgomp.so	4.5
LLVM Clang	11.1.0	libomp.so	5.0
ICX	12.0.0	libiomp5.so	4.5
ICC	20.2.2.20210228	libiomp5.so	5.0

TABLE II
Benchmarked OpenMP implementations

B. Tests

Within the following tests, we present speed-ups versus the number of threads executing a particular benchmark, in selected cases for several variants and settings.

1) Batch-Norm: With the small problem size for Batch-Norm, the best performance improvement was achieved on processor (b) - peak performance was observed using only 32 threads and it was 25 times faster than sequential run. Using only 32 threads also gives the best result on processor (a), for the rest, using all available threads constituted an optimal solution.

With increased size the best performance for all examined CPUs was achieved using all available threads (Figure 1). The rapid performance loss that occurs when employing one more thread than half of those available is an interesting phenomenon – at this point, hyper-threading begins to function. Despite this problem, performance increases linearly when using more and more threads. This does not apply to the result of the GCC compiler on machine (b) where performance started to decline progressively once more than 32 threads were used.

2) Convolution: Results from Figure 2 demonstrate how crucial it is to employ the collapse clause when appropriate. If CPUs have enough threads to consume the first loop entirely, the rest of available threads will be idle. Using the `collapse` clause generates many more tasks which can be distributed among different threads, which results in better scalability than without this clause – characteristic speed-up when using divisible number of threads in relation to the iteration count of first loop no longer exists. Overall, deciding whether it is always better to use 2-level or 3-level collapsing cannot be done, as it depends on the used compiler and the machine – e.g. for the Clang compiler on machine (a), best improvement is for collapse(2), but on machine (b) for collapse(3).

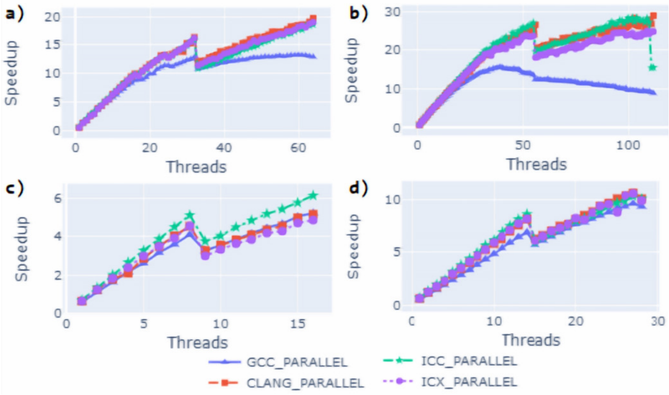


Fig. 1. Results of Batch-Norm with size N=32, C=2048, H=7, W=7

The results performed for N=256, H=112, W=112, kernel=3x3 indicate that when the first level loop has a large enough number of iterations to distribute tasks for each thread it is worthwhile to consider not using the `collapse` clause at all. It can be observed on machine (c) with GCC compiler and machine (d) ICC compiler.

3) Matrix Multiplication: For matrix-vector multiplication, in the case of desktop processors – (c) and (d), the results are satisfying and linear improvement can be observed for all compilers. Using the `collapse` clause has neither beneficial nor negative effect there. However, in server type CPUs differences show up. On machine (a) using the `collapse` clause causes performance degradation for every compiler. On machine (b) linear speed-up was disrupted by occurred anomalies after using more than 28 threads, which indicates the use of the second NUMA node.

For small square matrix-matrix multiplication desktop CPUs scale well and only a characteristic performance drop becomes apparent when hyper-threading comes into play. For configuration (a) and (b) compilers ICC and ICX allow good scaling and positive effect of using the `collapse` clause can be observed. For other configurations, scaling is much more irregular - especially for Clang on machine (b).

Increasing the size by an order of magnitude in each dimension causes all configurations but (b) to suffer from using hyper-threading as performance drops dramatically. Additionally, differences between using or not-using collapse construction are not visible for this size. The best scaling can be observed for configuration (b), but again, with anomalies visible in Figure 3 - when using HT threads.

4) Linear function: Tests performed for size=10000 showed very limited speed-ups. When the size is two orders of magnitude larger, the results are significantly better. In this example, letting the OpenMP implementation to determine chunk size automatically produces far better results than using

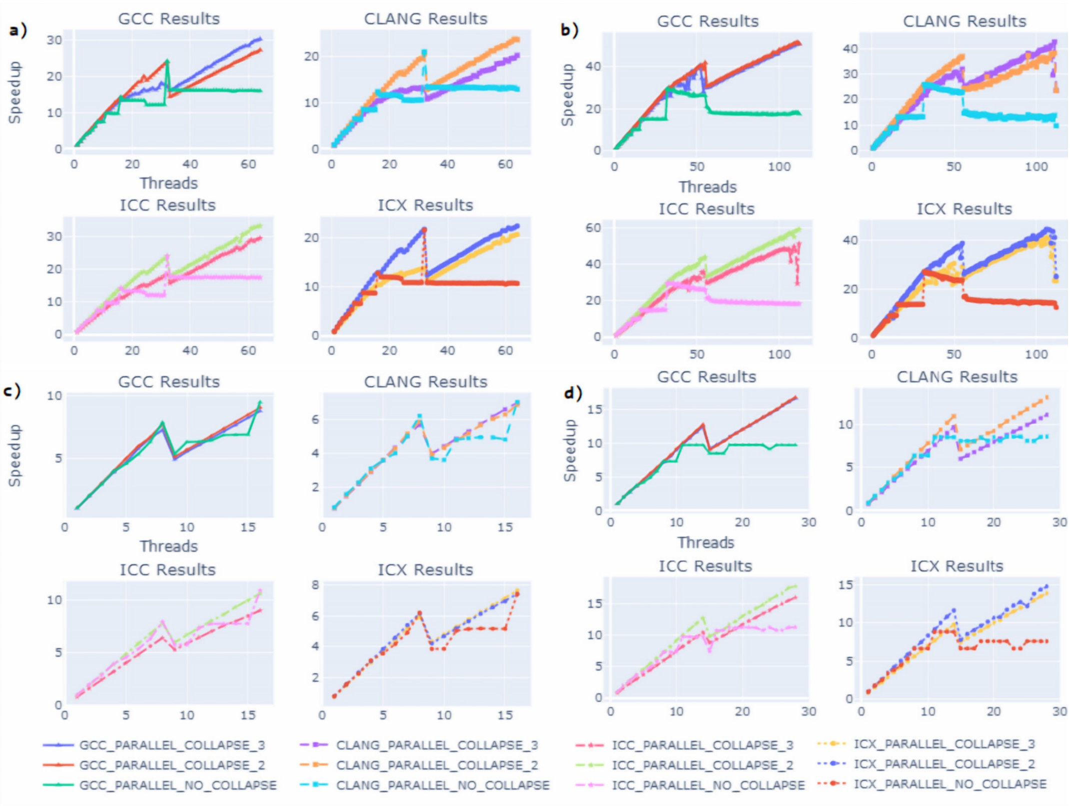


Fig. 2. Results of Convolution with size N=32, H=224, W=224, kernel=7x7

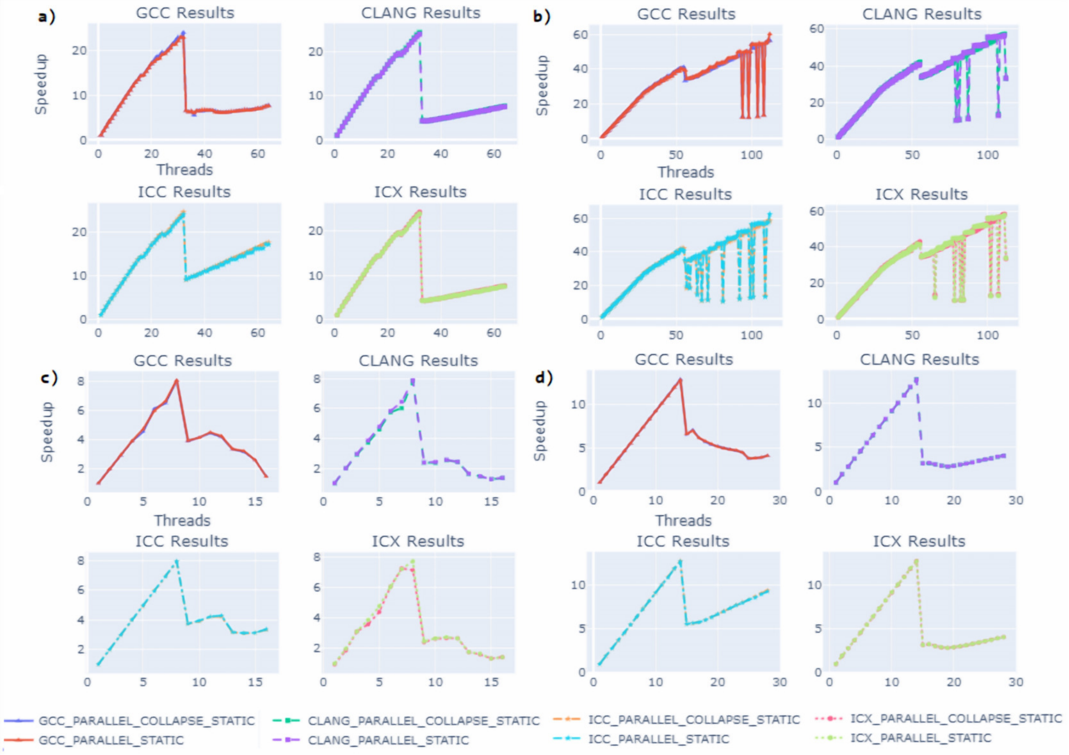


Fig. 3. Results of Matrix Multiplication with size N=1000, M=1000, K=1000

the `static` clause with a manually set chunk size. When it comes to the SIMD construction, in almost all cases it does not matter, but using this clause is beneficial when using ICC, but only on machine (d).

5) PrimeTest: Testing whether a given number is prime or not with a plain algorithm produces unbalanced amounts of work for particular threads. Results for size=10000 show that using the `guided` scheduling clause is the most stable one for every compiler. Poor performance for dynamic scheduling is visible for GCC and ICC for configurations (a), (b) and (d), but not for configuration (c), where it gives the best improvement. Another interesting observation is that a characteristic performance drop when CPU starts using hyper-threading vanished with the usage of guided scheduling.

For larger vectors of numbers to test, charts in Figure 4 are reasonably smooth and regular. Differences between using the `dynamic` and `guided` clauses are not visible and in the end, almost every configuration achieves the same level of parallelization when using all threads (except for dynamic scheduling using GCC and ICC compilers). Performance drop for static scheduling and hyper-threading still appears.

6) Wave Equation: The final benchmark determines whether it is better to place the `parallel` directive within or outside of a time-step loop, as the second iteration depends on first iteration's results. Results shown in Figure 5 are ambiguous. In most cases placing the `parallel` directive together with work-sharing for-loop gives better results. One exception to this appears on machine (b) when compiling with Clang – the chart is very irregular, but it is clear that placing parallel outside of the time-step loop produces better results.

For the larger sizes of the problem (N=5000, M=5000, T=100 tested) the differences are smaller and for desktop CPUs are almost not visible. Similar conclusions can be drawn for the server CPU from configurations where charts are overlapping each other. Only for configuration (a) some differences occurs – slightly better performance is observed when parallel clause is inside time-step loop, but scaling is then more irregular.

V. Summary and Future Work

Results show that no single best compiler nor OpenMP implementation can be chosen. In many cases compilers showed similar speed-up patterns on charts, however they differed in speed-up values. Different results and rankings were collected for various problems and various sizes of problems – consequently best configurations need to be considered on a case by base basis. In line with expectations, better scaling was achieved for bigger data sizes of problem – that suggests that data size must be large enough to get satisfying speed-ups. Often for small data sizes better performance can be achieved by using relatively few cores. It is especially visible when comparing desktop CPUs (a small number of cores) with server CPUs (a large number of cores) – in some cases exceeding a certain number of used cores caused degradation of performance. Consequently, it is recommended to benchmark own program with different OpenMP implementations in a production environment to get the best results in terms of performance before final deployment.

For future work, it would be valuable to benchmark the workloads also under power caps and determine performance-energy trade-offs [11] including optimization goals EDP, EDS as well as percentage wise performance loss for energy gains. Additionally, other benchmarks would also be of interest, such as: parallel similarity measure computations for large vectors [12] or image processing [13].

References

[1] P. Czarnul, *Parallel Programming for Modern High Performance Computing Systems.* CRC Press, Taylor & Francis, 2018, iSBN 9781138305953.

[2] A. Prabhakar, V. Getov, and B. Chapman, "Performance comparisons of basic openmp constructs," in *High Performance Computing*, H. P. Zima, K. Joe, M. Sato, Y. Seo, and M. Shimasaki, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. ISBN 978-3-540-47847-8 pp. 413–424.

[3] Q. Luo, C. Kong, Y. Cai, and G. Liu, "Performance evaluation of openmp constructs and kernel benchmarks on a loongson-3a quad-core smp system," in *2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2011. doi: 10.1109/PDCAT.2011.66 pp. 191–196.

[4] N. R. Fredrickson, A. Afsahi, and Y. Qian, "Performance characteristics of openmp constructs, and application benchmarks on a large symmetric multiprocessor," in *Proceedings of the 17th Annual International Conference on Supercomputing*, ser. ICS '03. New York, NY, USA: Association for Computing Machinery, 2003. doi: 10.1145/782814.782835. ISBN 1581137338 p. 140–149. [Online]. Available: https://doi.org/10.1145/782814.782835

[5] P. Czarnul, "Assessment of openmp master–slave implementations for selected irregular parallel applications," *Electronics*, vol. 10, no. 10, 2021. doi: 10.3390/electronics10101188. [Online]. Available: https://www.mdpi.com/2079-9292/10/10/1188

[6] K. Fürlinger and M. Gerndt, "Analyzing overheads and scalability characteristics of openmp applications," in *High Performance Computing for Computational Science - VECPAR 2006*, M. Daydé, J. M. L. M. Palma, Á. L. G. A. Coutinho, E. Pacitti, and J. C. Lopes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. ISBN 978-3-540-71351-7 pp. 39–51.

[7] V. G. M. Vergara, W. D. Joubert, M. G. Lopez, and O. R. Hernandez, "Early experiences writing performance portable openmp 4 codes," in *Cray User Group Conference*, London, United Kingdom, May 2016.

[8] W. Zhu, J. del Cuvillo, and G. R. Gao, "Performance characteristics of openmp language constructs on a many-core-on-a-chip architecture," in *OpenMP Shared Memory Parallel Programming*, M. S. Mueller, B. M. Chapman, B. R. de Supinski, A. D. Malony, and M. Voss, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN 978-3-540-68555-5 pp. 230–241.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, p. 448–456.

[10] B. Gawrych and P. Czarnul, "Performance investigation of openmp constructs and benchmarks using modern compilers and multi-core cpus," September 2021, https://github.com/bgawrych/openmp_benchmark.

[11] A. Krzywaniak, J. Proficz, and P. Czarnul, "Analyzing energy/performance trade-offs with power capping for parallel applications on modern multi and many core processors," in *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, pp. 339–346.

[12] P. Czarnul, P. Rościszewski, M. Matuszek, and J. Szymański, "Simulation of parallel similarity measure computations for large data sets," in *2015 IEEE 2nd International Conference on Cybernetics (CYBCONF)*, 2015. doi: 10.1109/CYBConf.2015.7175980 pp. 472–477.

[13] P. Czarnul, A. Ciereszko, and M. Frązak, "Towards efficient parallel image processing on cluster grids using gimp," in *Computational Science - ICCS 2004*, M. Bubak, G. D. van Albada, P. M. A. Sloot, and

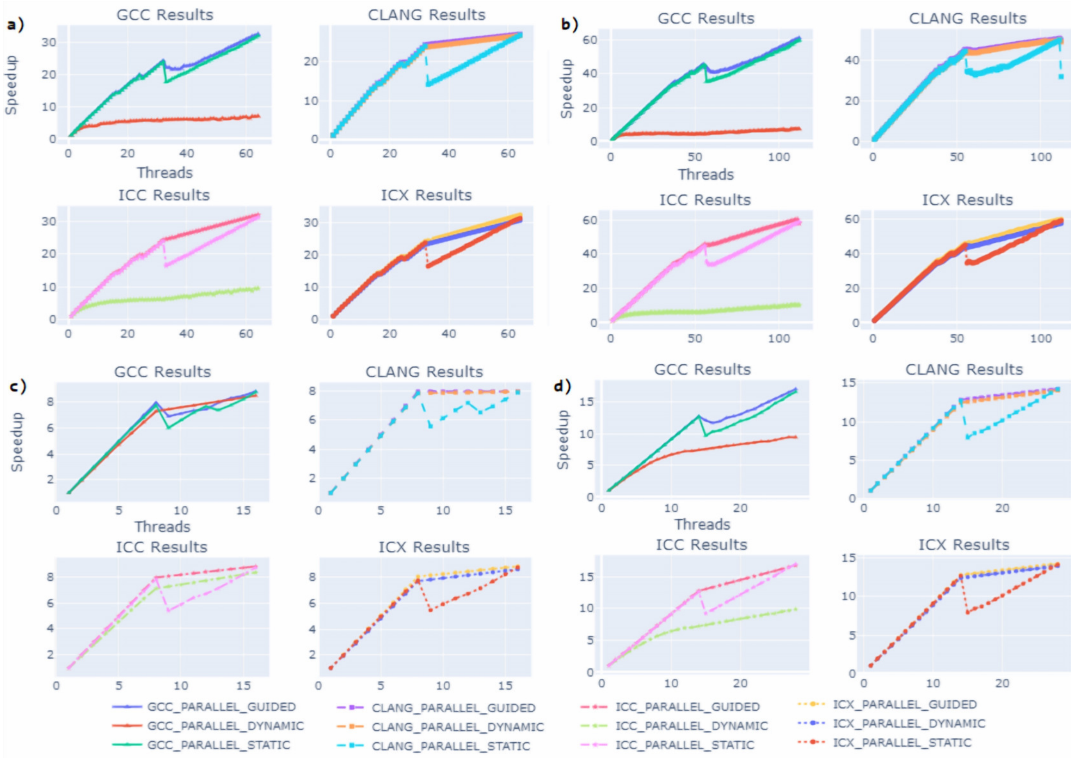


Fig. 4. Results of Prime Test with size=10000000

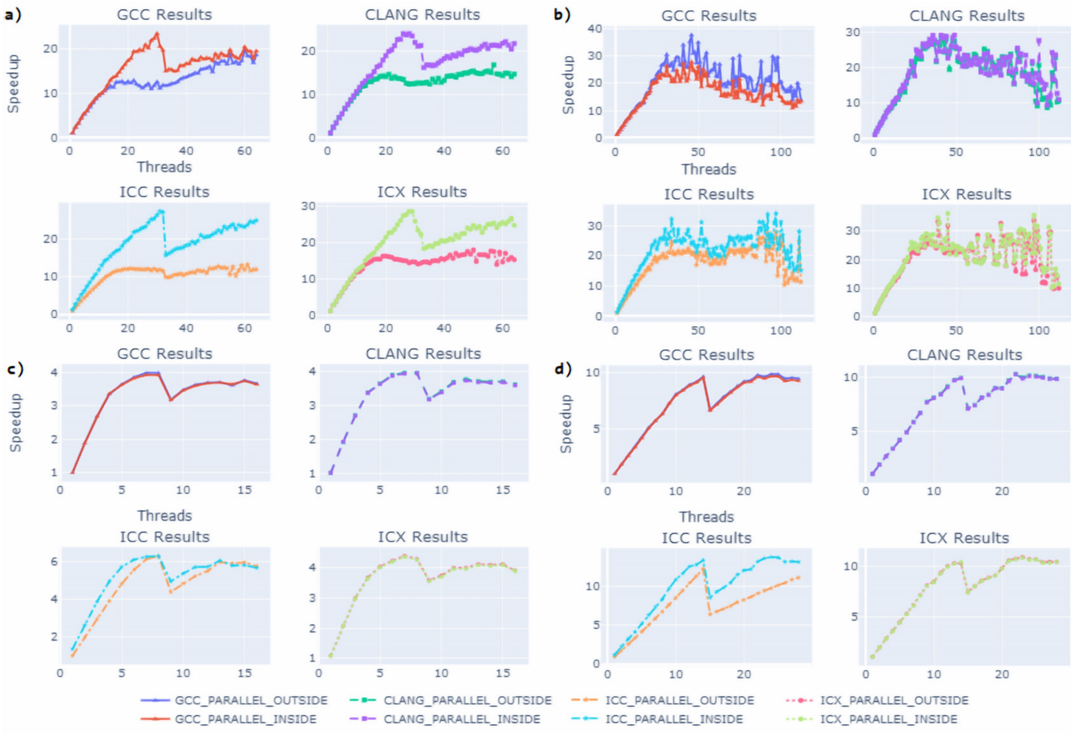


Fig. 5. Results of Wave Equation with N=1000, M=1000, T=10

J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. ISBN 978-3-540-24687-9 pp. 451–458.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 979–982
DOI: 10.15439/2023F5568
ISSN 2300-5963 ACSIS, Vol. 35

A short note on computing permutations

Pawel Gburzynski
Department of Computer Engineering
Vistula University
ul. Stokłosy 3, 02-787 Warsaw, Poland
ORCID: 0000-0002-1844-6110

Janusz Zalewski
Department of Informatics
Ignacy Mościcki State Professional College
ul. Narutowicza 9, 06-400 Ciechanów, Poland
ORCID: 0000-0002-2823-0153

***Abstract*—We discuss an algorithm for generating all permutations of numbers between 1 and N. The algorithm is short and efficient, yet its behavior is not obvious from the code, mostly owing to the recursion. The discussion touches upon a few interesting methodological issues and brings in an educational case study in recursion.**

***Index Terms*—algorithms, recursion, permutations, algorithm analysis**

I. Introduction

NOVEL programs for generating permutations are not in big demand today, as the issue is deemed to have been “settled” by the opus of D. E. Knuth [1]. The algorithm we are about to present has been known to us since 1980, although it has never been published, except for a brief mention in [2]. It comes with a story which is best told in a narrative less formal than demanded by a research paper because announcing upfront the algorithm’s purpose removes from the yarn the essential element of suspense.

When the algorithm was first introduced to an audience of students in an introductory programming course, it caused a bit of confusion. During an exam, the students were asked to guess what the program was doing, explain its flow control, and describe the output produced, i.e., tell the ordering of the resulting sequence of permutations. The era of portable communication/computing gadgets (so nightmarish from the viewpoint of a contemporary examiner) was still far ahead, so the students were left to their own “devices.” To the one of us who devised the exam and the question the problem seemed non-trivial but well within the grasp of a university student in computer science who had acquired the understanding of recursion in programming. But it was in fact a disaster. When two local and accomplished faculty experts in algorithm design and analysis were subsequently shown the question, their reflex, after a brief deliberation, was to run to the computer terminal and see what happens. That incident left us with a feeling that has persisted to this day, that some important questions deserve a thought.

We introduce the algorithm as a case study in algorithm design. First, the nature of recursion employed in it is nontrivial and educational, as it involves both categories of data (global, local) in a manner that makes them both relevant to the workings of the algorithm. The recursion is essential to the program’s dynamics, e.g., in contrast to artificial examples where it can be trivially eliminated (like in calculating the Fibonacci function). While other recursive (and also optimal) algorithms for generating permutations are known [3], they (as most of their non-recursive relatives) assume element swapping as the basic operation. This way, the problem immediately receives an algebraic flavor and becomes that of transforming a given permutation into another permutation in such a way that all permutations are eventually mentioned in the transformation sequence. This is not necessarily the most natural expression of the problem from the viewpoint of a programmer. Our algorithm, in contrast, simply *generates* all permutations by making sure that all the elements are eventually *permuted*, i.e., each of them appears in all the possible slots, while giving all the other elements every possible chance to do the same.

Second, the basic variant of the algorithm, while being relatively easy to explain and instructional from the viewpoint of its correctness proof, is suboptimal from the viewpoint of performance. With some creativity, the algorithm can be improved such that its complexity matches the best (possible) solutions to the problem. The improved version appears more complicated (if introduced alone, it would have been considerably more difficult to analyze); however, owing to its descent from the basic variant, its analysis (both in term of correctness and performance) can be naturally carried over from the easier case. Then, we show how the algorithm can be transformed into a function that, instead of generating all the permutations in response to its zero-level call, can be invoked multiple times to yield consecutive permutations on individual demand. Overall, we believe it amounts to a case for beauty in programming, as per the views expressed in [4].

II. The basic algorithm

The algorithm has the form of a recursive procedure listed in Figure 1. It operates on three global variables:

```
var N, k : integer; A : array [1..N] of integer;
```

The array will contain consecutive permutations generated by the algorithm, and its effective size is N. We want to express the algorithm in a simple, generally understood programming language such that the code is (almost) immediately runnable. It makes sense to use a Pascal lookalike because it is convenient to have the array indexed from 1.

```
 1  procedure F ( );
 2      var i : integer;
 3  begin
 4      if k > N then
 5          ready ( )
 6      else
 7          for i := 1 to N do
 8              if A [i] = 0 then
 9                  A [i] := k; k := k + 1;
10                  F ( );
11                  k := k – 1; A [i] := 0;
12              end if
13          end for
14      end if
15  end;
```

Fig. 1. The basic variant

Before $F()$ is invoked for the first time (externally), N is set to the requisite parameter (and henceforth appears as a constant), the array A is initialized to zeros (for the indices from 1 to N), and k is set to 1.

Each call to *ready()* in $F()$ marks the moment when A contains a new permutation which can be printed out or otherwise used. Thus, the algorithm (in its boilerplate variant listed in Figure 1) *generates* all permutations, e.g., as opposed to returning them one by one on subsequent invocations [5]. Function *ready()* should be viewed as the consumer of the output produced by the algorithm. It is convenient to start the presentation with a variant where the consumer is intertwined with the procedure. Later we shall show how the two can be disentangled.

Before looking into the algorithm's behavior, let us reflect on the author's inspiration. An exam was being devised, most of the questions had been written down, and the one remaining topic to be addressed was recursion. The problem had to be stated as briefly as possible, and it had to touch upon all the essential aspects of data from the viewpoint of a recursive algorithm. Thus, we needed at least one local variable and at least one global variable (both of them relevant), and (of course) a non-trivial recursive invocation. In this respect, the two global variables k and A (N can be treated as a constant), and the local variable i nicely fit the bill. Consequently, one advantage of our algorithm is that it provides an educational case study in recursive programming, even if its practical significance is not transparent.

III. Correctness

Formal correctness proofs of our algorithm have been the topic of several studies in Algorithmic Logic [6]. Here, for the sake of brevity, we shall confine ourselves to informal arguments. Our goal is to convince the reader that the algorithm terminates and in fact generates all permutations of the numbers from 1 to N, with each permutation appearing exactly once.

The following snippet illustrates the way to invoke $F()$ as to account for the required initialization:

```
...
readln (N);
for k := N downto 1 do A [k] := 0;
...
F ( );
...
```

Note that the loop setting the array elements to zero has the side effect of initializing k to 1. Thus, when the procedure is called from the outside (as opposed to its recursive invocation within itself), A is filled with zeros and k contains 1.

The global variable k is only modified in lines 9 and 11. It is incremented just before the internal (recursive) invocation of $F()$ and brought back to the previous value when the procedure returns. As it starts from 1, before the first (outer) call to $F()$ is made, it can be viewed as the counter of the recursive levels going from 1 until $N+1$. Note that the last level ($N+1$) is special: the function uses it to present its result, which action is represented by the call to *ready()*. In lines $7-13$, the procedure executes a loop going through all elements of A. Those elements that contain nonzero values are skipped, and the same value of k is consecutively inserted into the remaining positions in the array. Then, for every possible insertion of k, the procedure is called recursively with k incremented by one. Everything is undone when the recursive invocation of $F()$ returns. Looking globally, we see that the procedure inserts 1 in all places in A (initially, when $k=1$, all of them are empty), then, for every configuration from the previous level, inserts 2 into all the remaining positions, and so on, all the way until N. This is how the problem of generating all permutations of the numbers from 1 to N is in fact defined. The procedure just literally fulfills this prescription; thus, in a sense, it can be viewed as the most natural (naive) solution to the problem.

IV. The time cost

Is our naive solution practical? To answer this question, we should gauge it against the best known algorithms for generating permutations which are known as loopless (or loop-free) ones [5], [7]. This term is somewhat unfortunate (no algorithm generating permutations can be truly loop-free) and refers to the constant average cost per permutation. In other words, the cost of generating all $N!$ permutations should be bounded by $c \times N!$ where c is some constant. Besides, a useful procedure should be able to generate *a* permutation when asked for it, i.e., one permutation at a time [1], [5], [7]–[10], as opposed to producing them all in response to a single invocation.

Let us start by calculating how many times $F()$ is called to produce all $N!$ permutations. We shall ignore the last-level call (for $k=N+1$) because its is special; its sole purpose is to present a ready permutation available in A. Denoting the number of (nontrivial) calls of the procedure by $U(N)$, we have:

$$\begin{array}{rcl} U(N) & = & N \times (1 + U(N-1)) \\ U(1) & = & 1 \end{array} \qquad (1)$$

One can easily show by induction that:

$$U(N) = N! \times \sum_{i=1}^{N} \frac{1}{i!} \qquad (2)$$

which means that:

$$U(n) < (e-1)N! \quad \text{and} \quad \lim_{N \to \infty} \frac{U(N)}{N!} = e \qquad (3)$$

V. The asymptotically optimal variant

The number of invocations of *F()* needed to solve the problem for a given N is of order $N!$ with the factor $c < e-1$. If we could prevent the *for* loop from iterating over the nonzero entries in A, which simply have to be skipped and ignored, and make it proceed directly to the next free entry on every turn, we would bring the complexity of our algorithm down to $O(N!)$. To accomplish that, in addition to the original array A, we introduce another array acting as a representation of the list of free entries in A available at the current level. The new set of global declarations becomes this:

```
var N, k : integer; A : array [1..N] of integer;
    X : array [0..N] of integer;
```

The role of A is now reduced to storing the permutation being constructed by the procedure, while X keeps track of the unoccupied slots in A. The initialization/invocation sequence is replaced with this code:

```
...
readln (N);
for k := N+1 downto 1 do X [k – 1] := 0;
...
G ( );
...
```

The new variant of the procedure shown in Figure 2 is named $G()$. We claim that it generates all permutations of values $1, \ldots, N$ in an asymptotically constant number of steps per permutation, i.e., its time complexity is bounded from above by:

$$T(N) = cN! \qquad (4)$$

To see this notice that initially the values in X describe the straightforward succession of indices in A where the head points to element 1, every subsequent element of X, for $i = 1, \ldots, N-1$ points to the next element $(i+1)$, and the last element contains a special value $(N+1)$ indicating that the list ends there. Thus, immediately after the initialization (when $k = 1$) traversing the array through the links in X will amount to going through all its elements in exactly the same order as with the straightforward loop in $F()$. When a value is inserted into A, the corresponding index in X is replaced with its successor, which has the effect of removing the index from the list for all the subsequent recursive calls. The index is restored upon return from the recursive call, equivalent to zeroing the corresponding element of A in $F()$. This implies that $G()$ carries out the same series of nonzero insertions into A as its previous version, but the total number of instructions associated with every invocation of $G()$ is now constant.

```
procedure G ( );
    var b, d : integer;
begin
    if k > N then
        ready ( )
    else begin
        b := 0;
        while X [b] <= N do begin
            d := X [b]; A [d] := k; X [b] := X [d];
            k := k + 1;
            G ( );
            k := k – 1;
            X [b] := d; b := X [b];
        end while
    end if
end
```

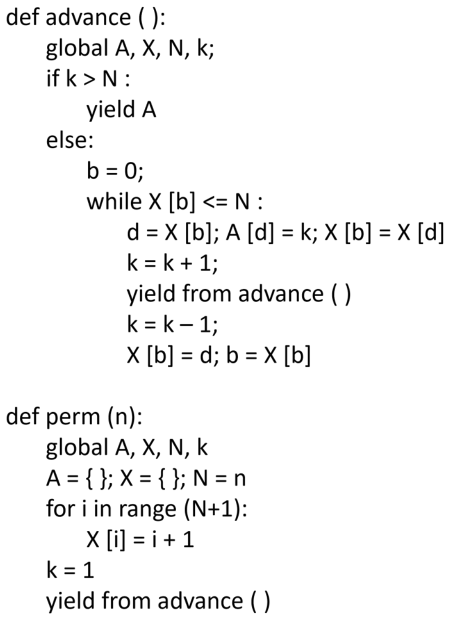
Fig. 2. The “loopless” variant

VI. One permutation at a time

The algorithm is inherently recursive which a practical programmer may see as a disadvantage. One would prefer a function that could be called from an external program each time a new permutation is needed. [5], [7]. Of course, as any recursive procedure, the algorithm can be reprogrammed in a non-recursive manner, but one can argue that the recursive form is its essential feature.

Modern programming languages and environments offer tools which make the adaptation of our algorithm to a practical usage natural and easy while retaining its essentially recursive form. These tools, under the name of coroutines, originated historically with Simula 67 [11], becoming useful features of many contemporary platforms and being available in several guises offering handy shortcuts for typical applications.

Figure 3 presents a modern-flavor coroutine-like variant of our algorithm implemented in Python. The implementation consists of a Python function *perm()*, providing the actual callable generator, and its helper function *advance()* taking care of the recursive part. The semantics of the *yield* operation [12] consist in suspending the execution of the current function and returning to its caller in such a way that on a subsequent invocation of the same function its execution will continue from the point of the last interruption. The operation *yield from* f_2 carried out by a function f_1 invokes the specified function f_2 and, when that function returns via *yield*, carries



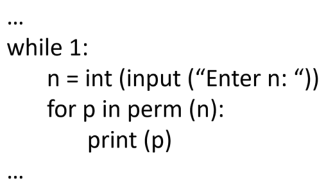
```
def advance ( ):
    global A, X, N, k;
    if k > N :
        yield A
    else:
        b = 0;
        while X [b] <= N :
            d = X [b]; A [d] = k; X [b] = X [d]
            k = k + 1;
            yield from advance ( )
            k = k – 1;
            X [b] = d; b = X [b]

def perm (n):
    global A, X, N, k
    A = { }; X = { }; N = n
    for i in range (N+1):
        X [i] = i + 1
    k = 1
    yield from advance ( )
```

Fig. 3. A coroutine-style implementation in Python

over the effect to its original caller. Consequently, when the original caller calls f_1 again, it will continue within f_2 from the place where f_2 last yielded.

We can easily see that *advance()* is basically a straightforward rewrite of *G()* in Python, except that: 1) the function yields with the value of array A whenever *G()* would produce a complete new permutation; 2) the yield has to be carried over recursively, so the recursive call is appropriately replaced with a *yield from*.

The following code illustrates the usage of the generator:



```
...
while 1:
    n = int (input (“Enter n: ”))
    for p in perm (n):
        print (p)
...
```

In a serious project, the generator would be encapsulated into a structure isolating the namespace of its global variables.

VII. Final comments

One intriguing feature of our algorithm is the apparent difficulty to see its function at first sight and the wrong intuitions that it tends to connote for a first-time viewer, if presented without the spoiler. Our discussions, involving students as well as experts, have raised these questions:

1) Why can the designer of a few-line program (devised for educational purposes and with no malicious intentions) see things much clearer than a competent reader subsequently looking at the same piece?
2) How to best convey the “obvious” idea behind the design that, ideally, should be present there, in the very code, plain for everyone to see?
3) How to prevent misunderstandings and misrepresentations of the ideas implanted into programs by their designers? In other words, how to ensure that programs are correct?
4) How to think about programs, so the right and correct ideas can materialize and find their way into the code in a manner that will make them transparent, so they can be seen and comprehended when the code is scrutinized?

The design of procedure $F()$ began with a simple narrative: “I am going to generate all permutations of the values from 1 to N by inserting 1 into all possible places, and then, for every such insertion, inserting 2 into all places that still remain unoccupied, and so on, continuing doing so until all the values have been inserted.” This sentence seems to explain everything there is to see about the algorithm. It can also be viewed as the most straightforward plain-language specification of the problem and, at the same time, rather precisely explains the programmer’s intention. According to the paradigm of literate programming [13], it should thus be incorporated into the procedure’s code and become its integral component. Viewed in this light, $F()$ merely follows its simple specification to the letter. Considering that its efficiency is not worse than that of the most refined solutions known in the area, our algorithm should probably be viewed as the most natural solution to the problem of generating all permutations.

References

[1] D. E. Knuth, *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations (Art of Computer Programming)*. Addison-Wesley Professional, 2005.

[2] L. Banachowski, A. Kreczmar, and W. Rytter, *Analysis of Algorithms and Data Structures*. Addison-Wesley Longman Publishing Co., Inc., 1991.

[3] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge University Press, 2009.

[4] A. Oram and G. Wilson, *Beautiful code: Leading programmers explain how they think*. O’Reilly Media, Inc, 2007.

[5] G. Ehrlich, “Loopless algorithms for generating permutations, combinations, and other combinatorial configurations,” *Journal of the ACM*, vol. 20, no. 3, pp. 500–513, 1973.

[6] G. Mirkowska and A. Salwicki, *Algorithmic Logic*. PWN, Warszawa, 1987. [Online]. Available: http://lem12.uksw.edu.pl/wiki/Algorithmic_Logic

[7] N. Dershowitz, “A simplified loop-free algorithm for generating permutations,” *BIT Numerical Mathematics*, vol. 15, no. 2, pp. 158–164, 1975.

[8] C. W. Ko and F. Ruskey, “Generating permulations of a bag by interchanges,” *Information Processing Letters*, vol. 41, no. 5, pp. 263–269, 1992.

[9] D. R. van Baronaigien and F. Ruskey, “Generating permutations with given ups and downs,” *Discrete Applied Mathematics*, vol. 36, no. 1, pp. 57–65, 1992.

[10] S. Effler and F. Ruskey, “A CAT algorithm for generating permutations with a fixed number of inversions,” *Information Processing Letters*, vol. 86, no. 2, pp. 107–112, 2003.

[11] O.-J. Dahl and K. Nygaard, “Simula,” in *Encyclopedia of Computer Science*. Wiley, 2003, pp. 1576–1578.

[12] D. Beazley and B. K. Jones, *Python cookbook: Recipes for mastering Python 3*. ”O’Reilly Media, Inc.”, 2013.

[13] D. E. Knuth, “Literate programming,” *The Computer Journal*, vol. 27, no. 2, pp. 97–111, 1984.

DOI: 10.15439/2023F4831
ISSN 2300-5963 ACSIS, Vol. 35

Association Rule Mining for Requirement Elicitation Techniques in IT Projects

Denys Gobov
0000-0001-9964-0339
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prosp. Peremohy, Kyiv, Ukraine
Email: d.gobov@kpi.ua

Nikolay Sokolovskiy
0009-0000-1282-5665
Independent researcher
Email: sokolovskynik@gmail.com

***Abstract*—Selecting suitable techniques for requirements elicitation in IT projects is crucial to the business analysis planning process. Typically, the determining factors are the preferences of stakeholders, primarily business analysts, previous experience, and company practices, as well as the availability of sources of information. The influence of other factors is not as evident. One of the possible ways to form recommendations for using techniques is the analysis of industrial experience. This paper is intended to analyze the application of association rules mining to define factors influencing technique selection and predict the usage of a particular elicitation technique depending on the project context and specialist background. The dataset for experiments was formed based on a survey of 328 specialists from Ukrainian IT companies. The associations found to make it possible to speed up the process of choosing elicitation techniques and improve the elicitation process efficiency.**

***Index Terms*—associations rules mining, requirements elicitation, IT project, business analysis.**

I. Introduction

REQUIREMENTS elicitation is the effort expended by the Requirements Engineer to turn implicit desires, demands, wishes, needs, and expectations — which until now were hidden in their sources — into explicit, understandable, recognizable, and verifiable requirements [1]. The outputs of elicitation serve as input for the following tasks from the core business analysis cycle: current state analysis, risk assessment, and requirement specification and modeling [2]. Elicitation activities can be divided into three tasks: preparing, conducting, and result confirming. The effectiveness of elicitation directly depends on the quality of the first – preparation. The requirement engineer/business analyst should define the available source of information, a subset of stakeholders, who should be involved in the following elicitation activities and select appropriate elicitation techniques. Professional guides and standards recommend many techniques practitioners use in IT projects. Due to time and budget constraints, specialists can't use them all and should select a set of techniques best suited to the particular project's conditions. The set of predefined elicitation techniques significantly influences the business analysis, project plan, and the associated costs and resources needed. This study was conducted to analyze the current practices of using elicitation techniques in IT projects and to find associations between project context, specialist's profile, and techniques used for requirement elicitation via Association Rule Mining. The dataset for analysis was gathered via a survey of 328 IT specialists employed by Ukrainian and international companies with branches in Ukraine via a survey [3]. The strong associations identified with Association Rule Mining made it possible to formulate recommendations on using requirements elicitation techniques in IT projects.

II. Problem Statement

The task of selecting best-suited techniques, particularly requirements elicitation techniques, is performed by a business analyst at the start of the project due to defining and estimating a list of business analysis-related activities. But that does not mean it is a one-time task, and a list of used techniques can be updated based on the efficiency monitoring results and project context changes. Considering that the requirements elicitation lays the foundation for further analysis and development activities, the optimal technique selection is an essential business analysis task. The emergence of new techniques and their development in the process of business analysis evolution, as well as the continuously changing business environment, can lead to the complication of this task. A recommendation system that considers the accumulated experience of practicing business analysts and requirements engineers can be applied to solve this problem. An important condition is the explainability of these recommendations, which will allow for checking their applicability in the unique context of each project.

III. The Best Existing Solution

There are many studies regarding solving the choosing appropriate requirement elicitation technique problem using different approaches and models.

Hatim Dafaalla et al. [4] built a model based on an artificial neuronal network (ANN). The model was learned based on the collected dataset with 1684 records about selecting the elicitation technique. By choosing the ROC AUC metric as a score of the model, the authors achieved significant accuracy of the model, which was equal to 82%. Despite good forecasting by modeling, as with any other ANN, this model has a significant weakness. ANN is a net of perceptron (miniature models of neurons). The perceptron is organized in layers, which are connected to each other. The connections might have a different architecture. Each connection of each perceptron has a weight coefficient. The learning process is a process to optimize these coefficients. Unfortunately, a single coefficient and a set of coefficients don't have meaning and can't be explained in business terms. Similarly, connections, layers, and perceptions do not have any sense separately and don't explain how ANN solved a problem. That is the way some decisions of an ANN might be seen as strange, unexplained, and untrusted [5].

Nagy Ramadan Darwish et al. [6] suggested a hybrid approach. The manuscript describes a pipeline of methods. The feature is manually selected based on literature reviews. Then multiple linear regression model was built to select critical attributes influencing technique selection. In the last stage, the ANN was built. The accuracy of the final model was declared as 81%. Despite the remarkable result, the final model has the same limitations as discussed previously. Ihor Bodnarchuk et al. [7] applied goal function for assessment and selection architecture design in the context of "light-weighted" requirements techniques.

Different machine learning approaches were applied not only to technique selection but to related areas as well. Fadhl Hujainah and others [8] suggested using a semi-automated attribute measurement criteria method for requirement prioritization and selection.

Similar method - attributes-based decision making was described by Jinyu Li [9]. Remarkably, semi-automated methods bring a possibility of bias since experts conducted the first assessment.

IV. The Proposed Solution

Associate Rule Mining (ARM) method is a machine-learning technique that combines several remarkable advantages. Firstly it doesn't require data annotation because it is an unsupervised method. Secondly, the method and output are intuitive and could be understood by domain experts and business people, which is a rare property of a machine-learning algorithm.

ARM, also known as basket analysis, was applied first in retail, but now it is widely applied in other areas. For example, Giovanna Castro and colleagues in [10] applied association rules to study the comorbidity of bipolar disorder and premenstrual dysphoric disorder. Chad Creighton [11] used association rules to discover hidden gene expression patterns. Ahmad Mirabadi and Shabnam Sharifian [12] applied the ARM to Iranian Railways data to discover patterns leading to incidents and create management manuals and guidelines. Finally, the method could detect credit card fraud [13]. The Association rules are even included in other algorithms, such as Lamma and other [14] embedded AR, as part of the SLA algorithm.

V.Conditions of the Analysis to Follow

Considered methods are applied to the particular dataset for extraction association rules. It means that if the initial dataset is biased, the found association rules will also have bias.

Moreover, as you will see in the following sections, ARM requires settled initial (apriori) hyperparameters that influence the number of found rules. According to mentioned studies above, there is no standard practice to calculate the metrics, and usually, it comes from the business perspective and domain expert knowledge. During the study, we considered various combinations of rules to find a balance between the number of rules and the reasonability in order to find the most appropriate set of rules.

During the study, we worked with two hyperparameters: support and confidence (see definitions in the next section). We began with a support level of 0.5, increasing by 0.1 while reaching 1.0. We chose 0.5, which means a rule is true for 50% of cases. We obtained an itemset with confidence levels from 0.1 to 1.0 with increments of 0.1 for each new support. Each obtained dataset was estimated among the following questions:

- How many association rules are found out?
- Does an entirely differential rule in the top 100 rules disappear compared with the previous values of hyperparameters?

We stopped the process when we obtained a set with completely differential rules at the top of the list.

VI. Details of the Proposed Solution

A. Association Rule Mining

The problem of discovering association rules was proposed by Agrawal et al. [15]. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of m items. Let T be a set of transaction $\{t_1, t_2, \dots, t_n\}$, where each t_i is set of items in which $t_i \subseteq I$. Association rules are implication rules:

$$A \Rightarrow B,$$

which is interpreted as "if A, then B". The following statements must be met: $A \subset I, B \subset I$ and $A \cap B = \emptyset$. The A term is an antecedent of the rule. The B term is a consequent of the rule.

The number of rules might be huge, so we need some mechanism for selecting strong rules from weak ones. To do that, let's postulate the following hyperparameters:

- Confidence is a measure that counts how many transactions in T that contain A also contain B. It is the probability of B being true when we already know that A is true:

$$Confidence(A \Rightarrow B) = \frac{Occurence\ of\ A\ and\ B}{Occurences\ of\ A}$$

- Support is a measure of the frequency of the transaction patterns that occur in the T:

$$Support(A \Rightarrow B) = \frac{Occurence\ of\ A\ and\ B}{Total\ transaction\ in\ T}$$

- Lift is a value that gives us information about the increase in the probability of the "then" (consequent) given the "if" (antecedent) part. If the lift equals one, we consider there are no dependencies, but if the lift is more than one, we can consider a dependency. Additionally, the lift can demonstrate the "power" of dependency: the larger the lift, the stronger the rule.

$$Lift(A \Rightarrow B) = \frac{Support(A \Rightarrow B)}{Support(A) * Support(B)}$$

Now we can define the minimal support and confidence values to select strong rules. The rules which have confidence more than the selected minimal value are called strong rules.

B. Apriori Algorithm

The Apriori, proposed by Agrawal et al. in [15], is an algorithm for discovering association rules. The algorithm is based on searching frequent itemsets. It assumes that if rule X has a confidence level of C and $X \subset Y$, so rule Y has a confidence level not less than X. In this way, we can dramatically reduce calculations by excluding many weak rules from consideration based on the frequency of every single *i* in *I*.

VII. Analysis

A. Input data

To discover association rules, we used the survey result conducted in 2020 [16]. After data cleaning, the dataset has 324 answers, which will be treated as a transaction. To describe a project context, we asked respondents about the following:

- project size;
- project domain;
- company type (IT-outstaff, IT-outsource, IT product, non-IT);
- company size;
- class of the developed system (business software, embedded software, scientific, etc.);
- belonging to the co-located or distributed team;
- role in the project;
- years of experience;
- passing certification in the chosen role;
- using adaptive, hybrid, or predictive ways of working on the project;
- project category (developing from scratch, reengineering, product or platform customization, etc.);
- involving in different Types of BA activities.

The dataset is available at the link https://data.mendeley.com/datasets/svzv7rs279.

Together the answer's options produced 96 possible items in the itemset.

Before running the apriori algorithm, we discovered the support (frequency) of single items of elicitation techniques. We decided not to consider items (and consequently rules) with a frequency less than 50% (Table 1).

The apriori algorithm was launched across the dataset with the following hyperparameters: minimal support 0.5 and minimal confidence 0.8. After removing autogenerated rules with empty antecedents, there were left 86 association rules.

Table I.
Elicitation techniques with a frequency of more than 50%

Elicitation Technique	Support level %
Interviews	87.3
Document analysis	85.5
Interface analysis	71.3
Brainstorming	69.2
Process analysis/modeling	66.1
Prototyping	66.1
Business rules analysis	54.4

The first look at consequent showed that only two techniques have strong antecedents: Document analysis and Interviews. It means that despite the frequency of other consequents, there is not a strong enough implication between any project context aspects under interest and the consequent itself. Perhaps, the choice of rest elicitation techniques is managed by factors that lay off the considered dataset.

Remarkable that both mentioned methods are often used in pairs. Rule "Document analysis → Interviews" has one of the biggest (0.77) support levels and similar "Interviews → Document analysis". This fact makes sense: a business analyst uses different sources of information due to business analysis information elicitation. Usually, documents and people are the most valuable and accessible sources.

B. Document analysis association rules

First, some rules state implications based on other elicitation methods presented in Table 2.

Table II.
Document analysis association rules

Association Rule	Support level %
Interface analysis → Document analysis	0.65
Process analysis & modeling → Document analysis	0.6
Brainstorming → Document analysis	0.6
Prototyping → Document analysis	0.58

Also, a small subset of rules combines different elicitation methods and another aspect of the project context. For example (here and further, the number in parentheses is a support level): (Business software, Interviews) → Document analysis (0.69), (Interviews, BA Role) → Document analysis (0.69), (Business software, Interviews, BA Role) → Document analysis (0.62), (Interface analysis, Role: BA) → Document analysis (0.58), (Interface analysis, Interviews) → Document analysis (0.58). But these rules have support levels smaller than in rules without other components.

Consider other strongest association rules in this group. Remarkable that BA's role in the project implicates using Document analysis: BA Role → Document analysis (0.76). And the rule includes the class of the system under interest: Business software → Document analysis also has a high (0.74) support level. The situation with mixed rules for role and class system is the same as for mixed rules of elicitation techniques: they have more minor support levels and confidence than the short version. For example, Business software, Role: BA → Document analysis (0.67), Role: BA, Requirements analysis and design definition 0.58

Behind the discovered rules, one more group influences the choice of elicitation techniques. The rule with the strongest support level is (Requirements analysis and design definition, Elicitation & Collaboration) → Document analysis (0.57)

C. Interviews association rules

The Interview's association rules are presented in table 3.

TABLE III.
INTERVIEW ASSOCIATION RULES

Association Rule	**Support level %**
Business software → Interviews	0.77
BA Role → Interviews	0.76
Elicitation & Collaboration → Interviews	0.63
Interface analysis → Interviews	0.63
Brainstorming → Interviews	0.62
Process analysis & modeling → Interviews	0.60
Team distributed → Interviews	0.55

As well as for the previous group, there are many more complex rules with three and more antecedents. However, the support level of these rules is less than the listed above, while their confidence level stays the same. Several examples illustrate the thesis: (Business software, BA Role, Document analysis) → Interviews (0.62), (Requirements analysis and design definition, Elicitation & Collaboration) → Interviews (0.57), (BA Role, Requirements analysis and design definition) → Interviews (0.57), (Business software, Requirements analysis and design definition, Elicitation & Collaboration) → Interviews (0.51), (Business software, Document analysis, Process analysis & modeling) → Interviews (0.5)

That could mean that a significant and essential implication in choosing the elicitation technique is laid out in less complex rules. Remarkable that here we can observe rules that postulate implications based on another elicitation technique, such as Interface analysis and Brainstorming.

VIII. CONCLUSION

We analyzed datasets obtained from the survey. The dataset includes 324 transactions containing items from itemset with 96 items. The apriori algorithm was used for discovering association rules. The algorithm's hyperparameters were defined as minimal support equals 0.5 and minimal confidence equals 0.8. We considered only rules with left bigger than 1. The algorithm discovered 86 associated rules.

The most frequently used elicitation techniques are Interviews, Document analysis, Brainstorming, Process analysis and modeling, Prototyping, and Business rules analysis.

The main discovering facts and rules are:

- Among all frequent rules, only two techniques - Document analysis and Interviews- form strong association rules with project context.
- Interviews and Document analysis are used together pretty often.
- Class of developing system (business software) and BA role and BA activity make using Document Analysis elicitation technique.
- Class of developing system (business software) and BA role, distributed team, Process analysis & modeling, and BA activity such as Elicitation and Collaboration and make using Interview technique.
- Some elicitation techniques (Brainstorming, Interface analysis, Process analysis & modeling) implicate using Interview technique.
- The combination class of developing system, role in the project, team distribution, and activity with other aspects of project context have more minor support levels than less complex rules having only one antecedent and could be considered a sub-option.

The following recommendations can be proposed based on found association rules:

- If a person who performs requirements elicitation uses only Document Analysis or only Interview, they might consider Interview or Document Analysis accordingly.
- If a business analyst uses Interface analysis, Process analysis & modeling, Brainstorming, or Prototyping, they might consider Document Analysis as an additional technique;
- If the system under development is business software, then Document analysis and Interview are reasonably chosen;
- If Interface analysis, Process analysis & modeling, or Brainstorming are used, Interview should be considered as an additional technique;
- Interview is a suitable technique in case of a distributed team.

REFERENCES

[1] K. Pohl, "Requirements engineering: fundamentals, principles, and techniques", Springer, New York, USA, 2010, 182 p.

[2] D. Gobov, V. Yanchuk, "Network Analysis Application to Analyze the Activities and Artifacts in the Core Business Analysis Cycle," *2021 2nd International Informatics and Software Engineering Conference (IISEC)*, Ankara, Turkey, 2021, pp. 1-6, doi: 10.1109/IISEC54230.2021.9672373.

[3] D. Gobov, "Practical Study on Software Requirements Specification and Modelling Techniques". International Journal of Computing, 22(1), pp. 78-86, 2023. https://doi.org/10.47839/ijc.22.1.2882.

[4] H. Dafaalla, et al., "Deep Learning Model for Selecting Suitable Requirements Elicitation Techniques, Applied Science, vol. 12 (18), pp. 9060, 2022. https://doi.org/10.3390/app12189060

[5] V. Sharma, S. Rai, A Dev, "A comprehensive study of artificial neural networks." International Journal of Advanced research in computer science and software engineering, vol 2, no. 10, pp. 278-284, 2012

[6] N Darwish, A. Mohamed, A. Abdelghany, "A hybrid machine learning model for selecting suitable requirements elicitation techniques", International Journal of Computer Science and Information Security, vol. 14, no. 6, pp. 1-12, 2016.

[7] I. Bodnarchuk, et al., "Adaptive Method for Assessment and Selection of Software Architecture in Flexible Techniques of Design", IEEE, 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 292-297, 2018. https://doi.org/10.1109/stc-csit.2018.8526620

[8] F. Hujainah, R. B. A. Bakar, M. A. Abdulgabber, "StakeQP: A semi-automated stakeholder quantification and prioritization technique for requirement selection in software system projects", Decision Support Systems, vol. 121, pp. 94-108, 2019. https://doi.org/10.1016/j.dss.2019.04.009

[9] J. Li, et al., "Attributes-based decision making for selection of requirement elicitation techniques using the analytic network process", Mathematical Problems in Engineering, vol. 2020, pp. 1-13, 2020. https://doi.org/10.1155/2020/2156023

[10] G. Castro, et al., "Applying Association Rules to Study Bipolar Disorder and Premenstrual Dysphoric Disorder Comorbidity," 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE), Quebec, QC, Canada, 2018, pp. 1-4. https://doi.org/10.1109/ccece.2018.8447747

[11] C. Creighton, S. Hanash, "Mining gene expression databases for association rules", Bioinformatics, vol. 19., no. 1, pp. 79-86, 2003. https://doi.org/10.1093/bioinformatics/19.1.79

[12] A. Mirabad, S. Sharifian, "Application of association rules in Iranian Railways (RAI) accident data analysis", Safety Science, vol. 48, no. 10, pp. 1427-1435, 2010. https://doi.org/10.1016/j.ssci.2010.06.006

[13] D. Sánchez, et al., "Association rules applied to credit card fraud detection", Expert systems with applications, vol. 36, no. 2, pp. 3630-3640, 2009. https://doi.org/10.1016/j.eswa.2008.02.001

[14] E. Lamma, et al., "Improving the SLA algorithm using association rules", Springer Berlin Heidelberg, AI* IA 2003: Advances in Artificial Intelligence: 8th Congress of the Italian Association for Artificial Intelligence, Pisa, Italy, September 2003. Proceedings 8, pp. 165-175, 2003. https://doi.org/10.1007/978-3-540-39853-0_14

[15] R. Agrawal, et al., "Fast algorithms for mining association rules", Proceeding 20th international conference very large data bases, VLDB, vol. 1215., pp. 487-499, 1994.

[16] D. Gobov, I. Huchenko, “Influence of the software development project context on the requirements elicitation techniques selection”, In: Hu, Z., Petoukhov, S., Dychka, I., He, M. (eds) Advances in Computer Science for Engineering and Education IV. ICCSEEA 2021. Lecture Notes on Data Engineering and Communications Technologies, vol 83. Springer, Cham. https://doi.org/10.1007/978-3-030-80472-5_18.

DOI: 10.15439/2023F4012
ISSN 2300-5963 ACSIS, Vol. 35

Potentials and Challenges of Gamification in Recruiting

Jasmin Zerrer
University of Aalen,
73430 Aalen, Germany
Email: jasmin.zerrer@hs-aalen.de

Ralf-Christian Härting
University of Aalen,
73430 Aalen, Germany
Email: ralf.haerting@hs-aalen.de

Maren Gerst
University of Aalen,
73430 Aalen, Germany
Email: maren.gerst@studmail.htw-aalen.de

***Abstract*—Digitalization is changing many areas of society and has an impact on recruiting processes. Many companies are facing the challenge of making their application processes more attractive for potential candidates. The use of gamification is becoming a new trend, which is aimed particularly at Generation Y and Z, who have grown up in the age of digitalization. The potentials and challenges of gamification are wide-ranging for applicants and companies. The framework of a qualitative empirical study according to Mayring is used to investigate this research topic. A literature review provides current findings of former research studies. The research design is based on interviews with HR experts. The potentials are reflected in a higher employer attractiveness, applicant quality and more efficiently designed processes. At the same time the implementation of gamification involves some challenges, including additional requirements, like digitalization and data protection, deterrent effects and finding the fitting target group.**

***Index Terms*—Gamification, Recruiting, Digitalization, Human Resources, Qualitative Study**

I. Introduction

NEW technologies and the progress of digitalization are important impacts on further development of companies. Due to the Covid-19 crisis an increasing focus on digital processes has been required [1]. Besides the opportunities of Big Data and Artificial Intelligence (AI), companies are facing the challenge of finding qualified employees and trying to avoid follow-up costs of wrong recruiting decisions. To attract attention, companies can invest in traditional campaigns or explore innovative concepts [2]. Gamification offers a possibility for attracting and retaining new employees, as well as increasing their motivation in recruiting processes [3]. It deals with understanding and assessing human behavior. The goal of gamification in recruiting is to bring real processes into a playful context for better selection of suitable candidates [4]. Because companies are already using gamification, research needs to address the potentials and challenges. Based on dynamic developments in research and business, gamification could be established as a significant opportunity for attracting professionals [5]. With the increasing digitalization, the use of gamification will play an important role for companies. Especially for younger generations, this represents an interesting challenge to prove their skills against other applicants. For this reason, this research study examines the potentials and challenges of using gamification in recruiting for applicants and companies.

II. Research Design

The empirical investigation is based on a qualitative research design. Because the topic is very current and there is limited literature available on the potentials and challenges for applicants and the company itself, an inductive method according to Mayring was selected [14]. Based on the aim of investigating the research question a detailed literature review was conducted first. Afterwards, eleven semi-structured expert interviews were executed for the data collection. The data were subsequently transcribed and coded by using the software MAXQDA. As a result, hypotheses were created, and a hypothesis model was derived. The results are interpreted with the findings from the literature review and finally lead to answer the research question.

III. Literature review

The recruiting process is about identifying and classifying the potential of human resource for the personnel supply and the effective use of this knowledge. Recruiting is a continuous process. In total there are five phases: planning human resources, determining the strategy, evaluation of recruitment sources, implementing recruitment methods and strategies as well as feedback and control. The aim is to identify and hire the most suitable candidate for the advertised position. A well-planned and well-structured recruitment process is important for a company to attract high quality candidates [1].

The term "Gamification" was first used by various players at the end of 2010 [6]. The term is still very controversial and many game designers and user experience designers use other terms, such as "Gameful design" or "Gamefulness" [4]. Today, games have become a growing trend in society. Due to the huge market size that is constantly increasing, non-gaming fields are also trying to take advantage of it. Gamification is therefore not only a good concept to motivate users, but it also brings the health, business, and education sectors closer to the attractive gaming world. Among other things, this can help increase the attractiveness of the company to applicants and make the application process more attractive for potential candidates [7]. Since gamification is a new subject, there is currently no valid definition. However, it can be said that gamification is an interplay of psychology and user experience design [2]. By using specific game mechanisms, it attempts to increase the engagement and enjoyment of participants in each environment. The mechanisms used are directly linked to the provision of a reward. For example, when players complete a task in a specified time, they receive a reward depending on the difficulty level. Gamification is a suitable strategy for influencing and motivating applicants [8]. Due to the close connection between gamification and recruiting, the term “Recrutainment” has been formed. It defines the merging of methods of cognitive assessment, aptitude assessment and gamification elements, which are subsequently embedded in the recruiting process of a company. The goal is to attract more younger applicants and at the same time achieve better candidate results [7].

Gamification is particularly attractive and suitable for the target group Generation Y and Z. Generation Z includes everyone born after 1995 and is also referred to as "digital natives" because they have completely grown up with digitalization [9]. Even in their early years, this generation played on their parents' laptops and smartphones. Of course, dealing with digital communication and media technologies matters for "digital natives" and is an integral part of their everyday lives. In contrast, the previous Generation Y only encountered the new technologies in childhood or adolescence but was able to acquire media skills through the development of social networks, such as Facebook. Both generations differ significantly in their communication behavior, as Generation Z's communication takes place online to an even greater extent than is the case with Generation Y [10]. However, since both generations enjoy using new technologies, gamification is a particularly good recruiting strategy for this target group. As a result, many potential applicants can be attracted to join the company [11].

By implementing gamification as a recruiting tool, the company can test certain skills and abilities, such as creativity, time management or innovative thinking, before a hiring decision is made. This way, unsuitable applicants can be quickly identified and sorted out at the initial stage, speeding up the recruiting process [11]. Before integrating gamification into their own recruiting process, it is advisable for companies to check whether enriching their own application with game-like elements will motivate candidates to further their interest in applying [2]. Gamification in recruiting is based on elements such as high scores, awards, point lists or different game levels. With gamification, the basic needs are addressed, such as receiving a reward, the desire for success, striving to be superior, or to project a certain self-image and increase one's notoriety. Gamified (online-) assessments represent one possibility for personnel selection. In a gamified assessment, applicants are exposed to a game-like environment or a virtual world. For example, the virtual world can resemble a real work environment and employees can be represented by avatars. This is used to elicit job-relevant behavior in situations that may take place in the real work environment. Gamified assessments or serious games do not necessarily have to present a realistic and work-related scenario. Gamification can be integrated into the recruiting process in two ways, either as a one-to-one assessment of applicants or as an extension to existing situational tests by adding game elements. Integrating game elements into the selection process can reduce deception, thereby improving the quality of candidate information and predicting job performance. At the same time, transparency, fun and interaction are promoted. The games can be programmed in two ways. Team based games provide the candidates with the opportunity to interact and compete with other candidates while individual based games can be used for individual evaluation, completing the task alone [10]. Due to the competition in the game, applicants are eager and strive to obtain the maximum score/reward to be achieved in the game so that they quickly advance to the next level. The use of gamification allows companies to attract and engage a variety of applicants in addition to the other types of recruitment. From an HR perspective, evaluating the skills and qualifications of a large pool of applicants following the game is a way to select only the best, who have successfully demonstrated their suitability and value to the company [6]. However, the game elements can also be incorporated into psychometric tests, for example, to test situational judgement to better evaluate candidates' soft skills [3].

When it comes to gamification, there are two different groups among experts: proponents and opponents. Proponents believe that this new tool can help improving the user experience and at the same time offer added value for the practicing company. Gamification seems particularly suitable for internal training, but also as a tool to increase employee motivation. Game thinking can help companies to engage employees more strongly [12]. The potential work performance of an applicant in the intended position can be predicted well by a gamified personnel selection. Game elements and virtual games make it difficult for applicants to pretend or embellish their behavior in the recruiting process. This increases the authenticity of the test participants and contributes to a more reliable prediction of potential job performance. In contrast to other recruiting methods, such as traditional personality tests, gamification allows more targeted statements to be made about the personality and behavior of candidates. This is because traditional personality tests have a high probability of candidates changing their behavior by giving appropriate answers to the recruiters' questions, which are not accurate [8]. Opponents argue that gamification is a currently prevalent trend among many companies. They argue that making a game out of everything is pointless. While people do many things voluntarily and out of a sense of drive, companies use gamification for the purpose of trying to control behavior that was originally intrinsically motivated with extrinsic rewards. In the end, this only achieves the opposite. The extrinsic incentives ensure that intrinsic motivation is undermined, and ultimately activities are only performed because of the prospect of a reward and not out of genuine interest in the company itself. The tricky thing here is that a reward system, like gamification, initially promises to increase engagement. In short term, this is true. Long term, employees will only do a certain activity because of the reward. Therefore, gamification opponents criticize that proponents propagate a "loyalty-for-little-effort" philosophy, which communicates that it would be easy to control the behavior of users at will [12]. Additionally, as shown in the study by J. Koivisto and J. Hamari, the usability of gamification consistently declines with age [13].

IV. Data collection

The data collection includes eleven semi-structured interviews which were conducted between January 2022 and February 2022. All interview partners are from Germany, so the interviews were also done in German language. The experts are representing companies, working with gamification in their recruiting process, HR experts and HR consultants, offering gamification solutions for recruiting processes. It was ensured that participants from different areas were interviewed to get a broader understanding of the situation and to minimize individual bias.

Table 1 provides an overview of all participants. Eleven interviews were conducted, because the minimum number of interviews required for qualitative useful results is at least ten. The qualitative interviews have been selected as research method because they provide the advantage of being able to spontaneously go into more depth on certain topics and questions. This is not possible in a survey. The knowledge gained from the literature review was used to formulate questions regarding the research question. During the formulation of the questions, it was considered that the experts would be able to answer as open and free as possible to obtain a maximum of relevant information. Therefore, the questionnaire contains open, closed and hybrid questions.

The questionnaire was divided into five categories. In the first part, the experts answer the section of "demographic questions" such as gender, age, academic background, and company questions. The second section deals with “the status of digitalization” in the company and then specifically queries for recruiting. The third section contains questions about the “recruiting process”. The focus was on personnel selection and digitalization in the recruiting process. In the section "changes in the recruiting process due to corona pandemic", challenges during the pandemic were addressed. The last section was "gamification in the recruiting process". The practical state of experience regarding gamification in human resources is surveyed here. The experts were interviewed via the platform Zoom and all interviews were digitally recorded. The duration of the interviews was between 40 and 75 minutes.

V. Data analysis

The authors focused on the content of the interviews and not on linguistic aspects. Any dialect was transferred into written language and any personal data were made anonymous. To evaluate the interviews a structured content analysis was used. This enables a more objective view of the data. Following the rules of a qualitative structural content analysis the analysis is reproducible and the intersubjectivity is verifiable. When building categories inductively, certain criteria were defined in advance to structure the coding process. With the coding method of Mayring the gathered data were analyzed, and certain categories were identified [14]. This allows to filter out correlations and commonalities between the different raw interview data and to derive various categories. In the first step, the interviews were compared and examined for relationships and contradictions. In the process of constant comparison sentence by sentence, the authors were searching for patterns to build initial categories. Later, these categories were analyzed in more detail. Throughout the first phase of coding 525 codes were generated to structure

Table I.
Interviewee overview

No.	Academic Background	Professional Field	Industry	Age	Gender
1	Diploma Business Admin.	CEO	Consumer Goods	52	Male
2	Intl. Business	Head of Recruiting	Industrial Solutions	36	Male
3	Diploma Business Admin.	CEO	HR Consulting	52	Male
4	Diploma Business Admin.	Director HR, EU & USA	Automotive Supplier	36	Male
5	Master Management	Head of Group HR	Construction Industry	37	Female
6	Diploma Business Admin.	HR Employer Branding	IT-Services	55	Female
7	Diploma Business Admin.	CEO Managing Director	HR Consulting	49	Male
8	Diploma Psychology	Director Recruiting	Consulting Digitalization	51	Female
9	Diploma Business Admin. and Economics	CEO	HR Consulting	40	Male
10	Master Intl. Management	HR Employer Branding	IT Consulting	36	Female
11	Diploma Psychology	Head of Recruiting	Transport and Logistics	47	Female

the collected data. Each aspect mentioned by the interviewees was taken into account. After coding each given statement, the second step of coding was executed. In the ongoing process of comparatively analyzing data, all the categories were constantly revised and backchecked. Characteristics in one dimension have been deleted if they were too differentiated or added if the level of differentiation should have been increased. In the end, eleven main categories incorporated 26 subcategories. During the inductive category formation, certain criteria were established to structure the coding process. This allows to filter out correlations and commonalities between the different interviews and to derive various categories.

VI. Results

In this section the results of the expert interviews about Gamification of the recruitment process are described. The conceptual model is illustrated by the findings from the expert interviews, followed by the derived hypotheses. All eleven categories which were generated through the content analysis are presented in Figure 1. Each of these categories presents either a potential or a challenge of gamification in recruiting.

Potentials of gamification

Process Efficiency. At the beginning, the implementation of gamification in recruiting is associated with additional effort. Filling the same positions lead to standardization, which minimizes the high expense. In this context, recruiting costs are saved because managers' or specialists' working time is not required for this purpose. Follow-up costs of a wrong hiring can be minimized by gamified approaches. Additionally, the feature of preselection accelerates the process as well. Some experts stated that a faster decision-making contributes to this. According to the interviews, gamification serves the purpose of an application funnel. Mostly, it saves companies from screening many application documents as well as conducting unnecessary interviews. Therefore, the company can prioritize more effectively which candidates are invited for a personal interview.

H1: Gamification positively influences process efficiency in recruiting.

Trustworthy Process. Gamification provides an objective basis for the evaluation. Nearly every expert stated that fairness is included in the consideration of applicants by lowering recruiting bias. The fact that each applicant knows that they must overcome a hurdle increases fairness and transparency. If gamified tests are valid and follow a standardized process, gamification can increase objectivity, due to a higher volume of collected data, which are considered during the evaluation.

H2: Gamification positively influences a trustworthy process in recruiting.

Employer Attractiveness. Most experts stated that gamification enhances the company image. Currently, it offers an opportunity to stand out from the competition through innovation. The gamified tasks create a candidate experience, which has a positive effect on potential applicants. Another influencing factor is the target group-oriented approach through gamified assessments. Consequently, gamification acts as a marketing instrument that increases the number of incoming applicants by making it easier to address them.

H3: Gamification positively influences employer attractiveness in recruiting.

Additional Insights. One potential of gamification is the proof of certain qualifications. The statements made clear reference that skills and hidden talents are acquired independently of oral expression. Regarding the action behavior, concentration and attention span can be tested, which allows conclusions about the psychological security of

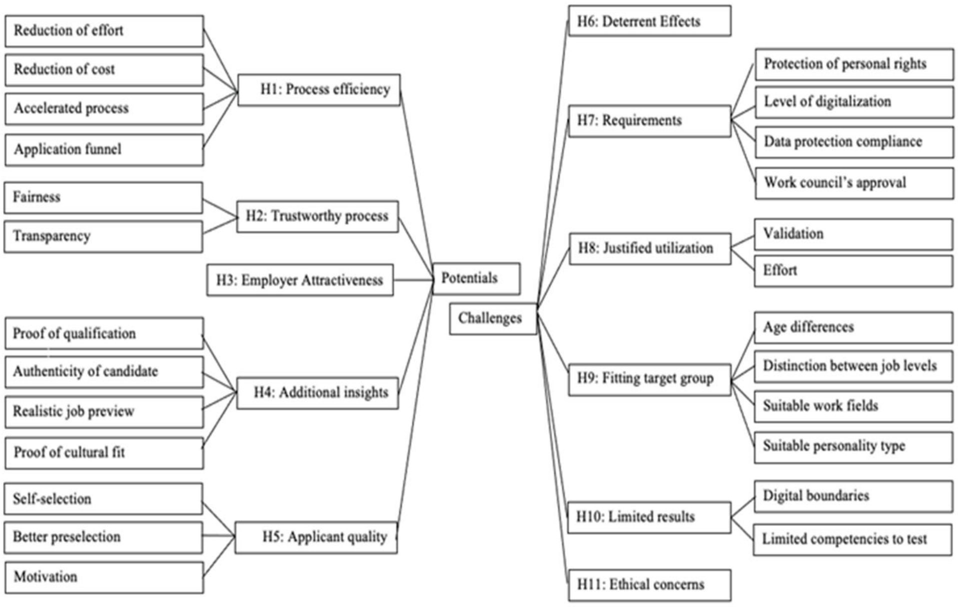


Fig 2. Conceptual Model

the applicant. Gamification creates immersive situations that engage the capacities of the brain. The applicant has no time to think but has to act spontaneously. Overall, the behavior of the applicant is more significant. The interviewees agreed, that gamification provides the packaging for a realistic job preview. Gamification promotes accuracy of expectations by experiencing and testing a workday. The applicant is assured of the job requirements, the workplace, and the company's products. Most experts claim that gamification significantly minimizes the risk of termination after hiring.
H4: Gamification provides additional insights in recruiting.

Applicant Quality. The use of gamification in self-assessment is most effective. The risk of making a wrong decision is reduced for the company and the applicant. The transparency of required skills encourages self-selection. Therefore, most interviewees favored gamification in career orientation to show the applicant which job is suitable. Findings show that gamification is especially suitable at the beginning of the recruiting process. Experts agree that gamification can select qualifications in advance. In the interviews it was mentioned that gamification increases motivation through various factors. When used as a marketing tool, it increases motivation to apply. Furthermore, motivation is raised through active engagement in the game. Most experts agreed that motivation is also achieved through a target group-oriented approach.
H5: Gamification positively influences applicant quality.

Challenges of gamification

Deterrent Effects. A common issue was the possible deterrent effect of gamification on applicants. It was frequently cited that gamification represents an additional hurdle for applicants and that their willingness to make an additional effort could be limited. Given the ongoing shortage of skilled specialists the loss of suitable candidates could be an excessive risk for some companies. Moreover, reservations regarding the scientific respectability of gamification or fear of failing could present further deterrent effects for some candidates.
H6: Deterrent effects hamper the use of gamification in recruiting.

Requirements. Almost all interviewees agree that data protection regulations result in a high level of requirements for gamification in recruiting. This includes clear communication towards the candidates as well as communication with colleagues from intertwined departments. In addition, a certain level of digitization may be required for the implementation of gamification in recruiting. This is imperative for integrating the collected data in systems and efficiently manage high numbers of participants. Additionally, gamification needs to uphold personal rights, which includes a clear communication to candidates. Most interviewees suspected that getting approval of the work council represents a challenge. Therefore, the work council should be involved early in the decision-making process. For this reason, the introduction of gamification in smaller companies without work councils might be easier and more open towards innovation.
H7: High requirements hamper the use of gamification in recruiting.

Justified Utilization. All interviewees agree that the use of gamification needs to be justified by a valid utilization. Foremost in this regard is the scientific validation of the gamified test as well as validating the method through connecting the gamified test to future work elements. Additionally, the use of gamification needs to truthfully reflect the organizational structure and overall modernness to paint a valid picture. Moreover, all interviewees agreed that gamification takes a lot of effort and involves high costs. To justify this effort, many interviewees suspect that gamification is only worth-while, when a certain number of candidates take part in the process. Therefore, to justify gamification in recruiting the use needs to be valid and the effort put in needs to be reasonable for organizations.
H8: Justification of utilization hampers the use of gamification in recruiting.

Fitting Target Group. Due to a higher gaming affinity and being more comfortable in a digital setting gamification might be more fitting for a younger generation of candidates, that grew up in a digital age. Questioning the fit of gamification for all job levels, most interviewees see more potential in using gamification for recruiting junior positions. Only few interviewees state that it is not depending on the job level. Lower levels often include more candidates and might be more suitable for gamification from an economic view. In this context, distinctions between the levels need to be considered. Furthermore, gamification might work best in job fields that involve a high level of numerical or technological understanding, like IT. In addition to that, there could be a certain personality type that gamification works better for, involving traits like a preference for gaming, motivation through gamified element, performance, or power orientation. Companies might be challenged by finding such a target group.
H9: Target group fitting hampers the use of gamification in recruiting.

Limited Results. It is feared that with losing one dimension in a digital setting, you only get limited results from digital gamification. Additional insights could be restricted by a loss of facial expressions, gesticulations, senses, or chemistry. Candidates might also get limited insights into the workplace. Less information and technological issues might lead to a bad decision-making on both sides. Therefore, more innovation might be needed. Theoretically gamification can be used to test methodical, professional, or social competencies. Most interviewees agree that not all these competencies can be adequately tested with gamified elements. Especially challenging might be testing social skills in a gamified setting.
H10: Limited results hamper the use of gamification in recruiting.

Ethical Concerns. In the interviews, ethical questions for gamification, like ageism or ableism, were discussed. Values and attitudes of the gamification developer as well as selected courses of action within gamified elements, could result in underlying developer bias. In addition, algorithms that make decisions autonomously could be classified as unethical. Therefore, a challenge of gamification in recruiting is ensuring ethical utilization.

H11: Ethical concerns hamper the use of gamification in recruiting.

VII. Conclusion and limitations

Implementing gamification in recruiting processes provide potentials and challenges for companies. For a successful implementation it is necessary to fulfill a multitude of legal, organizational, and systematic requirements. Furthermore, it is essential to ensure a certain level of digitalization. The implementation takes a lot of effort, in addition to time involving high costs. To justify the effort, it must be secured that gamified tests are scientifically valid and designed for a specific target group. Regarding the economic view, gamification is appropriate for selecting candidates from a large number of applicants. When developing gamified tests for recruiting processes, it must be ensured that underlying developer bias is excluded. In general gamification might be most suitable for younger applicants, who have a higher gaming affinity. This conclusion was also made by the study of J. Koivistro and J. Hamari [13]. Through gamification both sides involved in the recruiting process can benefit of the added information given about the applicant and the vacant position. This leads to a better fit and a higher quality of suitable candidates and simultaneously reduces fluctuation. In addition, this promotes the opportunity for better self-selection by candidates. If gamification has already been implemented in recruiting, even time savings can be generated, due to automated processes, accelerated decision making and a better preselection. Gamification can increase objectivity, if gamified tests are valid and follow a standardized process, due to a higher volume of data collected during the evaluation. Gamification in recruiting processes can be used as a tool to increase the motivation of applicants, through active engagement and a gamified way to convey content alongside the recruiting process. A model of the potentials and challenges of gamification in recruiting was developed, based on empirical data from German-speaking experts using the content analysis according to Mayring. These generated data show some important influencing factors like process efficiency, data protection, target groups, expenditure of cost and time, as well as the quality of applicants. Experts in HR and gamification mentioned all these factors repeatedly. Therefore, they can be considered as a good basis for the model. To extend the current scientific view of potentials and challenges of gamification in recruiting, current researchers can use these results additionally. There are also some practical implications that should be considered. Companies can benefit from this research by evaluating their level of digitization and their personal fit for gamification in their own recruiting processes. Besides the view of gamification potentials, this research demonstrates equally challenging aspects of gamification, which can have a negative impact on the recruiting processes of companies. Due to the fact, that this qualitative research only focuses on a small sample of experts, there are some limitations, which must be considered. This sample includes different perspectives from experts in different industries and was created to provide reliable information. To prove this qualitative method, a model validation with a quantitative approach is necessary. The influencing factors, which have been identified in this research could be an appropriate starting point for this purpose. Evaluating this model in more countries and focusing on varying aspects would be a great opportunity for future research. Further, a differentiated investigation of single German-speaking states would be interesting as well as a comparison to other counties in the EU or internationally.

References

[1] R. Härting, K. Bilge, L. Fleischer, N. Landgraf, F. Özcakir and M. Wicher. 2021. “Impact of the COVID-19 Crisis on Digital Business Models—Contactless Payments”, in Smart innovation, systems and technologies pp. 143–153. Springer Nature. https://doi.org/10.1007/978-981-16-2994-5_12.

[2] J. Diercks and K. Kupka. 2013. “Recrutainment”, Springer, Wiesbaden. https://doi.org/10.1007/978-3-658-01570-1.

[3] D. M. Küpper, K. Klein and F. Völckner. 2021. “Gamifying employer branding: An integrating framework and research propositions for a new HRM approach in the digitized economy”, Human Resource Management. Review, 31(1),100686. https://doi.org/10.1016/j.hrmr.2019.04.002.

[4] S. Dale. 2014. “Gamification: Making work fun, or making fun of work?”, Business Information Review, 31(2), 82–90. https://doi.org/10.1177/0266382114538350.

[5] G. H. Lowman. 2016. “Moving Beyond Identification: Using Gamification To Attract and Retain Talent. Industrial and Organizational Psychology”, 9(3), 677–682. https://doi.org/10.1017/iop.2016.70.

[6] S. Stieglitz, C. Lattemann, S. Robra-Bissantz, R. Zarnekow and T. Brockmann. 2017. Gamification. Springer International Publishing. https://doi.org/10.1007/978-3-319-45557-0.

[7] O. Korn, F. Brenner, J. Börsing, F. Lalli, M. Mattmüller and A. Müller. 2017. „Defining Recrutainment: A Model and a Survey on the Gamification of Recruiting and Human Resources”, in Advances in intelligent systems and computing (S. 37–49). Springer Nature. https://doi.org/10.1007/978-3-319-60486-2_4.

[8] T. Reiners and L. C. Wood. 2015. “Gamification in Education and Business“. Springer eBooks. https://doi.org/10.1007/978-3-319-10208-5.

[9] G. Hesse and R. Mattmüller. 2019. „Perspektivwechsel im Employer Branding“, Springer eBooks. https://doi.org/10.1007/978-3-658-26208-2.

[10] S. C. Woods, S. Ahmed, I. E. Nikolaou, A. C. Costa and N. Anderson. 2020. “Personnel selection in the digital age: a review of validity and applicant reactions, and future research challenges”, European Journal of Work and Organizational Psychology, 29(1), 64–77. https://doi.org/10.1080/1359432x.2019.1681401.

[11] S. Strahringer and C. Leyh. 2017. „Gamification und Serious Games: Grundlagen, Vorgehen und Anwendungen“ Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-16742-4.

[12] P. Durai. 2010. „Human Resource Management“. Pearson Education.

[13] J. Koivisto and J. Hamari, "Demographic differences in perceived benefits from gamification", Comput. Hum. Behav., vol. 35, pp. 179-188, 2014/06/01/ 2014, https://doi.org/10.1016/j.chb.2014.03.007.

[14] P.Mayring. 2015. „Qualitative Inhaltsanalyse: Grundlagen und Techniken“, (twelfth edition). Beltz, Weinheim and Basel, Germany.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 995–999
DOI: 10.15439/2023F2610
ISSN 2300-5963 ACSIS, Vol. 35

Recognition of Weeds in Cornfields

Gábor Hartyányi
ORCID ID: 0009-0002-0524-0495
Image Processing Research Laboratory
University of Pannonia, 8200 Veszprém
Egyetem u. 10, Hungary
also with Axial Ltd, 6500 Baja
Email: hartyanyi.gabor@gmail.com

László Czúni
ORCID ID: 0000-0001-7667-9513
Image Processing Research Laboratory
University of Pannonia, 8200 Veszprém
Egyetem u. 10, Hungary
Email: czuni.laszlo@mik.uni-pannon.hu

***Abstract*—In terms of weed control, existing precision spraying solutions seek to reduce the unwanted impact of spraying by separate field scanning from mostly birds' eye view. In our study, we propose a hybrid approach in which the mechanical hoeing and the spraying is done simultaneously accomplished by weed recognition from a lower position where the plants' leaves do not cover weeds. We demonstrate the line and the weed recognition methods on a dataset collected from corn fields and compare different convolutional neural networks. We also investigate the feasibility on two widely known embedded platforms.**

I. Introduction

WHILE the population of Europe and world is still increasing the possible incorporation of new lands to agriculture is very limited. Weed is a major factor to limit crop yields, beside their physical clearing, spraying is a main general solution to this problem. However, the EU Green Deal agreement [1] aims 50% reduction of applied chemicals, thus conventional spraying techniques should be revised. In traditional, large field applications whole areas are being sprayed resulting in wasted chemicals on healthy and intact plants and on bare soil. The utilization of machine vision techniques for the detection of weed has been a target for decades, a good overview of different approaches can be found in [2]. Computer vision techniques have to face lots of problems if applied on the field. The leaves of weeds and crops often overlap each other at late growth stages, especially if images are taken from above, making them indistinguishable. Additionally, the plant's leaves may be obscured or damaged by unwanted material including dead leaves or clay, making identification difficult. Maize is the most produced grain in the world, with more than 1.2 billion tons produced in 2021. In Europe, the area affected by corn cultivation was approximately 20 million hectares [3]. Thus, the development of solutions for corn alone can have a significant impact on environmental protection, and the various techniques can be adapted to other crops as well. The main contributions of our article are: We are proposing a hybrid approach combining hoeing and spraying. Cameras, fixed to the cultivator, are to capture the areas between the stems of corn (or between lines); convolutional neural networks (CNNs) recognize weeds and spraying is concentrated only on those areas near stems. Hoeing is made between the lines simultaneously, there is no need for multiple scanning of the fields. Fig. 1 illustrates the region of interest (ROI) areas for possible spraying. We introduce a new free annotated dataset of images of corn lines. Binary labels indicate the presence of weeds. We investigate the use of a popular scalable DNN (EfficientNet [18]) and less complex CNNs for weed recognition. The feasibility on micro-controllers is also part of our study. To narrow down the target area for spraying the physical setup is calibrated with image homography.

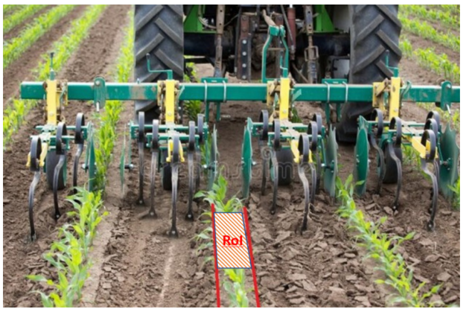


Fig. 1: Corn fields being hoed and only the ROI areas to be sprayed where weed is detected.

II. Literature Overview

Traditional approaches typically consist of four main steps: pre-processing, segmentation, feature extraction, and classification. Pre-processing tries to "standardize" global image properties, while segmentation is to separate vegetation from background. Since weed and crops have similar properties it is difficult to find the proper features and their representation to achieve the best possible classification. Four feature categories can be identified in papers, such as spectral features (mean and standard deviation of RGB, HSV, and chlorophyll vegetation index values), textural features, morphological features, and spatial contexts.

For textural features common techniques can be utilized, for example in [4] single-level Haar discrete wavelet decomposition was used to obtain four sub-images (approximate image,

We acknowledge the financial support of the Hungarian Scientific Research Fund grant OTKA K-135729. We are grateful to the NVIDIA corporation for supporting our research with GPUs obtained by the NVIDIA Hardware Grant Program.

vertical details, horizontal details, and diagonal details), and then gray level co-occurrence matrices were extracted from these sub-images for weed detection. In [5] Gabor filters were applied but also the co-occurrence matrix features were finally calculated. The shape of leaves can be very characteristic for recognition and dozens of such traditional descriptors have already been utilized for weed recognition: eccentricity, circularity, convexity, elongatedness, invariant moments, just to mention a few. The above mentioned approaches did not consider the spatial context, while the sowing pattern of crops is typically very specific: they are sowed or planted in almost straight lines thus the spatial contexts or position information could help to improve the recognition process. It is natural to think of the variants of the Hough transformation, linear regression, the vanishing point, or the frequency analysis of the lines or repetitive patterns. More sophisticated approaches (such as [6] which uses dynamic programming and energy optimization) can also handle curved lines. However, applying strict assumptions about crop positions can result in false detections. In [7] the upper limit of detection accuracy was investigated when using information about sowing geometry and positions. The uncertainty in real crop positions and the disturbing effect of weeds can have a significant effect on detection accuracy, thus complex solutions are required.

Beside RGB cameras, special sensors such as depth cameras can also be used for weed recognition. For example in [8], beside color, position, and texture features also depth features, obtained by a special RGB-D camera, were also utilized to recognize weeds in wheat fields. The AdaBoost algorithm was employed for the integrated learning of multiple classifiers. Experimental results showed accuracy between 81 and 88%, depending on the growth phase of wheat (the different experiments used 50-600 images).

Considering the theoretical and practical problems of the above specified four main steps, there is no surprise for the breakthrough of deep learning methods. As an early attempt to overcome the weak generalization ability of manually designed features [9] used K-means clustering to construct a feature dictionary, fed to a single-layer network, to create an identification model. The approach in [10] can be considered as a hybrid solution where both hand crafted features (requiring segmentation) and DNN features, generated by a pre-trained GoogLeNet [11] network, were used with four kinds of clustering methods. Interestingly, the technique was used to cluster four kinds of weeds but the number of test images were much below a thousand. In [12] an embedded system on a UAV was introduced utilizing the YOLOV3-tiny network to detect the pixel coordinates of weeds in images. The mean Average Precision (mAP) was 72.5% at 2FPS on a mobile device. The average positioning error was 10.31 cm. Tests were carried out on a total of 2000 images, taken at 2 m high, of winter wheat with 5 types of weeds. The most relevant paper from our point of view is [13] where a classification approach of Zea mays L. (corn), narrow-leaf weeds, and broadleaf weeds from multi-plant images are presented. Compared to previously discussed articles, a large image dataset was generated: 13,000 recordings were made in natural field conditions, at different locations and at different stages of plant growth. The ROIs were detected using connected component analysis, whereas the classification was based on VGG [14] and Xception [15] CNNs (and alternatively by SVMs). The best method for weed classification, at early stages of growth and in natural corn field environments, was the CNN-based approach, as indicated by the 97% accuracy obtained.

For the reader interested in hand-designed feature methods we propose to read the review of Wang et al. in [2] while for more recent DNN approaches go for [16].

III. A Maize Image Dataset

Contrary to hand-designed approaches machine learning methods, especially DNNs, don't require much pre-processing but large datasets with enough generality are a must. While some articles were trying to recognize the different types of weeds (e.g. [10]) or tried to increase the variety of viewpoints (f.e. [13]) we have different purposes: Since between the lines of corn hoeing is made, weeds are to be detected (and sprayed if found) only between maize stems. Weed types are out of interest and three types of images are to be classified: only weed, only maize, and weed and maize. The cameras can be placed on the cultivators approximately 25 cm high and 25 cm laterally from the corn row, with the optical axis of 45 degrees to the ground plane.

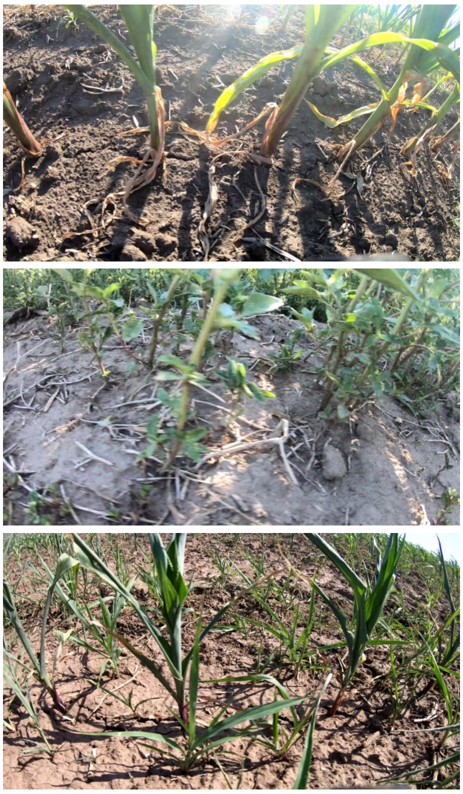


Fig. 2: Example images of the "Corn and Weed" dataset. Top: clear maize. Middle: only weeds. Bottom: maize and weeds.

Images of our publicly available dataset (downloadable at https://keplab.mik.uni-pannon.hu/images/caw/) were made in a second-sown corn field. Sowing time was late May - early June 2022. The average row spacing was usually 70 cm, while the distance between the stems was on average 25 cm. The height of the plants varied between 20-50 cm depending on the nutrient and water supply of the area. The shots were made with a GoPro 7 camera at 2704 × 1520 resolution and at an average speed of 4 km/h, with different corn line orientations. The dataset contains 816 images with only weed, 1231 images with only corn, and 1796 images with corn and weed. Original images are downscaled to 640 × 480, example photos are in Fig. 2.

IV. Detection of Corn Lines and Safety Margin Area

For the most accurate localization of the ROI we make the following steps: First we segmented corn stems. For the instance segmentation of corn stems we used Mask-R CNN [17] pre-trained on the COCO dataset. For transfer learning with two classes (background and corn stem) 50 images (with circa 200 corn plants) were manually annotated. We used stem bottom endings as the lowest points of Mask-R CNN masks to fit lines with linear regression. ROI was set with planar homography (see Subsection IV-A).

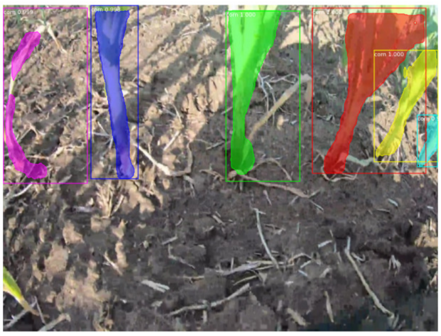


Fig. 3: Segmented maize stems with Mask R-CNN.

A. Planar Homography for ROI Designation

To find the border lines of the ROI the size of the safety margin should be considered. In our layout 10 cms were given on both sides of corn lines. To determine the border lines in the image space we computed the homography matrix with the help of ArUco markers. By applying plain homography we assumed the smoothness of the ground (the relative pose of the camera plane and soil at the stem endings is constant). Naturally, this is not always true but considering the spread of the spray we accepted the resulting inaccuracy. The result is illustrated in Fig. 4.

Starting from our initial dataset now we arrived to a smaller set: there are only two labels (weed free and with weed) and to avoid a very unbalanced configuration the number of weed free images were limited. Tab. I gives the number of images per category in our experiments.

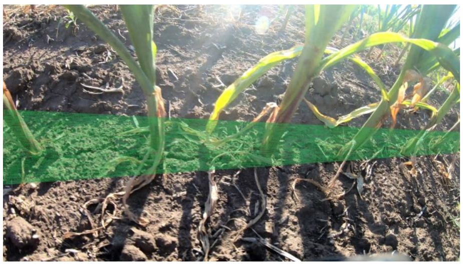


Fig. 4: ROI defined by stem endings and homography of safety margins.

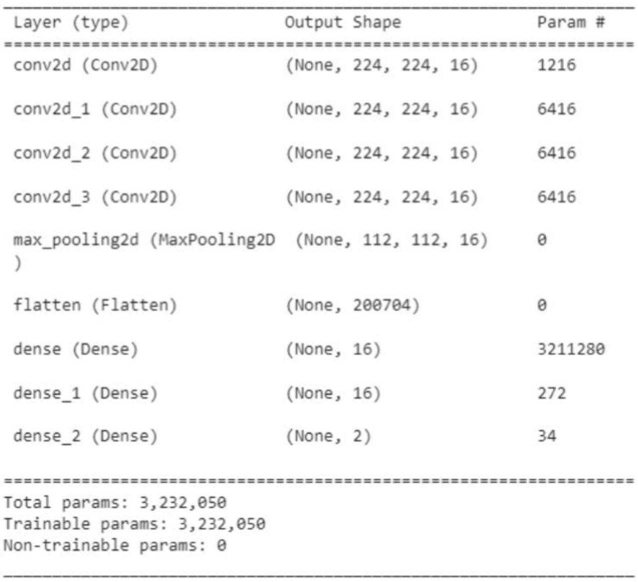
TABLE I: The ROI based dataset used in experiments.

	Weed free	With weed	Total	Percentage
Training images	851	808	1659	72%
Validation images	230	209	439	19%
Test images	59	160	219	9%
Total	1140	1177	2317	100%

V. Comparison of Different Weed Recognition Models

Assuming approximately 15 km/h average speed of the cultivator, circa 70° viewing angle, and 10% overlapping of images at least 4 FPS processing speed should be reached. There are two main purposes of the following experiments: First, to investigate the effect of masking: what happens if the whole area (i.e. the context) is considered during the classification at the ROI. Second, to find the limit to minimize the complexity of the applied CNNs so to increase the processing speed without a painful degradation of accuracy. In 2019, Google Brain published the open source EfficientNet [18] network family for image classification. The members of the family are the differently scaled versions (from B0 to B7) of the base model, B7 being the largest variant achieving state-of-the-art Top-1 accuracy on ImageNet in 2019. It was created with a compound scaling method to scale the depth (number of layers), the width (number of kernels in a layer), and resolution (size of input image) of an existing model and a baseline network with fine-tuned layers, in a balanced manner, to consider the computation limits. We used the ImageNet pre-trained B0 version without the top classification parts after adding two dense hidden layers with 512 and 128 neurons and two output neurons.

Tab. III compares results showing almost perfect classification accuracy on both masked and whole area images. Thus our next step was to create CNNs with decreasing number of parameters to reach the smallest size without a significant drop in accuracy. We started with a network (named CNN 2) specified in Fig. 5 and then decreased the number of convolutional blocks and dense layers as given in Tab. II. Each convolutional block had 16 filters of size 5 × 5, all images are downscaled to 224 × 224. As given in Tab. III the experiments showed that the information from the context could help the classification accuracy (or there is a strong correlation in the presence of weeds between the lines and between the neighboring stems in the ROI area). While we can see a decreasing trend in accuracy from CNN 2 to CNN 5, the reduction of number of parameters is not significant (see Tab. II). Thus we made further variants of CNNs: reduced the number of neurons in dense layers and reduced the number of convolutions. In this process we generated 7 models, namely CNN 3.2, 3.3, 3.4, 3.5, 5.2.1, 5.2.2, and 5.3. The number of parameters and accuracy of these networks are visible in Tab. IV, Fig. 6, and Fig. 7. It is clear to see that there is a significant drop in accuracy for CNN 3.5, and halving the number of neurons in the dense layers of CNN 5.2.1 was not a good idea. Many of these simplified networks produces rather good results, the question is their computational power needed.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 16)      1216

 conv2d_1 (Conv2D)           (None, 224, 224, 16)      6416

 conv2d_2 (Conv2D)           (None, 224, 224, 16)      6416

 conv2d_3 (Conv2D)           (None, 224, 224, 16)      6416

 max_pooling2d (MaxPooling2D  (None, 112, 112, 16)     0
 )

 flatten (Flatten)           (None, 200704)            0

 dense (Dense)               (None, 16)                3211280

 dense_1 (Dense)             (None, 16)                272

 dense_2 (Dense)             (None, 2)                 34

=================================================================
Total params: 3,232,050
Trainable params: 3,232,050
Non-trainable params: 0
_________________________________________________________________
```

Fig. 5: The structure of the CNN 2 model.

TABLE II: Different base CNNs included in our study. Each convolutional layer has 16 filters.

Models	# parameters	# dense layers	# conv. layers
EffNetB0	4,779,045	3	
CNN 2	3,232,050	3	4
CNN 3	3,219,218	3	2
CNN 4	3,218,946	2	2
CNN 5	3,212,530	2	1

TABLE III: Accuracy of initial networks on images with/without masking.

	Whole images		**Masked images**		
	Val. acc.	**Test acc.**	**Val. acc.**	**Test acc.**	**Test acc. diff.**
EffNetB0	0,9977	1	0,9977	1	*0*
CNN 2	0,9909	0,9909	0,9658	0,9863	*-0,0046*
CNN 3	0,9932	0,9954	0,9408	0,968	*-0,0274*
CNN 4	0,9954	0,9863	0,9112	0,9452	*-0,0411*
CNN 5	0,9863	0,9954	0,7306	0,9315	*-0,0639*

TABLE IV: The main parameters and accuracy values of CNN 3 and CNN 5 variants.

	filters by layer	neurons in dense layers	params	tr.acc.	val.acc.	test.acc.
CNN3	16	16	3.219M	0.9741	0.9932	0.9954
CNN3.2	16	8	1,613M	0.9542	0.9818	1.0000
CNN3.3	8	8	805,130	0.9367	0.9658	0.9909
CNN3.4	4	4	201,446	0.8993	0.9431	0.9772
CNN3.5	2	2	50,444	0.5130	0.5239	0.2694
CNN 5	16	16	3.122M	0.9554	0.9863	0.9954
CNN 5.2.1	16	8	1,607M	0.5130	0.5239	0.2694
CNN 5.2.2	8	16	1,606M	0.9470	0.9863	1.0000
CNN 5.3	4	16	803,170	0.8981	0.9431	0.9817

VI. Performance on Embedded Systems

All in the previous experiments we used cloud services with massive GPU support which is not very typical in field applications often far from high-bandwidth networks. Luckily there are different embedded system platforms for application developers which could be operated in cultivators.

A. Experiments on the Jetson AGX Xavier Development Platform

The NVIDIA Jetson AGX Xavier Series is an industrial platform for massively parallel computations reaching up to 32 TOPS. We run our tests on a 512 cores Volta architecture with 64 Tensor cores. As given in Tab. V, all models could run at high speed.

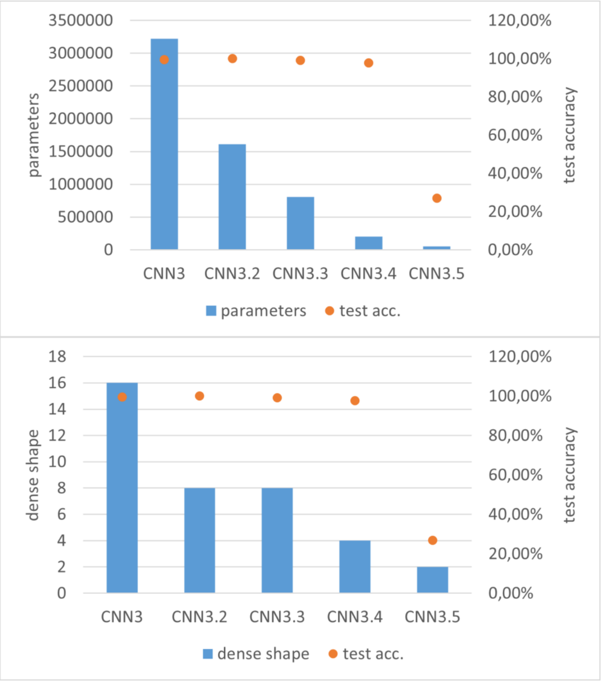


Fig. 6: Top: Accuracy and number of parameters of the CNN models created from model CNN 3. Bottom: Accuracy and shape of the same models.

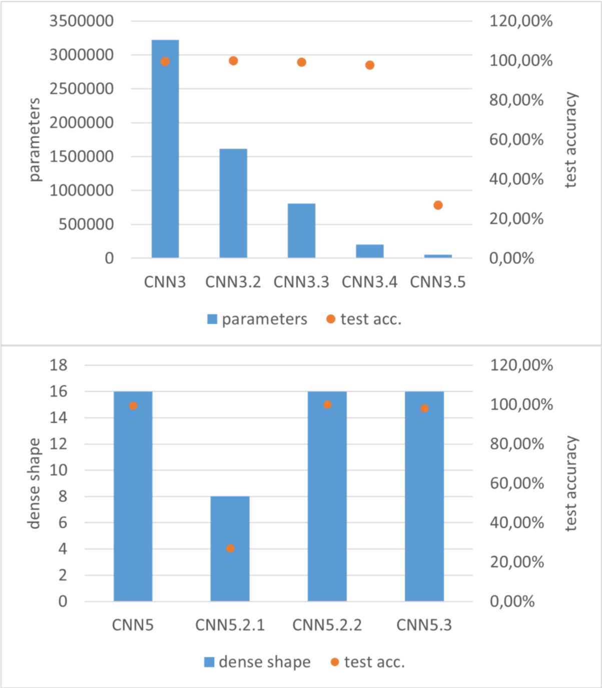


Fig. 7: Top: Accuracy and number of parameters of models created from model CNN 5. Bottom: Accuracy and shape of the same models.

TABLE V: The running performance of different CNN models on the Jetson AGX Xavier platform.

CNN Model	2	3.2	3.3	3.4	3.5	4	5	5.2	5.2.2	5.3
FPS	15.63	28.57	37.04	45.45	33.33	25.0	41.67	47.67	45.45	50

B. Experiments on the STM32 Platform

STMicroelectronics produces different boards built on ARM cores which can be possible platforms for on-field weed recognition. It is possible to test different DNN models in a cloud service of STMicroelectronics at https://stm32ai-cs.st.com/home. Uploaded models can be optimized for speed, for memory usage, or for both. We have chosen the third option to test four of the previous CNN models on two platforms. According to Tab. VI acceptable FPS could be achieved with model CNN 5.3 only on the STM32H735G-DK platform.

TABLE VI: The running performance (FPS) of different CNN models on two specific STM32 boards.

CNN Model:	2	3.2	5.2	5.3
STM32H735G-DK	0.065	0.173	2.15	4.26
STM32F469I-DISCO	0.012	0.032	0.29	0.57

VII. Conclusion

We have outlined a hybrid weed control approach where hoeing is combined with spraying, ensuring that the amount of applied chemicals is very low. We found that the localization of lines can be achieved by Mask-R CNN segmentation of corns, while the recognition of weeds can be done with relatively small size CNNs. Considering the typical speed of cultivators both the NVidia AGX Xavier platform and the STM32H735G-DK board of STMicroelectronics are applicable for recognition. The detection of maize lines with Mask-R CNN on the Xavier platform is for future work as well as capturing new field images under different weather conditions and growth phases of corns. Since corn is the most produced cereal worldwide, our study shows that environmental friendly hybrid approaches can significantly contribute to reaching short term aims of agriculture.

References

[1] Constanze Fetting, “The European Green Deal," ESDN Report, December 2020, ESDN Office, Vienna.

[2] Aichen Wang, Wen Zhang, Xinhua Wei, “A review on weed detection using ground-based machine vision and image processing techniques,” *Computers and Electronics in Agriculture,* vol. 158, 2019, pp. 226–240.

[3] Food and Agriculture Organization of the United Nations, https://www.fao.org/faostat

[4] Bakhshipour, A., Jafari, A., Nassiri, S.M., Zare, D., “Weed segmentation using texture features extracted from wavelet sub-images,” *Biosyst. Eng.,* vol. 157, 2017, pp. 1-–12.

[5] Kumar, D.A., Prema, P., “A novel wrapping curvelet transformation based angular texture pattern (WCTATP) extraction method for weed identification,” *ICTACT J. on Image Video Process.,* vol. 6, no. 3, 2016

[6] García-Santillán, I., Guerrero, J.M., Montalvo, M., Pajares, G., “Curved and straight crop row detection by accumulation of green pixels from images in maize fields.” *Precis. Agric.,* vol. 19, 2018, pp. 18-–41.

[7] Midtiby, H.S., Åstrand, B., Jørgensen, O., Jørgensen, R.N.,“Upper limit for context–based crop classification in robotic weeding applications.” *Biosyst. Eng.* vol. 146, 2016, pp. 183-–192.

[8] Xu, K., Li, H., Cao, W., Zhu, Y., Chen, R., and Ni, J., “Recognition of weeds in wheat fields based on the fusion of RGB images and depth images.” *IEEE Access,* vol. 8, 2020, pp. 110362–110370.

[9] Tang, J., Zhang, Z., Wang, D., Xin, J., He, L., “Research on weeds identification based on K-means feature learning.” *Soft Comput.* vol. 22, 2018, pp. 7649–7658.

[10] Hall, D., Dayoub, F., Kulk, J., McCool, C., “Towards unsupervised weed scouting for agricultural robotics.” *in Robotics and Automation (ICRA),* 2017 IEEE International Conference On. IEEE, pp. 5223-–5230.

[11] Szegedy, C, Liu W., Jia Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., “Going Deeper with Convolutions,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2015, pp. 1—9.

[12] Zhang, R., Wang, C., Hu, X., Liu, Y., and Chen, S., “Weed location and recognition based on UAV imaging and deep learning.” *International Journal of Precision Agricultural Aviation*, 2020, vol. 3, no. 1, pp. 23–29.

[13] Garibaldi-Márquez, F., Flores, G., Mercado-Ravell, D. A., Ramírez-Pedraza, A., and Valentín-Coronado, L. M., “Weed Classification from Natural Corn Field-Multi-Plant Images Based on Shallow and Deep Learning, ” *Sensors,* vol. 22, no. 8, 3021.

[14] Simonyan, K.; Zisserman, A., “Very deep convolutional networks for large-scale image recognition,” *In Proceedings of the 3rd International Conference on Learning Representations,* San Diego, USA, 7–9 May 2015.

[15] Chollet, F. “Xception: Deep learning with depthwise separable convolutions,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, USA, 21-26 July 2017, pp. 1800—1807.

[16] Venkataraju, A., Arumugam, D., Stepan, C., Kiran, R., and Peters, T., “A Review of Machine Learning Techniques for Identifying Weeds in Corn,” *Smart Agricultural Technology,* 2022, 100102.

[17] He, K., Gkioxari, G. , Dollár, P., and Girshick, R., “Mask R-CNN,” *in Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961-–2969.

[18] Tan, M., and Le, Q., “EfficientNet: Rethinking model scaling for convolutional neural networks.” International Conference on Machine Learning, PMLR, 2019, pp. 6105–6114.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1001–1005
DOI: 10.15439/2023F6642
ISSN 2300-5963 ACSIS, Vol. 35

Price-Shaped Optimal Water Reflow in Prosumer Energy Cascade Hydro Plants

Przemysław Ignaciuk, *IEEE Senior Member*
0000−0003−4420−9941
Institute of Information Technology
Lodz University of Technology
Łódź, Poland
przemyslaw.ignaciuk@p.lodz.pl

Michał Morawski
0000−0002−8902−1259
Institute of Information Technology
Lodz University of Technology
Łódź, Poland
michal.morawski@p.lodz.pl

***Abstract*—In the face of the recent surge in energy prices, intensified use of free renewable sources of energy (RSE) gains much importance. Unfortunately, the operation of RSE highly depends on weather conditions, which perturb the balance between the industrial and home energy dissipation patterns. This disparity induces price fluctuations or even destabilizes the energy supply system, yet can be alleviated by the installation of energy depots. While electrochemical depots are hardly cost-effective, they may be supplemented or replaced by small hydro plants with the ponds located above the plant recognized as energy reservoirs. However, inappropriate use of the plant is likely to cause floods or droughts down the river. In this paper, following a rigorous mathematical argument, a cost-optimal controller of a cascade of hydro plants is designed and its properties are formally proved. It is shown to flatten the price pattern, by reducing the load fluctuation of the legacy supply system, as well as provide a concrete revenue for prosumers.**

***Keywords*—hydro plants, green energy, optimal control, networked systems, time-delay systems**

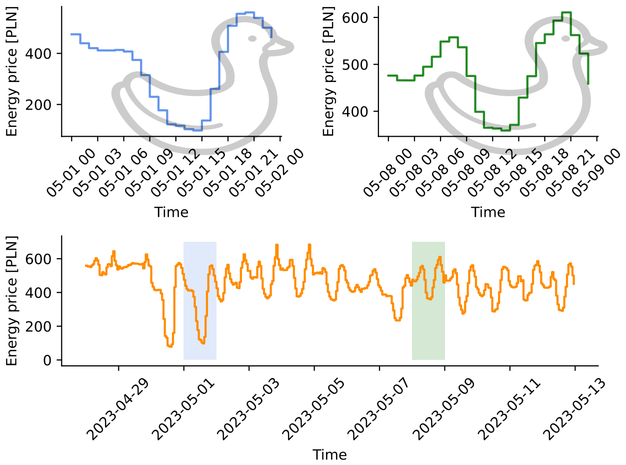


Fig. 1. Fluctuation of energy prices [PLN/MWh] on the Polish market in May 2023 [3] *following 'duck curve'.*

I. Introduction

The energy demand grows day by day, including home users, industry, transportation, building cooling, heating, etc. Unfortunately, the price of electrical energy increases even faster, thwarting domestic budgets. The use of fossil fuels is more and more penalized in Europe, thus the only way to decrease operational costs is to generate energy from renewable sources (RSE), e.g., the sun, wind, waves, geothermal sources, and water flow [1]. Unfortunately, RSEs are capricious in the sense that the amount of retrieved energy heavily depends on weather conditions, whereas the dissipation depends on human activities. These factors oppose each other which leads to substantial price fluctuations. In essence, an RSE generates inexpensive energy around noon and maximum demands (thus high prices) are in the evening when costly fuel-sourced plants need to be engaged. As a result, one obtains the energy price variation resembling the so-called 'duck curve', illustrated in Fig. 1 for a 2-week evolution of the Polish market in May 2023. On May 1st the price in the evening was over 6 times higher than at noon on that day. Meanwhile, on May 8th, this ratio was less than 2, and overall prices have been higher.

A typical business objective from the grid owner's viewpoint is to level the supply-demand disparity – to "behead the duck"" [2] – by decreasing the imbalance in the evening. The only practical way available today to achieve this goal is via energy depots. However, such systems, e.g., pumped hydroelectric storage, or chemical batteries, are expensive to install and later on to maintain. The business objective for plant owners is revenue, thus, despite the government stimulus, the energy depots are introduced infrequently. The problem can be mitigated if different kinds of power plants mutually cooperate [4]. However, it is the domain of commercial plants rather than prosumer ones.

Not all RCSs are susceptible to the influence of weather conditions. A good example is hydro plants [5]. In Poland, there are numerous former mills, currently abandoned, that can be converted into small power plants without significant expenditures from prosumers, i.e., in the same way, the photovoltaic installations have been engaged. To increase the revenue and speed up the return on investment, instead of keeping a constant flow through generators, one may suppress the flow when energy is cheap and boost it when the price is high. Then, the water in the reservoirs located above the plant dams constitutes an energy store. Currently, in Poland, less than 5% of possible installations are used for energy production [6], thus the application area of research presented here is meaningful.

Contrary to broad and deep artificial lakes built on major rivers, the prosumer ponds are relatively small and can be filled up or drained quickly, yet the water supply from upstream reservoirs is subject to delay. To capture this effect, the dynamical model constructed in the work explicitly incorporates the information about different delays among the water flows on the links connecting the reservoirs. Using the system's dynamical representation, an optimization

This work has been performed in the framework of a project "Robust control solutions for multi-channel networked flows" no. 2021/41/B/ST7/00108 financed by the National Science Centre, Poland.

problem is stated and solved analytically. The optimal control law is expressed in an easy-to-implement closed form. The proposed solution brings profits not only to prosumers by increasing their economic gain, but also to the power grid operators by reducing the load changes of standard power plants.

The remaining part of the paper is organized as follows. In Section II, a discrete-time dynamic model of water flow in a multi-hydro-plant system is constructed. It explicitly takes into account different plant characteristics, e.g., capacity, and delays on the conduits linking the reservoirs. Based on the mathematical formulation of system dynamics, an optimization problem is defined and solved. The analytical solution is detailed in Section III and its properties are illustrated in a numerical example presented in Section IV. Conclusions are drawn in Section V.

II. System Model

Usually, the modeling of hydropower plants concentrates on the optimization of the work of generators, leaving the supply system apart [7]. Here, the generator is considered a black box, and the focus is placed on the hydrological aspects of the water plant operation. A key point to consider in a water reflow system is the nonnegligible time between issuing the control action at one reservoir before it influences the water level at a downstream one. Therefore, as opposed to the earlier models of storage networks, e.g., [8], in the approach advocated here, the control principles from time-delay storage systems [9, 10], will be applied. However, the models proposed in [9, 10] assume continuous-time control adjustment, which is difficult to realize in a water control system owing to the specifics of mechanical components steering the dam weirs. The model in this work explicitly covers the effects of finite sample time and will be constructed directly in the discrete-time domain.

A. Single-plant system

Let us consider the model of a single hydro plant illustrated in Fig. 2. The water budget dynamics at the plant will be described via the recursive relation

$$s_j(k+1) = s_j(k) - f_j(k) + \sum_{i \in \text{plants upstream}} f_i(k - T_{ij}) + r_j(k), \qquad (1)$$

where

- $s_j(k)$ is the water volume (water level) of the reservoir near plant j,
- $f_j(k)$ is the amount of water used to drive power generators installed at dam j between time instants k and $k + 1$,
- $r_j(k)$ is the supply from external hydrological sources like rain (and its runoff), melting snow, uncontrolled tributaries, vaporization, etc. The values of $r_j(k)$ can be obtained from the weather forecast and hydrological models within the planning horizon of m periods. $r_j(k)$ is assumed known.

The tributaries supply the pond with water previously used by the plants upstream. The water from upstream plant i arrives at plant j with $T_{ij} > 0$ delay. The period length Δk, i.e., the time between instants k and $k + 1$, can be selected arbitrarily, but according to the pace of price changes, it is reasonable to choose 1 hour (or 15 minutes in the near future). Similarly, the planning horizon m usually covers a 24-hour window of known energy prices (the next-day market). The initial flow $f_j(k \le 0)$ and the initial water level $s_j(0)$ are assumed known. The terminal condition $s_j(m)$ can be selected arbitrarily.

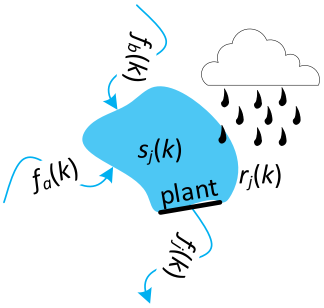


Fig. 2. Model of a single waterpower plant: $f_a(k)$, $f_b(k)$ – water inflow from reservoirs a and b to pond j with the current level $s_j(k)$; $r_j(k)$ – water supply from exogenous sources; $f_j(k)$ – outflow supplying the power generators.

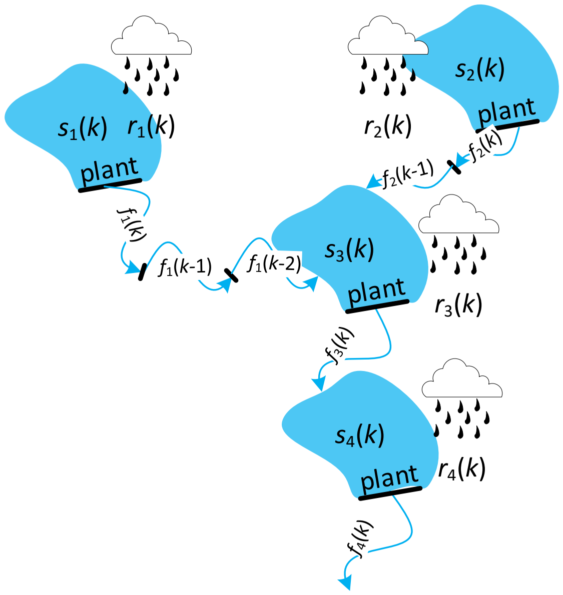


Fig. 3. Model of connected hydro plants. Different canal length inflicts different delay of water reflow between the plants.

The period income from the plant may be calculated as

$$J_j(k) = \eta_j p(k) f_j(k) \Delta k, \qquad (2)$$

where

- $p(k)$ is the energy price at instant k. Usually, it reflects the duck curve, but the actual profile may be subjected to specific local demands.
- η_j is the efficiency of power generators, including the impact of the dam height. For prosumer generators in the lowlands, the flow of 1 m^3/s corresponds to power generation of 5-7 kW.

B. Multi-plant system

In the case of n power plants under common management (an example system with four plants illustrated in Fig. 3), it is convenient to describe the model in a vector form. Let

$$\mathbf{s}(k)=\begin{bmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_n(k) \end{bmatrix}, \mathbf{f}(k)=\begin{bmatrix} f_1(k) \\ f_2(k) \\ \vdots \\ f_n(k) \end{bmatrix}, \mathbf{r}(k)=\begin{bmatrix} r_1(k) \\ r_2(k) \\ \vdots \\ r_n(k) \end{bmatrix}, \tag{3}$$

denote the vector of reservoir water level, the water volume of inter-reservoir flows, and the water volume from exogenous sources, respectively.

The proposed state-state representation is given as

$$\begin{aligned} \mathbf{s}(k+1) &= \mathbf{s}(k)-\mathbf{f}(k)+\sum_{t=1}^{T}\boldsymbol{\Theta}_t\mathbf{f}(k-t)+\mathbf{r}(k) \\ &= \mathbf{s}(k)+\sum_{t=0}^{T}\boldsymbol{\Theta}_t\mathbf{f}(k-t)+\mathbf{r}(k). \end{aligned} \tag{4}$$

Introducing $\mathbf{x}(k)$ as a controlled part of the flow, i.e., $\mathbf{x}(k) = \mathbf{f}(k) - \mathbf{f}^{\mathrm{ref}}$, where $\mathbf{f}^{\mathrm{ref}}$ is the vector of natural flows, the system dynamics becomes

$$\mathbf{s}(k+1)=\mathbf{s}(k)+\sum_{t=0}^{T}\boldsymbol{\Theta}_t\mathbf{x}(k-t)+\mathbf{r}(k), \tag{5}$$

where T is the maximum delay, matrix $\boldsymbol{\Theta}_0 = -\mathbf{I}$, $\mathbf{I}$ being the $n\times n$ identity matrix, and matrices $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_T$ group the information about flow delays,

$$\boldsymbol{\Theta}_t=\left[\theta_{ij}\right]_{n\times n} \tag{6}$$

with $\theta_{ij} = 1$, if the flow from reservoir j reaches reservoir i with delay t, and 0, otherwise. The entries on the main diagonal $\theta_{ii} = 0$. Contrary to [11], here, the distance between plants is nonnegligible. For the example from Fig. 3, the longest delay $T = 3$ (the flow between reservoirs 1 and 3) and

$$\boldsymbol{\Theta}_1=\begin{bmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&1&0 \end{bmatrix}, \boldsymbol{\Theta}_2=\begin{bmatrix} 0&0&0&0\\0&0&0&0\\0&1&0&0\\0&0&0&0 \end{bmatrix}, \boldsymbol{\Theta}_3=\begin{bmatrix} 0&0&0&0\\0&0&0&0\\1&0&0&0\\0&0&0&0 \end{bmatrix}.$$

III. Optimization problem

The optimal values of $\mathbf{x}(k)$ can be obtained from numerical procedures, like [12-16]. However, here, we present an analytical solution, based on the theory of optimal control systems. The obtained closed form is amenable to physical interpretation and can be directly and efficiently implemented in dam control systems.

A. Problem statement

With the initial water level $\mathbf{s}(0)$ and initial flow $\mathbf{x}(k \leq 0)$, the task is to reach level $\mathbf{s}(m)$ within m periods so that imposed cost criteria are fulfilled. Formally, the optimization problem may be stated as

$$\max_{\mathbf{x}(k)} J_E(p(k),\mathbf{x}(k))=\frac{1}{2}\sum_{k=0}^{m-1}p(k)\mathbf{x}'(k)\mathbf{N}\mathbf{x}(k), \tag{7}$$

where $\mathbf{N} = diag\{\eta_1, \eta_2, \ldots, \eta_n\}$ is a positive definite matrix of weighting coefficients that correspond to the efficiency of energy conversion at the plants. []' denotes transposition.

The considered problem is difficult to treat analytically owing to the delays in water reflow. For that reason, an alternative, equivalent system description will be used. Let $\mathbf{y}(t)$ denote the overall system resource level, i.e., the sum of water volume accommodated in the reservoirs and the water flowing between them subjected to control $\mathbf{x}$,

$$\mathbf{y}(k)=\mathbf{s}(k)+\sum_{j=1}^{T}\sum_{t=j}^{T}\boldsymbol{\Theta}_t\mathbf{x}(k-t). \tag{8}$$

Using a similar approach as in [17], it can be shown that the dynamics of $\mathbf{y}(t)$ follows

$$\mathbf{y}(k+1)=\mathbf{y}(k)+\boldsymbol{\Theta}\mathbf{x}(k)+\mathbf{r}(k). \tag{9}$$

B. Solution

For the performance index in problem (7), the Hamiltonian can be defined as

$$\mathbf{H}(k)=\frac{1}{2}p(k)\mathbf{x}'(k)\mathbf{N}\mathbf{x}(k)+\boldsymbol{\lambda}'(k+1)\left[\mathbf{y}(k)+\boldsymbol{\Theta}\mathbf{x}(k)+\mathbf{r}(k)\right], \tag{10}$$

where $\boldsymbol{\lambda}'(t+1)$ is a row vector of Lagrange multipliers.

The necessary conditions are as follows:

- state equation

$$\mathbf{y}(k+1)=\frac{\partial\mathbf{H}(k)}{\partial\boldsymbol{\lambda}(k+1)}=\mathbf{y}(k)+\boldsymbol{\Theta}\mathbf{x}(k)+\mathbf{r}(k), \tag{11}$$

- costate equation

$$\boldsymbol{\lambda}(k)=\frac{\partial\mathbf{H}(k)}{\partial\mathbf{y}(k)}=\boldsymbol{\lambda}(k+1), \tag{12}$$

- stationarity condition

$$\begin{aligned} 0 &= \frac{\partial\mathbf{H}(k)}{\partial\mathbf{x}(k)}=\frac{1}{2}p(k)(\mathbf{N}+\mathbf{N}')\mathbf{x}(k)+\boldsymbol{\Theta}'\boldsymbol{\lambda}(k+1) \\ &= p(k)\mathbf{N}\mathbf{x}(k)+\boldsymbol{\Theta}'\boldsymbol{\lambda}(k+1). \end{aligned} \tag{13}$$

Solving (13) for $\mathbf{x}$, yields

$$\mathbf{x}(k)=-p^{-1}(k)\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(k+1). \tag{14}$$

Note that since $\mathbf{N}$ is positive definite (a diagonal matrix with all positive entries), its inverse does exist.

Then, substituting (14) into (11), gives

$$\mathbf{y}(k+1)=\mathbf{y}(k)-p^{-1}(k)\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(k+1)+\mathbf{r}(k). \tag{15}$$

Equation (12) is a homogeneous difference equation. Its solution with the terminal condition $\boldsymbol{\lambda}(m)$ is

$$\boldsymbol{\lambda}(k) = \boldsymbol{\lambda}(m)\,. \tag{16}$$

Substituting (16) into (15), yields

$$\mathbf{y}(k+1) = \mathbf{y}(k) - p^{-1}(k)\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(m) + \mathbf{r}(k)\,. \tag{17}$$

With the initial resource level $\mathbf{y}(0)$ the solution of (17) is

$$\mathbf{y}(k) = \mathbf{y}(0) + \sum_{i=0}^{k-1}\left[\mathbf{r}(i) - p^{-1}(i)\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(m)\right]. \tag{18}$$

The initial state $\mathbf{y}(0)$ and the final state $\mathbf{y}(m)$ are fixed, so their first derivatives are equal to zero. Using (18), the final resource level may be calculated as

$$\mathbf{y}(m) = \mathbf{y}(0) + \sum_{i=0}^{m-1}\left[\mathbf{r}(i) - p^{-1}(i)\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(m)\right]. \tag{19}$$

Hence, the terminal value of the Lagrange multiplier vector

$$\boldsymbol{\lambda}(m) = -\left(\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\right)^{-1}\left[\mathbf{y}(m) - \mathbf{y}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right] / \sum_{i=0}^{m-1} p^{-1}(i), \tag{20}$$

and, using (16),

$$\boldsymbol{\lambda}(k) = \boldsymbol{\lambda}(m) = -\frac{\left(\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\right)^{-1}\left[\mathbf{y}(m) - \mathbf{y}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right]}{\sum_{i=0}^{m-1} p^{-1}(i)}. \tag{21}$$

Note that since $\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'$ is symmetric and $\mathbf{N}^{-1} = diag\{\eta_1^{-1}, \eta_2^{-1}, \ldots, \eta_n^{-1}\}$ positive definite, $\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'$ is positive definite, thus invertible.

Using (14) and (21), the optimal control

$$\begin{aligned}\mathbf{x}(k) &= -p^{-1}(k)\mathbf{N}^{-1}\boldsymbol{\Theta}'\boldsymbol{\lambda}(k+1)\\ &= \frac{p^{-1}(k)\mathbf{N}^{-1}\boldsymbol{\Theta}'\left(\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\right)^{-1}\left[\mathbf{y}(m) - \mathbf{y}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right]}{\sum_{i=0}^{m-1} p^{-1}(i)}.\end{aligned} \tag{22}$$

Since $\boldsymbol{\Theta}$ is invertible and $\mathbf{N}^{-1}$ a diagonal matrix with non-zero entries

$$\mathbf{N}^{-1}\boldsymbol{\Theta}'\left(\boldsymbol{\Theta}\mathbf{N}^{-1}\boldsymbol{\Theta}'\right)^{-1} = \mathbf{N}^{-1}\boldsymbol{\Theta}'\left(\mathbf{N}^{-1}\boldsymbol{\Theta}'\right)^{-1}\boldsymbol{\Theta}^{-1} = \boldsymbol{\Theta}^{-1}. \tag{23}$$

Therefore, (22) simplifies to

$$\mathbf{x}(k) = p^{-1}(k)\boldsymbol{\Theta}^{-1}\left[\mathbf{y}(m) - \mathbf{y}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right] / \sum_{i=0}^{m-1} p^{-1}(i). \tag{24}$$

It follows from (8) that

$$\mathbf{y}(m) = \mathbf{s}(m) + \sum_{j=1}^{T}\sum_{t=j}^{T}\boldsymbol{\Theta}_t\mathbf{x}(m-t), \tag{25}$$

so the control system is noncausal. However, when m » T, then $\mathbf{y}(m) \cong \mathbf{s}(m)$, which results in the following control law

$$\mathbf{x}(k) \cong \frac{p^{-1}(k)}{\sum_{i=0}^{m-1} p^{-1}(i)}\boldsymbol{\Theta}^{-1}\left[\mathbf{s}(m) - \mathbf{s}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right]. \tag{26}$$

C. *System properties*

Looking at how the flow control signal in (26) is established, a few observations can be made:

1) The current flow value depends on the current price, yet not on the water level. Thus, prone-to-error water level measurements are not needed for control law implementation.
2) Since $-\boldsymbol{\Theta}$ is a positive matrix and the price is also positive, the flow control signal does not change the sign in the entire planning horizon. It either reduces the water inflow in the case of heavy rainfall and a risk of flood or magnifies the flow intensity for users to gain profit.
3) The flow intensity does not depend on the temporary rainfall intensity, but on its cumulative value $\sum_{i=0}^{m-1}\mathbf{r}(i)$, only. It improves the control system robustness to weather condition fluctuations. In fact, it is resistant to temporal changes of opposite polarity.
4) With

$$\mathbf{K} = \frac{\boldsymbol{\Theta}^{-1}}{\sum_{i=0}^{m-1} p^{-1}(i)}\left[\mathbf{s}(m) - \mathbf{s}(0) - \sum_{i=0}^{m-1}\mathbf{r}(i)\right], \tag{27}$$

substituting (26) for $\mathbf{x}(k)$ in (5), one obtains

$$\mathbf{s}(k+1) = \mathbf{s}(k) + \sum_{t=0}^{T}\boldsymbol{\Theta}_t p^{-1}(k-t)\mathbf{K} + \mathbf{r}(k). \tag{28}$$

Therefore, the closed-loop system with control (26) does not lose the integrating property. For any k, one has

$$\mathbf{s}(k) = \mathbf{s}(0) + \sum_{i=0}^{m-1}\sum_{t=0}^{T}\boldsymbol{\Theta}_t p^{-1}(i-t)\mathbf{K} + \sum_{i=0}^{m-1}\mathbf{r}(i). \tag{29}$$

The water level exhibits neither oscillations nor overshoots. It is confined to the interval determined by the initial $\mathbf{s}(0)$ and final value $\mathbf{s}(m)$.

IV. Numerical example

In order to verify the analytic considerations from the previous sections, a series of tests for the topology from Fig. 3 and the price profile from May 1st Fig. 1 has been conducted. The system is supplied with the precipitation and corresponding runoff depicted in Fig. 4. The system experiences the following input: the initial pond occupancy $\mathbf{s}(0)$ = [6, 9, 15, 18] * 10^3 [m^3], and $\mathbf{s}(m)$ = [1.4, 2.1, 3.5, 4.2] * 10^4 [m^3]. The evolution of $\mathbf{x}(k)$ and $\mathbf{s}(k)$ computed according to (26) is presented in Fig. 5.

In the considered example, one observes the accumulation of energy in the ponds. Each plant in the cascade 1-3-4 and 2-3-4 throttles the flow all the more it is located down the river, which is intuitively justified. All the propitious system properties described in Section III.C are evidenced in graphs from Fig. 5.

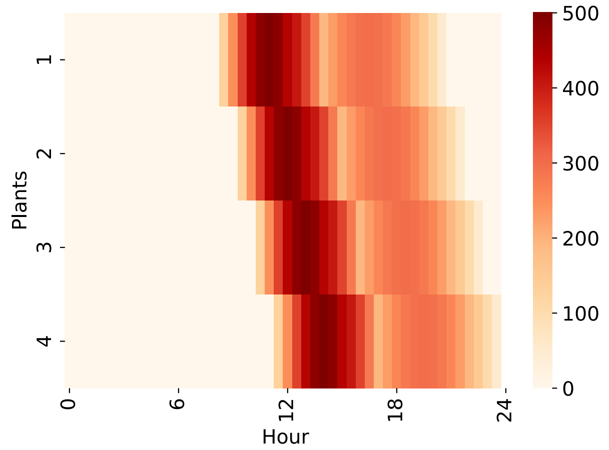


Fig. 4. Moving wave of rain and resulting runoff $\mathbf{r}(k)$ *[m^3/period].*

V. Conclusions

The paper's focus was to design an optimal control strategy to steer the system of connected hydro plants so that the power grid operators benefit from price profile flattening ("beheading the duck), and, at the same time, the plant owners gain monetary profit from their installations. In this way, the natural small rivers and reservoirs form a set of distributed, short-term energy depots deployable with low capital and operational expenditures. Additionally, the ponds, which slow down the precipitation runoff, elevate resilience to floods and droughts, whose risk grows as the climate changes.

A closed-form expression of the designed control law allows for a formal study of system properties. In particular, it has been shown that oscillations and overshoots are avoided so that the capacity constraints of reservoirs and riverbeds can be maintained. The control law is straightforward to implement and recompute for different system settings and weather conditions. No involving numerical treatment is required.

References

[1] S. Kolosok, Y., Bilan, T. Vasylieva, A. Wojciechowski, and M. Morawski, „A scoping review of renewable energy, sustainability and the environment." *Energies* 14(15): 4490, 2021.

[2] J. Wirfs-Brock. "IE Questions: Why Is California Trying To Behead The Duck?," [Online]. Available: https://insideenergy.org/2014/10/02/ie-questions-why-is-california-trying-to-behead-the-duck/, 2014, Accessed: 2023-05-12.

[3] PSE, "Market Energy Prices." [Online]. Available: https://www.pse.pl/dane-systemowe/funkcjonowanie-rb/raporty-dobowe-z-funkcjonowania-rb/podstawowe-wskazniki-cenowe-i-kosztowe/rynkowa-cena-energii-elektrycznej-rce.

[4] D. Borkowski, D. Cholewa, and A. Korzeń, „A run-of-the-river Hydro-PV battery hybrid system as an energy supplier for local loads." *Energies*, 14, 5160, 2021.

[5] A. Wijesinghe, and L. Lai, "Small hydro power plant analysis and development," *Proc. 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT)*, 25-30, 2011.

[6] K. Wyszkowski, Z. Piwowarek, and Z. Pałejko, „Małe elektrowne wodne w Polsce", Technical Report, UN Global. In Polish, [Online]. Available: https://ungc.org.pl/wp-content/uploads/2022/03/Raport_Male_elektrownie_wodne_w_Polsce.pdf, 2022.

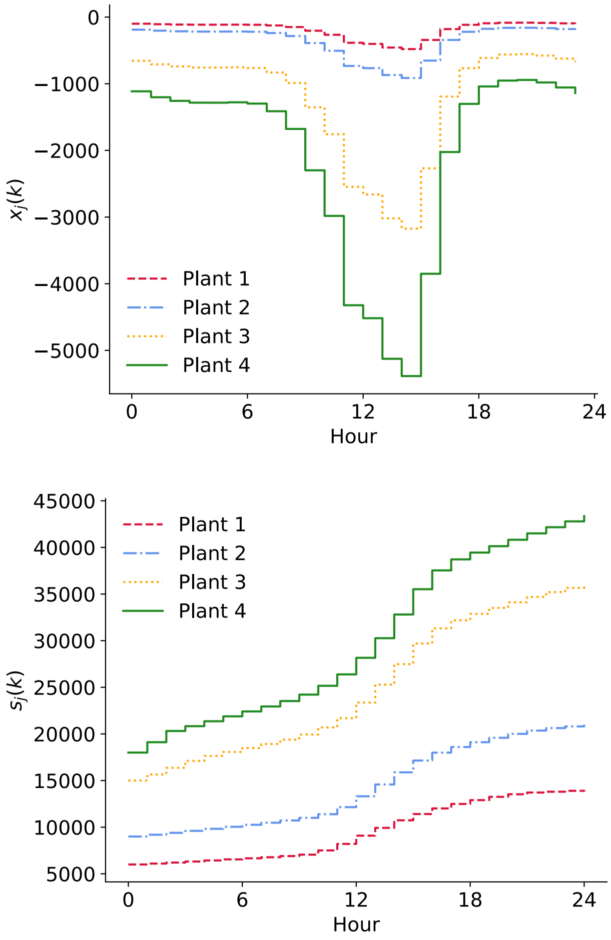


Fig. 5. Regulated flow $\mathbf{x}(k)$ *[m^3/h] and pond occupancy* $\mathbf{s}(k)$ *[m^3]*

[7] G. Alvarez, "An optimization model for operations of large scale hydro power plants," *IEEE Latin America Transactions*, 18(9), 1631–1638, 2020.

[8] C. Danielson, F. Borrellia, D. Oliver, D. Anderson, and T. Phillips, "Constrained flow control in storage networks: Capacity maximization and balancing," *Automatica*, 49(9), 2612–2621, 2013.

[9] M. V. Basin, F. Guerra-Avellaneda, and Y. B. Shtessel, "Stock management problem: Adaptive fixed-time convergent continuous controller design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(12), 4974–4983, 2020.

[10] P. Ignaciuk, "Dead-time compensation in continuous-review perishable inventory systems with multiple supply alternatives," *Journal of Process Control* 22(5), 915–924, 2012.

[11] Y. Yu, Y. Wu, and Q. Sheng, "Optimal scheduling strategy of cascade hydropower plants under the joint market of day-ahead energy and frequency regulation." *IEEE Access*, 2021.

[12] A. R. Shaw, et al. "Hydropower optimization using artificial neural network surrogate models of a high-fidelity hydrodynamics and water quality model," *Water Resources Research*, 53, 9444–9461, 2017.

[13] P. Ignaciuk and Ł. Wieczorek, "Continuous genetic algorithms in the optimization of logistic networks: Applicability assessment and tuning," *Applied Sciences*, 10(21), 7851, 2020.

[14] I. Ahmadianfar, A. Samadi-Koucheksaraee, and M. Asadzadeh, "Extract nonlinear operating rules of multi-reservoir systems using an efficient optimization method," *Sci. Rep.* 12, 18880, 2022.

[15] J. Bernardes, Jr., at all. "Hydropower operation optimization using machine learning: A systematic review," *AI*, 3(1), 78–99, 2022.

[16] Ł. Wieczorek and P. Ignaciuk, "Hydropower operation optimization using machine learning," *International Journal of Shipping and Transport Logistics*, 15(1–2), 111–143, 2022.

[17] P. Ignaciuk, "DARE solutions for LQ optimal and suboptimal control of systems with multiple input-output delays," *Journal of The Franklin Institute* 353(5), 974–991, 2016.

DOI: 10.15439/2023F8946
ISSN 2300-5963 ACSIS, Vol. 35

Semi-Active Control of a Shear Building based on Reinforcement Learning: Robustness to measurement noise and model error

Aleksandra Jedlińska*, Dominik Pisarski, Grzegorz Mikułowski,
Bartłomiej Błachowski, Łukasz Jankowski
0009-0003-0644-0760, 0000-0002-0515-3298, 0000-0003-4215-8650, 0000-0001-6021-0374, 0000-0002-9773-0688
Institute of Fundamental Technological Research (IPPT PAN),
Polish Academy of Sciences 02-106 Warsaw, Poland
Email: ajedlins@ippt.pan.pl | ljank@ippt.pan.pl

***Abstract*—This paper considers structural control by reinforcement learning. The aim is to mitigate vibrations of a shear building subjected to an earthquake-like excitation and fitted with a semi-active tuned mass damper (TMD). The control force is coupled with the structural response, making the problem intrinsically nonlinear and challenging to solve using classical methods. Structural control by reinforcement learning has not been extensively explored yet. Here, Deep-Q-Learning is used, which appriximates the Q-function with a neural network and optimizes initially random control sequences through interaction with the controlled system. For safety reasons, training must be performed using an inevitably inexact numerical model instead of the real structure. It is thus crucial to assess the robustness of the control with respect to measurement noise and model errors. It is verified to significantly outperform an optimally tuned conventional TMD, and the key outcome is the high robustness to measurement noise and model error.**

***Index Terms*—structural control, semi-active control, reinforcement learning, tuned mass damper (TMD).**

I. Introduction

In this paper, a novel control strategy for reducing structural vibrations in shear-type building structures under seismic excitation is presented and assessed. To achieve this, machine learning techniques, specifically reinforcement learning (RL), were customized, developed, and applied. Structural vibrations in engineering structures can have a negative impact on structural condition and operation, and they can negatively impact structural integrity. Various approaches have been developed to mitigate these effects, including passive, active, and semi-active control methods [1], [2]. The semi-active methods are appealing, since they do not require significant power sources and can be designed to be failure-safe. However, the control forces are coupled with the structural response, which leads to formulations that are challenging to be solved using classical methods [3], [4]. This paper focuses on semi-active control through the use of a semi-active tuned mass damper (TMD). The TMD is a classical device used to mitigate structural vibrations by adding a secondary mass that opposes the motion of the main structure [5], [6]. The semi-active TMD applied here is controllable through a switchable level of viscous damping.

The main aim of this contribution is to test the application potential of reinforcement learning (RL) in semi-active structural control, and in particular, the robustness of the trained agent to measurement noise and structural errors. This is a crucial problem for potential practical applications in civil engineering, since for safety reasons the RL agent must be trained using a numerical model instead of the physical target structure. The structure investigated here is an 11-story shear-type building equipped with a semi-active TMD. The structure is modeled using the finite element (FE) method, and the specific parameters of the models are taken from literature [7]. The TMD is controlled by switching its viscous damping coefficient in an on/off manner (bang–bang), as suggested by the Pontryagin minimum principle [8]. The structure is subjected to an earthquake-like random base excitation. A Deep Q Learning (DQN) algorithm is applied. The trained RL agent reduces the structural vibrations effectively and to a greater extent than a conventional tuned mass damper. Importantly, the contribution demonstrates and evaluates *also* the robustness of the trained agent with respect to measurement noise and model error.

II. Reinforcement Learning – the technique and system architecture

Reinforcement learning is a set of machine learning techniques that aim to teach an agent to determine the most effective actions by engaging in trial-and-error interactions with its environment. During the process the agent receives feedback in the form of rewards or punishments, which it uses to enhance its decision-making abilities over time. This research investigates the capability of reinforcement learning (RL) to enhance semi-active structural control. Unlike supervised learning that depends on optimal control sequences, which are often unknown in semi-active control, and unlike unsupervised learning, which solely relies on exploring input data, RL enables learning from interactions and seems to be well-tailored to the needs of structural control. However, despite the large successes of RL in mastering other complex tasks [9], including control-like problems [10]-[13], it is still a novel and very scarcely explored approach in structural control with only a handful of publications [14]-[16].

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

In this study, the reinforcement learning (RL) agent employs a dense artificial neural network (ANN) to learn and encode the Q-function. The ANN is implemented in the Python programming language using two popular open-source libraries, TensorFlow and Keras. TensorFlow is a low-level library used for building and training machine learning models, while Keras is a high-level API that simplifies the process of building neural networks. The ANN used in this study consists of six hidden sequential dense layers, each with 40 neurons. The input layer provides the network with measurements of structural response, while the output layer consists of two neurons corresponding to the possible states of the control signal. The activation function used in the neural network is rectified linear unit (ReLU) [17].

III. Shear building and excitation

The structure analyzed in this study is a shear-type building consisting of 11 stories with a semi-active tuned mass damper (TMD) attached to the top story (Fig. 1). The TMD is a well-known classical engineering device that comprises a mass, spring, and viscous damper and is widely used to reduce vibrations in structures subjected to external excitations, such as earthquakes [5]. Such a setup results in a total of twelve degrees of freedom (DOFs) which correspond to each of the eleven stories and the TMD. The equation of motion for the building model under seismic excitation can be expressed as:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = -[M]\{r\}a(t) \quad (1)$$

The vector $\{u\}$ has 12 rows and represents the absolute displacements of each DOF, while the vector $\{r\}$ also has 12 rows and represents the displacement resulting from unit horizontal ground displacement for each DOF. The ground acceleration is denoted by a(t), while the matrices [M], [C], and [K] are 12 × 12 in dimension, and represent the mass, damping, and stiffness of the structure, respectively. The material damping model is assumed, and the damping matrix $[C]$ is proportional to the stiffness matrix with the proportionality coefficient chosen to achieve 2% critical damping for the first mode of vibration of the structure without the TMD. The control directly affects the entries in $[C]$ that correspond to the damping of the TMD, see Fig. 1, by switching it between zero and a large value. The mass matrix $[M]$ is diagonal. The masses of each story and the TMD are thus listed on the diagonal of the mass matrix, assuming lumped masses at each floor level. Building specifications, including the number of stories, their masses, and stiffnesses, are based on the literature data [7]. The first undamped natural frequency is 0.89 Hz for the building with the TMD and 1.05 Hz for the building without the TMD.

In this study, an effort has been made to safeguard the RL agent from acquiring a limited response pattern conditioned on a particular collection of ground movements. This restriction is essential to avoid overfitting, a common issue in supervised learning. To address this concern, the seismic load a(t) is assumed to be the white Gaussian noise. Consequently, it is generated afresh for every training and evaluation episode, guaranteeing that the proposed control system is exposed to diverse ground motions without any bias towards specific patterns [7].

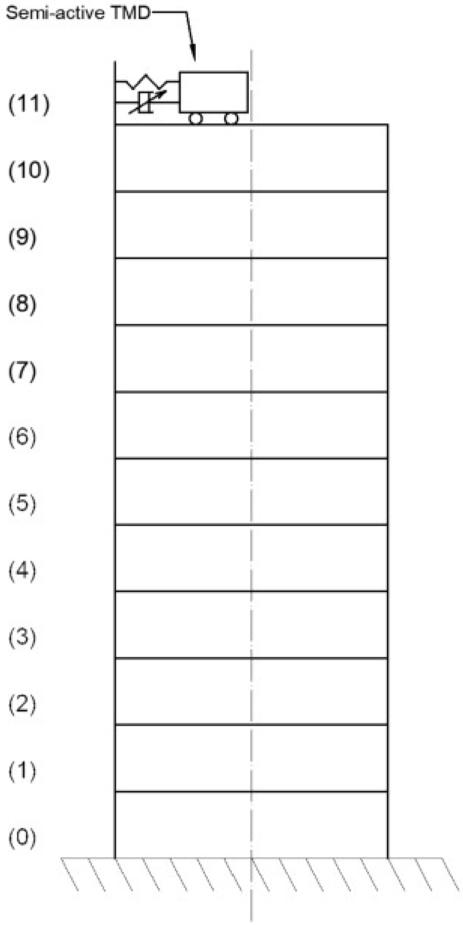


Fig. 1. The investigated 11-DOF structure with a semi-actively controlled TMD placed on the top level

IV. RL training

The state of the RL environment employed for training and control purposes is based on linearly transformed full structural state vector, and it is comprised of the relative displacements and velocities between the ground, subsequent floors, and the TMD. Such a choice is practical, as the relative interstory displacements and velocities are relatively easy to be measured in a real setting.

The training proceeds in episodes. Each episode consists of 1000 RL steps of 25 ms each, and it corresponds to about 25 periods of the fundamental structural vibration. For fidelity of structural response simulation, each RL step is internally further subdivided into 5 simulation steps, each of 5 ms.

The aim of the control is to reduce the oscillations experienced by the highest floor of the structure. For structural control purposes, the control efficiency is usually assessed using the root mean square (RMS) of the displacements in each episode. Consequently, the agent's training is based on the rewards it receives in each step of the interaction episodes, and the rewards are evaluated using the displacement level of the top floor. The maximum reward of 1 is assigned when the displacement is zero. The rewards are used to update the agent's Q-function, allowing it to improve its performance in future episodes.

The total reward signal at the end of each episode reflects the agent's performance and the distance from the equilibrium point over all time steps. Fig. 2 shows the total reward per training episode, together with its EMA50, which increases as the agent learns. The value of 1000 denotes a perfectly stationary top floor, and the chart includes the effect of a 10% exploration rate (10% of actions, on average, is selected randomly to ensure ongoing exploration of the action space).

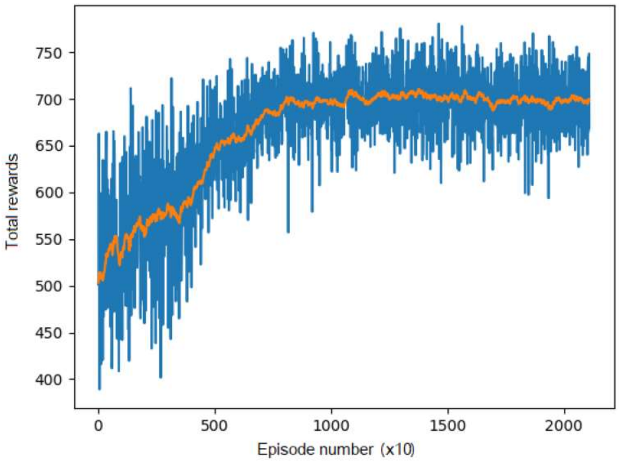


Fig 2. Total rewards per training episode (blue line) and its EMA50 (orange line)

V. Robustness to measurement noise and model error

The intended ultimate application scenario involves a real physical environment (building structure) rather than just its idealized mathematical model. There are two main factors that inevitably differentiate a physical structure from its numerical model: 1) measurement noises overlaid on signals from physical sensors, and 2) model errors that represent the modeling inaccuracies. These factors can negatively affect the control efficiency applied by an RL agent trained using an idealized environment.

The first test involves applying simulated measurement noise to the agent's observations, which is modeled as a Gaussian white noise and added to the input to the neural network (sensor measurements). The test examines increasing larger levels of noise, which is quantified in the signal RMS terms (noise standard deviation related to the RMS of the original sensor signal). The control effectiveness is assessed in terms of the ratio of the top floor displacement RMS in the controlled structure to the top floor displacement RMS in the structure equipped with an optimally tuned passive TMD. Values smaller than 1.0 denote a better effectiveness in comparison to the passive system. To account for the random character of the earthquake-like base excitation, 1000 episodes of 2000 time steps are simulated for each noise level. Fig. 3 plots the mean value of the RMS ratio (blue line) together with its 1 sigma band (yellow).

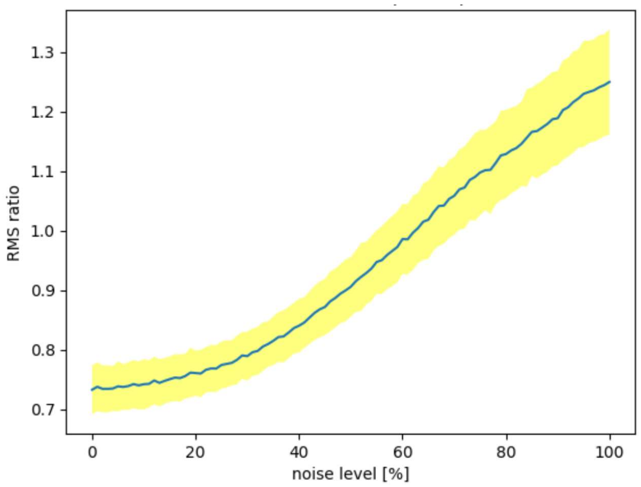


Fig 3. Control efficiency for various measurement noise levels, assessed in terms of the ratio of the top floor displacement RMS between the controlled structure and the structure equipped with an optimally tuned passive TMD: mean value (blue) and 1 sigma band (yellow)

The next evaluation assesses how the trained agent handles model errors, which are possible deviations of physical engineering structures from their ideal mathematical models. The stiffness and mass of individual floors are subject to random change. The generated error follows a normal distribution, limited at 10% of the original level to avoid near-zero or negative values. The evaluation results are presented in Fig. 4. Similarly as in Fig 3, the figure depicts the top floor displacement mean RMS ratio (blue) and its 1 sigma range (yellow), evaluated at each error level using 1000 episodes of 2000 steps each.

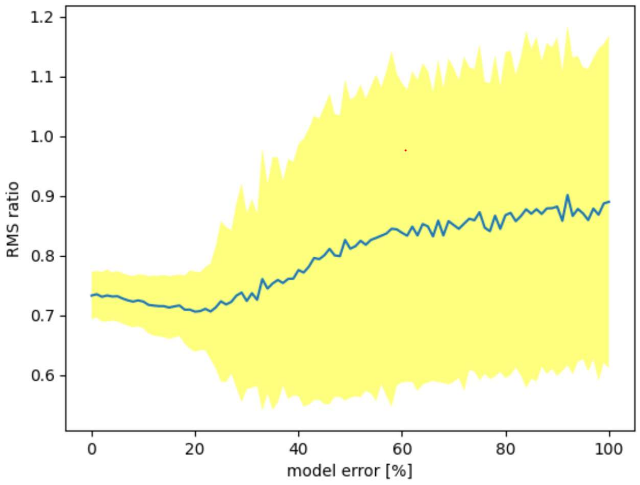


Fig 4. Control efficiency for various model error levels, assessed in terms of the ratio of the top floor displacement RMS between the controlled structure and the structure equipped with an optimally tuned passive TMD: mean value (blue) and 1 sigma band (yellow)

The control performance was not significantly affected by even large levels of measurement noise and model error, as indicated by the observed stable RMS ratio. In particular, the control was more effective than the optimal passive TMD up to measurement noise of about 60% rms. In case of model errors, the mean control effectiveness remained surprisingly good in the entire tested error range; however, the variability of the results increased considerably for model errors above the level of 20%. Such results suggest that the trained model possesses a certain degree of tolerance to disturbances in the form of measurement noise and model errors, allowing it to maintain reliable performance even in the presence of real-world environmental variations.

One possible reason for the model's low sensitivity to disturbances could be attributed to its neural network architecture. Neural networks, particularly those with deeper structures, are known for their ability to learn and extract meaningful features from noisy data. The network layers and parameters might have been optimized during training to capture relevant patterns and generalize well, enabling the model to disregard irrelevant noise components. Additionally, the random character of the base excitation could also contribute to the model resilience, as it prevents the agent from overfitting the specific characteristics of the model and signals and allows it to explore the entire control space.

Further analysis and experimentation can provide deeper insights into the model's robustness and shed light on the specific architectural and training aspects that contribute to its noise tolerance. Understanding these factors will not only enhance our understanding of the model's behavior but also guide the development of more resilient and reliable models in various scientific and engineering domains.

VI. Conclusion

This contribution studied the efficiency of an RL-based semi-active control scheme applied to a shear-type building subjected to an earthquake-like base excitation. The evaluation revealed a noteworthy characteristic of the control system, namely its remarkable insensitivity to measurement and model errors. Despite potential deviations or inaccuracies in the mathematical model used for control, the system demonstrated a high level of robustness and stability. This implies that the control algorithm could effectively compensate for discrepancies between the actual system behavior and the idealized mathematical representation, ensuring reliable performance in real-world scenarios. The observed low insensitivity to noises and errors highlights the effectiveness and practical applicability of the RL-based control methodology in the considered context.

The promising results provide initial insights into the potential of reinforcement learning for improving and ensuring the performance of semi-active damping systems, even though the use of RL in structural control, particularly in semi-active control, is not yet widespread.

Acknowledgment

The authors gratefully acknowledge the support of the National Science Centre, Poland, granted under the grant agreement 2020/39/B/ST8/02615. For the purpose of Open Access, the authors have applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

References

[1] B. F. Spencer Jr., S. Nagarajaiah. 2003. “State of the art of structural control.” *J. Struct. Eng.* 129:845–856, https://doi.org/10.1061/(ASCE)0733-9445(2003)129:7(845)

[2] B. Basu, O. S. Bursi, F. Cascati, S. Cascati, A. E. Del Grosso, M. Domaneshi, L. Faravelli, J. Holnicki-Szulc, H. Irshik, M. Krommer, M. Lepidi, A. Martelli, B. Ozturk, F. Pozo, G. Pujol, Z. Rakicevic, and J. Rodellar. 2014. “A European Association for the Control of Structures joint perspective. Recent studies in civil structural control across Europe,” *Struct. Contrl Health Monit.* 21:1414–1436, https://doi.org/10.1002 / stc.1652

[3] F. Casciati, J. Rodellar, and U. Yildirim. 2012. “Active and semi-active control of structures-theory and applications: A review of recent advances.” *J. Intell. Mater. Syst. Struct.* 23:1181–1195, https://doi.org/10.1177/1045389X12445029

[4] N. R. Fisco, H. Adeli. 2011. “Smart structures: Part I—Active and semi-active control.” *Sci. Iran.* 18(3):275–284, https://doi.org/10.1016/j.scient.2011.05.034

[5] M. Gutierrez Soto, H. Adeli. 2013. “Tuned mass dampers.” *Arch. Comput. Meth. E.* 20:419–431, https://doi.org/10.1007/s11831-013-9091-7

[6] S. Elias, V. Matsagar. 2017. “Research developments in vibration control of structures using passive tuned mass dampers.” *Annu. Rev. Control* 44:129–156, https://doi.org/10.1016/j.arcontrol.2017.09.015

[7] S. Pourzeynali, H. H. Lavasani, and A. H. Modarayi. 2007. “Active control of high rise building structures using fuzzy logic and genetic algorithms.” *Eng. Struct.* 29:346–357, https://doi.org/10.1016/j.engstruct.2006.04.015

[8] D. E. Kirk. *Optimal control theory.* Courier Corporation, 2004.

[9] G. Rypeść, Ł. Lepak, P. Wawrzyński. 2022. “Reinforcement Learning for on-line Sequence Transformation,” in *Proc. 17th Conf. on Computer Science and Intelligence Systems*, Sofia, ACSIS 30, pp. 133–139. https://doi.org/10.15439/2022F70

[10] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. 2018. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play.” *Science* 362(6419):1140–1144, https://doi.org/10.1126/science.aar6404

[11] A. El Sallab, M. Abdou, E. Perot, and S. Yogamani. 2017.” Deep reinforcement learning framework for autonomous driving.” in *Proc. IS&T International Symposium on Electronic Imaging Science and Technology*, Burlingame, 2017, pp. 70–76, https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023

[12] G. Reddy, A. Celani, T. Sejnowski, and M. Vergassola. 2016.” Learning to soar in turbulent environments.” *PNAS* 113(33):E4877–E4884, https://doi.org/10.1073/pnas.1606075113

[13] H. Shi, Y. Zhou, X. Wang, S. Fu, S. Gong, and B. Ran. 2022. “A deep reinforcement learning-based distributed connected automated vehicle control under communication failure.” *Comput.-Aided Civ. Inf.* 37(15):2033–2051, https://doi.org/10.1111/mice.12825

[14] A. Bernard, I. F. Smith. 2008. “Reinforcement learning for structural control.” *J. Comput. Civil Eng.* 22(2):133–139, https://doi.org/10.1061/(ASCE)0887-3801(2008)22:2(133)

[15] A. Khalatbarisoltani, M. Soleymani, and M. Khodadadi. 2019. “Online control of an active seismic system via reinforcement learning.” *Struct. Contrl Health Monit.* 26(3):e2298, https://doi.org/10.1002/stc.2298

[16] Z.-C. Qiu, G.-H. Chen, and X.-M. Zhang. 2021. “Reinforcement learning vibration control for a flexible hinged plate.” *Aerosp. Sci. Technol.* 118:107056, https://doi.org/10.1016/j.ast.2021.107056

[17] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis. 2012. “Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers.” *IEEE Contr. Syst. Mag.* 32(6):76–105, https://doi.org/10.1109/MCS.2012.2214134

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1011–1015
DOI: 10.15439/2023F4485
ISSN 2300-5963 ACSIS, Vol. 35

Gender-aware speaker's emotion recognition based on 1-D and 2-D features

Włodzimierz Kasprzak, Mateusz Hryciów
0000-0002-4840-8860, 0000-0000-0000-0000
Warsaw University of Technology, Institute of Control and Computation Eng.
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
Email: wlodzimierz.kasprzak@pw.edu.pl

***Abstract*—An approach to speaker's emotion recognition based on several acoustic feature types and 1D convolutional neural networks is described. The focus is on selecting the best speech features, improving the baseline model configuration and integrating in the solution a gender classification network. Features include a Mel-scale spectrogram and MFCC-, Chroma-, prosodic- and pitch-related features. Especially, the question whether to use 2-D maps of features or reduce them to 1-D vectors by averaging, is experimentally resolved. Well–known speech datasets RAVDESS, Tess, Crema-D and Savee are used in experiments. It appeared, that the best performing model consists of two convolutional networks for gender-aware classification and one gender classifier. The Chroma features have been found to be obsolete, and even disturbing, given other speech features. The f1 accuracy of proposed solution reached 73.2% on the RAVDESS dataset and 66.5% on all four datasets combined, improving the baseline model by 7.8% and 3%, respectively. This approach is a serious alternative to other proposed models, which reported accuracy scores of 60% - 71% on the RAVDESS dataset.**

I. Introduction

A HUMAN is able to effectively recognize the emotions of the speaker, but attempts to create automatic systems of this type (SER - speaker's emotion recognition) give quite limited results – their intensive development is still carried out [1]. SER systems can find a variety of applications: detecting the emotions of mobile phone users, call center operators and customers, car drivers and other participants of human-machine communication [2]. In some situations, this would allow computer-generated characters to be used to have natural conversations by appealing to human character. Only with this ability is it possible to achieve a fully meaningful dialogue between man and machine.

Human emotionality includes personality, character, temperament, and inspiration as the main psychological parameters that drive human emotions." It can therefore be concluded that there are different sources of communication through which a person expresses his emotions. These include, among others, facial expressions, gestures, speech and writing. On the basis of each of these methods, models can be constructed that will enable the recognition of emotions. One can expect, that a reliable emotion recognition system will require the use of various information modalities, like face videos synchronized with speech and wearable sensor recordings [3].

In the theory of basic emotions [4], it is assumed that people have a limited number of emotions (e.g. joy, anger, fear) that are biologically and psychologically basic. Each of them manifests in the majority of society in an organized, repeating pattern of related behavioral components [5]. Thanks to this, it is possible to label them. However, for this purpose, the number of possible classes must be specified. One of the theories was developed by Robert Plutchik. He distinguished eight basic emotions: joy, acceptance, fear, surprise, sadness, anger, disgust, and anticipation [6].

Despite the relatively easy task of classifying emotions into several categories, it can be problematic even for people if they are not exaggerated. It turns out that in the case when they need to determine the emotional state of an unknown person, the recognition rates are about 60% [7]. Errors in labelling can also affect the quality of automatic classification systems at a later stage.

Currently developed systems for speaker's emotion classification are based on acoustic modeling used in speaker recognition (speaker identification and verification) systems [8]. The basic classic machine learning methodologies used for this problem are UBM-GMM (Universal Background Model - Gaussian Mixture Model) and "i-vectors" [9]. An early solution based on deep neural networks is the ”x-vector” [10] network.

In section 2, related work and database issues are introduced in more details. The implemented system SER (speaker's emotion recognition) is presented in section 3. The main experiments and emotion classification results are shown in section 4. At the end, in section 5, we conclude the work.

II. Related work

A. Speech features and recognition techniques

The development of SER systems is inseparably connected with the use of machine learning techniques. It turns out that the vast majority of solutions used to recognize emotions based on speech are based on these solutions. These include [11]: neural networks (NN), convolutional neural networks (CNN), deep neural networks (DNNs), hidden Markov models (HMM), support vector machines (SVM) [12], decision trees and random forests [13].

This work was supported by ”Narodowe Centrum Badań i Rozwoju”, Warszawa, Poland, grant No. CYBERSECIDENT/455132/III/NCBR/2020. The publication was funded by Warsaw University of Technology.

In various studies, it has been shown that reducing the number of features has a positive effect on classification. It increases the generalization abilities of individual models, and the time of their training decreases. What's more, it turned out that reducing the number of features did not negatively affect the accuracy of emotion prediction, and sometimes even led to its improvement. For example, using the SVM algorithm, reducing the number of attributes from 276 to 75 resulted in an increase in the recognition rate by 5% [12]. In other studies, using the random forest method, selecting 16 traits out of 84 dropped the accuracy by 5%[13].

From the analysis of existing methods we can conclude the following motivation for our research work:

1) Acoustic features of man and women usually differ. For example, the fundamental frequency (pitch) of women voice is usually higher than man.
2) Prosodic features are useful in emotion recognition. The correlation between prosodic features and emotional states of has been already demonstrated [14]. Thus, our work will take such features into account.
3) The use of pitch classes for emotion recognition need to be evaluated. A pitch class is a set of notes with a given halftone pitch from all octaves. Chroma features associated with pitch classes are most often used to recognize emotions in musical works, but some works indicate their correlation with emotional states expressed by speech [15].
4) As the problem is similar to the classification of speaker groups, the solution can well be based on speaker identification methods, like GMM-UBM, JFA, i-vectors or x-vectors. However, we focus on lightweight neural network models using network types, like MLP and CNN.

B. Datasets

Four databases of annotated speech recordings were used for training and testing of the proposed speaker's emotion recognition (SER) system - the ”Ryerson Audio-Visual Database of Emotional Speech and Song” (RAVDESS) [16], the ”Toronto emotional speech set” (TESS) [17], the ”Crowd-sourced Emotional Multimodal Actors” dataset (Crema-D) [15], and the ”Surrey Audio-Visual Expressed Emotion” dataset (Savee) [18]. We use the RAVDESS and TESS databases in initial experiments, dealing with the selection of feature sets and the tuning of network models. An overall evaluation of the proposed solution is also given on a combination of the four datasets. The RAVDESS set is most important for us, as it contains recordings of 24 actors (12 Male and 12 Female voices) and annotates the full number of 8 emotional states: *angry, disgust, fear, happy, neutral, sad, surprise* and *calm*. The TESS database is larger, but uses only the first 6 emotion classes and contains samples of 2 female actors only. Savee holds samples of 7 emotion classes (no *calm* class) from 4 male speakers and Crema-D – samples of first 6 classes only from 91 actors (48 male and 43 female actors). In RAVDESS, every actor delivered 60 sentences with the content "Kids are talking by the door" or "Dogs are sitting by the door", giving a total of 1440 recordings with an average recording length of approximately 3.7 s.

III. System SER

A. Structure

The SER (System for Emotion Recognition) solution was designed with a general structure shown in Figure 1. There are three basic stages of processing:

1) Signal segmentation and detection of acoustic features;
2) Gender classification;
3) Two emotion classifiers - trained separately for male and female speakers.

The results of studies of models with different configurations indicate that it is useful to pre-classify the gender of the speaker and train separate models for male and female voices. Hence, the target configuration of the emotion classifier has three networks - one gender model for binary classification into Male and Female speaker and two emotion models for male and female emotion classification.

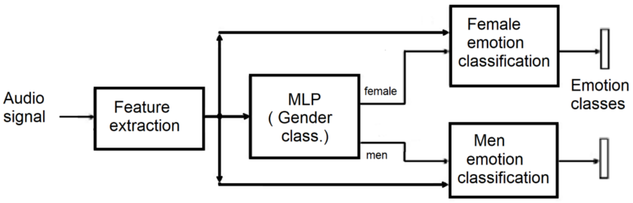


Fig. 1. Structure of the SER solution

B. Acoustic features

The feature vector in our solution can contain features of six types: zero-crossing rate (ZCR), Chroma, RMS value of the signal (root mean square), MFCC–based features, Mel-spectrogram coefficients, and prosodic features (e.g., mean and variance of the fundamental frequency). To determine the above features, the audio signal is processed by functions from the librosa library [19]. Different combinations of the above characteristics were investigated as well as individual types of features. In a final chosen solution, based on 1D convolutional layers, the features are averaged over time (over signal frames) to give a 1D input data – each recording may be represented by a vector of up to 162 features. We also experimented with solutions based on true 2D CNNs and LSTM networks – the feature averaging was omitted to provide a map of 2D features as input to the neural networks (Fig. 2).

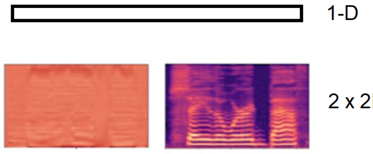


Fig. 2. Alternative input features of SER: a 1D vector of features averaged over time, two 2D maps of features

Regarding a prosody feature, we tracked the basic frequency (”pitch", F0) over time. In general, there are useful prosody features, like: waveform F0 (mean, minimum, maximum, variance), average energy of the voiced and voiceless parts, tempo of speech (inverse of the average time of voiced parts in a statement).

C. Gender model

For gender classification, we use an MLP network with 1D input (Figure 3). The model has three fully connected hidden layers containing 400 neurons each followed by a 0.1 dropout for each layer.

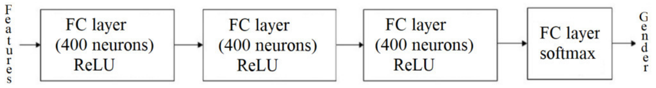

Features
FC layer (400 neurons) ReLU
FC layer (400 neurons) ReLU
FC layer (400 neurons) ReLU
FC layer softmax
Gender

Fig. 3. Structure of the MLP network for gender classification

D. 1D-based SER

For emotion classification, we use a baseline solution built from 1D convolutional layers [20]. There are four layers with 1x5 masks followed by ”MaxPooling1D” layers. After the flattening layer there are two Dense layers - one with ReLu activation and the other with softmax. During experiments with various feature sets and model parameters, under a gender-aware policy, the performance of this baseline model was improved by several percent. The final architecture of our emotion classifier (used both for man and female emotions) differs from the base network mainly by the final two layers and the input vector (Table I). The last convolutional layer has 96 filters (replacing 64), while the fully connected layer has 128 neurons (replacing 32). The input data consists of 150 features, as the Chroma features have been found useless for this network configuration.

TABLE I
THE MODEL ”LAYER 128-96” FOR EMOTION RECOGNITION

Layer (type)	Output (shape)	Param. number
conv1d (Conv1D)	(None, 150, 256)	1536
max_pooling1d (MaxPooling1D)	(None, 75, 256)	0
conv1d_1 (Conv1D)	(None, 75, 256)	327936
max_pooling1d_1 (MaxPooling1D)	(None, 38, 256)	0
conv1d_2 (Conv1D)	(None, 38, 128)	163968
max_pooling1d_2 (MaxPooling1D)	(None, 19, 128)	0
dropout (Dropout)	(None, 19, 128)	0
conv1d_3 (Conv1D)	(None, 19, 96)	61536
max_pooling1d_3 (MaxPooling1D)	(None, 10, 96)	0
flatten (Flatten)	(None, 960)	0
dense (Dense)	(None, 128)	123008
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 8)	1032
Total params: 679016		
Trainable params: 679016		
Non-trainable params: 0		

E. 2-D based SER

For a 2-D data input, models based on CNN networks were studied. We considered two processing streams - one for the MFCC-based feature map and one for the Mel-spectrogram. Each 2D map is processed by a CNN model with convolutional layers and 3 maxpooling layers. The outputs of both models are concatenated and processed by a fully connected layer with softmax activation.

F. Data augmentation

A well-known approach in machine learning is data augmentation [21]. For automatic increase of the number of annotated recordings, there will be synthetic secondary recordings generated from existing recordings by following operations: adding noise, stretching or compressing the signal over time, and changing the frequency of the basic tone (F0). The operation of adding noise does not change the envelope of the signal, so it also does not change the type of emotion in the recording. Other operations can change the class, so the scope of changes has been limited.

G. Evaluation metrics

The metrics used to evaluate a classification system is typically based on the following counts of model prediction: TN – True Negative, TP – True Positive, FN – False Negative, FP – False Positive. The appropriate relationships of the above results lead to evaluation metrics of the system:

- Precision – the correctness degree of the positive prediction result of a given class: $Precision = TP/(TP+FP)$ Observe that ”false positive rate”: $FPR = 1 - Precision$.
- Recall = TPR (true positive rate) – the degree of correct prediction of examples of a given class: $Recall = TP/(TP+FN) = TPR$
- F1 score – the degree of correct positive predictions: $F1 = 2 \cdot (Recall \cdot Precision)/(Recall + Precision)$
- ROC (receiver operating characteristic) – a curve, $TPR = f(FPR)$ that relates true positive rate (TPR) versus false positive rate (FPR) (where such pairs of values are obtained for the same decision thresholds).
- AUC (area under the ROC curve) – it determines the probability that the classifier will rank a random example of a positive class higher than a random example of a negative class. Ideally, its value is 1.

IV. EXPERIMENTAL RESULTS

A. Gender classification

The gender classifier (Fig. 3) was tested on the RAVDESS set (24 speakers). The accuracy on the test set was 98.7% and the AUC value was 0.9993. These results show a high quality of the proposed model.

B. Effect of recording times

The effect of time (length) of the recording onto the emotion classification was tested using the MLP model as emotion classifier (with 8 outputs) and all the 165 features (Table II). For the two datasets (RAVDESS and TESS), the classification accuracy increased with increasing recording time and saturated between 4 and 6 seconds. In further experiments, sequences with a length of 4 s were chosen for analysis.

TABLE II
DIFFERENT LENGTHS OF RECORDINGS AND THEIR INFLUENCE ONTO CLASSIFICATION ACCURACY

Length	TESS		RAVDESS	
[s]	Test accuracy	AUC field	Test accuracy	AUC field
2	0,78	0,9725	0,43	0,8274
3	0,92	0,9945	0,55	0,9029
4	0,98	0,9998	0,60	0,9117
6	0,99	1,0000	0,62	0,9418

C. Effect of input features

In early experiments with the MLP-based emotion classification on the RAVDESS dataset, we studied the effect of using single types of features. The following test accuracies (F1 score) were observed: for prosodia features 0.49, MFCC 0.39, Mel-spectrogram 0.37, Chroma 0.20, all features 0.62. Thus, when combining all the features into an input vector for the 1D emotion classifier, the effect of cancelling Chroma features was tested. It turned out, that the modified baseline network, in every configuration performs better without the Chroma features (Table III). The presented results of training and test (validation) accuracies come after 50 or 100 training epochs, with batch size 64.

TABLE III
EFFECT OF MODIFYING THE FEATURE SET AND THE BASE MODEL ON EMOTION CLASSIFICATION ACCURACY

Model	All features (162)		No Chroma (150)	
	Avg. recall	F1	Avg. recall	F1
Baseline (100 ep.)	0.64	0.654	0.67	0.681
Layer 128-64 (100 ep.)	0.69	0.696	0.71	0.716
Layer 128-96 (100 ep.)	0.69	0.700	0.72	0.719
Baseline (50 ep.)	0.66	0.661	0.66	0.664
Layer 64-128 (50 ep.)	0.66	0.678	0.67	0.684
Layer 96-64 (50 ep.)	0.66	0.655	0.67	0.682
Layer 64-64 (50 ep.)	0.67	0.672	0.68	0.690
Layer 64-96 (50 ep.)	0.67	0.669	0.68	0.685
Layer 96-96 (50 ep.)	0.67	0.676	0.68	0.693

D. Effect of layer size modification

The experimental results, collected in Table III, also show, that by increasing the number of neurons in the FC layer to 128, and the number of filters in the last convolutional layer to 96, a significant increase of the performance can be achieved. This configuration is denoted as ”Layer 128-96”.

E. Gender-aware emotion classification

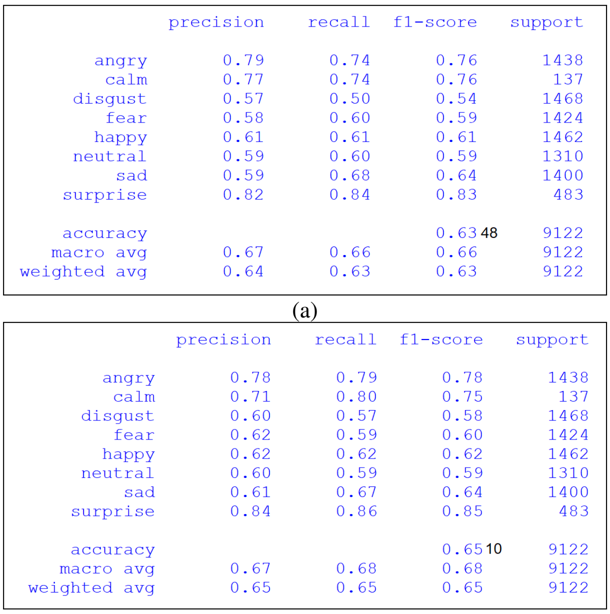
We complete the experiments on our 1D SER approach by training and testing two separate emotion classifiers – one for Males and one for Females. The results provided in Table IV are twofold. There is no significant increase of the performance of the baseline model when applied under perfect gender classification conditions. The practical observable performance is even slightly lower for the gender-aware (G-A) solution. A different effect is observed for our best modified model (Layer 128-96) used for separate gender emotion modelling – the F1 accuracy is now increased by 2.3% (in theory) and 1.3% (in practice) on the RAVDESS dataset.

TABLE IV
EFFECT OF GENDER-CONTROLLED EMOTION CLASSIFICATION

Model	All features (162)		No Chroma (150)	
	Avg. recall	F1	Avg. recall	F1
Baseline female	0.68	0.7019	0.73	0.7426
Baseline male	0.58	0.6019	0.61	0.6222
2x Baseline theoretic	0.63	0.6519	0.67	0.6824
2x Baseline real	0.622	0.645	0.664	0.675
Female Layer 128-96	0.69	0.7241	0.76	0.7796
Male Layer 128-96	0.66	0.6722	0.70	0.7037
G-A 2x Layer 128-96	0.675	0.6982	0.73	0.7417
Real G-A 2x Layer 128-96	0.667	0.689	0.722	0.732

F. Comparison on 4 datasets

Finally, the baseline solution has been compared with our best modified model (Layer 128-96, no Chroma) by training and testing both on the four available datasets: RAVDESS, TESS, Crema-D and Savee. Classification reports are given in Figure 4. Again, our modified model keeps an advantage of 2% in the F1 score.



```
              precision    recall  f1-score   support

       angry       0.79      0.74      0.76      1438
        calm       0.77      0.74      0.76       137
     disgust       0.57      0.50      0.54      1468
        fear       0.58      0.60      0.59      1424
       happy       0.61      0.61      0.61      1462
     neutral       0.59      0.60      0.59      1310
         sad       0.59      0.68      0.64      1400
    surprise       0.82      0.84      0.83       483

    accuracy                           0.6348    9122
   macro avg       0.67      0.66      0.66      9122
weighted avg       0.64      0.63      0.63      9122
```

(a)

```
              precision    recall  f1-score   support

       angry       0.78      0.79      0.78      1438
        calm       0.71      0.80      0.75       137
     disgust       0.60      0.57      0.58      1468
        fear       0.62      0.59      0.60      1424
       happy       0.62      0.62      0.62      1462
     neutral       0.60      0.59      0.59      1310
         sad       0.61      0.67      0.64      1400
    surprise       0.84      0.86      0.85       483

    accuracy                           0.6510    9122
   macro avg       0.67      0.68      0.68      9122
weighted avg       0.65      0.65      0.65      9122
```

(b)

Fig. 4. Classification reports of (a) the baseline model and (b) our modified baseline model, when trained and evaluated on four emotion datasets

G. The 2D SER

Two-dimensional data is in the form of a feature map, where one axis represents discrete time (indexes of subsequent frames) and the other feature indexes for one frame. We studied maps representing the mel-spectral features and cepstral features of the MFCC. Each feature map was obtained from a recording with a duration of 4 seconds. We applied CNN or CNN+LSTM models for emotion classification. The best result was achieved for a Mel-spectrogram input – an accuracy of 37%, based on RAVDESS. For the MFCC feature map, an accuracy of 34% was reached. We also uses the well-known pretrained VGG16 convolutional model. Here the best result was an accuracy of 43%, achieved on the Mel-spectrograms. The results of the classification based on the LSTM network have reached 34% only.

H. Comparison with other works

It should also be mentioned that for the RAVDESS set, the highest accuracy values achieved by different authors are 60% – 71%, but using much more complex models than ours [22], [23], [24], [25]. Some recent results, obtained on the RAVDESS dataset, are listed in Table V. Our best solutions — the single model "Layer 128-96" and the gender-aware configuration of three models – have outperformed other known solutions.

TABLE V
COMPARISON WITH OTHER WORKS EVALUATED ON THE RAVDESS DATASET

Model [ref]	Aver. Recall (%)	F1 (%)
CVT+SVM [22]	-	60.1
ResNet [23]	50.3	53.3
GResNet [23]	59.7	60.35
VGG16 [24]	71.0	-
Our "Layer 128-96"	72.0	71.86
Our "G-A 2x Layer 128-96"	72.2	73.2

V. CONCLUSIONS

In our research, we have confirmed a good performance of models processing a 1D feature vector. The approach to emotion classification based on 2D feature maps has failed. The use of a proper subset of speech features (without Chroma), a modification of the baseline network and the gender-aware approach have all contributed to a final result, that outperforms other known approaches validated on the RAVDESS dataset. The aim of our future research will be to explore more deeply prosodic features in emotion recognition.

REFERENCES

[1] M. Lech, M. Stolar, C. Best and R. Bolia, "Real-Time Speech Emotion Recognition Using a Pre-trained Image Classification Network: Effects of Bandwidth Reduction and Companding," *Frontiers in Computer Science*, vol. 2, 2020, article 14, https://doi.org/10.3389/fcomp.2020.00014.

[2] E. Andre, M. Rehm, W. Minker and D. Buthler, "Endowing spoken language dialogue systems with emotional intelligence," in *Affective Dialogue Systems. ADS 2004,* Lecture Notes in Computer Science, vol. 3068, 2004, pp. 178–187, Springer, Berlin, Heidelberg, https://doi.org/10.1007/978-3-540-24842-2_17.

[3] C. Guo, K. Zhang, J. Chen, R. Xu and L.Gao, "Design and application of facial expression analysis system in empathy ability of children with autism spectrum disorder", *Proceedings of the 16th Conference on Computer Science and Intelligence Systems*, Annals of Computer Science and Information Systems, vol. 25, 2021, pp. 319–325, http://dx.doi.org/10.15439/2021F91.

[4] S. Gu, F. Wang, N. P. Patel, J. A. Bourgeois and J. H. Huang, „A Model for Basic Emotions Using Observations of Behavior in Drosophila," *Frontiers in Psychology*, vol. 10, 2019, article 781, https://doir.org/10.3389/fpsyg.2019.00781.

[5] J. A. Russel, "Emotions are not modules," *Canadian Journal of Philosophy*, vol. 36, 2006, sup1, pp. 53–71, Routledge Publ. https://doi.org/10.1353/cjp.2007.0037.

[6] E.Y. Bann, "Discovering Basic Emotion Sets via Semantic Clustering on a Twitter Corpus," arXiv:1212.6527 [cs.AI], December 2012, https://doi.org/10.48550/arXiv.1212.6527.

[7] S. Lugovic, I. Dunder and M. Horvat, „Techniques and Applications of Emotion Recognition in Speech," *in 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO),* 2016, pp. 1278–1283, https://doi.org/10.1109/MIPRO.2016.7522336.

[8] U. Kamath, J. Liu and J. Whitaker, *Deep Learning for NLP and Speech Recognition,* Springer Nature Switzerland AG, Cham, 2019. https://doi.org/10.1007/978-3-030-14596-5.

[9] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, "Front-End Factor Analysis for Speaker Verification", *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, 2011, no. 4, pp. 788–798. http://dx.doi.org/10.1109/TASL.2010.2064307.

[10] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition", in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2018, pp. 5329–5333, http://dx.doi.org/10.1109/ICASSP.2018.8461375.

[11] B. J. Abbaschian, D. Sierra-Sosa and A. Elmaghraby, „Deep Learning Techniques for Speech Emotion Recognition, from Databases to Models," *Sensors*, mdpi, 2021, 21(4), https://doi.org/10.3390/s21041249.

[12] B. Schuller, S. Reiter, R. Muller, M. Al-Hames, M. Lang and G. Rigoll, "Speaker independent speech emotion recognition by ensemble classification," in *IEEE International Conference on Multimedia and Expo,* 2005, https://doi.org/10.1109/icme.2005.1521560.

[13] J. Rong, G. Li and Y.-P. P. Chen, „Acoustic feature selection for automatic emotion recognition from speech," *Information Processing and Management,* Elsevier, vol. 45, 2009, issue 3, pp. 315–328, https://doi.org/10.1016/j.ipm.2008.09.003.

[14] M. B. Akcay and K. Oguz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Communication,* Elsevier, vol. 116, January 2020, pp. 56–76, https://doi.org/10.1016/j.specom.2019.12.001.

[15] M. B. Er and I. B. Aydilek, "Music Emotion Recognition by Using Chroma Spectrogramand Deep Visual Features," *International Journal of Computational Intelligence Systems,* vol. 12, Issue 2, 2019, pp. 1622–1634, https://doi.org/163410.2991/ijcis.d.191216.001.

[16] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLoS ONE*, 13(5): e0196391, 2018, https://doi.org/10.1371/journal.pone.0196391.

[17] K. Dupuis and K. M. Pichora-Fuller, *Toronto emotional speech set (TESS),* University of Toronto, Psychology Department, Borealis data publisher, 2010, https://doi.org/10.5683/SP2/E8H2MF.

[18] S. Haq and P.J.B. Jackson, "Multimodal Emotion Recognition", in *W. Wang (ed), Machine Audition: Principles, Algorithms and Systems,* IGI Global Press, 2011, chapter 17, pp. 398–423, https://doi.org/10.4018/978-1-61520-919-4.

[19] LibRosa documentation, https://librosa.org/doc/.

[20] S. Burnval, *Speech emotion recognition*, (https://www.kaggle.com/shivamburnwal/speech-emotion-recognition)

[21] M. A. Kutlugün, Y. Sirin and M. A. Karakaya, "The Effects of Augmented Training Dataset on Performance of Convolutional Neural Networks in Face Recognition System", *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*, Annals of Computer Science and Information Systems, vol. 18, 2019, pp. 929–932, http://dx.doi.org/10.15439/2019F181.

[22] P. Shegokar and P. Sircar, "Continuous wavelet transform based speech emotion recognition", in *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS),* Dec. 2016, IEEE, pp. 1–8. https://doi.org/10.1109/ICSPCS.2016.7843306.

[23] Y. Zeng, H. Mao, D. Peng and Z. Yi, "Spectrogram based multi-task audio classification", *Multimedia Tools and Applications,* vol. 78 (2019), no. 3, pp. 3705–3722, https://doi.org/10.1007/s11042-017-5539-3.

[24] A. S. Popova, A. Rassadin and A. Ponomarenko, "Emotion Recognition in Sound", *International Conference on Neuroinformatics,* vol. 736, 2018, pp. 117–124, 2018, http://dx.doi.org/10.1007/978-3-319-66604-4_18.

[25] D. Issa, F. M. Demirci and A. Yazici, "Speech emotion recognition with deep convolutional neural networks", *Biomedical Signal Processing and Control,* Elsevier, vol. 59, 2020, 101894, doi:10.1016/j.bspc.2020.101894.

DOI: 10.15439/2023F3083
ISSN 2300-5963 ACSIS, Vol. 35

Detecting type of hearing loss with different AI classification methods: a performance review.

Michał Kassjański[1], Marcin Kulawiak[1], Tomasz Przewoźny[2], Dmitry Tretiakow[2],
Jagoda Kuryłowicz[2], Andrzej Molisz[2], Krzysztof Koźmiński[4],
Aleksandra Kwaśniewska[3],Paulina Mierzwińska-Dolny[4], Miłosz Grono[4]
[1] Department of Geoinformatics, Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology, Gdansk, Poland
[2] Department of Otolaryngology, Medical University of Gdańsk, Poland
[3] Department of Otolaryngology, Laryngological Oncology and Maxillofacial Surgery, University Hospital No. 2, Bydgoszcz, Poland
[4] Student's Scientific Circle of Otolaryngology, Medical University of Gdańsk, Poland
Email: {michal.kassjanski, markulaw}@pg.edu.pl, {tprzew, d.tret}@gumed.edu.pl, jagoda.kurylowicz@gmail.com,
{andrzej.molisz, krzyk}@gumed.edu.pl,
kwasniewska.aleks@gmail.com, {paulinamierzwinska, milosz.grono}@gumed.edu.pl

Abstract—**Hearing is one of the most crucial senses for all humans. It allows people to hear and connect with the environment, the people they can meet and the knowledge they need to live their lives to the fullest. Hearing loss can have a detrimental impact on a person's quality of life in a variety of ways, ranging from fewer educational and job opportunities due to impaired communication to social withdrawal in severe situations. Early diagnosis and treatment can prevent most hearing loss. Pure tone audiometry, which measures air and bone conduction hearing thresholds at various frequencies, is widely used to assess hearing loss. A shortage of audiologists might delay diagnosis since they must analyze an audiogram, a graphic representation of pure tone audiometry test results, to determine hearing loss type and treatment. In the presented work, several AI-based models were used to classify audiograms into three types of hearing loss: mixed, conductive, and sensorineural. These models included Logistic Regression, Support Vector Machines, Stochastic Gradient Descent, Decision Trees, RandomForest, Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Graph Neural Network (GNN), and Recurrent Neural Network (RNN). The models were trained using 4007 audiograms classified by experienced audiologists. The RNN architecture achieved the best classification performance, with an out-of-training accuracy of 94.46%. Further research will focus on increasing the dataset and enhancing the accuracy of RNN models.**

I. Introduction

HEARING is considered an essential sensory organ since it provides us with valuable information about the external environment. In addition, it enables us to interact with the outside world, communicate with others, remain safe, and derive enjoyment from a variety of auditory experiences. Hearing complements our other senses, such as sight and sensation, to provide a complete understanding of our surroundings.

According to the World Health Organization (WHO), more than 1.5 billion persons worldwide suffer from hearing loss, of which 430 million have moderate or severe hearing loss in their better hearing ear. According to the projections of the World Health Organization, by 2050 nearly 2.5 billion people will have hearing loss and at least 700 million will require rehabilitation services. Fortunately, many instances of hearing loss can be prevented through early detection and intervention [1].

Although the majority of ear diseases are curable, accurate diagnosis is a significant barrier to effective treatment. Audiologists, who are essential for the execution and interpretation of testing, are scarce worldwide. Approximately 93% of low-income countries have fewer than one audiologist per million people [1]. Given the disparity between the supply and demand for hearing specialists, artificial intelligence (AI) has the potential to resolve this problem. AI employs algorithms that enable computers to recognize particular data analysis patterns and make conclusions. The most prevalent AI application in tonal audiometry is hearing aid personalization, in which AI systems assist both the hearing-care expert and the patient in more precisely and efficiently adjusting hearing aids to the client's preferences [2, 3, 4].

Another possible application of expert systems in audiology is interpreting results of pure-tone audiometry, which is the standard method for diagnosing hearing loss. Typically, the examination is conducted while situated in an anechoic chamber. It entails conveying increasing-intensity pure tones through headphones and determining the threshold for air and bone conduction. In general, the results of the pure-tone audiometry test are presented as an inverted graph called an audiogram, which allows for identifying hearing impairment.

When describing hearing loss, three aspects are considered: the type of hearing loss, the degree of hearing loss, and the configuration of hearing loss. Three types of hearing loss are distinguished: sensorineural, conductive, and mixed. The pattern of hearing loss across frequencies is determined by the configuration (shape) of the audiogram, whereas the severity is determined by the degree of hearing loss [5].

Classification of automated audiometry data has been investigated for a very long time. In the past ten years, there have been a number of initiatives to develop an automated classification system sufficiently accurate for clinical application. The most successful have been presented by Elbaşı and Obali [6], who compared Decision Tree, Naive Bayes, and Neural Network Multilayer Perceptron (NN) models for determining hearing loss. The research was conducted on a data set containing 200 samples divided into four categories: normal hearing, conductive hearing loss, sensorineural hearing loss, and mixed hearing loss. The accuracy of the classification algorithms was 95.5% for Decision Tree, 86.5% for Naive Bayes, and 93.5% for NN. While that work used raw audiometry test results, Crowson et al. [7] applied the ResNet models to classify rasterized results in the form of audiogram images into four categories of hearing (normal, sensorineural hearing loss, conductive hearing loss, mixed hearing loss) on a set of 1007 audiograms. Instead of completely training the classifier from scratch, the authors used transfer learning to train the classifier using widely recognized raster classification models. This method achieved a classification accuracy of 97.5%, but it is limited to image analysis.

In conclusion, the combination of machine learning and increased computational resources in innovative hardware architectures has the potential to generate faster overall test results and more exhaustive evaluations in audiology [8]. Despite the type of hearing loss, the classification accuracy of the currently proposed solutions ranges from 86 to 97%, which, while extremely high, still leaves a substantial margin of error. Moreover, while the best available audiogram classifier, presented by Crowson et al. [7], achieved 97.5% accuracy, it cannot be applied to the original data series produced by tonal audiometry due to being an image classifier. This means that before classification the datasets would need to be converted into a particular format of audiogram images (although the structure of audiograms is generally analogous, audiograms generated by different software can vary quite significantly). Additional problems would stem from the fact that some types of software generate two audiograms (one for each ear), while other software combines the information from both ears into a single audiogram, posing a great difficulty in universal analysis. Consequently, an image classifier cannot form the core of a versatile solution for classifying tonal audiometry results. Moreover, the abovementioned studies on determining the type of hearing loss were carried out with a relatively small data set, ranging from 200 test results in Elbaşı & Obali [6] to 1007 in Crowson et al. [7], which might have led to an optimistic and uncertain evaluation of model performance.

This study establishes the benchmark for machine learning and deep learning algorithms using a large set of discrete tonal audiometry data series. Throughout the course of this investigation, multiple AI models were trained and evaluated using 4007 audiogram data series analyzed and classified by professional audiologists. The purpose of this study was to investigate the performance of various AI solutions when applied to raw tonal audiometry data.

II. MATERIALS & METHODS

A. Data

The study was carried out on 4007 data series containing the results of pure tone audiometry tests performed between 2017 and 2021 by clinicians at the Otolaryngology Clinic of the University Clinical Centre in Gdansk, Poland. The data class proportion is presented in Fig. 1. Conductive hearing loss only has 674 examples, while mixed hearing loss has 1594 and sensorineural hearing loss has 1739. Each patient provided a maximum of two test results, one for the left ear and one for the right, resulting in no duplication of data from the same patient and ensuring adequate data variety.

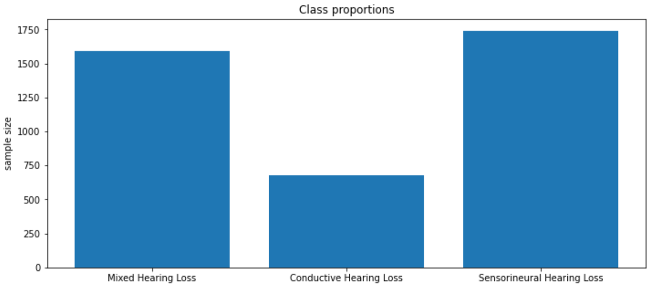


Figure 1. The three forms of hearing loss represented in the dataset, along with their respective proportions.

Tonal audiometry was used to evaluate patients' hearing according to the American Speech-Language-Hearing Association (ASHA) guidelines. All tests were conducted in soundproof chambers (ISO 8253, ISO 8253). The TDH39P headphones were utilized for air conduction testing, while the Radioear B-71 bone-conduction vibrator was used for bone conduction testing [9].

Experienced audiologists labeled the morphologies of hearing loss on the audiometry test results, dividing the set into three classes according to established methodology [5]: mixed hearing loss, conductive hearing loss and sensorineural hearing loss.

Typically, the results of pure-tone audiometry are depicted as an audiogram, which is a graphical representation of how loud sounds must be at various frequencies for them to be audible. In addition to a graphical representation, audiology software generates XML files that comprise all information regarding tonal points in the audiogram. This study processes raw audiometry data using XML files, analyzing five primary frequencies (250, 500, 1000, 2000, 4000 Hz) from both air conduction and bone conduction.

B. Methodology

The aim of the study was to test the performance of several different machine learning algorithms at the task of classifying tonal audiometry data. The goal of each method was to accurately categorize each dataset as mixed hearing loss (M), conductive hearing loss (C) or sensorineural hearing loss (S).

a) Machine learning algorithms

The initial phase of research involved testing the following machine learning classification algorithms: Gaussian Naive Bayes, K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machines (SVMs), Stochastic Gradient Descent (SGD), Decision Tree and Random Forest. The second phase of the study involved testing the following ANN architectures: Feedforward Neural Network (FNN), Convolutional Neural Network (CNN), Graph Neural Network (GNN), and Recurrent Neural Network (RNN). These techniques were previously applied to the classification problem of medical data [10, 11].

b) Data preprocessing

The input data series consisted of vertical information about tonal points of air and bone conduction, defined as volume (dB) for a given frequency (Hz), obtained from XML files. The frequency range of the dataset included 250Hz, 500Hz, 1000Hz, 2000Hz, and 4000Hz. Each frequency tested has been designated a loudness level between -10dB and 120dB. The dataset did not contain any empty values.

Since GNN requires graph input, the vector was turned into a directed graph with 10 nodes and 18 edges. Frequency and loudness values have been assigned to nodes. Figure 2 shows a graphical depiction of the graph.

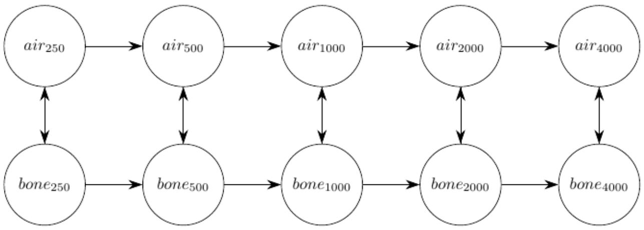


Figure 2. The GNN architecture's input graph structure.

c) Model evaluation

The performance of the tested models was evaluated using K-fold Cross-Validation, which is the process of splitting a dataset into K folds, using K-1 datasets for training and one for validation. The datasets are then rotated in consecutive tests, allowing for more accurate assessment of best, worst and average classification performance. Based on the magnitude of the dataset and the available computational resources, K was set to 5 in this study. Consequently, the ratio of train to test datasets is 80% to 20%, respectively.

III. Results and Discussion

The initial stage of research tested the classification performance of a set of machine learning algorithms. The results have been expressed in terms of accuracy, precision, recall, and F1 score. Due to the aforementioned class imbalance, macro averaging was calculated. The outcome of those tests is presented in Table I.

Receiver Operating Characteristics (ROC) curves with corresponding Area Under the Curve (AUC) parameters, displaying the discrimination performance of the tested machine learning models in terms of true positives vs false positives are presented in Fig. 3. The ROC Curve and the ROC AUC score are essential tools for evaluating binary classification models, but they can also be applied to multi-classification problems. OvR method was selected, which stands for "One versus the Rest" and is a method for evaluating multiclass models that evaluates each class in comparison to the others simultaneously. In this scenario, one class is deemed the "positive" class, while the other classes are deemed the "negative" class. This reduces the multiclass classification output to a binary classification output, allowing the use of all known binary classification metrics to assess this scenario [12].

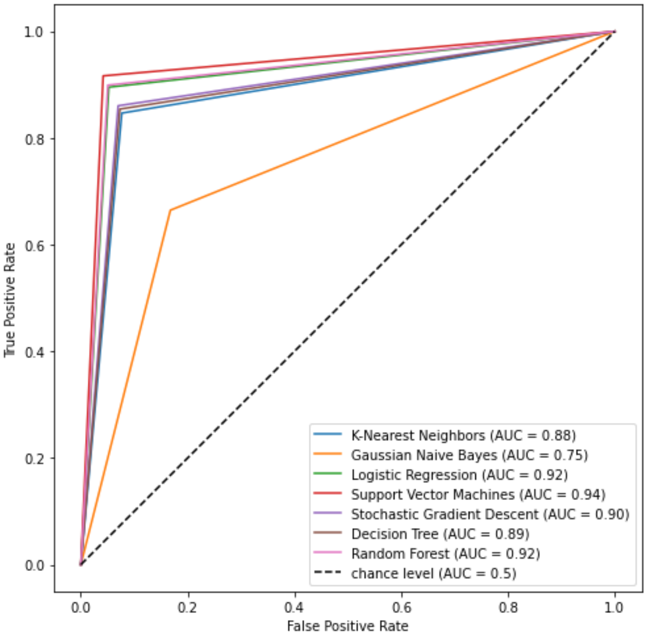


Figure 3. ROC curves with the AUC parameters for machine learning models.

As far as machine learning algorithms are concerned, the best results have been achieved by the Support Vector Machine classifier, which earned 83.38% accuracy. The algorithm also received best scores in precision, recall, F1, and AUC. The Logistic Regression and Random Forest models, which closely followed SVM, also scored above 80% accuracy.

TABLE I.
COMPARISON OF PERFORMANCE RESULTS OF MACHINE LEARNING MODELS. BEST RESULTS IN EACH CATEGORY HAVE BEEN HIGHLIGHTED IN GREEN

Algorithm	**Gaussian Naive Bayes**	**K-Nearest Neighbors**	**Logistic Regression**	**Support Vector Machines**	**Stochastic Gradient Descent**	**Decision Trees**	**Random Forest**
Accuracy	62.14% (+/- 8.43%)	74.40% (+/- 7.29%)	82.48% (+/- 7.21%)	**83.38%** **(+/- 6.21%)**	76.81% (+/- 7.78%)	79.49% (+/- 2.16%)	81.26% (+/- 4.46%)
Precision	87.68% (+/- 9.95%)	92.51% (+/- 5.92%)	94.74% (+/- 5.69%)	**94.97%** **(+/- 4.08%)**	90.96% (+/- 7.77%)	92.99% (+/- 5.68%)	94.27% (+/- 4.52%)
Recall	62.14% (+/- 8.43%)	74.40% (+/- 7.29%)	82.48% (+/- 7.21%)	**83.38%** **(+/- 6.21%)**	76.81% (+/- 7.78%)	79.49% (+/- 2.16%)	81.26% (+/- 4.46%)
F1	71.06% (+/- 5.32%)	81.12% (+/- 4.51%)	87.38% (+/- 5.62%)	**88.05%** **(+/- 3.76%)**	80.51% (+/- 9.62%)	85.16% (+/- 2.35%)	86.58% (+/- 2.70%)

Stochastic Gradient Descent and K-Nearest Neighbors achieved accuracy of 76.81% and 74.40%, respectively, which puts them well behind the three leading methods, but still a league above Gaussian Naive Bayes which scored only 62% accuracy.

It is worth noting that tree-based classifiers have shown the best accuracy stability in terms of 5-Fold validation, with approximately 2% standard deviation in Decision Tree and around 4.5% in Random Forest, whereas for all other models this parameter exceeds 6%. The problem of unbalanced data, which is definitely present in this study, is one of the elements that could have a negative impact on the scores of machine learning algorithms, which is particularly evident e.g. in the poor performance of Gaussian Naive Bayes.

The second phase of research involved deep learning architectures such as FNN, CNN, GNN, and RNN, which were examined using the same criteria as machine learning models. The results of these tests are shown in Table II. The ROC curves with AUC parameters are presented in Fig. 4.

Concerning the tested artificial neural network models, RNN performed best in terms of accuracy, precision, recall, F1 score and AUC, with 94.46% accuracy and 94.45% F1 score. This was to be expected, as the input datasets could be considered sequential data, which is a known strength of RNN [13]. These results also confirm the findings of a recent study [14], which evaluated different neural network designs in order to develop a binary classifier for normal and pathological hearing loss based on similar data, where the best results were also achieved by the RNN architecture. The second best model was CNN with roughly one percentage point less, which may be a little surprising given that CNNs are generally employed to evaluate images. This may be explained by the fact that CNNs perform best when processing data matrices, and the input datasets could be interpreted as small (5x2) matrices. FFN generally achieved third place, while GNN achieved the worst scores.

The overall performance differences between machine and deep learning models are largely in favor of artificial neural networks, with the exception of GNN, which remained at the level of machine learning techniques. The achieved results differ significantly from previous research (performed by Elbaşı and Obali [6]), which achieved 95.5 % accuracy in classifying raw audiometry data with Decision Tree. It should be noted, however, that the validity of those results may be questioned because they were obtained on only 200 samples, which is 20 times less than the dataset used in the current work. Furthermore, there is no information on the class proportion and the employed cross validation process.

TABLE II.
COMPARISON OF PERFORMANCE RESULTS OF DEEP LEARNING MODELS. BEST RESULTS IN EACH CATEGORY HAVE BEEN HIGHLIGHTED IN GREEN

Model	**FFN**	**CNN**	**GNN**	**RNN**
Accuracy	89.67% (+/-2.12%)	93.46% (+/- 0.83%)	83.15% (+/- 9.09%)	**94.46%** **(+/-0.91%)**
Precision	90.27% (+/-1.78%)	93.50% (+/- 0.83%)	86.04% (+/- 4.68%)	**94.50%** **(+/- 0.91%)**
Recall	89.67% (+/-2.12%)	93.46% (+/- 0.83%)	83.15% (+/- 9.09%)	**94.46%** **(+/- 0.91%)**
F1	89.71% (+/-2.09%)	93.46% (+/- 0.83%)	82.15% (+/- 11.02%)	**94.45%** **(+/- 0.91%)**

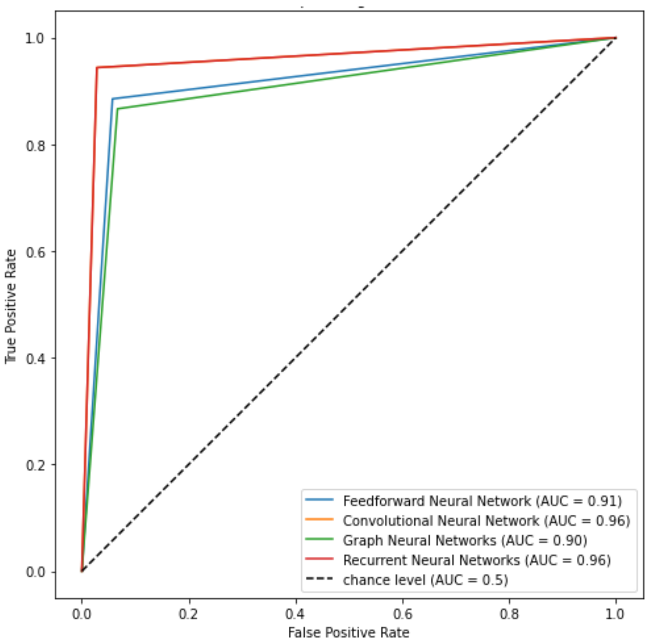


Figure 4. ROC curves with the AUC parameters for deep learning models.

In the above context, while best accuracy of 94,46%, achieved by RNN, is lower than the current state of the art in classification of audiometry test results (97.5%) held by Crowson et al. [7] for raster datasets, that score could be put in question as well. The most significant challenge with training deep learning models from scratch is that it must be done on a large dataset, or else it may miss important patterns. Reliable training of ANN classification models usually requires datasets consisting of at least 10000 samples. For raster datasets this may be alleviated somewhat by employing augmentation of much smaller datasets (which was the strategy applied by Crowson et al. [7]). Unfortunately, this method works best if the input dataset was sufficiently representative. In this case, various types of audiometry software can generate significantly different images, ranging from minor differences in plot color and measurement point indicator size to changes that can significantly impair the performance of an automated classifier, such as displaying test results from both ears on a single plot. As a result, unless an appropriately comprehensive audiogram database is constructed (which would require collection and classification of hundreds of thousands of audiograms produced by all types of audiometry software), image-trained classification models will only work with certain types of audiometry data. In comparison, a classifier which operates on raw audiometry data allows for more flexible and wider application in the clinical environment. This being said, the best classification accuracy of 94,46%, which was achieved in this test by RNN, could be considered too low for clinical application due to a prohibitively large number of false negatives. The latter would suggest that producing a reliably accurate raw audiometry data classifier will require constructing an appropriately large and representative training dataset.

IV. Conclusion

The presented work aimed to test several AI-based algorithms for classification of discrete tonal audiometry data series into three types of hearing loss: sensorineural, conductive, and mixed. In the course of this study, several different machine and deep learning models, including Gaussian Naive Bayes, K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Stochastic Gradient Descent, Decision Trees, Random Forest, Feedforward Neural Network, Convolutional Neural Network, Graph Neural Network, and Recurrent Neural Network, have been trained and tested with the use of 4007 audiometry data series analyzed and classified by professional audiologists. The highest classification accuracy was achieved with Recurrent Neural Network at 94.46% (+/-0.91%). The results of the study verified the general hierarchy of classification performance established by prior research, however they also suggest that the previously reported levels of classification accuracy (achieved for vastly inferior dataset sizes) might have been overly optimistic. In the above context, further work will concentrate on expanding the dataset and improving RNN models in terms of accuracy.

References

[1] World Health Organization. 2021. World report on hearing. https://www. who.int/publications/i/item/world-report-on-hearing.

[2] Guo, R., Liang, R., Wang, Q. et al. 2023. Hearing loss classification algorithm based on the insertion gain of hearing aid. Multimed Tools Appl, http://dx.doi.org/10.1007/s11042-023-14886-0

[3] Belitz, C., Ali, H., Hansen, J. H. L. 2019. A Machine Learning Based Clustering Protocol for Determining Hearing Aid Initial Configurations from Pure-Tone Audiograms. Interspeech, 2325–2329, http://dx.doi.org/10.21437/interspeech.2019-3091

[4] Elkhouly, A., Andrew, A.M., Rahim, H.A. et al. 2023. Data-driven audiogram classifier using data normalization and multi-stage feature selection. Sci Rep 13, 1854, http://dx.doi.org/10.1038/s41598-022-25411-y

[5] Margolis, R. H., Saly, G. L. 2007. Toward a standard description of hearing loss. International journal of audiology, 46(12), 746–758, http://dx.doi.org/10.1080/14992020701572652

[6] Elbaşı, E.¸ Obali, M. 2012. Classification of Hearing Losses Determined through the Use of Audiometry using Data Mining, Conference: 9th International Conference on Electronics,Computer and Computation

[7] Crowson, M.G., Lee J.W., Hamour A., Mahmood, R., Babier, A., Lin, V., Tucci, D.L., Chan, T.C.Y. 2020. AutoAudio: Deep Learning for Automatic Audiogram Interpretation. J Med Syst. 44(9):163, http://dx.doi.org/10. 1007/s10916-020-01627-1

[8] Barbour, D. L., Wasmann, J. W. 2021. Performance and Potential of Machine Learning Audiometry, The Hearing Journal: Volume 74 - Issue 3 - p 40,43,44, http://dx.doi.org/10.1097/01.HJ.0000737592.24476.88

[9] Guidelines for manual pure-tone threshold audiometry. (1978). ASHA, 20(4), 297–301

[10] Ciszkiewicz A., Milewski G., Lorkowski J., 2018. Baker's Cyst Classification Using Random Forests, 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), Poznan, Poland, 2018, pp. 97-100, http://dx.doi.org/10.15439/2018F89

[11] Kučera E., Haffner O., Stark E., 2017. A method for data classification in Slovak medical records, 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Czech Republic, 2017, pp. 181-184, http://dx.doi.org/10.15439/2017F44.

[12] Landgrebe, T.C., Duin, R.P. 2006. A simplified extension of the Area under the ROC to the multiclass domain

[13] Al-Askar, H., Radi, N. MacDermott, A. 2016. Chapter 7 - Recurrent Neural Networks in Medical Data Analysis and Classifications, In Emerging Topics in Computer Science and Applied Computing, Applied Computing in Medicine and Health, Morgan Kaufmann,147-165, 9780128034682, http://dx.doi.org/10.1016/B978-0-12-803468-2.00007-2

[14] Kassjański, M., Kulawiak, M., Przewoźny, T. 2022. Development of an AI-based audiogram classification method for patient referral, 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, pp. 163-168, http://dx.doi.org/10.15439/2022F66.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1023–1027

DOI: 10.15439/2023F539
ISSN 2300-5963 ACSIS, Vol. 35

Dynamic SITCOM: an innovative approach to re-identify social network evaluation models

Bartłomiej Kizielewicz
0000-0001-5736-4014
Dept. of Artificial Intelligence Methods and Applied Mathematics,
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: bartlomiej-kizielewicz@zut.edu.pl

Jarosław Jankowski
0000-0002-3658-3039
Dept. of Information Systems Engineering,
West Pomeranian University of Technology in Szczecin
ul. Żołnierska 49, 71-210 Szczecin, Poland
Email: jjankowski@zut.edu.pl

Abstract—**Complex networks attract attention in various scientific fields due to their ability to model real world phenomena and potential for problem-solving. It is essential to evaluate these networks to simulate and solve various issues. Evaluating social networks is challenging due to the unequal status of nodes and their unknown impact on everall characteristics. Existing measures of centrality often need to consider the global structure of the network, which requires the involvement of experts and creates space for multi-criteria decision-making methods usage. Unfortunately, more access to established decision-making models is often needed for various reasons. In this article, we propose an innovative approach called Dynamic Stochastic IdenTifiCation Of Models (Dynamic SITCOM), which considers the preferences of characteristic objects and the characteristic values of criteria, enabling the re-identification of multi-criteria decision models. The approach evaluates nodes in Facebook's complex social network, focusing on prediction accuracy using similarity measures and Mean Absolute Error. The study shows that a stable decision model can be created and applied to evaluate nodes in complex networks.**

I. Introduction

COMPLEX networks have attracted the interest of researchers from various scientific fields, such as biology, sociology, physics, and computer science [1]. Their occurrence in a wide variety of fields makes evaluating and analyzing these networks of great importance. Evaluating complex networks is essential in many fields because it allows for simulating and solving various problems [2].

For example, assessing the importance of nodes in the context of power networks makes it possible to identify critical points whose failure could lead to network shutdown. In the case of communication networks, assessing the importance of nodes makes it possible to optimally maintain connections and prevent disruptions in the flow of information. Complex networks also have applications in preventing the spread of rumors or diseases [2]. By identifying and assessing the importance of crucial nodes, it is possible to influence the control and limit the propagation of such phenomena. In addition, complex networks are widely present in the social domain, where assessing the importance of nodes is crucial for identifying opinion leaders, influential individuals, or experts in a community.

Due to the growing popularity of social media, it has become one of the most effective marketing tools. Using visual content on platforms such as Twitter, Facebook, and Instagram can help companies build their image and increase the reach of their brand. Social media allows companies to connect directly with customers and monitor product and service feedback [3]. However, one of the main problems with social networks is viral marketing, which involves using social networks to spread information about a product or service through users who pass the information on to their friends.

In addition, companies promoting themselves on social media have begun using virtual influencers to advertise products and services. Through them, companies can reach new audiences and increase the reach of their brand. However, the use of virtual influencers is controversial among consumers, who believe it is a scam and lacks authenticity.

Therefore, evaluating social networks is particularly important to identify critical nodes that play a crucial role in the network. This, in turn, allows us to control and limit the spread of rumors or use them for marketing purposes [4]. By identifying key nodes, we can also increase the reach of positive information and make positive community changes.

Due to the statuses of nodes found in complex networks, which are unequal and different, a problem arises in evaluating them. The main methods used are methods of evaluating nodes based on measures of node centrality. The most popular centrality measures used to evaluate complex networks are degree centrality, inter-node centrality, proximity centrality, PageRank centrality [5], Katz centrality [6], and k-shell [7].

Unfortunately, although centrality criteria are widely used, they have some shortcomings and deficiencies [2]. Measures of node centrality, such as degree centrality, do not always consider the global structure of the network [8], [9]. Therefore, it is common to use the knowledge of domain experts to evaluate key nodes based on the information gathered by selected centrality metrics. Multi-criteria decision analysis methods are also used, using some aggregation of centrality metrics to determine network node ratings [10].

Therefore, this article proposes a new Dynamic SITCOM approach to re-identify the decision model based on the evaluated decision variants. The main novelty of the Dynamic

SITCOM approach is the two-stage optimization. In the case of the baseline Stochastic IdenTifiCation Of Models (SITCOM) approach, optimization is based only on the preferences of characteristic objects that reflect the decision maker's preferences [11], [12]. In contrast, the proposed approach also optimizes the characteristic values responsible for the position of characteristic objects in the decision grid. This leads to the possibility of considering more non-linear problems.

The proposed approach will be applied to the problem of evaluating Facebook nodes in a complex network. For this problem, the expert evaluates nodes with a specific model based on four criteria reflected in the form of centrality metrics. This model is unavailable, so the Dynamic SITCOM approach used is to re-identify it. The study focuses on the accuracy of this approach using similarity measures of rankings and Mean Absolute Error (MAE).

The rest of the article is organized as follows. Section II presents the state of the art of MCDA/MCDM methods related to the topic of node evaluation in complex networks and a brief introduction of Stochastic IdenTifiCation Of Models (SITCOM). Section III presents a proposal for the Dynamic SITCOM approach. Section IV presents research related to the accuracy of the Dynamic SITCOM approach in the problem of evaluating the nodes of the Facebook complex network. The V section presents conclusions and future research directions.

II. Preliminaries

A. State of the art

Multi-criteria decision analysis/Multi-criteria decision-making (MCDA/MCDM) methods assess node importance in complex networks by using centrality metrics, such as node degree, betweenness, and closeness. These methods aggregate multiple criteria to provide a comprehensive evaluation of node significance, aiding in the identification of crucial nodes for network performance. Table I presents examples of the use of MCDA/MCDM methods for complex network evaluations. Khaoula et al. proposed a novel seed-centered approach based on TOPSIS and the k-means algorithm to find communities in a social network [13]. Zhang and Ng proposed a ranking method based on entropy weights and the TOPSIS approach, named EWM-TOPSIS, to evaluate the criticality of nodes considering various node characteristics in complex public transportation networks (PTNs) [14]. Lu used the TOPSIS method to evaluate and compare the ARPA network and the standard IEEE 39 bus system [15]. Meng et al. used the WTOPSIS approach to evaluate complex networks in urban rail transit (URT) [16]. Mi et al. used the VIKOR approach to evaluate a road network with 28 intersections in Shenzhen [17]. Kharanagh et al. proposed using SAW, TOPSIS, and ELECTRE I approaches to analyze social networks in water resources management [18]. Lin et al. used the CRITIC approach to assess the importance of nodes in reconfiguring the electricity grid backbone network [19].

B. Stochastic IdenTifiCation Of Models (SITCOM)

Stochastic IdenTifiCation Of Models (SITCOM) is a new approach to re-identify a decision model based on evaluated decision variants [11], [12]. This approach's operation mechanism is based on the logic of the selected stochastic optimization algorithm and Characteristic Object METhod (COMET). The stochastic algorithm determines the preferences of the Characteristic Objects (CO), which in the case of the COMET method, represents the preferences of the decision maker. Then, when selecting appropriate values of characteristic object preferences is over, it is possible to evaluate new decision variants. A full description of the algorithm can be found in the initial papers [11], [12].

III. Dynamic SITCOM

In this article, we propose to extend the above SITCOM approach with additional optimization. Since the base SITCOM approach only uses characteristic object preference values for optimization, the model may not consider some non-linearity occurring in expert knowledge. Therefore, the present proposition is based on the characteristic objects building factor, i.e., the characteristic values of the criteria. The characteristic values are mainly responsible for the model's grid and irregularity. In addition, the starting and ending values included in the vector of characteristic values of the COMET method are responsible for the boundaries of the model. Therefore, in the proposed approach, in addition to optimizing the preference of characteristic objects, we will focus on optimizing the middle characteristic values of the model.

The proposed method is based on a two-stage optimization. The first optimization, as in the case of the original SITCOM, will be based on the search for the best possible preferences of characteristic objects. The second optimization, on the other hand, will focus on the search for the best possible middle values for the characteristic values of the considered criteria. Due to the grid change, a loop was applied to query the optimized models to change the preferences of the characteristic objects with the newly found middle values for the characteristic values and vice versa.

IV. Study case

In this paper, we will focus on investigating the accuracy of the Dynamic SITCOM approach based on the problem of evaluating nodes of complex networks. First, it will get the selected dataset presented, and the study will be conducted in the next section.

A. Dataset description

The selected dataset concerns a complex network, which consists of nodes that are Facebook profiles. This dataset is anonymized and derived from the paper [20]. It consists of 4039 nodes and 88234 edges that connect the selected nodes. The network and the degree of the given nodes can be represented by Fig. 1.

TABLE I
EXAMPLES OF THE USE OF MCDA/MCDM METHODS IN EVALUATING NODES OF COMPLEX NETWORKS.

MCDA approach	No. of nodes	No. of criteria	Problem	Year	Reference
TOPSIS	4039	4	Evaluation of nodes from Facebook's network	2023	[13]
EWM–TOPSIS	95	3	Evaluation of the MTR network in Hong Kong	2021	[14]
TOPSIS	21	7	Evaluation of the ARPA network and the standard IEEE 39-bus system	2020	[15]
WTOPSIS	118, 132, 166	4	Evaluation of the Shenzhen Metro System	2020	[16]
VIKOR	28	3	Analysis of traffic safety at intersections	2020	[17]
SAW, TOPSIS, ELECTRE I	54, 30, 32, 26	12	Social network analysis of water resources management	2019	[18]
CRITIC	66	7	Evaluation of the Guangdong power system in China	2017	[19]

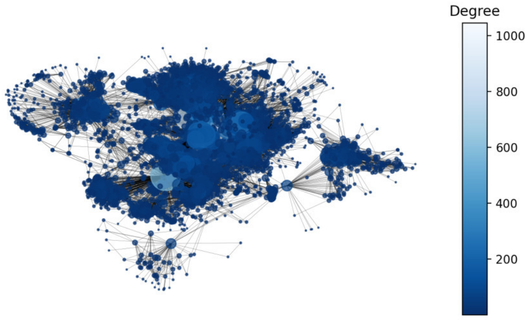


Fig. 1. Facebook's complex network of anonymized profiles [20].

Due to the problem of evaluating the nodes of the present network, four metrics were selected to serve as criteria. The selected network centrality metrics are degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality.

A decision matrix incorporating criteria and preferences, including expert ratings, will guide the re-identification of the decision model. It utilizes min-max normalization for all criteria, employing stochastic optimization in the training process. Table II displays the first ten decision variants with normalized values for criteria and preferences.

TABLE II
EXAMPLE 10 ALTERANTIVES.

$\mathbf{A_i}$	$\mathbf{C_1}$	$\mathbf{C_2}$	$\mathbf{C_3}$	$\mathbf{C_4}$
A_1	3.044754e-01	0.331418	0.000356	0.622104
A_2	5.792237e-06	0.015326	0.000006	0.295339
A_3	1.580590e-07	0.008621	0.000002	0.294918
A_4	3.506768e-06	0.015326	0.000007	0.295339
A_5	3.829891e-07	0.008621	0.000002	0.294918
A_6	4.590804e-06	0.011494	0.000012	0.295098
A_7	5.106522e-08	0.004789	0.000002	0.294678
A_8	3.544060e-04	0.018199	0.000269	0.342924
A_9	5.744837e-07	0.006705	0.000002	0.294798
A_{10}	3.424270e-05	0.053640	0.000023	0.297750
...	...	...	...	...

B. Research on the accuracy of the approach

This research case will investigate the accuracy of the proposed Dynamic SITCOM approach. For this purpose, the selected stochastic algorithm for re-identification of the multi-criteria model is the genetic algorithm. For optimization in the genetic algorithm in determining the preferences of characteristic objects and the means of characteristic values, 50 chromosomes were selected. On the other hand, for each criterion, the characteristic values were defined as a set of 0, 0.5, 1 because the criterion values were normalized. The implementations used in this study are from the *mealpy* library (genetic algorithm: `BaseGA`) [21] and the *pymcdm library* (COMET method) [22]. The related study evaluates subsets of test collections derived from tenfold cross-validation. The entire set in this study is divided into two parts, i.e., the training part (90 percent of the original set) and the testing part (10 percent of the original set). This division made was 10 times, where the selection of Folds is generated using the sklearn library. The subsets of the train and test sets are drawn 1,000 times and have 10,15,25,50,100 decision variants. These subsets evaluated were to use the learned SITCOM model on the selected Fold training set. Their evaluation is then compared with a reference evaluation determined subjectively by the expert using the MAE measure. In addition, the output evaluation from the learned decision variant model and the expert evaluation ranked is, and their similarity examined is using the r_w and WS measures. The training set has high quality and was similar to the test set, so it was decided to present only the research on the test set.

The results of the test set presented are in Tables III, IV and V for different numbers of randomly selected alternatives: 10, 15, 25, 50, and 100. The tests repeated were 1000 times. The Tables contain information on accuracy, expressed by the r_w, WS, and MAE metrics. Analyzing the MAE metric in the present case, the minimum MAE was smallest for 10 alternatives and was 0.002259, while the largest minimum error occurred for 100 alternatives and was 0.003226. The average values of the MAE for all the numbers of alternatives considered ranged from 0.021608 to 0.021710. As for the maximum values, the largest value was reached for 10 alternatives, while the smallest value occurred for 100 alternatives. The standard deviation indicates the spread of the results around the mean value of the MAE, which was approximately 0.012 for all cases.

TABLE III
MAE VALUES FOR SELECTED 1000 DRAWS OF GIVEN NUMBERS OF ALTERNATIVES FROM THE TEST SET.

No. of alts.	Min	Mean	Max	Std
10	0.002259	0.021665	0.059992	0.012677
15	0.002419	0.021710	0.057319	0.012469
25	0.002580	0.021608	0.055091	0.012236
50	0.002832	0.021657	0.050384	0.012166
100	0.003226	0.021674	0.048271	0.012082

The results of the r_w ranking similarity metric for randomly selected alternatives from the test set shown are in the following table. The table contains the minimum, mean, and maximum values of the r_w metric and the standard deviation. The table shows that the smallest minimum values of the r_w metric achieved were for 10 random alternatives, where they amounted to 0.388430. In comparison, the most significant minimum values occurred for 100 alternatives, where they reached a value of 0.929653. The average values of the r_w metric for all the considered numbers of alternatives range from 0. 985462 to 0.992024. Virtually all maximum values obtained were equal to 1, except for 100 alternatives, where the highest value was 0.999972. The standard deviation indicates the spread of results around the average value of the r_w metric, which ranged approximately from 0.010770 to 0.037956 for the different cases.

TABLE IV
r_w VALUES FOR SELECTED 1000 DRAWS OF GIVEN NUMBERS OF ALTERNATIVES FROM THE TEST SET.

No. of alts.	Min	Mean	Max	Std
10	0.388430	0.985462	1.000000	0.037956
15	0.638393	0.987587	1.000000	0.028707
25	0.656923	0.989946	1.000000	0.019518
50	0.871222	0.991282	1.000000	0.014129
100	0.929653	0.992024	0.999972	0.010770

Examining Table V, it can be seen that the minimum values of the WS metric for the various numbers of randomly selected alternatives range from 0.418711 to 0.888070. The average values of the WS metric for all the numbers of alternatives considered ranged from 0. 978925 to 0.994398. All the obtained maximum values of the WS metric equal 1. The standard deviation shows the spread of the results around the average value of the WS metric, which ranged from 0.007709 to 0.048627 for the different cases.

TABLE V
WS VALUES FOR SELECTED 1000 DRAWS OF GIVEN NUMBERS OF ALTERNATIVES FROM THE TEST SET.

No. of alts.	Min	Mean	Max	Std
10	0.418711	0.978925	1.0	0.048627
15	0.493393	0.980259	1.0	0.041311
25	0.511516	0.983272	1.0	0.032187
50	0.571031	0.988155	1.0	0.020861
100	0.888070	0.994398	1.0	0.007709

The visualizations associated with the r_w, WS, and MAE measures for randomly selected alternatives from the test set, repeated 1,000 times, are shown in Figs. 2, 3, and 4. Analyzing the MAE measure, the most accurate model was obtained for Fold number 2, while the least accurate model obtained was for Fold number 10. The smallest number of outliers was observed for Fold number 8, while the most significant was for Fold number 9. Turning to the similarity measure of rankings r_w, the lowest similarity of rankings occurred for Fold numbers 6 and 10, while the highest similarity of rankings observed was for Fold number 7.

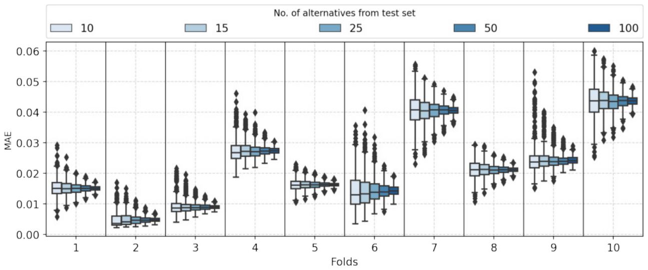


Fig. 2. Distributions of MAE values for selected 1000 draws of given numbers of alternatives from the test set for 10-fold crosvalidation.

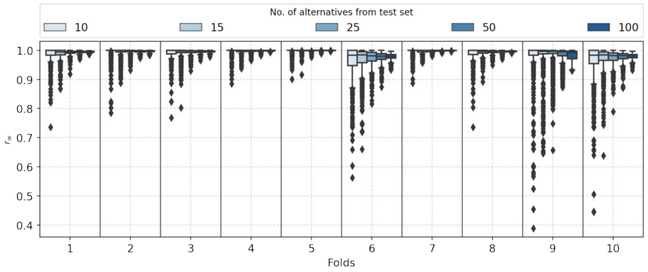


Fig. 3. Distributions of r_w values for selected 1000 draws of given numbers of alternatives from the test set for 10-fold crosvalidation.

V. CONCLUSIONS AND FUTURE RESEARCH

The study related to Dynamic SITCOM shows that it is possible to create a stable decision model for complex network nodes. Several conclusions can be taken by analyzing the presented research results related to the proposed Dynamic SITCOM approach. The first is that the larger the number of randomly selected alternatives, the smaller the value of the

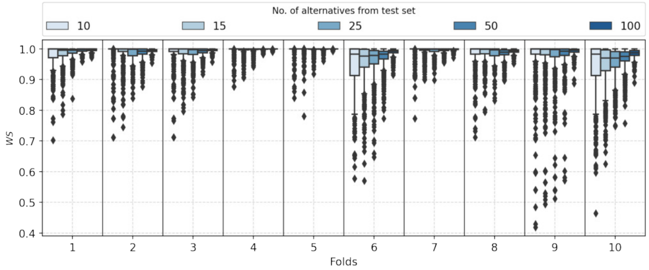


Fig. 4. Distributions of WS values for selected 1000 draws of given numbers of alternatives from the test set for 10-fold crosvalidation.

maximum MAE, suggesting that a more significant number of alternatives leads to better prediction accuracy. However, the average values of the MAE for all the numbers of alternatives considered are very close, indicating the stability of the model.

A similar trend observed is for the similarity measures of the r_w and WS rankings, where a more significant number of alternatives has larger minimum, average and maximum values. The maximum values are close to 1 for all cases, meaning the model represents reality well.

The conclusion is that a more significant number of randomly selected nodes presented as decision alternatives lead to better prediction accuracy and a more accurate reflection of the decision maker's preferences in the Dynamic SITCOM approach. At the same time, the models achieve stable results, as evidenced by the low variability of mean values and low standard deviation.

Future research directions could focus on considering more characteristic values for optimization. In addition, compromise solutions should be considered for characteristic objects with similar criterion values. It is also necessary to consider the applicability of the Dynamic SITCOM approach to other multi-criteria problems, such as selecting suppliers or creating a stable recommendation system.

References

[1] Z. Liu, C. Jiang, J. Wang, and H. Yu, "The node importance in actual complex networks based on a multi-attribute ranking method," *Knowledge-Based Systems*, vol. 84, pp. 56–66, 2015.

[2] Y. Yang, L. Yu, Z. Zhou, Y. Chen, T. Kou *et al.*, "Node importance ranking in complex networks based on multicriteria decision making," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[3] L. Wang, Z. Yu, F. Xiong, D. Yang, S. Pan, and Z. Yan, "Influence spread in geo-social networks: a multiobjective optimization perspective," *IEEE Transactions on Cybernetics*, vol. 51, no. 5, pp. 2663–2675, 2019.

[4] A. Zareie, A. Sheikhahmadi, and K. Khamforoosh, "Influence maximization in social networks based on TOPSIS," *Expert Systems with Applications*, vol. 108, pp. 96–107, 2018.

[5] M. Zhang, T. Huang, Z. Guo, and Z. He, "Complex-network-based traffic network analysis and dynamics: A comprehensive review," *Physica A: Statistical Mechanics and its Applications*, p. 128063, 2022.

[6] J. Zhao, T.-H. Yang, Y. Huang, and P. Holme, "Ranking candidate disease genes from gene expression and protein interaction: a Katz-centrality based approach," *PloS one*, vol. 6, no. 9, p. e24306, 2011.

[7] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," *Nature physics*, vol. 6, no. 11, pp. 888–893, 2010.

[8] P. Yang, X. Liu, and G. Xu, "A dynamic weighted TOPSIS method for identifying influential nodes in complex networks," *Modern Physics Letters B*, vol. 32, no. 19, p. 1850216, 2018.

[9] J. Zhang, Q. Zhang, L. Wu, and J. Zhang, "Identifying influential nodes in complex networks based on multiple local attributes and information entropy," *Entropy*, vol. 24, no. 2, p. 293, 2022.

[10] Y. Du, C. Gao, Y. Hu, S. Mahadevan, and Y. Deng, "A new method of identifying influential nodes in complex networks based on TOPSIS," *Physica A: Statistical Mechanics and its Applications*, vol. 399, pp. 57–69, 2014.

[11] B. Kizielewicz, "Towards the identification of continuous decisional model: the accuracy testing in the SITCOM approach," *Procedia Computer Science*, vol. 207, pp. 4390–4400, 2022.

[12] B. Kizielewicz and W. Sałabun, "A new approach to identifying a multi-criteria decision model based on stochastic optimization techniques," *Symmetry*, vol. 12, no. 9, p. 1551, 2020.

[13] A. Khaoula, M. MACHKOUR, and J. ANTARI, "Unsupervised Learning-based New Seed-Expanding Approach using Influential Nodes for Community Detection in Social Networks," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023.

[14] Y. Zhang and S. T. Ng, "Identification and quantification of node criticality through EWM–TOPSIS: a study of Hong Kong's MTR system," *Urban Rail Transit*, vol. 7, no. 3, pp. 226–239, 2021.

[15] M. Lu, "Node importance evaluation based on neighborhood structure hole and improved TOPSIS," *Computer Networks*, vol. 178, p. 107336, 2020.

[16] Y. Meng, X. Tian, Z. Li, W. Zhou, Z. Zhou, and M. Zhong, "Exploring node importance evolution of weighted complex networks in urban rail transit," *Physica A: Statistical Mechanics and its Applications*, vol. 558, p. 124925, 2020.

[17] X. Mi, C. Shao, C. Dong, C. Zhuge, and Y. Zheng, "A framework for intersection traffic safety screening with the implementation of complex network theory," *Journal of advanced transportation*, vol. 2020, pp. 1–12, 2020.

[18] S. G. Kharanagh, M. E. Banihabib, and S. Javadi, "An MCDM-based social network analysis of water governance to determine actors' power in water-food-energy nexus," *Journal of Hydrology*, vol. 581, p. 124382, 2020.

[19] Z. Lin, F. Wen, H. Wang, G. Lin, T. Mo, and X. Ye, "CRITIC-based node importance evaluation in skeleton-network reconfiguration of power grids," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 2, pp. 206–210, 2017.

[20] J. Leskovec and J. Mcauley, "Learning to discover social circles in ego networks," *Advances in neural information processing systems*, vol. 25, 2012.

[21] N. V. Thieu and S. Mirjalili, "MEALPY: a Framework of The State-of-The-Art Meta-Heuristic Algorithms in Python," Jun. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6684223

[22] B. Kizielewicz, A. Shekhovtsov, and W. Sałabun, "pymcdm—The universal library for solving multi-criteria decision-making problems," *SoftwareX*, vol. 22, p. 101368, 2023.

DOI: 10.15439/2023F6946
ISSN 2300-5963 ACSIS, Vol. 35

IoTrust - a HW/SW framework supporting security core baseline features for IoT

Mateusz Korona*, Bartosz Zabołotny†, Fryderyk Kozioł ‡, Mateusz Biernacki§,
Radosław Giermakowski¶, Paweł Rurka‖ Marta Chmiel**, Mariusz Rawski††
*0000-0002-9718-9684, †0000-0002-3364-1766, ‡0000-0002-6312-2406, §0009-0008-7765-800X,
¶0009-0001-3997-4369, ‖0009-0000-5014-8255, **0000-0002-9718-9684, ††0000-0002-7489-0785,
Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
Email:{mateusz.korona, bartosz.zabolotny, fryderyk.koziol, mateusz.biernacki,
radoslaw.giermakowski, pawel.rurka, marta.chmiel, mariusz.rawski}@pw.edu.pl

***Abstract*—The rapid growth of the Internet of Things has significant security implications. In the current IoT security landscape, many institutions and entities are defining security requirements, but no industry-wide standard has been agreed upon. There are solutions in the present state-of-the-art that fulfill a subset of secure IoT device requirements, but none adheres to all of them. However, the existing technologies introduced by those solutions could be combined to create a design framework which provides security baseline features to support requirements of a secure IoT device. In this paper, a configurable and comprehensive hardware-software security framework is proposed, that, when applied in the process of designing System on Chip for IoT, will ensure its cybersecurity by providing security core baseline features. The proposed sollution is CPU-agnostic, in the sense that no assumptions are made about the CPU's support for privilege levels, memory protection schemes, or any security mechanisms.**

I. Introduction

IoT devices are becoming an increasingly important aspect of our lives and can be sensed everywhere around us.

Due to the inherent characteristics of IoT devices, data is continuously transmitted, processed, and stored in the cloud. Studies have indicated that many IoT devices that have been compromised lack adequate security measures. IoT security is not just device security, as all elements need to be considered, including the device, cloud, mobile application, network interfaces, software, use of encryption, use of authentication, and physical security. Recent research directions in IoT focus on addressing these challenges and improving the performance and security of IoT systems [1].

Much research focuses on software, network, and cloud security; however, hardware security in these devices has been overlooked. Although software-based solutions are less expensive to implement and update, they have their limitations and are also more vulnerable to attacks. Hardware-based solutions may be more expensive and time-consuming to implement, but integrating hardware security can significantly strengthen the system's defenses against attackers and in the long run, this type of solution is better positioned to protect sensitive communications and personal data from exposure.

Many renowned institutions have already come forward with their security guidelines for developers, distributors, and users. Among the first was the National Institute of Standards and Technology (NIST), that in [2] has defined an Internet of Things (IoT) device cybersecurity capability core baseline, which is a set of device capabilities generally needed to support common cybersecurity controls that protect an organization's devices as well as device data, systems, and ecosystems. This core baseline provides organizations a starting point to use in identifying the device cybersecurity capabilities for new IoT devices they will manufacture, integrate, or acquire.

In [3] the authors provide an overview of security guidelines for IoT proposed by various organizations and evaluate some of the existing technologies applied to ensure IoT security against these guidelines. In the paper, recommendations proposed by selected government organizations, international associations, and advisory groups are gathered and compiled into a set of the most common and important considerations, divided into eight categories. Then the authors chose a number of representative examples from IoT security technologies and evaluated them against these criteria. Conclusions captured in that paper show that there is no exhaustive and CPU-agnostic solution. While none of the examined solutions fulfill all recommendations on their own, the existing technologies introduced by those solutions could be combined to create a design framework that satisfies all the requirements of a secure IoT device.

In this paper the concept of an IoTrust framework has been proposed. This hardware-software solution, when applied in the process of designing System on Chip (SoC) for IoT, will ensure its cybersecurity by providing security core baseline features. The proposed framework architecture assumes the combination of the mechanisms from the area of Hardware Root of Trust (HWRoT), Trusted Execution Environment (TEE), and Trusted Computing in order to ultimately create a configurable and comprehensive solution ensuring the security of IoT nodes.

This work was funded by The Polish National Centre for Research and Development under project No. CYBERSECIDENT/456446/III/NCBR/2020.

II. State of the Art

The security of IoT devices is an area of active research due to unsatisfactory levels of safety and the immense range of applications. The constrained resource nature of many IoT devices increases the challenge of an all-in-one solution. That is why there are many solutions that solve pinpointed areas of the SoC, but the security of the entire device is still a novel topic which has yet to have an industry-wide accepted solution.

A. Key enabling technologies

Security architectures defined for traditional embedded systems are also currently used in IoT devices, solving some security issues. However, further improvements to these architectures are necessary to address new varieties of device vulnerabilities in the IoT ecosystem. Trustworthy computing is a major challenge in the field of cybersecurity.

1) Trusted Execution Environments: Solution which provides a secure environment for applications to run, regardless of the security in the rest of the system. TEEs are complex systems that consist of both hardware and software components and offer an enhanced execution environment [4]. TEE is a tamper-resistant computing environment running a separation kernel that guarantees the authenticity of program code, integrity of crucial system assets (processor registers, secured memory), and confidentiality of code and data stored in persistent memory [4]. Additionally, a TEE is useful in providing authentication and identification of the system. To the outside world, TEE is a module that guarantees isolation between secure and non-secure environments for both code and data.

2) Hardware Root of Trust: A key technical challenge for TEEs is ensuring trust, meaning that the system behaves as expected by the user. To address this challenge, there has been significant support for the use of hardware-based root-of-trust (HRoT) implementations to establish trust in secure computing. Hardware RoTs are preferred over software RoTs due to their immutability, smaller attack surfaces, and more reliable behavior. They can provide a higher degree of assurance that they can be relied upon to perform their trusted functions [5].

3) Physically Unclonable Functions: A Physically Unclonable Function (PUF) is a physical random function that typically displays a unique challenge-response behavior for each of its instances. The response to a given challenge is randomly generated based on the intrinsic physical properties of the hardware in which it is embedded. Recently, PUFs have been proposed as key components in cryptographic mechanisms and security architectures [6]. They can be used for device identification and authentication, binding software to hardware in a platform, securely storing cryptographic secrets and designing secure protocols.

B. Existing solutions

Trusted computing solutions for IoT devices are essential for ensuring the security and integrity of IoT systems. To address this challenge, a number of solutions have been proposed. Some of them are already mature and currently in use for IoT devices, and some are emerging concepts that might bring new quality to the topic.

The evaluation of representative IoT security technologies against criteria presented in [3] shows that while there are solutions with the potential to meet all these recommendations, no solution currently addresses all requirements in an out-of-the-box capacity. This leaves room for further research in this field.

Here, we briefly describe a few example solutions.

1) ARM TrustZone: A security extension provided by ARM for both application processors (Cortex-A family) and microcontrollers (Cortex-M family) [7]. It is based on a TEE concept. TrustZone divides the system into secure and non-secure environments by providing two virtual processors with hardware-based access control. Memory isolation and a special processor mode dedicated to monitoring (the secure monitor) ensure complete separation of the two execution environments in hardware.

TrustZone offers a comprehensive security solution, but it requires a good understanding of the framework, creative implementation, and support from external IPs. It is important to note that TrustZone is designed for ARM infrastructure and relies on additional hardware such as CryptoCell and applications to do so. TrustZone alone is not an off-the-shelf, ready-to-use solution.

2) Intel SGX: A set of CPU instructions that enable the creation of isolated software containers called enclaves [8]. These enclaves provide a secure environment for a program's code, data, and stack through hardware-based access policy control and memory encryption. This isolation protects the program from other processes, even those with higher privilege levels. From a hardware perspective, Intel SGX isolates Processor Reserved Memory (PRM) and protects it against all memory accesses from outside an enclave. This includes access attempts by the kernel, hypervisor, system management mode, and DMA accesses requested by peripherals.

3) Keystone: An open-source framework designed for creating TEE environments based on an unmodified RISC-V architecture [9]. It uses RISC-V Physical Memory Protection (PMP) and the programmable machine mode (M-Mode) to implement a memory protection scheme. The Trusted Security Monitor (SM) is proposed at the M-Mode level and is responsible for managing secure hardware handling and context switching between enclaves. The SM should be executed entirely from on-chip memory and satisfies typical TEE requirements such as memory isolation and code/configuration attestation. Keystone does not provide direct resource management; this responsibility falls on the secure enclave application developer. Also, several platform requirements are listed by the authors, including support for a trusted boot process, an unique authentication key dedicated to this process, and a hardware source of randomness. Keystone can be a good starting point for securing an IoT device, but it is not sufficient on its own.

4) OpenTitan: An open-source Hardware Root of Trust implementation endorsed by leading non-profit, academic, and commercial organizations [10]. As an open-source project, its sources are available online for inspection by the broader community, which should improve its security. While OpenTitan's core is still under development, and several features are missing from its early-stage top-l el, its creators' intentions are well-documented. In its complete form, OpenTitan should be a robust solution for securing various systems as a HWRoT module that supports secure boot procedures and implements miscellaneous cryptographic primitives.

5) CURE: A security architecture providing Trusted Execution Environments with different types of enclaves: subspace enclaves provide vertical isolation at all execution privilege levels, user-space enclaves provide isolated execution to unprivileged applications, and self-contained enclaves allow isolated execution environments that span multiple privilege levels. CURE's protection mechanisms are based on new hardware security primitives on the system bus, the shared cache, and the CPU. It also enables the exclusive assignment of system resources, such as peripherals, CPU cores, or cache resources, to a single enclave [11]. The authors assume the CPU supports privilege levels to separate user space from the more privileged kernel space through virtual address spaces using a MMU. Moreover, it is assumed that the system performs a secure boot on reset, with the first bootloader stored in CPU Ready-Only Memory (ROM) and verifying the firmware through a chain of trust.

III. Concept of the IoTrust framework

In this work, we present a security framework called IoTrust that addresses security issues in a customizable way. The solutions presented are CPU-agnostic, meaning no assumptions are made about the CPU's support for privilege levels, memory protection schemes, or any security mechanisms.

A. Threat model

The presented framework focuses on securing code integrity, control flow integrity, and confidentiality and integrity of secrets of an application running on an IoT device installed in the field. The secrets to be protected include encryption keys, certificates, and hashes, as well as the application's sensitive data. The IoTrust architecture's trusted computing base consists of the system's on-chip hardware components as well as a dedicated hypervisor software component. While the source of data in on-chip memory is assumed to be correct, the framework does not trust off-chip memory, the operating system, the applications, or the physical protection provided by the device manufacturer. Side-channel attacks, cloud security, and network security are beyond the scope of this work. Potential attacks include cold boot attacks, physical code injection, and compromising the hypervisor.

B. IoTrust architecture

Fig. 1 shows the architecture of a sample SoC system for an IoT device based on a standard CPU, system bus fabric, typical off-chip memory blocks and input/output devices with additional specialized components of the IoTrust framework.

The IoTrust framework is intended to be a configurable SoC framework for IoT devices that leverages TEE environment concepts. A set of developed IP Cores is proposed, which enable the implementation of a HWRoT, proxy modules that filter interfaces that are connected to off-chip components, a secure DMA that encrypts based on per-enclave cryptographic keys and, a CPU agnostic module that allows compartmentalization of software execution space. The proposed solution can be adapted for IoT implementations using both FPGA (Field Programmable Gate Array) and ASIC (Application Specific Integrated Circuit) technologies.

The software part of the framework consists of the **Security Manager Software** (SM-SW). It is responsible for managing enclaves and their lifecycle while acting as a hypervisor. It allocates processor time to enclaves and queues enclaves that are waiting for CPU execution time. Using emulated software interrupts, enclaves are provided with a secure way to interact with the rest of the system through the SM-SW API and data exchange between enclaves is enabled. The SM-SW module also includes interrupt handling of interrupts generated by the Security Manager Hardware (SM-HW) module during active enclave switching. This procedure ensures a safe and secure switch between a potentially untrusted enclave and the hypervisor code, which has full access to all system components. For this implementation, the **enclave context** is defined as the extension of the processor's context (the state of the CPU registers used by the executed program) by adding the state of API registers included in the SM-HW.

Furthermore, the SM-SW ensures proper configuration of the hardware modules of the framework at the boot time. Before starting enclaves, it writes definitions of the privileges enforced by the hardware, such as the interrupt filtering or virtual address spaces, to SM-HW registers.

To guarantee a safe enclave switching procedure and successful preemption even if the CPU runs malicious code, SM-SW programs a **handshake sequence** based on handler execution and a watchdog timeout.

The software part includes libraries that provide trusted Application Programming Interfaces (APIs), enabling the implementation of typical cybersecurity mechanisms and modules that implement functionalities outlined in the aforementioned core baseline requirements. The solution can provide a comprehensive, configurable software/hardware framework for building a trusted TEE runtime environment together with hardware specific to IoT security solutions.

The hardware part of the IoTrust framework consists of the following main parameterizable modules that are added at the SoC level.

The **Security Manager Hardware** together with the Security Manager Software enables the implementation of Trusted Computing concepts by creating enclaves where code is executed and ensuring their physical isolation (each enclave has access to dedicated hardware resources), which, with the proper use of the HWRoT module, allows for the implementation of a state-of-the-art TEE environment. It manages transactions between CPU and system bus. Additionally, in-

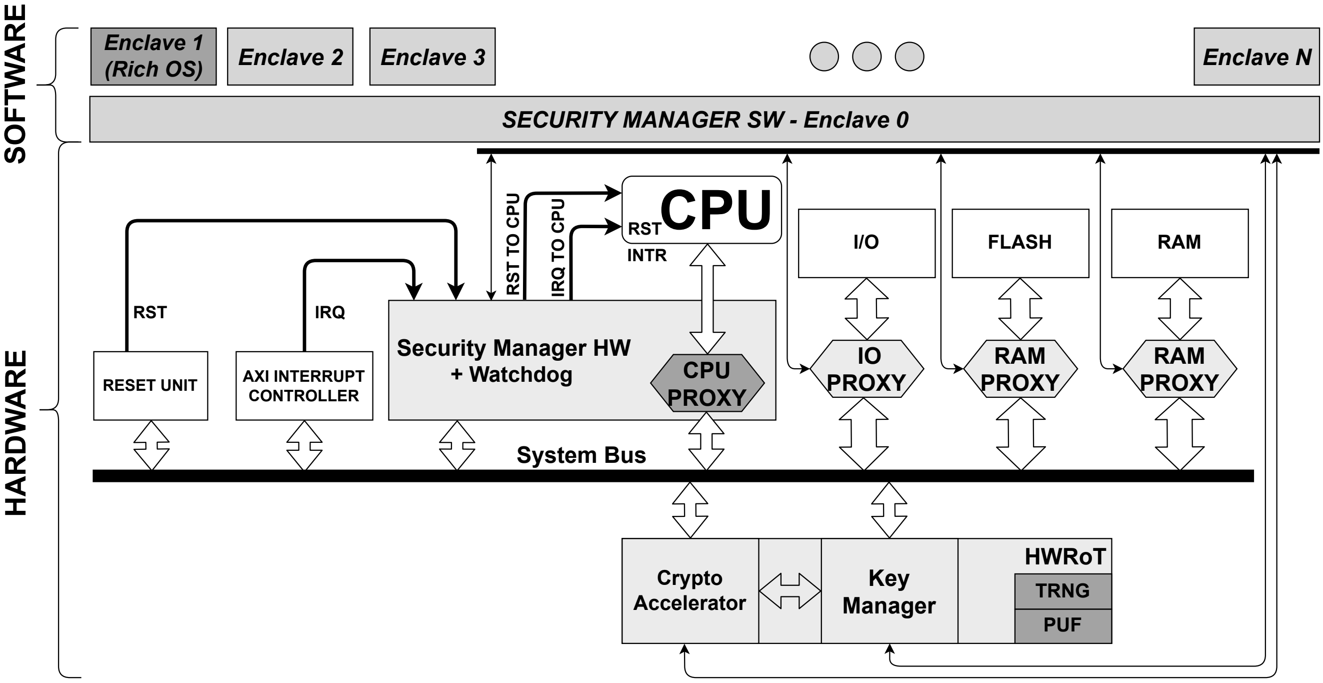


Fig. 1. Block level diagram of an example SoC integrating with the IoTrust framework

formation about the currently active enclave is passed to other IoTrust framework blocks to ensure synchronized isolation of processes from system devices and enable memory access (encryption/decryption) associated with the keys tied with the currently running enclave. Configuration changes and access violations are reported using a cybersecurity event logging module and the created logs are sealed to ensure their confidentiality and integrity when read by an authorized entity.

The **Hardware Root of Trust** (HWRoT) acts as an anchor of trust in the system and provides secure hardware implementations of necessary cryptographic algorithms. This component consists of hardware modules that provide security functions necessary to ensure trust within a platform (such as confidentiality, integrity, verification, authorization, secure storage, and updating). Its essential traits are immutability and predictability - it always behaves in the same way under known conditions. A hardware implementation enables meeting these conditions. HWRoT is treated as an inherently trusted element of the system. Thanks to the security services it offers, it can verify the correctness of subsequently launched software modules during the runtime of a Rich OS or other kernels. In this way, trust can be propagated within the platform and, the so-called Chain of Trust is formed, with HWRoT as the first element and the operating system or application running in it as the last. The secure system startup process including the creation of the Chain of Trust is called secure boot.

The **RAM Proxy** handles accesses to an external RAM with data protection. The task of the RAM Proxy module is to secure and manage accesses to external RAM. It is an essential component that mediates between the SM-HW and the memory controller during data exchange. Additionally, it works closely with the HWRoT module for data encryption and decryption.

IV. Implementation

The IoTrust framework's components have been developed using Verilog HDL and C/assembler. An example SoC based on the Xilinx Microblaze soft CPU and Vivado system platform has been designed as a proof of concept.

A Trusted Execution Environment of the IoTrust framework is a secure and isolated execution area within a computing unit that provides the authenticity of executed code, integrity of resources (such as CPU registers, memory, or input/output devices), and confidentiality of code, data, and states of non-volatile memory. In the IoTrust framework a TEE is implemented using the concept of enclaves.

An **enclave** is defined as a secure runtime environment managed by the Security Manager software and hardware, along with its metadata (a set of hardware access permissions, checksums, and digital signatures for the code) and an isolated address space where verified application instances are launched. Each enclave running on the processor operates in a separate virtual address space. Access to peripherals is limited by the Security Manager Hardware and is configurable per enclave.

A. IoTrust Hardware

1) Security Manager Hardware: The Security Manager Hardware module (Fig. 2) is directly connected to the CPU and is responsible for implementing Trusted Computing features. It is composed of:

- Enclave switch control – a module that monitors the CPU instruction bus to detect the sequence of instructions indicating enclave switch. The sequence can be programmed in the register file. The CPU Proxy is instructed to change the enclave context when the sequence is detected.
- CPU Proxy – a module responsible for hardware translation of transaction addresses from the virtual space seen by a given enclave to the physical space of the system bus. Transactions that violate access rights are rejected, and security breach logging processes are started. The module labels each AXI transaction with the number of the currently active enclave using AXI USER signals. This information is used by other IoTrust framework blocks to ensure isolation of enclaves from each other and enable memory access by encryption/decryption of its content using the keys tied with the currently running enclave.
- Interrupt Proxy – system interrupts are intercepted and subsequently forwarded to the computing unit only if they are authorized to be handled by the currently executing enclave. The Current Enclave register and the Watchdog interrupt mechanisms enable run-time security control of the system and detect anomalies. If the designated enclave fails to handle a specific interrupt within the appropriate time, the Watchdog module signals a system malfunction due to an error or a cyber attack. The event is then reported in a sealed log, and the system is reset.

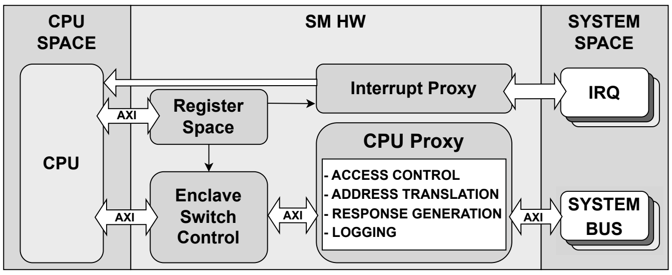


Fig. 2. Block level diagram of the SM HW module

2) Hardware Root of Trust: The Hardware Root of Trust of the IoTrust platform (Fig. 3) consists of three main components. The first one is a read-only memory for the first bootloader, which performs the initial system initialization and does the initial configuration of the HWRoT blocks during the secure boot procedure. The next component is the crypto accelerator block, which implements necessary cryptographic primitives (including lightweight algorithms suitable for hardware resource-constrained devices) and includes a DMA unit for efficient data operations. This module assists in the encryption of RAM for individual enclaves. The last component is the **secret top** block, which manages cryptographic keys and other secrets (such as physical and logical device identifiers, owner identity). It also generates secure cryptographic random sequences (to generate cryptographic keys within the HWRoT itself) and includes a Physical Unclonable Function block, which provides a unique fingerprint for each device.

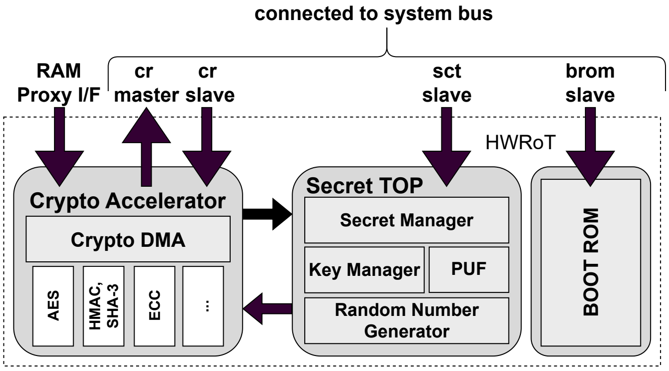


Fig. 3. Block level diagram of the HWRoT module

3) RAM Proxy: RAM Proxy is a component which connects system and the RAM. It intercepts memory accesses and is responsible for data protection. At first, appropriate data block is read from the memory and passed to HWRoT with enclave identifier (for cryptographic key selection) to be decrypted. If the requested operation is read, the decrypted data block is transmitted on the system bus. In the case of write operation, the corresponding data fragment is replaced, and the data block is then re-encrypted in HWRoT, and passed to the memory controller. Module has cache buffer that stores recently read memory blocks and increases the system's efficiency, because it allows to reduce the number of interactions with off-chip memory and the HWRoT.

4) IO Proxy: Allows hardware-based control of interface access in the SoC. For example, a local JTAG debug interface, Bluetooth, or Ethernet. It can filter transactions to and from peripherals. It contains a register that must be accessed by the SM-SW before usage due to hardware access restrictions when the peripheral is not enabled.

B. *IoTrust Software*

The SM-SW is functionally split into several modules. The Enclave Manager contains the main loop and is responsible for enclaves' management. Memory manager allocates/deallocates memory regions (*memory slices*) and is responsible for their safe clearing. It translates between virtual space and physical system bus addresses (Fig. 4). The hypervisor implements dedicated mechanisms for scheduling, safe switching of the enclaves, and data exchange between enclaves.

V. Synthesis results

Synthesis and implementation were performed using Vivado tools version 2022. The results presented are for the Trenz evaluation board TE0712 [12] equipped with a Xilinx Artix-7 XC7A200T FPGA. Table I presents the resource utilization of the entire system by assessing the number of LUTs, Slice, DSPs and block RAMs.

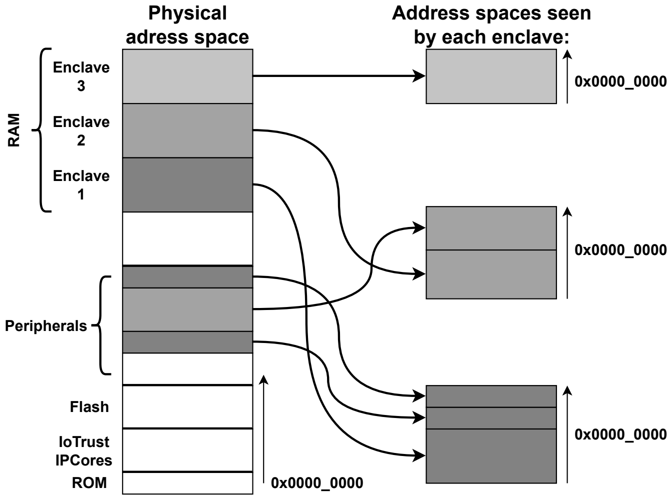


Fig. 4. Virtual address spaces of the enclaves are mapped onto the system's physical address space.

TABLE I
UTILIZATION RESULTS

	Used	**Available**	**Util [%]**
Slice LUT	59719	133800	44.63
LUT as Logic	58105	134600	43.19
LUT as Memory	2064	46200	4.47
Slice Registers	31023	269200	11.52
Slice	18363	33650	54.57
Block RAM	41	365	11.23
DSPs	4	740	0.54

The article presents the Proof of Concept of the IoTrust framework. For this reason, the target solution may be optimized for utilization, power consumption, or frequency. The results presented do not include the HWRoT module. Considering the number of logic cells used, the solution can be classified as a lightweight solution.

The power estimation analysis indicates, that the whole SoC consumes about $1.5\ W$ of power. Since the DDR memory controler consumes about $1\ W$ of power, the rest of the system components use $0.5\ W$ of power.

VI. CONCLUSIONS

The IoT ecosystem presents new security challenges beyond traditional data security. There is a need for IoT security guidelines, and many organizations worldwide have proposed recommendations to help ensure secure IoT infrastructure. While device designers and vendors have their own proprietary solutions that address some issues, they fall short in others. Evaluation of representative examples of IoT security technologies shows that while there are solutions with the potential to meet all recommendations, none currently do so in an out-of-the-box capacity.

In this paper, we proposed the concept of the hardware-software security framework that, when applied in the process of designing System on Chip for IoT device, will ensure its cybersecurity by providing security core baseline features. The proposed IoTrust framework consists of custom hardware IP Cores designed in Verilog HDL as well as C/assembler software procedures that can be included in SoC design, enabling the combination of the mechanisms from the area of Hardware Root of Trust, Trusted Execution Environment and Trusted Computing. This ultimately creates a configurable and comprehensive solution ensuring the security of IoT nodes.

The solution discussed does not make any assumptions about the CPU's support for privilege levels, memory protection schemes, or security mechanisms and is therefore CPU-agnostic.

A proof-of-concept implementation has been demonstrated where the IoTrust framework has been applied to SoC based on MicroBlaze soft-processor. As the prototype platform the Trenz evaluation board TE0712 equipped with a Xilinx Artix-7 XC7A200T FPGA has been used. Example execution scenarios have been included, that demonstrate basic functionalities of proposed solution.

REFERENCES

[1] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, "Securing the internet of things: Challenges, threats and solutions," *Internet of Things*, vol. 5, pp. 41–70, 2019. doi: 10.1016/j.iot.2018.11.003.

[2] M. Fagan, K. N. Megas, K. Scarfone, and M. Smith, "IoT device cybersecurity capability core baseline," tech. rep., National Institute of Standards and Technology, Gaithersburg, MD, may 2020. doi: 10.6028/NIST.IR.8259a.

[3] M. Chmiel, M. Korona, F. Kozioł, K. Szczypiorski, and M. Rawski, "Discussion on IoT Security Recommendations against the State-of-the-Art Solutions," *Electronics*, vol. 10, no. 15, 2021. doi: 10.3390/electronics10151814.

[4] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 57–64, 2015. doi: 10.1109/Trustcom.2015.357.

[5] A. Ehret, E. Del Rosario, K. Gettings, and M. A. Kinsy, "A hardware root-of-trust design for low-power soc edge devices," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6, 2020. doi: 10.1109/HPEC43674.2020.9286164.

[6] T. Idriss, H. Idriss, and M. Bayoumi, "A puf-based paradigm for iot security," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 700–705, 2016. doi: 10.1109/WF-IoT.2016.7845456.

[7] S. Pinto and N. Santos, "Demystifying arm trustzone: A comprehensive survey," *ACM Comput. Surv.*, vol. 51, jan 2019. doi: 10.1145/3291047.

[8] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative technology for cpu based attestation and sealing," in *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy*, vol. 13, p. 7, ACM New York, NY, USA, 2013.

[9] D. Lee, D. Kohlbrenner, S. Shinde, K. Asanović, and D. Song, "Keystone: An open framework for architecting trusted execution environments," in *Proceedings of the Fifteenth European Conference on Computer Systems*, EuroSys '20, (New York, NY, USA), Association for Computing Machinery, 2020. doi: 10.1145/3342195.3387532.

[10] "Opentitan - open source silicon root of trust." https://opentitan.org/. Accessed on 30.06.2021.

[11] R. Bahmani, F. Brasser, G. Dessouky, P. Jauernig, M. Klimmek, A.-R. Sadeghi, and E. Stapf, "CURE: A security architecture with CUstomizable and resilient enclaves," in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1073–1090, USENIX Association, Aug. 2021. doi: 10.48550/arXiv.2010.15866.

[12] T. Electronics, "TE0712 WIKI." https://wiki.trenz-electronic.de/display/PD/TE0712+Resources, 2023. Accessed on 21.05.2023.

DOI: 10.15439/2023F3575
ISSN 2300-5963 ACSIS, Vol. 35

On Gower Similarity Coefficient and Missing Values

Marzena Kryszkiewicz
0000-0003-4736-4031
Warsaw University of Technology,
Institute of Computer Science,
Nowowiejska 15/19,
00-665 Warsaw, Poland
Email:
Marzena.Kryszkiewicz@pw.edu.pl

***Abstract*—The Gower similarity coefficient is a popular measure for comparing objects with possibly mixed-type attributes and missing values. One of its characteristics is that it calculates the coefficient value without considering attributes with missing values. In this article, we explore the properties of the coefficient in detail, including the consequences of omitting attributes with missing values. We also introduce strict lower and upper bounds on the actual similarity value on an attribute and strict lower and upper bounds on the actual value of the Gower similarity coefficient, derive a number of their properties and propose a new coefficient as a solution to the identified problem with the Gower similarity coefficient.**

***Index Terms*—Gower similarity coefficient, mixed-type attributes, quantitative attributes, qualitative attributes, dichotomous attributes, missing values.**

Introduction

THE Gower similarity coefficient [4] is a popular measure for comparing objects with possibly mixed-type attributes (quantitative, qualitative and/or dichotomous) and missing values. One of its characteristics is that it calculates the coefficient value without considering attributes with missing values. The approach is easy and intuitive and finds many applications (see, e.g. [1], [2], [3], [5], [6], [8]). It is also considered as an easily extensible template of calculating (dis)similarities of objects with mixed-type attributes [2], [5], [7]. However, as we show in the article, Gower similarity coefficient has some deficiencies. In particular, we show that in the case of objects with missing values, the coefficient may take a similarity value impossible to obtain with any replacement of missing values with values from the domains of attributes.

Our main contribution in the article includes:

- Introduction of strict lower and upper bounds on the actual similarity value on an attribute and strict lower and upper bounds on the actual value of the Gower similarity coefficient, which are obtainable after replacing missing values with respective attribute domain values.
- Showing that in the case of a pair of objects one of which has missing value for at least one quantitative attribute, the Gower similarity coefficient may take an incorrect value, which will be less than the lower bound on the actual value of the Gower similarity coefficient.
- Derivation of a number of properties of similarity value of objects on the attribute, the Gower similarity coefficient and the introduced bounds.
- Proposing new similarity coefficient G' as a correction of the Gower similarity coefficient, which eliminates the problem found for quantitative attributes with missing values.

The layout of the article is as follows: First, we recall the definitions of attribute value similarities, their weights and the Gower similarity coefficient, as well as introduce additional basic notions that are used throughout the article. Then, we show example objects for which the Gower similarity coefficient takes an incorrect value, caused by the occurrence of a missing value of a quantitative attribute for one of them. We also illustrate the consequences of the occurrence of missing values for qualitative and dichotomous attributes. Next, we introduce strict lower and upper bounds on the actual similarity value on an attribute and on the actual value of the Gower similarity coefficient, as well as derive a number of their properties. In addition, the coefficient G', being the modification of the Gower similarity coefficient, is proposed, which, unlike the original Gower similarity coefficient, always returns similarity values that do not exceed the presented lower and upper bounds.

Basic notions related to Gower Similarity Coefficient

Gower proposed a measure of objects' similarity, which can be applied in the case of qualitative attributes, quantitative attributes, dichotomous attributes or their mixtures [4]. In the measure, only the attributes for which it is possible to determine their similarity are taken into account; the other are ignored. In particular, if for a pair of objects, an attribute value for at least one of these objects is missing, then the two objects are treated as not comparable on this attribute and the Gower similarity coefficient is calculated without taking this attribute into account.

In the remainder of the article, we assume that objects are characterized by n, where $n \geq 1$, attributes whose domains contain at least two different values. The missing value will be denoted by *. The value of attribute i of object u will be denoted by u_i.

The function $w_i(.,.)$ is used to indicate whether two objects are comparable on attribute i or not. Let u and v are objects under consideration. If u and v are comparable on attribute i, then $w_i(u,v) = 1$; otherwise $w_i(u,v) = 0$. We already mentioned that two objects u and v are incomparable on attribute i if the value of at least one of the objects is missing and so, $w_i(u,v) = 0$. However, in the case of a dichotomous attribute (indicating whether or not a feature is present), the objects may also be incomparable, even if their values are known (this happens when two objects do not have the feature represented by the dichotomous attribute).

The Gower similarity coefficient [4] for objects u and v is denoted by $G(u,v)$ and is defined as follows:

$$G(u,\, v) = \frac{\sum_{i=1}^{n} w_i(u,v) \times s_i(u,v)}{\sum_{i=1}^{n} w_i(u,v)},$$

where $s_i(u,v)$ is a coefficient determining similarity of two objects on attribute i, $i = 1..n$, taking values from the interval $[0,1] \cup \{undefined\}$. It is assumed that whenever $w_i(u,v) = 0$, then $w_i(u,v) \times s_i(u,v) = 0$. Thus, the Gower similarity coefficient is the average similarity of two objects on the attributes on which they are comparable.

In the case when the values of attribute i are not missing for both objects u and v, then $w_i(u,v)$ and coefficient $s_i(u,v)$ are determined as follows:

- If attribute i is qualitative:
  - $w_i(u,v) = 1$,
  - $s_i(u,v) = \begin{cases} 1, \text{if } u_i = v_i \\ 0, \text{if } u_i \neq v_i \end{cases}$;
- If attribute i is quantitative:
  - $w_i(u,v) = 1$,
  - $s_i(u,v) = 1 - \frac{|u_i - v_i|}{range_i}$;

  where $range_i = max_i - min_i$, where max_i is the maximal value of attribute i, while min_i is the minimal value of attribute i.
- If attribute i is dichotomous:
  - $w_i(u,v) = \begin{cases} 1, \text{ if } (u_i = +) \text{ and } (v_i = +) \\ 1, \text{ if } (u_i = +) \text{ and } (v_i = -) \\ 1, \text{ if } (u_i = -) \text{ and } (v_i = +) \\ 0, \text{ if } (u_i = -) \text{ and } (v_i = -) \end{cases}$,
  - $s_i(u,v) = \begin{cases} 1, \text{ if } (u_i = +) \text{ and } (v_i = +) \\ 0, \text{ if } (u_i = +) \text{ and } (v_i = -) \\ 0, \text{ if } (u_i = -) \text{ and } (v_i = +) \\ 0, \text{ if } (u_i = -) \text{ and } (v_i = -) \end{cases}$.

In the case when the value of attribute i is missing for at least one of the objects u or v, then $w_i(u,v)$ and the coefficient $s_i(u,v)$ is determined for any type of attribute i in the same way as follows:

- $w_i(u,v) = 0$,
- $s_i(u,v) = undefined$.

Now, we are ready to formally define *comparable* and *incomparable objects on an attribute*. *Objects u and v* are defined as *incomparable on attribute i* if:

- either the value of attribute i is missing for at least one the two objects
- or attribute i is dichotomous and the values of both objects are equal to $-$.

Otherwise, *objects u and v* are *comparable on attribute i.*

Property 1.

a) Objects u and v are incomparable on attribute i iff $w_i(u,v) = 0$.

b) Objects u and v are comparable on attribute i iff $w_i(u,v) = 1$.

c) If objects u and v are incomparable on attribute i, then $w_i(u,v) \times s_i(u,v) = 0$.

d) If objects u and v are comparable on attribute i, then $w_i(u,v) \times s_i(u,v) = s_i(u,v)$.

e) $s_i(u,v) = s_i(v,u)$ and $w_i(u,v) = w_i(v,u)$.

In the remainder of the article, we will use the following notation:

- *CMP_ATT*(u,v) denotes the set of attributes on which u and v are comparable; that is, *CMP_ATT*(u,v) = {attribute i| $w_i(u,v) = 1$}.
- *INCMP_ATT*(u,v) denotes the set of attributes on which u and v are not comparable; that is, *INCMP_ATT*(u,v) = {attribute i| $w_i(u,v) = 0$}.
- *INCMP*_ATT*(u,v) denotes the set of attributes on which either u or v or both have missing values.
- $INCMP^d\_ATT(u,v)$ denotes the set of dichotomous attributes on which both u and v have value $-$.

Property 2.

a) $G(u,v) = \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v)}{|CMP\_ATT(u,v)|}$.

b) $|CMP\_ATT(u,v)| + |INCMP\_ATT(u,v)| = n$.

c) $INCMP{*}\_ATT(u,v) \cap INCMP^d\_ATT(u,v) = \emptyset$.

d) $INCMP\_ATT(u,v) = INCMP{*}\_ATT(u,v) \cup INCMP^d\_ATT(u,v)$.

e) $|CMP\_ATT(u,v)| + |INCMP{*}\_ATT(u,v)| \leq n$.

Objects u and v are defined as *comparable* if they are comparable on at least one attribute; that is, if $\sum_{i=1}^{n} w_i(u,v) > 0$ (or equivalently, if $|CMP\_ATT(u,v)| > 0$).

Otherwise, *objects u and v are* defined as *incomparable*; that is, when $\sum_{i=1}^{n} w_i(u,v) = 0$ (or equivalently, if $|CMP\_ATT(u,v)| = 0$). Please note that the value of $G(u,v)$ is not defined for incomparable objects. Otherwise, if u and v are comparable, then $G(u,v) \in [0, 1]$.

NEW RESULTS

A. What's Wrong with Gower Similarity Coefficient?

Though Gower similarity coefficient is appreciated by the ease and intuitiveness of dealing with attributes on which objects are incomparable, we will show that it may take an unacceptable value if the values of attributes are missing (see Example 1).

Example 1. Table I presents Set 1 of example objects characterized by qualitative attribute 1 (*colour of hair*) and quantitative attribute 2 (*age*). Let $max_2 = 100$, $min_2 = 0$, so $range_2 = 100$.

Objects u and v are comparable and different on attribute 1 (so, $w_1(u,v) = 1$ and $s_1(u,v) = 0$) and are not comparable on attribute 2 (so, $w_2(u,v) = 0$, $s_2(u,v) =$ *undefined*). Hence, $G(u,v) = (1 \times 0 + 0 \times$ *undefined*$) / (1 + 0) = 0 / 1 = 0$.

TABLE I.
SET 1 OF EXAMPLE OBJECTS

object o	1 (*colour of hair*)	2 (*age*)	$w_2(u,o)$	$s_2(u,o)$	$G(u,o)$
u	brown	**40**	1	1–|50–50|/100=1	2/2=1
v	blond	*	**0**	***undefined***	**0/1=0**
v_1	blond	0	1	1–|40–0|/100=0.6	0.6/2=0.3
v_2	blond	10	1	1–|40–10|/100=0.7	0.7/2=0.35
v_3	blond	20	1	1–|40–20|/100=0.8	0.8/2=0.4
v_4	blond	30	1	1–|40–30|/100=0.9	0.9/2=0.45
v_5	blond	**40**	**1**	**1–|40–40|/100=1**	**1/2=0.5**
v_6	blond	50	1	1–|40–50|/100=0.9	0.9/2=0.45
v_7	blond	60	1	1–|40–60|/100=0.8	0.8/2=0.4
v_8	blond	70	1	1–|40–70|/100=0.7	0.7/2=0.35
v_9	blond	80	1	1–|40–80|/100=0.6	0.6/2=0.3
v_{10}	blond	90	1	1–|40–90|/100=0.5	0.5/2=0.25
v_{11}	blond	**100**	**1**	**1–|40–100|/100=0.4**	**0.4/2=0.2**

Now we will consider what would be the Gower similarity coefficient of objects u and v_i, where v_i represents v after replacing its missing value of attribute 2 with some value from the domain range [0, 100]. Objects $v_1, \ldots, v_{11}$ in Table I represent object v under assumption that its actual value of attribute 2 is 0, 10, …, 100, respectively. Clearly, each instance v_i of object v is comparable with u on both attributes and is different from u on attribute 1, which is qualitative (so similarity of v_i to u on attribute 1 equals 0). Hence, $G(u,v_i) = (1 \times 0 + 1 \times s_2(u, v_i)) / (1 + 1) = s_2(u,v_i) / 2$.

Clearly, $G(u,v_i)$ reaches maximum for the greatest value of $s_2(u,v_i)$. This happens for object v_5, for which $s_2(u,v_5) = 1$ and, in consequence, $G(u,v_5) = 0.5$.

$G(u,v_i)$ reaches minimum for the least value of $s_2(u,v_i)$ (that is, for the largest absolute value of the difference between *age* of u and v_i). This happens for object v_{11}, for which $s_2(u,v_{11}) = 0.4$ and so, $G(u,v_{11}) = 0.2$. Please note that this least achievable value of 0.2 of $G(u,v_i)$ is greater than $G(u,v)$, which equals 0.

As shown in Example 1, $G(u,v)$ may take a value that is not obtainable for any actual completions of missing values of quantitative attributes of objects u and v.

In the further part of the article, we introduce strict lower and upper bounds on the actual similarity value of any objects u and v on an attribute from the set $INCMP^*\_ATT(u,v)$ and on the actual value of the Gower similarity coefficient for these objects. The bounds will make it possible to check when the Gower similarity coefficient takes values unattainable for any completions of missing values.

B. Lower and Upper Bounds on Actual Similarity Value on an Attribute

Let us recall that objects u and v are not comparable on attribute i either because at least one of the objects has missing value for this attribute (i.e. $i \in INCMP^*\_ATT(u,v)$) or the attribute is dichotomous and both objects have value – for it (i.e. $i \in INCMP^d\_ATT(u,v)$). If u and v are incomparable on attribute i, then $w_i(u,v) = 0$, and so attribute i does not contribute to the value of $G(u,v)$. Nevertheless, in the case of attribute $i \in INCMP^*\_ATT(u,v)$, u and v may become comparable on attribute i if the actual values of attribute i become known for both objects. Then, $w_i(u,v)$ can become equal to 1, and so, $s_i(u, v)$ can contribute to the value of $G(u,v)$. Example 1 illustrates how replacing missing value of quantitative attribute i affects the values of $w_i(u, v)$, $s_i(u, v)$ and $G(u,v)$. This influence is also illustrated for a qualitative attribute and a dichotomous attribute in Examples 2 and 3, respectively.

Example 2. Table II presents Set 2 of example objects characterized by qualitative attribute 1 (*colour of hair*) and quantitative attribute 2 (*age*). Let $max_2 = 100$, $min_2 = 0$, so $range_2 = 100$.

Objects u and v are not comparable on attribute 1 ($w_1(u,v) = 0$ and $s_1(u,v) =$ *undefined*) and are comparable on attribute 2 ($w_2(u,v) = 1$, $s_2(u,v) = 0.9$). Hence, $G(u,v) = (0 \times$ *undefined* $+ 1 \times 0.9) / (0 + 1) = 0.9 / 1 = 0.9$.

TABLE II.
SET 2 OF EXAMPLE OBJECTS

object o	1 (*colour of hair*)	2 (*age*)	$w_1(u,o)$	$s_1(u,o)$	$G(u,o)$
u	**brown**	40	1	1	2/2=1
v	*	30	**0**	***undefined***	**0.9/1=0.9**
v_1	**brown**	30	**1**	**1**	**1.9/2=0.95**
v_2	**blond**	30	**1**	**0**	**0.9/2=0.45**

Objects v_1 and v_2 in Table II present instances of object v after replacing its missing value of attribute 1 with some value from the domain of this attribute. Clearly, unlike v, v_1 and v_2 are comparable with u on attribute 1. Since, u and v_1 have identical value of attribute 1, their similarity on this attribute is the greatest possible; namely, $s_1(u,v_1) = 1$. Since,

u and v_2 differ on attribute 1, their similarity on this attribute is the least possible; namely, $s_1(u,v_2) = 0$. Please note that $G(u,v) \in [G(u,v_2), G(u,v_1)] = [0.45, 0.95]$.

Example 3. Table III presents Set 3 of example objects characterized by qualitative attribute 1 (*colour of hair*), quantitative attribute 2 (*age*) and dichotomous attribute 3 (*has a car*). Let $max_2 = 100$, $min_2 = 0$, so $range_2 = 100$.

Objects u and v are comparable on attributes 1 and 2 ($w_1(u,v) = w_2(u,v) = 1$, $s_1(u,v) = 0$, $s_2(u,v) = 0.9$) and are not comparable on attribute 3 ($w_3(u,v) = 0$, $s_3(u,v) = undefined$). Hence, $G(u,v) = (1 \times 0 + 1 \times 0.9 + 0 \times undefined) / (1 + 1 + 0) = 0.9 / 2 = 0.45$.

TABLE III.
SET 3 OF EXAMPLE OBJECTS

object o	1 (*colour of hair*)	2 (*age*)	3 (*has a car*)	$w_3(u,o)$	$s_3(u,o)$	$G(u,o)$
u	brown	40	–	0	0	2/2=1
v	blond	30	*	**0**	***undefined***	**0.9/2=0.45**
v_1	blond	30	–	**0**	**0**	**0.9/2=0.45**
v_2	blond	30	+	**1**	**0**	**0.9/3=0.3**

Objects v_1 and v_2 in Table III present instances of object v after replacing its missing value of attribute 3 with either – or +. Since, both u and v_1 have value – of attribute 3, they are not comparable on this attribute (so, $w_3(u,v_1) = 0$) and $s_3(u,v_1) = 0$. This means that attribute 3 does not contribute to the value of $G(u,v_1)$ even though its value is known both for u and v_1. Now, since, u and v_2 have values – and +, respectively, on attribute 3, they are comparable on attribute 3 (so, $w_3(u,v_1) = 1$) and their similarity on this attribute is the least possible; namely, $s_3(u,v_1) = 0$. Please note that $G(u,v) \in [G(u,v_2), G(u,v_1)] = [0.3, 0.45]$.

In Examples 1, 2 and 3, we considered instances of example object u, with known values for all attributes, and object v, with missing value only for one given attribute i. We considered all or some instances of object v in which missing value was replaced by possible actual values including those instances of object v whose similarity on attribute i was the least and greatest, respectively. Clearly, these least and greatest values are lower and upper bounds, respectively, on similarity values of objects u and v on the examined attributes.

Let $i \in INCMP^{*}\_ATT(u, v)$. *Lower bound on the actual similarity value of u and v on attribute i* will be denoted by $\underline{s}_i(u,v)$, while *upper bound on the actual similarity value of u and v on attribute i* will be denoted by $\overline{s}_i(u,v)$. The associated *weights for the bounds* will be denoted as $\underline{w}_i(u,v)$ and $\overline{w}_i(u,v)$, respectively.

In Table IV, we provide the values of the similarity bounds $\underline{s}_i(u,v)$ and $\overline{s}_i(u,v)$ and their weights, respectively, under assumption that the value of attribute i is missing for at least one object. In fact, $\underline{s}_i(u,v) = \underline{s}_i(v,u)$, $\underline{w}_i(u,v) = \underline{w}_i(v,u)$, $\overline{s}_i(u,v) = \overline{s}_i(v,u)$ and $\overline{w}_i(u,v) = \overline{w}_i(v,u)$, thus, without loss of generality, we assume that the value of attribute i is missing for object v. The results are provided for quantitative, qualitative and dichotomous attributes. We also indicate for which possible actual values of v and eventually u, $s_i(u,v) = \underline{s}_i(u,v)$ and $s_i(u,v) =\overline{s}_i(u,v)$, respectively. Thus, we show that $\underline{s}_i(u,v)$ and $\overline{s}_i(u,v)$ are strict lower and upper bounds on the actual similarity value of objects u and v on each attribute $i \in INCMP^{*}\_ATT(u, v)$.

Please note that $\underline{w}_i(u,v)$ equals 1 for each attribute $i \in INCMP^{*}\_ATT(u, v)$. On the other hand, the lower bound $\underline{s}_i(u,v) = 0$ in all cases considered in Table IV except for quantitative attribute i whose value is missing for only one of the two compared objects. In that exceptional case, $\underline{s}_i(u,v)$ depends on the known value of the other object and can be

TABLE IV.
STRICT SIMILARITY BOUNDS $\underline{s}_i(u,v)$, $\overline{s}_i(u,v)$ AND THEIR ASSOCIATED WEIGHTS $\underline{w}_i(u,v)$ AND $\overline{w}_i(u,v)$ FOR MISSING VALUE OF OBJECT v AND KNOWN OR MISSING VALUE OF OBJECT u.

Type of attribute *i*	**Value of attribute *i* for object *u***	$\underline{w}_i(u,v)$	$\underline{s}_i(u,v)$	**When $s_i(u,v) = \underline{s}_i(u,v)$?**	$\overline{w}_i(u,v)$	$\overline{s}_i(u,v)$	**When $s_i(u,v) = \overline{s}_i(u,v)$?**
qualitative	missing	1	0	when actual value of v is different from actual value of u	1	1	when actual value of v is equal to actual value of u
	x	1	0	when actual value of v is different from x	1	1	when actual value of v is equal to x
quantitative	missing	1	0	when actual value of v is minimal and actual value of u is maximal or vice versa	1	1	when actual value of v is equal to actual value of u
	x	1	$\min\{(x - min_i), (max_i - x)\} / range_i$	when the absolute value of the difference between x and actual value of v is the largest possible; that is, is equal to $max\{(x - min_i), (max_i - x)\}$.	1	1	when actual value of v is equal to x
dichotomous	missing	1	0	when actual value of v is different from actual value of v	1	1	when actual value of v and actual value of u are equal to +
	+	1	0	when actual value of v is equal to –	1	1	when actual value of v is equal to +
	–	1	0	when actual value of v is equal to +	0	0	when actual value of v is equal to –

greater than 0 (as shown in Table IV, in this case, $\underline{s}_i(u,v) = \min\{(x - min_i), (max_i - x)\} / range_{i.})$.

Property 3. Let i be a quantitative attribute. Let the value of attribute i be missing for object v and be equal to x for object u. Then:

a) $\underline{s}_i(u,v)$ reaches maximum, which is equal to 0.5, for $x = (min_i + min_i) / 2$.

b) $\underline{s}_i(u,v)$ reaches minimum, which is equal to 0, for $x = min_i$ or $x = max_i$.

Proof: Follows from $\underline{s}_i(u,v)$ for a quantitative attribute (see Table IV).

Note also that for each attribute $i \in INCMP*\_ATT(u, v)$, upper bound $\overline{s}_i(u,v) = 1$ and $\overline{w}_i(u,v) = 1$, unless attribute i is dichotomous and its value is equal to $-$ for one object, say u, and is missing for the other object, say, v. In that exceptional case, $\overline{w}_i(u,v) = 0$ and $\overline{s}_i(u,v) = 0$ (which corresponds to the situation when the actual value of v is also equal to $-$), while $\underline{w}_i(u,v) = 1$ and $\underline{s}_i(u,v) = 0$ (which corresponds to the situation when the actual value of v equals $+$). In the former case, attribute i does not contribute to the Gower similarity coefficient, while in the latter case, attribute i contributes to it with the least possible value of 0.

C. Lower and Upper Bounds on Actual Value of Gower Similarity Coefficient

We start with defining lower and upper bounds on the actual value of the Gower similarity coefficient, which are achievable after replacing all missing values in the compared objects with some values from the domains of corresponding attributes.

Lower bound on the actual value of $G(u,v)$ is denoted by $\underline{G}(u,v)$ and is defined as follows:

$$\underline{G}(u,v) = \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i \in INCMP*\_ATT(u,v)} \underline{w}_i(u,v) \times \underline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \sum_{i \in INCMP*\_ATT(u,v)} \underline{w}_i(u,v)}.$$

Upper bound on the actual value of $G(u,v)$ is denoted by $\overline{G}(u,v)$ and is defined as follows:

$$\overline{G}(u,v) = \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i \in INCMP*\_ATT(u,v)} \overline{w}_i(u,v) \times \overline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \sum_{i \in INCMP*\_ATT(u,v)} \overline{w}_i(u,v)}.$$

Clearly, if $|CMP\_ATT(u,v)| + \sum_{i \in INCMP*\_ATT(u,v)} \underline{w}_i(u,v) > 0$, then $\underline{G}(u,v)$ is the strict lower bound on the actual value of $G(u,v)$, which is obtainable for some completion of missing attribute values of objects u and v, while $\overline{G}(u,v)$ is the strict upper bound on the actual value of $G(u,v)$ provided $|CMP\_ATT(u,v)| + \sum_{i \in INCMP*\_ATT(u,v)} \overline{w}_i(u,v) > 0$.

As shown in Table IV, the weight $\underline{w}_i(u,v) = 1$ for each attribute $i \in INCMP*\_ATT(u,v)$. Hence, $\underline{G}(u,v)$ can be rewritten as presented in Property 4:

Property 4.

a) $$\underline{G}(u,v) = \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i \in INCMP*\_ATT(u,v)} \underline{s}_i(u,v)}{|CMP\_ATT(u,v)| + |INCMP*\_ATT(u,v)|}.$$

b) $$\underline{G}(u,v) \geq \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i \in INCMP*\_ATT(u,v)} \underline{s}_i(u,v)}{n}$$ if $|CMP\_ATT(u,v)| + |INCMP*\_ATT(u,v)| > 0$.

Proof: Ad a) By definition of $\underline{G}(u,v)$ and the fact that $\underline{w}_i(u,v) = 1$ for each attribute $i \in INCMP*\_ATT(u,v)$ (see Table IV).
Ad b) By Property 4a and Property 2e.

Property 5. If there are no quantitative attributes in $INCMP*\_ATT(u, v)$, then

a) $$\underline{G}(u,v) = \frac{\sum_{i \in CMP_{ATT(u,v)}} s_i(u,v)}{|CMP_{ATT(u,v)}| + |INCMP*\_ATT(u,v)|}.$$

b) $\underline{G}(u,v) \geq \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v)}{n}$ if $|CMP\_ATT(u,v)| > 0$.

c) $\underline{G}(u,v) \leq G(u,v)$ if $|CMP\_ATT(u,v)| > 0$.

Proof: Ad a) By Property 4a and the fact that $\underline{s}_i(u,v) = 0$ for each non-quantitative attribute i in $INCMP*\_ATT(u, v)$ (see Table IV).
Ad b) By Property 5a and Property 2e.
Ad c) By Property 5a and Property 2a.

Example 4. Let us consider again objects u and v from Example 1 (see also Table I), whose attribute 2 is quantitative. Then, $G(u,v) = 0$, $\underline{s}_2(u,v) = \min\{(40 - 0), (100 - 40)\} / 100 = 0.4$ (see Table IV), $\underline{G}(u,v) = (1 \times 0 + 1 \times 0.4) / (1 + 1) = 0.2$. Thus, $\underline{s}_2(u,v) > G(u,v)$ and $\underline{G}(u,v) > G(u,v)$.

Example 4 allows us to conclude what follows:

Property 6. Let u and v be comparable objects. Let i be a quantitative attribute with missing value for object u and known value for object v. Then:

a) It is probable that $\underline{s}_i(u,v) > G(u,v)$.

b) If $\underline{s}_i(u,v) > G(u,v)$, then it is probable that $\underline{G}(u,v) > G(u,v)$.

Corollary 1. It is probable that $\underline{G}(u,v) > G(u,v)$ when there is a missing value in u or v. If $\underline{G}(u,v) > G(u,v)$, then $G(u,v)$ takes an incorrect value, which cannot be obtained for any possible actual value of attribute i of object u.

To avoid the problem stated in Corollary 1, one may use, depending on an application, the lower bound $\underline{G}(u,v)$, the upper bound $\overline{G}(u,v)$ or an appropriately modified version of $G(u,v)$ instead of $G(u,v)$ itself. Below we introduce new $G'(u,v)$ similarity coefficient defined as follows:

$$G'(u,v) = \frac{\sum_{i \in CMP\_ATT(u,v)} s_i(u,v) + \sum_{\substack{i \in INCMP*\_ATT(u,v), \\ \underline{s}_i(u,v) > G(u,v)}} \underline{s}_i(u,v)}{|CMP\_ATT(u,v)| + |\{i \in INCMP*\_ATT(u,v) \mid \underline{s}_i(u,v) > G(u,v)|}.$$

In fact, $G'(u,v)$ can be regarded as an improved version of $G(u,v)$.

Let $INCMP*\_QNT\_ATT(u,v)$ be the set of the quantitative attributes in $INCMP*\_ATT(u,v)$. Now, we will express $G'(u,v)$ in terms of attributes in $CMP\_ATT(u,v) \cup INCMP*\_QNT\_ATT(u,v)$.

Property 7. Let |*CMP_ATT*(*u*,*v*)| > 0. Then:

$$G'(u,v) = \frac{\sum_{i\in CMP\_ATT(u,v)} s_i(u,v) + \sum_{\substack{i\in INCMP*\_QNT\_ATT(u,v,\\ \underline{s}_i(u,v)>G(u,v)}} \underline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \ |\{i\in INCMP*\_QNT\_ATT(u,v)|\ \underline{s}_i(u,v)>G(u,v)|}.$$

Proof: By assumption, u and v are comparable, so $G(u,v) \geq 0$. If i is a qualitative or dichotomous attribute in *INCMP*_ATT*(*u*, *v*), then $\underline{s}_i(u,v) = 0$ (see Table IV), and so, $\underline{s}_i(u,v)$ is not greater than $G(u,v)$. So, $\underline{s}_i(u,v)$ can be greater than $G(u,v)$ only if i is a quantitative attribute in *INCMP*_ATT*(*u*, *v*); i.e., if $i \in$ *INCMP*_QNT_ATT*(*u*,*v*).

Please note that $G'(u,v)$ differs from $G(u,v)$ in that the value of $G'(u,v)$ is calculated not only on the attributes on which u and v are comparable (as in the case of $G(u,v)$), but also on those quantitative attributes i on which u and v are not comparable provided $\underline{s}_i(u,v) > G(u,v)$.

Property 8. Let u and v be comparable objects. Then:

a) $G'(u,v) \geq G(u,v)$.

b) $G'(u,v) \geq \underline{G}(u,v)$.

Proof: Ad a) By definition of $G'(u,v)$ and Property 2a.
Ad b) By definition of $G'(u,v)$ and Property 4a.

Example 5. In the case of objects u and v from Example 1 (see also Table I), $G'(u,v) = \underline{G}(u,v) = 0.2 > G(u,v) = 0$.

We will consider now the properties of the upper bound on Gower similarity coefficient.

Property 9. If there are no dichotomous attributes in *INCMP*_ATT*(*u*, *v*), then:

$$\overline{G}(u,v) = \frac{\sum_{i\in CMP\_ATT(u,v)} s_i(u,v) + |INCMP*\_ATT(u,v)|}{|CMP\_ATT(u,v)| + \ |INCMP*\_ATT(u,v)|}.$$

Proof: By definition of $\overline{G}(u,v)$ and the fact that $\overline{s}_i(u,v) = 1$ and $\overline{w}_i(v,u) = 1$ for any non-dichotomous attribute i on which u and v are incomparable (see Table IV).

Finally, we check the relationship between $\overline{G}(u,v)$ and $G'(u,v)$ as well as between $\overline{G}(u,v)$ and $G(u,v)$.

Property 10. Let u and v be comparable objects. Then:

a) $\overline{G}(u,v) \geq G'(u,v)$.

b) $\overline{G}(u,v) \geq G(u,v)$.

Proof: Ad a) In the proof, we will use the property saying that $\overline{s}_i(u,v) = 1 \geq \underline{s}_i(u,v)$ and $\overline{w}_i(v,u) = 1$ for any attribute $i \in$ *INCMP*_QNT_ATT*(*u*,*v*) (*) and that for any attribute $j \in$ *INCMP*_ATT*(*u*,*v*) \ *INCMP*_QNT_ATT*(*u*,*v*) either: (i) $\overline{s}_j(u,v) = 1$ and $\overline{w}_j(u,v) = 1$ or (ii) $\overline{w}_j(u,v) = 0$ (**).

Thus, by definition,

$$G'(u,v) =$$

$$\frac{\sum_{i\in CMP\_ATT(u,v)} s_i(u,v) + \sum_{\substack{i\in INCMP*\_QNT\_ATT(u,v,\\ \underline{s}_i(u,v)>G(u,v)}} \underline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \ |\{i\in INCMP*\_QNT\_ATT(u,v)|\ \underline{s}_i(u,v)>G(u,v)|}$$

/ by (*) /

$$\leq \frac{\sum_{i\in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i\in INCMP*\_QNT\_ATT(u,v)} \overline{w}_i(u,v)\times\overline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \sum_{i\in INCMP*\_QNT\_ATT(u,v)} \overline{w}_i(u,v)}$$

/ by (**) /

$$\leq \frac{\sum_{i\in CMP\_ATT(u,v)} s_i(u,v) + \sum_{i\in INCMP*\_ATT(u,v)} \overline{w}_i(u,v)\times\overline{s}_i(u,v)}{|CMP\_ATT(u,v)| + \sum_{i\in INCMP*\_ATT(u,v)} \overline{w}_i(u,v)}$$

$$= \overline{G}(u,v).$$

Ad b) By Property 10a and Property 8a.

SUMMARY

In the article, we introduced lower and upper bounds on the actual similarity value on an attribute and on the actual value of the Gower similarity coefficient. We showed that the Gower similarity coefficient for two objects may take an incorrect value, which would be less than the lower bound on the actual value of the Gower similarity coefficient for those objects, if one of the objects has a missing value for at least one quantitative attribute. To solve this problem, we introduced coefficient *G'*, being a modification of the Gower similarity coefficient, that is free from this deficiency. A number of properties of similarity value of objects on the attribute, the Gower similarity coefficient, the introduced lower and upper bounds and the coefficient *G'* were derived.

REFERENCES

[1] B. Ben Ali, Y. Massmoudi, "K-means clustering based on gower similarity coefficient: A comparative study," 2013 5th International Conference on Modeling, Simulation and Applied Optimization, ICMSAO 2013. https://doi.org/10.1109/ICMSAO.2013.6552669.

[2] S. S. K J. Chae and W. Y. Yang, "Cluster analysis with balancing weight on mixed-type data," The Korean Communications in Statistics, vol. 13, no. 3, 2006, pp. 719–732, http:\\DOI:10.5351/CKSS.2006.13.3.719.

[3] J. Fontecha, R. Hervás, and J. Bravo, "Mobile Services Infrastructure for Frailty Diagnosis Support based on Gower's Similarity Coefficient and Treemaps," Mobile Information Systems, vol. 10, Article ID 728315, 20 pages, 2014. https://doi.org/10.1155/2014/728315.

[4] J. C. Gower, "A General Coefficient of Similarity and Some of Its Properties, Biometrics, " Vol. 27, No. 4. (Dec., 1971), pp. 857-871, https://doi.org/10.2307/2528823.

[5] S. Pavoine, J. Vallet, A.-B. Dufour, S. Gachet, and H. Daniel, "On the challenge of treating various types of variables: application for improving the measurement of functional diversity," Oikos, 118(3) 2009, pp. 391-402, https://doi.org/10.1111/j.1600-0706.2008.16668.x.

[6] G. Philip and B. S. Ottaway, "Mixed data cluster analysis: an illustration using cypriot hooked-tang weapons, " Archaeometry, vol. 25, no. 2, 1983, pp. 119–133, https://doi.org/10.1111/j.1475-4754.1983.tb00671.x.

[7] J. Podani and D. Schmera: "Generalizing resemblance coefficients to accommodate incomplete data," Ecological Informatics 66 (2021) 101473, https://doi.org/10.1016/j.ecoinf.2021.101473

[8] G. Tuerhong and S. B. Kim, "Gower distance-based multivariate control charts for a mixture of continuous and categorical variables," Expert Systems with Applications, 41(4 PART 2), 2014, pp. 1701–1707, https://doi.org/10.1016/j.eswa.2013.08.068.

DOI: 10.15439/2023F4827
ISSN 2300-5963 ACSIS, Vol. 35

Mechanism for detecting cause-and-effect relationships in court judgments

Łukasz Kurant
0000-0002-2523-5952
University of Maria Curie-Sklodowska
in Lublin
Pl. M. Curie-Skłodowskiej 5, 20-031 Lublin, Poland
Email: lukasz.kurant@mail.umcs.pl

***Abstract*—Among the solutions for the detection of cause-and-effect relationships are methods based on knowledge, statistical solutions or methods allowing the use of deep learning. The solution presented in the article uses bidirectional artificial neural networks LSTM to detect such relationships in legal texts in Polish. The analysis was performed at the sentence level, but due to the specific legal language and the focus on Polish, two separated networks were used in the experiment. The task of the first one is to classify whether a sentence contains a conditional, while the second one is to identify the elements of this relationship. Both use word embedding sets for the Polish language corpus. The results of the experiment prove that it is possible to perform such extraction with satisfactory results, and raise questions and point to further possible ways forward.**

I. Introduction

DETECTING cause-and-effect relationships in texts, is a task that requires advanced cognitive processes and is not a trivial problem. Inference itself can often be a very difficult task for human beings, so it is not surprising that attempts are made to automatically process and extract such relationships. Such data can be of significant value to many fields of science, including the field of law. Performing inference and argumentation in a proper and automatic manner can be used for many purposes and can assist those using legal texts in their daily work.

A. Causality relationship

We can define causality as a relationship between events e_1 and e_2, such that the occurrence of event e_1 results in the occurrence of event e_2 [1]. The following division of causality is made [2]:

- Explicit causality, which occurs in a sentence in the form of overt, often with conjunctions or causal phrases, such as in the sentence: “I did not attend the event because I was not invited.”
- Implicit causality, which does not occur in overt form overt, and can often be split into several sentences, such as in the sentences: “Drive slower. It’s slippery.”

It should be noted that in some cases the sentences, causes or effects may be unequal to each other, such as “The reason for the verdict was the evidence supporting the defendant’s guilt, but also the lack of cooperation on his part”. In this example, both “evidence supporting the defendant’s guilt” and “a lack of cooperation on his part” are causes in a sentence. Cause and effect can also be nested as in the sentence: “Refusal to testify or failure to appear at trial cause the court’s disfavor and the defense counsel’s concern.”. In this case, both “refusal to testify” and “failure to appear at trial” may cause further effects. Associations may also share certain parts with each other. In the next example, the effect of the first cause is also a cause for the next effect: “The defendant’s inappropriate behavior caused agitation in the courtroom, as a result of which the court had to cancel the hearing”.

Causality can be single-sentence or multi-sentence. Single-sentence is often combined with so-called overt causal conjunctions and phrases, which we can divide into:

- causal conjunctions: “because”, “as”, “cause”,
- result phrases: “as a result”, “due to”, "because of”,
- conditional phrases: “if ... then ...”.

In the case of implicit or multi-sentence causality, it is up to the reader to use basic knowledge to analyze and infer to detect it. These are much more complicated and therefore more difficult to analyze [3].

B. Practical uses

Detecting causal relationships in texts is of immense value and can be used in predictive and analytical tasks [3]. Having such information can be helpful in many fields [4], such as

- medicine, when analyzing medical cases,
- learning about the causes of security incidents,
- learning about the effects of natural disasters, etc.

In the context of the legal field, information about such relationships can carry a lot of value, for example, in the context of adjudicating court cases (especially in countries where the law of precedent is used, such as the US) and can be an important aid to judges in formulating a verdict. Also for prosecutors, or attorneys, such information can help in taking the right strategy in the courtroom.

II. Research status

Two main types of methods can be found in articles and scientific papers to detect cause-and-effect relationships [3]:

- methods based on patterns or rules [5], [6], [7],

- methods based on machine learning techniques [8], [17], [18], which we can divide into statistical methods (e.g. using decision trees, Naive Bayes algorithm or linear regression) and deep learning methods (neural networks).

The first type manifests weakness in many areas due to the need to create very sophisticated rules or patterns, thus requiring a lot of domain knowledge. Methods based on machine learning, on the other hand, although built without human intervention, need to be programmed and trained, thus requiring a lot of hardware and time resources.

A common approach appearing in the literature, is the use of a two-step causality extraction: first the detection of candidates is done, and then the classification of relationships. In this approach, there can be so-called cascading errors [19], i.e. errors that, when present in the first step, can significantly affect the results of the next step.

A. Data preparation

In order for the model to be trained, proper data preparation is required. The authors of [13] used the technique of labeling words using the Cartesian product of entity and relation tags, and then assigned a unique tag to the word. On the other hand, in [8] a new approach was proposed, the so-called "BIO and CEEmb" labeling of words based on tags: cause (C), effect (E) and embedded causality (Emb). An additional step is to mark each word as the beginning of the cause/effect (B), the continuation of the cause/effect (I), and another word (O). This approach makes it possible to formulate causal triples. Suppose we have a sentence, "The court refused to continue the trial due to the absence of the defendant." After analyzing this example, we can formulate a causal triple, where two events are divided by the type of relationship (in this case, cause-effect): "refusal to continue the trial, cause-effect, the absence of the defendant".

However, some tagging schemes, such as the one proposed in [17], cannot identify overlapping relationships. To solve this in [8], the authors use the "Tag2Triplet" algorithm, which allows the extraction of nested relationships in which individuals can be part of multiple ones. For example, the sentence "As a result of the incident, the plaintiff was unable to testify, leading to incorrect conclusions." contains an effect, i.e. the lack of testimony, which is also the cause of another effect, i.e. "incorrect conclusions."

B. Detecting and extraction

Following the determination process, the dominant approach is using recurrent LSTM neural networks in varieties with connection to conditional random fields [8] or in the Bi-LSTM type [4]. Some works in [21] or [22] have focused on detecting causality per se without dividing it into full relations (they detect sentences in which such a relation exists without dividing them into cause and effect), and some, e.g. in [4] or [23] focus on identifying linguistic expressions useful in describing causality (such as conjunctions and causal phrases).

In [8], [24], authors also point out that the use of word embedding layers makes a significant contribution to improving the performance and overall results of causality extraction. To improve the detection of relationships that remain far apart, various techniques are being introduced, such as the so-called self-suggestion mechanism [25], which, unlike the classical LSTM approach, can lead to a connection between arbitrarily distant words [26], and thus detect relationships between words in a more sophisticated way. This is because the meaning of a word is defined in the context of its entire surroundings, and not just (as in simple recurrent networks) based on what is immediately before or after it. The main problem in causality extraction is the embedding of such relationships in the text. On the other hand, extracting prepositions or effects without traditional conjunctions ("because", "since", "if", etc.) is an extremely difficult task [8].

III. Experiment

Causality can often be buried very deeply in a text, and even a person himself may have trouble pointing it out. Extracting such relationships from legal texts significantly narrows the corpus of words that can be used. In addition, the collection can be narrowed even further when focusing on a specific type of legal texts, such as the texts of court judgments. Among current studies, such experiments, i.e. causality analyses for legal texts, are lacking, especially when talking about Polish.

In the experiment, we focused on the extraction of explicit causality at the sentence level. This task is divided into two parts — the first goal is to indicate whether a sentence contains a causal relationship, while the second is to label and extract parts of such relationships. Not only semantic analysis becomes important here, but also the construction of the sentence itself. The biggest problem in this type of experiment is undoubtedly the lack of a suitable learning set in Polish. Therefore, it became necessary to manually prepare such a set before starting further analysis. In order to perform it, recurrent neural networks with LSTM-type cells were used, along with layers of word embeddings.

A. Data preparation

To conduct the experiment, it was necessary to prepare a dataset. For this purpose, legal texts were used, specifically court judgments from several open sources [9], [10], [11], [12]. The total number of judgment texts was 150. Using the author's script (adopting the beautifulsoup library in Python [13]), court judgments were downloaded from the above-mentioned sources in HTML format, and then converted to text and divided into sentences (using the NLTK library [14]). Each document has been marked accordingly (as indicated below). Two datasets were manually prepared for the experiment: the first set in order to perform binary classification on it — each sentence was assigned a positive or negative label, depending on the presence or absence of a cause-and-effect relationship in it. The second set was prepared based on sequence labeling, where each word in a sentence was assigned a label indicating its type in a sentence with causality. Both collections were prepared manually, requiring human intervention. We marked

Example 3.1 (Examples of elements of the first set):

Natomiast przedawnieniu podlega samo ustalenie odszkodowania, gdyż wg woli ustawodawcy następuje ono w formie decyzji administracyjnej.;1

Niezbędne jest dodatkowo wykazanie konieczności wyjaśnienia zakresu sprawy.;0

TABLE I
THE NUMBER OF CLASSES IN THE FIRST SET

	Number of elements	Percentage
Class 0 (no relation)	22060	92.01%
Class 1 (with a relationship)	1914	7.99%
Total	23974	100%

the data manually on our own, then we verified it (for this purpose we used Doccano software [15]).

1) First dataset: In the first set, each sentence was labeled, i.e. assigned a corresponding class, according to the presence of a causal relationship (class 1, positive) or its absence (class 0, negative), as shown in the Example 3.1. It is worth noting that this set is not a balanced set — the negative clause significantly dominates (Table I), which has implications for further text analysis.

2) Second dataset: Each sentence that contained a cause-effect relationship was additionally labeled, i.e. each word was given a membership in one of the groups: cause (class 0), effect (class 1), causal phrase (class 2), other (class 3). A collection of such sentences, divided into words, has been marked accordingly (Example 3.2). Each element was labeled in such a way that it could contain multiple consecutive words within it (Table II). The tagging method is a modified version of the method presented in the [20].

Example 3.2 (Example element of the second set):

w ocenie sądu okręgowego nagrody z zakładowego funduszu nagród wypłacone wnioskodawcy niepodlegają uwzględnieniu przy ustalaniu podstawy wymiaru renty gdyż nie były zaliczane do wynagrodzeń osobowych 11111111111111111111112000000 cause-effect-sentence

TABLE II
THE NUMBER OF CLASSES IN THE SECOND SET

	Number of elements	Percentage
Class 0 (cause)	1748	32.43%
Class 1 (effect)	1729	32.08%
Class 2 (causal phrase)	1774	32.91%
Class 3 (other)	139	2.58%
Total	5390	100%

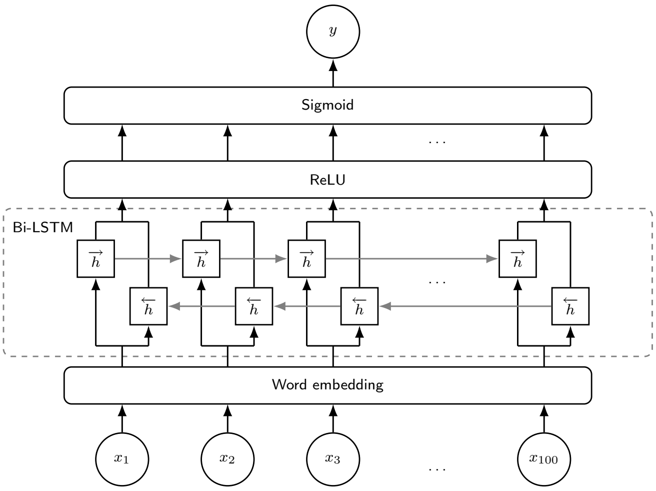


Fig. 1. The architecture of the network responsible for marking a sentence as having causality or not

B. RNNs structures

The program for detecting explicit cause-effect relationships was built on the basis of two separated neural networks. For this purpose, the Tensorflow library and the Keras interface were used [16]. The first network is responsible for binary classification of whether a cause-effect relationship is present in a sentence. The second network is tasked with performing cause-and-effect extraction, i.e. labeling a sentence with a cause-and-effect relation, assigning each word a token of the appropriate class. In both cases, validation of the correctness of the trained models is carried out at the end of the subroutines.

1) First network: The input data is properly prepared before entering the network, by dividing it into tokens, removing punctuation and whitespace characters. The set is divided in a 7:3 ratio into a learning set and a validation set. The next step is to transform the sentences into a dense feature vector using a set of word embeddings for the Polish language [27], [28], i.e. a 100-dimensional corpus containing all parts of speech, created using the CBOW architecture. Based on the subset counts, the weights of each class are calculated (due to the unbalanced dataset). The data then becomes the input for a recurrent neural network in the Bi-LSTM variant, in which the first layer is the word embedding layer (loaded earlier). The detailed architecture of the network is shown on Fig. 1.

The model was created using standard binary cross entropy as a loss function and the adam optimization algorithm. After training, the model is tested with a validation set and evaluated (precision, recall, F1 and accuracy values are calculated, as well as the ROC curve and the value under the AUC curve). The training process took place in 8 epochs, during which all the above metrics were measured.

2) Second network: The task of the second neural network is to extract cause-and-effect relationships from a sentence evaluated positively as containing causality. Each word must be assigned one of four classes: cause, effect, connective phrase or another word. The input sentences, as in the case

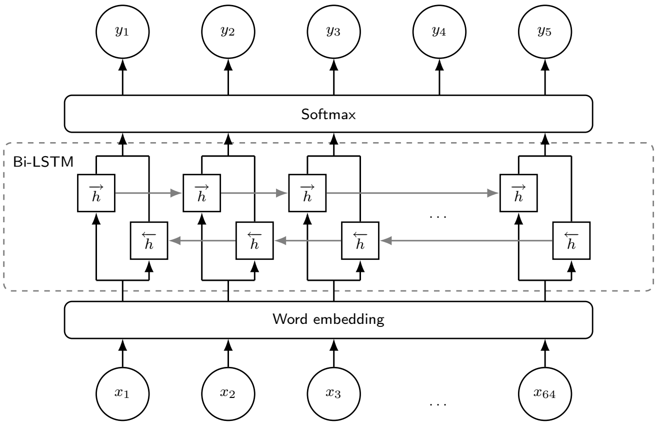


Fig. 2. The architecture of the network responsible for extracting the cause and effect parts

TABLE III
THE RESULTS OF THE FIRST NEURAL NETWORK

	Precision	**Recall**	**F1**
Class 0	0.81	0.89	0.85
Class 1	0.99	0.98	0.99
Macro	0.90	0.94	0.92
Weighted	0.98	0.98	0.98

of the first neural network, undergo preprocessing (identical to that described above), and then go as input to the neural network in the Bi-LSTM variant (Fig. 2). Due to the small size of the collection, cross-validation was used in the validation process, i.e. the collection was divided into ten parts and trained nine of them at a time, and tested the last one. After the training process, the results of the network are validated using the metrics of precision, accuracy, recall, and F1 index, both for each class and the entire collection. The training process took place over 10 epochs, during which all of the above metrics were measured.

IV. Results

The following tables present the values of the metrics for each set and each program. Table III shows the validation results of the first neural network tasked with binary classification. The accuracy for the entire set was 97.68%.

The value of AUC = 0.9822, which shows that the classifier can correctly distinguish class elements. The high precision is maintained throughout the learning period of the model due to issues related to the unbalanced dataset, described below. Details of the values of the metrics at training time (at a specific epoch) are shown in Fig. 3-7. Noteworthy, this curve gives an incomplete picture of the classifier, due to the unbalanced dataset. The more important information is the values for the class with causality sewn in, the results of which no longer look so good (as can be seen in the confusion matrix of validation set in Table IV).

As the results indicate, the classification of such relationships is not a simple task, but to some extent it is feasible.

TABLE IV
CONFUSION MATRIX OF THE FIRST NEURAL NETWORK

	Actually positive	**Actually negative**
Predicted positive	477	58
Predicted negative	109	6549

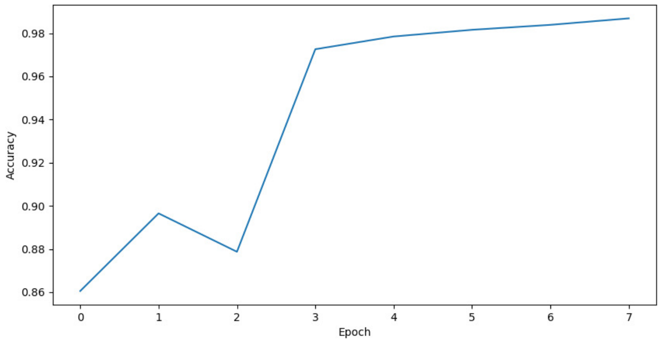


Fig. 3. Accuracy in training the first neural network

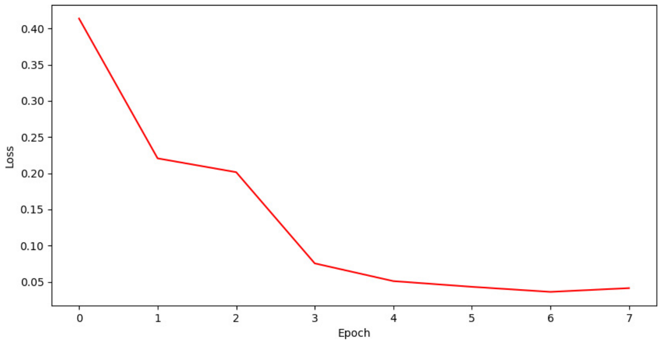


Fig. 4. Loss in training the first neural network

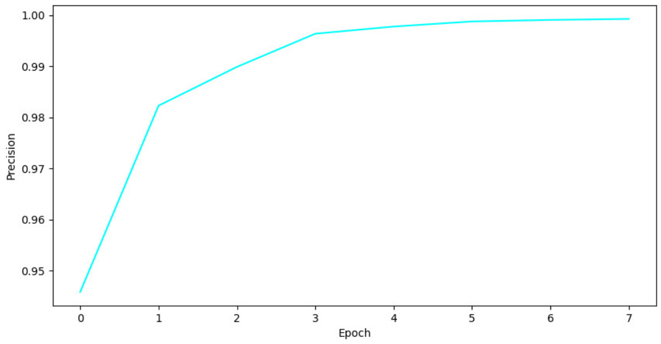


Fig. 5. Precision in training the first neural network

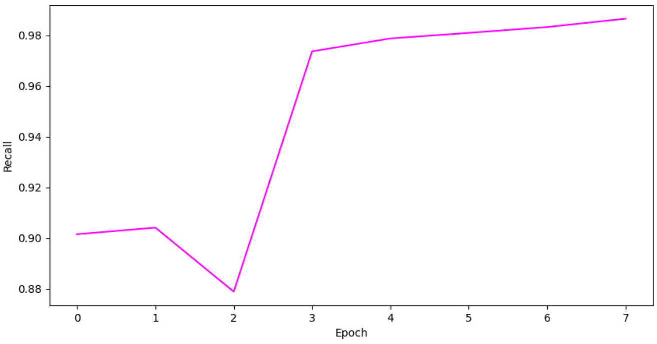


Fig. 6. Recall in training the first neural network

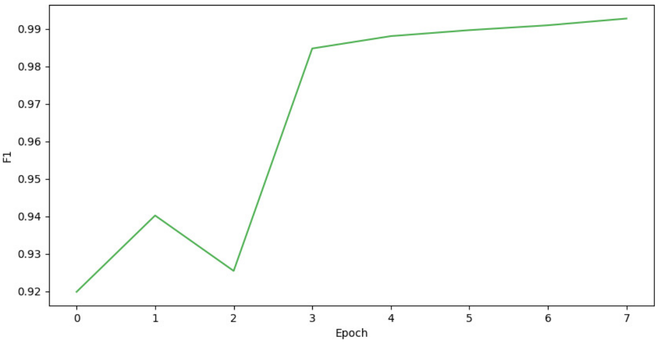


Fig. 7. F1 process of training the first neural network

This is greatly influenced by words that are parts of causal phrases, but it should be noted that there is never such certainty. For example, when a sentence contains the word “albowiem”, which often occurs in legal language, this may or may not indicate conditionality. The word “bowiem” in most cases separates the cause and effect parts, but there are also exceptions to this.

Table V shows the results for the second neural network, which was tasked with extracting the parts belonging to causal relationships. The results here are much worse. What stands out here is the better result of the causal phrase class, due to the frequent occurrence of the same phrases and words. The results here are probably also influenced by the small collection. In contrast, class with other words (class 3) performs in a clearly negative way, given the problem of indicating it in a sentence, because there are no special rules formulated in the experiment for the occurrence of such a class.

TABLE V
THE RESULTS OF THE SECOND NEURAL NETWORK

	Precision	Recall	F1
Class 0	0.58	0.58	0.58
Class 1	0.50	0.51	0.51
Class 2	0.72	0.89	0.80
Class 3	0.18	0.06	0.09
Macro	0.60	0.61	0.60
Weighted	0.83	0.83	0.83

It should also be noted that legal texts (especially court judgments) often have sentences that are very rote in their construction, i.e. contain many subordinate sentences, which also affects such analysis. The network also did not cope when a word or phrase indicating causality was located at the beginning of a sentence. In such a case, the word “ponieważ” does not separate the causal part from the effect, so the network’s results were subject to high error.

V. Conclusions

The biggest problem with the argument extraction experiment became the lack of a suitable training set. There is no such set for the Polish language in the sources, which made it necessary to create such a set manually. This was a tedious activity, but at the same time required adequate attention. Closing the corpus of words to the texts of court rulings, significantly simplified the analysis and marking of sentences, due to the orderly structure of the text, often containing similar causal phrases. Judgment texts, like other legal texts, are often written in correct language, but stylistic, punctuation and even spelling errors can be found among them (unlike, for example, the texts of statutes). The structure of a court decision itself looks very similar, regardless of the court or its type (division into a operative part, justification or cited provisions).

In the case of the first set (the input for the first neural network tasked with binary classification), sentences that have a causal relationship in them make up a small percentage of the set. Hence, it is necessary to set up the neural network in such a way as to notify it of the greater importance of certain elements of the set. The reason for using such a set is to reflect the real ratio of sentences that contain a causal relationship to those that do not. The use of a word embedding layer with a trained set of vectors for the Polish language also has a broad impact on better results.

In some cases, the word occurs with cause (without effect), indicating that causation is missing at the sentence level. Thus, it cannot be assumed that syntactic analysis alone would carry significant information about the semantics of the sentence, but it would be largely sufficient. Reviewing the results, we can note the following regularities. For example, a sentence containing a causal connective phrase has a high degree of certainty about the occurrence of a cause in it. On the other hand, a sentence that does not have such a phrase with the highest probability is assigned to a class with no such relationship.

VI. Future works

To develop the topic of causal relationship extraction in the future, it would therefore be important to create a suitably large and diverse test dataset. Semantic analysis at the level of the whole document, and not just at the sentence level, would also be an important element. This would allow detection of arguments that are implicit relationships (sewn into the text), often found in different parts of the document. As research in the field of detecting such relationships shows, this task is not easy. When analyzing texts in Polish, we often also have to pay attention to other elements absent in other languages, which makes such texts significantly more difficult to analyze semantically for causality. The resulting data from this experiment can successfully serve for further research and be the basis for other tasks in the area of machine learning in the field of law.

Acknowledgment

Special thanks to dr Tomasz Żurek for inspiration and his help with defining the described problem and dr Andrzej Bobyk for providing his knowledge and experience.

References

[1] Stanford Encyclopedia of Philosophy, *Causal Models*, 2022 https://plato.stanford.edu/entries/causal-models/

[2] E. Blanco, N. Castell, and D. Moldovan, *Causal relation extraction,* Proceedings of the Sixth International Conference on Language Resources and Evaluation, 2008, pp. 310

[3] Yang J, Han S. C. and Poon J. *A survey on extraction of causal relations from natural language text*, Knowledge and Information Systems 64, 2022, pp. 1161-1186, https://doi.org/10.48550/arXiv.2101.06426

[4] T. Dasgupta, R. Saha, L. Dey, and A. Naskar, *Automatic Extraction of Causal Relations from Text using Linguistically Informed Deep Neural Networks*, Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, Melbourne, Australia. Association for Computational Linguistics, 2018, pp. 306-316, http://dx.doi.org/10.18653/v1/W18-5035

[5] C. S. G. Khoo, J. Kornfilt, R. N. Oddy, and S. H. Myaeng, *Automatic Extraction of Cause-Effect Information from Newspaper Text Without Knowledge-based Inferencing,* Literary and Linguistic Computing, Volume 13, Issue 4, 1998, pp. 177–186, https://doi.org/10.1093/llc/13.4.177

[6] C. S. G. Khoo, S. Chan, and Y Niu, *Extracting Causal Knowledge from a Medical Database Using Graphical Patterns*, Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, Association for Computational Linguistics, 2000, pp. 336–343, http://dx.doi.org/10.3115/1075218.1075261

[7] R. Girju, D. Moldovan, *Text Mining for Causal Relations*, Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, 2002, pp. 360–364

[8] R. Girju, *Automatic Detection of Causal Relations for Question Answering*, Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering, Association for Computational Linguistics, 203, pp. 76–83, http://dx.doi.org/10.3115/1119312.1119322

[9] Portal Orzeczeń Sądów Powszechnych, https://orzeczenia.ms.gov.pl/

[10] Wyrok.org — Największa baza wyroków w Polsce, https://wyrok.org/

[11] Centralna Baza Orzeczeń Sądów Administracyjnych, https://orzeczenia.nsa.gov.pl/

[12] Dziennik wyroków i ogłoszeń sądowych, https://www.ebos.pl/

[13] Beautiful Soup Python library, https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[14] NLTK: Natural Language Toolkit, https://www.nltk.org

[15] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi and X. Liang, *Doccano: Text Annotation Tool for Human*, 2018, https://github.com/doccano/doccano

[16] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*, 2015, https://doi.org/10.48550/arXiv.1603.04467

[17] A. Sorgente, G Vettigli, and F. Mele: *Automatic extraction of cause-effect relations in Natural Language Text*, Proceedings of the 7th International Workshop on Information Filtering and Retrieval co-located with the 13th Conference of the Italian Association for Artificial Intelligence, 2013, pp. 37–48

[18] S. Zhao, T. Liu, S. Zhao, Y. Chen, and J. Nie, *Event causality extraction based on connectives analysis*, Neurocomputing 173, 2016, pp. 1943–1950, https://doi.org/10.1016/j.neucom.2015.09.066

[19] Z. Li, Q. Li, X. Zou, and J. Ren, *Causality Extraction based on Self-Attentive BiLSTM-CRF with Transferred Embeddings*, Neurocomputing 423, 2021, pp. 209, https://arxiv.org/abs/1904.07629

[20] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, *Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme*, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2017, pp. 1227–1236, http://dx.doi.org/10.18653/v1/P17-1113

[21] T. N. de Silva, X. Zhibo, Z. Rui, and M. Kezhi, *Causal Relation Identification Using Convolutional Neural Networks and Knowledge Based Features*, International Journal of Computer, Electrical, Automation, Control and Information Engineering 11 (6), 2017, pp. 697–702, https://doi.org/10.5281/zenodo.1130679

[22] C. Kruengkrai, K. Torisawa, C. Hashimoto, J. Kloetzer, J. H. Oh, and M. Tanaka, *Improving Event Causality Recognition with Multiple Background Knowledge Sources Using Multi-Column Convolutional Neural Networks*, Proceedings of the AAAI Conference on Artificial Intelligence 31(1), 2017, https://doi.org/10.1609/aaai.v31i1.11005

[23] J. Dunietz, J. Carbonell, and L. Levin, *DeepCx: A transition-based approach for shallow semantic parsing with complex constructional triggers*, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1691–1701, http://dx.doi.org/10.18653/v1/D18-1196

[24] A. Akbik, D. Blythe, and R. Vollgraf, *Contextual String Embeddings for Sequence Labeling*, Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1638–1649

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is All you Need,* Advances in Neural Information Processing Systems 30, 2017, pp. 6000–6010, https://doi.org/10.48550/arXiv.1706.03762

[26] Z. Tan, M. Wang, J. Xie, Y. Chen, and X. Shi: *Deep Semantic Role Labeling with Self-Attention*, Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, https://doi.org/10.48550/arXiv.1712.01586

[27] A. Przepiórkowski, M. Bańko, R. L. Górski, and B. Lewandowska-Tomaszczyk, *Narodowy Korpus Języka Polskiego*, Wydawnictwo Naukowe PWN, Warsaw, 2012

[28] A. Mykowiecka, M. Marciniak, and P. Rychlik, *Testing word embeddings for Polish*, 2017, http://dsmodels.nlp.ipipan.waw.pl

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1047–1052

DOI: 10.15439/2023F9859
ISSN 2300-5963 ACSIS, Vol. 35

Expectation-Maximization Algorithms for Gaussian Mixture Models Using Linear Algebra Libraries on Parallel Shared-Memory Systems

Wojciech Kwedlo
0000-0002-5040-2302
Faculty of Computer Science
Białystok University of Technology
Wiejska 45A, 15-351 Białystok, Poland,
w.kwedlo@pb.edu.pl

***Abstract*—In this paper the problem of parameter estimation of Gaussian mixture models using the expectation-maximization (EM) algorithm is considered. Four variants of the EM algorithm parallelized using the OpenMP standard are proposed. The main difference between the variants is the degree of usage of vendor-optimized linear algebra libraries. The computational experiments were performed using 25 large datasets on a system with two 12-core Intel Xeon processors. The results of experiments indicate that the EM variant using level 3 (matrix-matrix) operations and L3 cache blocking is the fastest one. It is 1.75–2.75 times faster than the naive version using level 2 (matrix-vector) operations. Its parallel efficiency relative to the sequential version is always greater than 83%.**

I. Introduction

FINITE mixture models [1] are a very versatile tool used for modeling complex probability distributions. Gaussian mixture models (GMMs) which assume multivariate normal density of a component, are arguably the most popular mixture models. GMMs have been successfully applied to many problems in engineering, finance, biology and data mining.

The maximum likelihood estimation (MLE), which seeks a maximum of the log-likelihood function, is a method of choice for GMM parameter estimation. The expectation-maximization (EM) algorithm [2] is the most common approach for MLE of GMM parameters. The algorithm is simple and easy to implement. Its important drawback is high computational complexity. The complexity of a single iteration is $O(NKd^2)$, where N is the number of data items, K is the number of mixture components, and d is the dimension of a feature space. These high computational requirements limit the usability of the EM, especially when d is large.

The problem of high computational requirements can be tackled a by parallel realization of the EM for GMMs (e.g., [3]). The importance of parallel formulations of the EM stems from ubiquity of relatively cheap multi-core processors. However, these processors have complex structures with multiple SIMD execution units and two- or three-level hierarchy of cache memory. This complexity makes an efficient implementation of the EM a tedious task. The difficulties in an efficient implementation can be alleviated by using a vendor-optimized matrix algebra libraries, for instance based on the BLAS standard [4].

This paper proposes four such parallel formulations, two of which use level 2 BLAS calls, and the remaining two leverage more efficient level 3 BLAS operations. The proposed algorithms are parallelized using the OpenMP standard, implemented in C++, and employ the Eigen template library[1] which seamlessly invokes the BLAS calls. We also investigate the use of blocking [5] for L3 cache. The computational experiments indicate that this optimization significantly improves the performance of the EM variant based on level 3 BLAS calls.

II. GMM parameter estimation

A finite mixture model with K components has the probability density function given by:

$$f(\mathbf{x}|\boldsymbol{\Theta}) = \sum_{m=1}^{K} \alpha_m \phi(\mathbf{x}; \boldsymbol{\theta}_m), \tag{1}$$

where $\phi(\mathbf{x}; \boldsymbol{\theta}_m)$ is the probability density function of the m-th component parameterized on $\boldsymbol{\theta}_m$, and $\alpha_1, \ldots, \alpha_K$ are the mixing proportions which must satisfy the following two conditions: $\alpha_1 + \cdots + \alpha_K = 1$ and $\alpha_m \geq 0$ for $m = 1, \ldots, K$. $\boldsymbol{\Theta} = \{\alpha_1, \ldots, \alpha_k, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ is the complete set of parameters defining the mixture.

In Gaussian mixture models each component has the following (multivariate normal) probability density function:

$$\phi(\mathbf{x}; \boldsymbol{\theta}_m) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \\ \frac{1}{(2\pi)^{d/2} \det(\boldsymbol{\Sigma}_{\mathbf{m}})^{1/2}} e^{\left[-0.5(\mathbf{x}-\boldsymbol{\mu}_m)\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)^T\right]}, \tag{2}$$

with the set of parameters $\boldsymbol{\theta}_m = [\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m]$, where d is the dimension of the feature space, $\boldsymbol{\mu}_m \in \mathbb{R}^d$ is the mean

This work was supported by the grant WZ/WI-IIT/4/2023 from Białystok University of Technology. Computations were carried out using the computers of Centre of Informatics Tricity Academic Supercomputer & Network, in Gdansk, Poland.

[1] http://eigen.tuxfamily.org

and $\mathbf{\Sigma_m}$ is the $d \times d$ covariance matrix. Thus, for a GMM the complete set of mixture parameters is given by $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K\}$.

Given a set of N independent and identically distributed feature vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$, the log-likelihood function, corresponding to a K-component mixture is given by:

$$\log f(X|\mathbf{\Theta}) = \sum_{i=1}^{N} \log \sum_{m=1}^{K} \alpha_m \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_m, \mathbf{\Sigma}_m). \tag{3}$$

The maximum likelihood estimate of parameters is obtained as: $\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmax}} \log f(X|\mathbf{\Theta})$.

The EM algorithm [2], [6] is an iterative procedure which, given an initial estimate of parameters $\mathbf{\Theta}^{(0)}$, produces a sequence of estimates with increasing log-likelihood (3). j-th iteration of the algorithm consists of two steps called expectation step (E-step) and maximization step (M-step).

In the E-step [7], using the parameters $\Theta = \{\alpha_1, \ldots, \alpha_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K\}$ from the previous iteration, for each feature vector $\mathbf{x}_i$, $i = 1, \ldots, N$ and for each mixture component m, $m = 1, \ldots, K$ the posterior probability that $\mathbf{x}_i$ was generated from m-th component is calculated as:

$$P(m|\mathbf{x}_i) = \frac{\alpha_m \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_m, \mathbf{\Sigma}_m)}{\sum\limits_{k=1}^{K} \alpha_k \mathcal{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k)}. \tag{4}$$

The M-step [7], using the posterior probabilities $P(m|\mathbf{x}_i)$, computes new estimate of parameters $\mathbf{\Theta}$ as $(m = 1, \ldots, K)$:

$$\alpha_m = \frac{1}{N} \sum_{i=1}^{N} P(m|\mathbf{x}_i), \tag{5}$$

$$\boldsymbol{\mu}_m = \frac{\sum\limits_{i=1}^{N} P(m|\mathbf{x}_i)\mathbf{x}_i}{\sum\limits_{i=1}^{N} P(m|\mathbf{x}_i)}, \tag{6}$$

$$\mathbf{\Sigma}_m = \frac{\sum\limits_{i=1}^{N} P(m|\mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T(\mathbf{x}_i - \boldsymbol{\mu}_m)}{\sum\limits_{i=1}^{N} P(m|\mathbf{x}_i)}. \tag{7}$$

The E-step and M-step are applied alternately until a convergence criterion is met.

III. Four formulations of the EM using matrix algebra libraries

All formulations of the EM algorithm for GMMs discussed in this section store the training set in a matrix (two-dimensional array in the C++ language) $\mathbf{X} = [x_{i,j}]_{1 \le i \le N, 1 \le j \le d}$, where i-th row, denoted by $x_{i,*}$ stores the feature vector $\mathbf{x}_i$. Similarly, the posterior probabilities are stored in a matrix $\mathbf{P} = [p_{i,j}]_{1 \le i \le N, 1 \le j \le K}$, where $p_{i,j} = P(m|\mathbf{x}_i)$.

Algorithm 1 shows a high-level overview of the EM. The equations (3) and (4) indicate that in order to perform both the convergence check and the E-step we need to compute Gaussian probability density function values multiplied by the corresponding mixing proportions. An obvious optimization is to compute densities weighted by mixing proportions once, store them in a matrix $\mathbf{W} = [w_{i,j}]_{1 \le i \le N, 1 \le j \le K}$, where $w_{i,j} = \alpha_j * \mathcal{N}(x_{i,*}; \boldsymbol{\mu}_j, \mathbf{\Sigma}_j)$ and use them in subsequent E-Step and computation of log-likelihood.

The four variants of the EM discussed in the paper follow this pattern. After the computation of weighted densities $\mathbf{W}$ (line 3 of Algorithm 1), the log-likelihood using (3) is computed (line 4). If the algorithm is not terminated in line 6, then the posterior probability matrix $\mathbf{P}$ using weighted densities $\mathbf{W}$ is obtained by equation (4) (line 8). The computation of the log-likelihood L and the matrix $\mathbf{P}$ based on $\mathbf{W}$ are very straightforward. We have implemented them using Eigen C++ library, which generates an efficient vectorized code. The iterations of the EM algorithm are performed until either the algorithm converges or the maximal number of iterations M is reached.

Algorithm 1 The pseudocode of the EM algorithm

```
Require: $\mathbf{X}$, $\mathbf{\Theta}^0$, $M$, $\varepsilon$
 1: $\mathbf{\Theta} \leftarrow \mathbf{\Theta}^0$
 2: for $i \leftarrow 1$ to $M$ do
 3:   $\mathbf{W} \leftarrow$ WeightedDensities($\mathbf{X}$, $\mathbf{\Theta}$)
 4:   $L' \leftarrow L$, $L \leftarrow$ Loglikelihood($\mathbf{W}$)
 5:   if $i > 1$ and ConvergenceCheck($L, L'$) then
 6:     Terminate the algorithm
 7:   end if
 8:   $\mathbf{P} \leftarrow$ EStep($\mathbf{W}$)
 9:   $\mathbf{\Theta} \leftarrow$ MStep($\mathbf{X}, \mathbf{P}$)
10: end for
11: return $\mathbf{\Theta}$
```

The four variants of the EM algorithm differ in implementation of `WeightedDensities` and `MStep` functions.

A. Variant I: EM-L2

This variant uses BLAS Level 2 (matrix-vector) calls in implementation of `WeightedDensities` and `MStep`, hence its name EM-L2. The pseudocode of `WeightedDensities` function is shown in Algorithm 2. The function starts (lines 1–3) with the computation of invariants which do not depend on the feature vector. The inverse and determinant of covariance matrices are calculated (line 2) using the Cholesky decomposition [8]. Next, the loop (lines 4–10) iterating over all rows of $\mathbf{X}$ and $\mathbf{W}$ is executed. In this loop, for each mixture component j a squared Mahalanobis distance between the i-th row $x_{i,*}$ and mean vector $\boldsymbol{\mu}_j$ is calculated in lines 6–7. The computations in line 7 are done using `cblas_dsymv` and `cblas_dot` calls [4]. Next (line 8) weighted normal density is calculated.

Algorithm 3 shows the pseudocode of `MStep` function. The function calculates sums in equations (5)–(7) in two passes over rows of matrices $\mathbf{X}$ and $\mathbf{P}$. In the first pass (lines 4–8),

Algorithm 2 `WeightedDensities` function in EM-L2

```
Require: $\mathbf{X}$, $\mathbf{\Theta}$
 1: for $j \leftarrow 1$ to $K$ do
 2:   $\mathbf{S}_j \leftarrow \mathbf{\Sigma_j}^{-1}$, $b_j \leftarrow 1/\left((2\pi)^{d/2}\det(\mathbf{\Sigma_j})^{1/2}\right)$
 3: end for
 4: for $i \leftarrow 1$ to $N$ do
 5:   for $j \leftarrow 1$ to $K$ do
 6:     $\mathbf{y} \leftarrow x_{i,*} - \boldsymbol{\mu}_j$
 7:     $w_{i,j} \leftarrow -0.5\mathbf{y}S\mathbf{y}^T$
 8:     $w_{i,j} \leftarrow \alpha_j b_j \exp(w_{i,j})$
 9:   end for
10: end for
11: return $\mathbf{W} = [w_{i,j}]$
```

Algorithm 3 `MStep` function in EM-L2

```
Require: $\mathbf{X}$, $\mathbf{P}$
 1: for $j \leftarrow 1$ to $K$ do
 2:   $\mathbf{\Sigma}_j \leftarrow \mathbf{0}$, $\boldsymbol{\mu}_j \leftarrow \mathbf{0}$, $s_j \leftarrow 0$
 3: end for
 4: for $i \leftarrow 1$ to $N$ do
 5:   for $j \leftarrow 1$ to $K$ do
 6:     $s_j \leftarrow s_j + p_{i,j}$, $\boldsymbol{\mu}_j \leftarrow \boldsymbol{\mu}_j + p_{i,j}x_{i,*}$
 7:   end for
 8: end for
 9: for $j \leftarrow 1$ to $K$ do
10:   $\boldsymbol{\mu}_j \leftarrow \boldsymbol{\mu}_j/s_j$, $\alpha_j \leftarrow s_j/N$
11: end for
12: for $i \leftarrow 1$ to $N$ do
13:   for $j \leftarrow 1$ to $K$ do
14:     $\mathbf{y} \leftarrow x_{i,*} - \boldsymbol{\mu}_j$
15:     $\mathbf{\Sigma_j} \leftarrow \mathbf{\Sigma_j} + p_{i,j}\mathbf{y}^t\mathbf{y}$
16:   end for
17: end for,
18: for $j \leftarrow 1$ to $K$ do
19:   $\mathbf{\Sigma_j} \leftarrow \mathbf{\Sigma_j}/s_j$
20: end for
21: return $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K\}$
```

for each component j, the sum of posterior probabilities s_j and the sums in the numerators of (6) are accumulated. Next, the final values of mixing proportions α_j and mean vectors $\boldsymbol{\mu}_j$ are obtained (lines 9–11). In the second pass (lines 12–17), using the mean vectors computed in the first pass, for each component j, the sum in the numerator of (7) is obtained. The computation in line 15 is performed using `cblas_dsyr` call [4], which calculates rank-1 update of a symmetric matrix. Finally, in lines 18–20, the covariance matrices are obtained from accumulated numerators of (7).

B. Variant II: EM-L2-reordered

Our initial experiments with EM-L2 indicated the abysmal performance, where both dimension of the feature space d and the number of mixture components K are high. However, a simple interchange of loops in lines 12–17 of Algorithm 3, which places the loop iterating over the mixture components first was able to significantly improve the performance. We call this variant EM-L2-reordered. It used the same formulation of `WeightedDensities` function as EM-L2.

C. Variant III: EM-L3-blocking

This variant uses Level 3 BLAS operations [4], which usually have have $O(n^2)$ memory complexity and much larger $O(n^3)$ computational complexity, which allows for higher reuse of data and higher level of optimization [5]. Additionally it uses employs blocking (called also loop tiling [5]) to further optimize the most time-critical operations of `WeightedDensities` and `MStep` functions. We apply this technique to process the data in smaller blocks that are more likely to fit in the last level of cache memory. The `WeightedDensities` and `MStep` functions are shown in Algorithms 4 and 5, respectively. These functions require additional parameter, which is the number of blocks. This parameter is denoted by β in `WeightedDensities` function and by γ in the `MStep` function.

In the `WeightedDensities` function the computation of squared Mahalanobis distance is performed for blocks of rows of the data matrix $\mathbf{X}$. To simplify description, we assume that the number of feature vectors N is divisible without remainder by the number of blocks β. In such case the size of each block equals $B = N/\beta$. The outermost loop (line 5) iterates over blocks. It starts by the computation of the indices of the first (i_s) and the last (i_e) row in current l-th blocks. These must satisfy the condition $i_e - i_s + 1 = N/\beta$. The inner loop (lines 7–13) is performed for submatrix of $\mathbf{X}$ consisting of rows $i_s, i_{s+1}, \ldots, i_e$, which we denote as $\mathbf{X}_{i_s:i_e,*}$. All the matrices involved in the inner loop nest including the submatrices of $\mathbf{X}$ and $\mathbf{W}$ have B rows. Since the computations in lines 8–12 are repeated K times, this approach allows for much greater reuse of data in the cache memories.

The code in lines 7-13 computes the squared Mahalanobis distances and stores them in a block of the matrix $\mathbf{W}$. Using the Cholesky decomposition of Σ_j^{-1} the squared distance between a feature vector in i-th row of X and j-th mixture component can be written as:

$$(x_{i,*}-\boldsymbol{\mu}_j)\mathbf{\Sigma}_j^{-1}(x_{i,*}-\boldsymbol{\mu}_j)^T = (x_{i,*}-\boldsymbol{\mu}_j)\mathbf{L}_j\mathbf{L}_j^T(x_{i,*}-\boldsymbol{\mu}_j)^T = \left[(x_{i,*}-\boldsymbol{\mu}_j)\mathbf{L}_j\right]\left[(x_{i,*}-\boldsymbol{\mu}_j)\mathbf{L}_j\right]^T. \quad (8)$$

Lines 8–12 implement this computation efficiently using two temporary matrices. The $B \times d$ temporary matrix $\mathbf{Y}$ is obtained by subtracting $\boldsymbol{\mu}_j$ from each row of the block of $\mathbf{X}$. The temporary $B \times d$ matrix $\mathbf{Z}$, where i-th row is given by $z_{i,*} = (x_{i,*} - \boldsymbol{\mu}_j)\mathbf{L}_j$ is computed in line 7 using `cblas_dtrmm` (Level 3) BLAS call [4], which multiplies a general matrix by a triangular matrix.

The pseudocode of `MStep` function is shown in Algorithm 5. The less time-consuming ($O(NKd)$ computational complexity) computation of mean vectors $\boldsymbol{\mu}_j$ and posterior sums s_j (lines 4–11) is performed similarly to EM-L2 version shown

Algorithm 4 `WeightedDensities` function in EM-L3-blocking

```
Require: $\mathbf{X}$, $\mathbf{\Theta}$, $\beta$
 1: for $j \leftarrow 1$ to $K$ do
 2:     $\mathbf{S}_j \leftarrow \mathbf{\Sigma_j}^{-1}$, $\mathbf{L}_j \leftarrow \texttt{chol}(\mathbf{S}_j)$
 3:     $b_j \leftarrow 1/\left((2\pi)^{d/2}\det(\mathbf{\Sigma_m})^{1/2}\right)$
 4: end for
 5: for $l \leftarrow 1$ to $\beta$ do
 6:     $i_s, i_e \leftarrow \texttt{BlockIndices}(N, \beta, l)$
 7:     for $j \leftarrow 1$ to $K$ do
 8:         $\mathbf{Y} \leftarrow \mathbf{X}_{i_s:i_e,*} - \boldsymbol{\mu}_j$
 9:         $\mathbf{Z} \leftarrow \mathbf{Y}\mathbf{L}_j$
10:         for $i \leftarrow i_s$ to $i_e$ do
11:             $w_{i,j} \leftarrow \sum_{k=1}^{d} z^2_{i-i_s+1,k}$
12:             $w_{i,j} \leftarrow b_j\alpha_j * \exp(-0.5 * w_{i,j})$
13:         end for
14:     end for
15: end for
16: return $\mathbf{W} = [w_{i,j}]$
```

Algorithm 5 `MStep` function in EM-L3-blocking

```
Require: $\mathbf{X}$, $\mathbf{P}$, $\gamma$
 1: for $j \leftarrow 1$ to $K$ do
 2:     $\mathbf{\Sigma}_j \leftarrow \mathbf{0}$, $\boldsymbol{\mu}_j \leftarrow \mathbf{0}$, $s_j \leftarrow 0$
 3: end for
 4: for $j \leftarrow 1$ to $K$ do
 5:     for $i \leftarrow 1$ to $N$ do
 6:         $s_j \leftarrow s_j + p_{i,j}$, $\boldsymbol{\mu}_j \leftarrow \boldsymbol{\mu}_j + p_{i,j}x_{i,*}$
 7:     end for
 8: end for
 9: for $j \leftarrow 1$ to $K$ do
10:     $\boldsymbol{\mu}_j \leftarrow \boldsymbol{\mu}_j/s_j$, $\alpha_j \leftarrow s_j/N$
11: end for
12: for $l \leftarrow 1$ to $\gamma$ do
13:     $i_s, i_e \leftarrow \texttt{BlockIndices}(N, \gamma, l)$
14:     for $j \leftarrow 1$ to $K$ do
15:         for $i \leftarrow i_s$ to $i_e$ do
16:             $y_{i-i_s+1,*} = (x_{i,*} - \boldsymbol{\mu}_j) * \sqrt{p_{i,j}}$
17:         end for
18:         $\mathbf{\Sigma}_j \leftarrow \mathbf{\Sigma}_j + \mathbf{Y}\mathbf{Y}^T$
19:     end for
20: end for
21: for $j \leftarrow 1$ to $K$ do
22:     $\mathbf{\Sigma_j} \leftarrow \mathbf{\Sigma_j}/s_j$
23: end for
24: return $\mathbf{\Theta} = \{\alpha_1, \ldots, \alpha_K, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K\}$
```

in Algorithm 3. The only change is the reordering of loops starting in lines 4–5.

However, sums in numerators of (7) are obtained (lines 12–17) in a completely different way. Similarly to Algorithm 4, the data are processed in γ blocks, with size of block equal $C = N/\gamma$. After a calculation of temporary $C \times d$ matrix $\mathbf{Y}$ in lines 12–15, the numerator of (7) is updated by a single `cblas_dsyrk` level 3 BLAS call [4].

The `WeightedDensities` and `MStep` functions of the EM-L3-blocking variant were designed to delegate the most time-consuming code fragments with $O(Nd^2)$ complexity to Level 3 BLAS calls. `MStep` requires additional $O(N)$ square root calculations (line 14 of Algorithm 5) per single covariance matrix.

The method for choosing numbers of blocks β and γ remains to be described. Denote by L the total capacity of last level cache in bytes. In our implementation, to conserve memory, we store $\mathbf{X}$ using single precision floating point numbers. $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{W}$, $\mathbf{P}$ are stored using double precision numbers. Taking into consideration that a single precision number needs 4 bytes of storage and a double precision 8 bytes, the loop in lines 7–13 of Algorithm 4 needs $4dB$ bytes for storage of $\mathbf{X}_{i_s:i_e,*}$, $8dB$ bytes for $\mathbf{Y}$ and $\mathbf{Z}$ and $8KB$ bytes for the submatrix of $\mathbf{W}$. Assuming, that the total working set in the loop should be equal to L bytes, we have:

$$\beta = \frac{(20 * d + 8K)N}{L}. \tag{9}$$

After performing a similar analysis of the loop in lines 14–19 of Algorithm 5 we get:

$$\gamma = \frac{(12 * d + 8K)N}{L}. \tag{10}$$

D. Variant IV: EM-L3

This variant is a simplification of EM-L3-blocking, which does not perform loop tiling, i.e., sets the number of blocks $\beta = 1$ and $\gamma = 1$. We implemented this variant in order to assess the influence of blocking on the performance of the variant III.

IV. Parallelization for shared-memory systems

All the EM variants described in the previous section can be parallelized using data decomposition approach. We have designed parallel formulation of the algorithms and implemented them using the OpenMP standard [9] for shared-memory architectures.

An OpenMP application can be viewed as a group of cooperating threads. At the begin, only a master thread executes. When this thread encounters the `#pragma omp parallel` directive, the execution of the following block of code is performed by a team of threads. When a team of threads encounters the `#pragma omp for` directive, a succeeding `for` loop is parallelized by the team of threads. In this case each thread executes a subset of the loop iterations. In our approach we use static loop scheduling, where each of t threads is assigned approximately n/t iterations, when n is the total number of loop iterations.

In the parallelization of the `WeightedDensities` and `EStep` functions we use the fact that the rows of output matrices ($\mathbf{W}$ and $\mathbf{\Theta}$, respectively) can be computed using the corresponding rows of the input matrices ($\mathbf{X}$ and $\mathbf{W}$, respectively) without the knowledge about the remaining rows. We employ the data decomposition of the matrices $\mathbf{X}$, $\mathbf{W}$, $\mathbf{P}$ in which each thread is responsible for a block of consecutive rows. In case of EM-L3-blocking, where data are processed in blocks we divide further each of β or γ blocks into t sub-blocks.

Similar decomposition scheme is applied to the parallelization of `Loglikelihood` function, where each thread computes the local sum (3) using its assigned block of rows. Next, the local sums computed by the team of threads are added up giving the final log-likelihood. This is an example of the reduction operation which, for a single variable, can be easily carried-out using the OpenMP `reduction` clause.

The above decomposition scheme can be easily applied to `WeightedDensities` in Algorithm 2, by placing a `#pragma omp parallel for` directive before loops starting in lines 1, 5, and 12. For the `WeightedDensities` function shown in Algorithm 4 we use `#pragma omp for` before lines 1 and 10, and manually divide the rows of matrices $\mathbf{Y}$ and $\mathbf{Z}$ (lines 8–9) into t threads of the OpenMP team.

The parallelization of `MStep` functions is also based on data decomposition. Additionally, we have to tackle the problem of computing the s_j and the sums in numerators of (5), (6) and (7). We use a similar approach to that in the `Loglikelihood` function. Each OpenMP thread calculates local sums which are added up using the reduction operation. Since OpenMP 4.5 does not provide reduction operation for user-defined datatypes we have used the binary tree reduction algorithm [10].

V. Experimental results

All the results reported in this section were obtained using single compute nodes of the Tryton cluster installed in Centre of Informatics Tricity Academic Supercomputer and Network in Gdansk, Poland. A single node of the cluster is equipped with two 12-core Intel Xeon E5-2670 v3 (2.3 GHz) CPUs and 128 GiB of DDR4 RAM. The programs were compiled using the Intel C/C++ compiler (icpc) version 2021.7.1 and linked with the Intel MKL library version 2022.2.1, which provided the BLAS calls. We run the sequential version of EM-L3-blocking on a single core, assuming in (9) and (10) the last level cache size $L = 30 * 2^{20}$ bytes, according to manufacturer specification of the processor. We run parallel versions of all four algorithms using all 24 cores and assuming the last level cache size of multiplied by two: $L = 60 * 2^{20}$ bytes, because two processors were used in the calculations.

The experiments were performed on synthetic datasets obtained by the MixSim simulator proposed in [11]. The experiments were executed as follows. First we chose $d \in \{20, 40, 60, 80, 100\}$ and $K \in \{20, 40, 60, 80, 100\}$. For each of 25 combinations of K and d we generated a single dataset using the MixSim simulator. The number of feature vectors N was chosen to set the total size of the dataset as close of 512MiB ($512 * 2^{20}$ bytes) as possible. Thus, all the datasets were much larger as the total size of last level cache memory in a compute node. For each dataset we generated a single initial solution of the EM algorithm. This solution was used to initialize all the variants of the EM algorithm. Because all the variants started from the same solution, they converged after the same number of iterations. We obtained the average iteration time by dividing the total execution time measured using a system high-precision real time clock by the number of iterations. The shorter average EM iteration time indicated the higher performance of the algorithm.

The results indicated that EM-L3-blocking is the fastest of all parallel algorithms in all the experiments. Due to space limitations we have to omit the presentation of average iteration times in a table. These times ranged from 1.13 second (EM-L3-blocking, $d = 20$, $K = 20$) to 150 seconds (EM-L2, $d = 100$, $K = 100$). Using these measurements we have calculated the algorithmic speedup and parallel efficiency of the EM-L3-blocking. Figure 1 shows the algorithmic speedup of the EM-L3-blocking variant over the EM-L3. For a given dataset. an algorithmic speedup S of EM-L3-blocking over another variant A is defined as: $S = t_{\mathrm{A}}/t_{\mathrm{EM-L3-blocking}}$, where t_{A} and $t_{\mathrm{EM-L3-blocking}}$ denote average iteration times of EM variant A and EM-L3-blocking, respectively. The figure indicates that EM-L3-blocking is faster than EM-L3 for all datasets and its advantage is increased with decreased feature space dimension d.

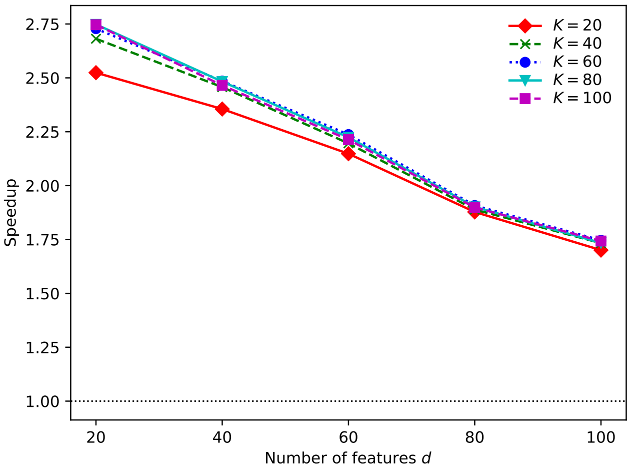


Fig. 1. Algorithmic speedup of the EM-L3-blocking variant over EM-L3 variant. The dotted horizontal line indicates equal speed of both algorithms.

Figure 2 shows the algorithmic speedup of the EM-L3-blocking over the faster of two variants (EM-L2 and EM-L2-reordered) using level 2 BLAS operations. The plots indicate that EM-L3-blocking is always faster and its advantage is increased with the increase of the dimension d.

We end the presentation of the results by showing the parallel efficiency of EM-L3-blocking with respect to its sequential version. A parallel efficiency (in percent) is defined as the ratio of measured parallel speedup to the ideal linear

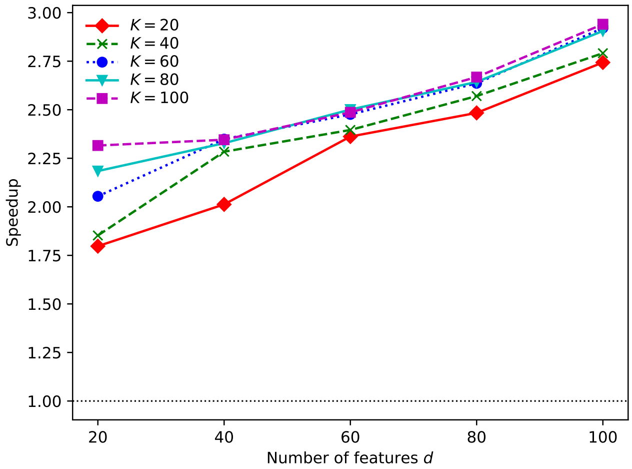


Fig. 2. Algorithmic speedup of the EM-L3-blocking variant over the fastest variant using L2 BLAS operations. The dotted horizontal line indicates equal speed of both algorithms.

speedup (equal to the number of the cores in a compute node). In turn, the parallel speedup is given by the ratio of the iteration time of the sequential version to iteration time of the parallel version of the algorithm. The plots indicate that the EM-L3-blocking variant scales very well with the efficiency higher then 83% in all the cases and higher than 90% where $K \geq 40$ and $d \geq 40$.

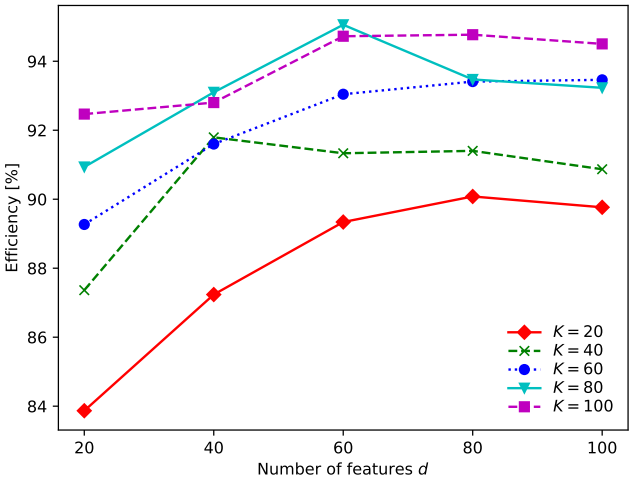


Fig. 3. Parallel efficiency of the EM-L3-blocking variant of the EM algorithm.

VI. Conclusions and future work

In the paper we described four variants of the EM algorithm. Two of them use level 2 BLAS operations while the remaining two are based on level 3 BLAS operations which can be implemented more efficiently on the contemporary hardware. We proposed a parallelization scheme for all the variants using OpenMP threads. The results of the study indicate that a combination of level 3 BLAS operations with the blocking for last level cache achieves the shortest runtime for all tested datasets. The resulting algorithm scales very well on a 24-core system.

An obvious extension our work would be a hybrid parallelization using many nodes of the cluster. In this method a parallel application could consists of processes communicating using a message-passing (e.g., MPI [12]) framework. One MPI process would be executed in each compute node of the cluster. Each process would execute in several OpenMP threads, with the number of threads equal to the number of cores in a compute node. A reduction operation in the `MStep` function would be performed hierarchically, first on the process level, then on the MPI application level. We have successfully applied this approach to multi-node parallelization of the well-known K-means algorithm [13].

References

[1] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Royal Stat. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[3] W. Kwedlo, “A parallel EM algorithm for Gaussian mixture models implemented on a NUMA system using OpenMP,” in *Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing PDP 2014*. IEEE CPS, 2014, pp. 292–298.

[4] L. Blackford, J. Demmel, J. Dongarra, I. Duff, S.Hammarling, G. Henry, M. Heroux, L. Kaufman, A. Lumsdaine, A. Petitet, R. Pozo, K. Remington, and R. Whaley, “An updated set of basic linear algebra subprograms (BLAS),” *ACM Transactions on Mathematical Software*, vol. 28, no. 2, pp. 135–151, 2002.

[5] M. Kowarschik and C. Weiß, *An overview of cache optimization techniques and cache-aware numerical algorithms*, ser. Lecture Notes in Computer Science, vol 2625. Springer, 2003, pp. 213–232.

[6] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: Wiley, 2008.

[7] R. A. Redner and H. F. Walker, “Mixture densities, maximum likelihood and the EM algorithm,” *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984. [Online]. Available: https://www.jstor.org/stable/2030064

[8] G. H. Golub and C. F. van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins, 1996.

[9] OpenMP Architecture Review Board, “OpenMP application program interface version 4.5,” http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf, 2015.

[10] E. Chan, M. Heimlich, A. Purkayastha, and R. van de Geijn, “Collective communication: theory, practice, and experience,” *Concurr. Comput. Pract. Exp.*, vol. 19, no. 13, pp. 1749–1783, 2007.

[11] R. Maitra and V. Melnykov, “Simulating data to study performance of finite mixture modeling and clustering algorithms,” *J. Comput. Graph. Stat.*, vol. 19, no. 2, pp. 354–376, 2010. [Online]. Available: https://doi.org/10.1198/jcgs.2009.08054

[12] Message Passing Interface Forum, “MPI: A Message-Passing Interface Standard Version 3.1,” 2015. [Online]. Available: http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

[13] W. Kwedlo and P. J. Czochański, “A hybrid MPI/OpenMP parallelization of K-means algorithms accelerated using the triangle inequality,” *IEEE Access*, vol. 7, pp. 42 280–42 297, 2019.

DOI: 10.15439/2023F5683
ISSN 2300-5963 ACSIS, Vol. 35

Performance Analysis of a 3D Elliptic Solver on Intel Xeon Computer System

Ivan Lirkov
0000-0002-5870-2588
Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
Acad. G. Bonchev, bl. 25A
1113 Sofia, Bulgaria
ivan.lirkov@iict.bas.bg
http://parallel.bas.bg/~ivan/

Marcin Paprzycki, Maria Ganzha
0000-0002-8069-2152, 0000-0001-7714-4844
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw, Poland,
paprzyck@ibspan.waw.pl, maria.ganzha@ibspan.waw.pl
http://www.ibspan.waw.pl/~paprzyck/,
http://pages.mini.pw.edu.pl/~ganzham/www/

***Abstract*—It was shown that block-circulant preconditioners, applied to a conjugate gradient method, used to solve structured sparse linear systems, arising from 2D or 3D elliptic problems, have very good numerical properties and a potential for good parallel efficiency. In this contribution, hybrid parallelization based on MPI and OpenMP standards is experimentally investigated. Specifically, the aim of this work is to analyze parallel performance of the implemented algorithms on a supercomputer consisting of Intel Xeon processors and Intel Xeon Phi coprocessors. While obtained results confirm the positive outlook of the proposed approach, important open issues are also identified.**

I. Introduction

IN THIS contribution, we are concerned with the numerical solution of linear boundary value problems of an elliptic type. After discretization, such problems are reduced to finding the solution of a linear systems in the form $A\mathbf{x} = \mathbf{b}$. In what follows, symmetric and positive definite problems are considered. Moreover, it is assumed that A is a large matrix. Obviously, the term “large” is relative, as what was large in the past, is no longer large. Therefore, it is assumed that the size of the linear system (matrix) is defined as large, in the context of capabilities of currently existing computers.

In practice, large problems of this class are often solved by iterative methods, such as the conjugate gradient (CG) method. At each step of such methods, a single product of A with a given vector $\mathbf{v}$ is needed. Therefore, to minimize number of arithmetic operations, the sparsity of the matrix A should be explored. On the other hand, exploration of sparsity may be in conflict with parallelization (for large number of processors and cores) of the iterative process.

Typically, the rate of convergence of CG methods depends on the condition number $\kappa(A)$ of the coefficient matrix A. Specifically, the smaller $\kappa(A)$ is, the faster the convergence. Unfortunately, for elliptic problems of second order, usually, $\kappa(A) = \mathcal{O}(n^2)$, where n is the number of mesh points in each coordinate direction. Hence, conditioning of the matrix grows rapidly (gets worse) with n. To accelerate the convergence of the iterative process, a preconditioner M is applied within the CG algorithm. The theory of the Preconditioned CG (PCG) methods says that M is a good preconditioner if it significantly reduces the condition number $\kappa(M^{-1}A)$ and, at the same time, if it allows one to efficiently compute the product $M^{-1}\mathbf{v}$, for a given vector $\mathbf{v}$. The third important aspect should be considered, namely, the need for efficient implementation of the PCG algorithm on modern parallel computer systems, see e.g. [1], [2]. Here, again, the question can be raised, what does it mean “modern” as the practical meaning of this term evolves. Establishing how the problem should be approached on a computer current to the time of conducted research is one of the issues that inspired this work.

II. The 3D elliptic problem

Let us now consider the following 3D elliptic problem:

$$-\frac{\partial}{\partial x_1}\left(k_1\frac{\partial u}{\partial x_1}\right)-\frac{\partial}{\partial x_2}\left(k_2\frac{\partial u}{\partial x_2}\right)-\frac{\partial}{\partial x_3}\left(k_3\frac{\partial u}{\partial x_3}\right)= f(x_1,x_2,x_3), \qquad \forall(x_1,x_2,x_3)\in\Omega, \tag{1}$$

$$0<\sigma_{\min}\le k_1(x_1,x_2,x_3), \quad k_2(x_1,x_2,x_3), \quad k_3(x_1,x_2,x_3)\le\sigma_{\max},$$

$$u(x_1,x_2,x_3)=0, \qquad \forall(x_1,x_2,x_3)\in\Gamma=\partial\Omega,$$

to be solved on the unit cube $[0,1]^3$. Let the domain be discretized by a uniform grid with n grid points in each coordinate direction.

A. Finite Difference Method

Let us consider the usual seven-point centered difference approximation for problem (1). This discretization leads to a system of linear algebraic equations

$$A\mathbf{x}=\mathbf{b}$$

where the vector of unknowns $\mathbf{x}$ has size n^3. If the grid points are ordered along the x_3 and x_2 directions first, the resulting matrix A admits a standard block-tridiagonal structure. Here, the diagonal blocks are block-tridiagonal matrices, while the off-diagonal blocks are diagonal matrices. Overall, the matrix A can be written in the following form

$$A = tridiag(A_{i,i-1}, A_{i,i}, A_{i,i+1}) \qquad i = 1, 2, \ldots, n,$$

where $A_{i,i}$ are block-tridiagonal matrices which corresponds to one x_1-plane. For details see [3], [4], [5], [6].

III. Circulant Block-Factorization Preconditioning

Let us recall that a circulant matrix C has the form $(C_{k,j}) = \left(c_{(j-k) \bmod m}\right)$, where m is the size of C. Moreover, for any given coefficients $(c_0, c_1, \ldots, c_{m-1})$, let us denote by $C = (c_0, c_1, \ldots, c_{m-1})$ the circulant matrix

$$\begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{m-1} \\ c_{m-1} & c_0 & c_1 & \cdots & c_{m-2} \\ \vdots & \vdots & \vdots & & \vdots \\ c_1 & c_2 & \cdots & c_{m-1} & c_0 \end{bmatrix}.$$

Any circulant matrix can be factorized as

$$C = F \Lambda F^*,$$

where Λ is a diagonal matrix containing the eigenvalues of C, F is the Fourier matrix

$$F = \frac{1}{\sqrt{m}} \left\{ e^{2\pi \frac{jk}{m} \mathbf{i}} \right\}_{0 \le j,k \le m-1}$$

and $F^* = \overline{F}^T$ denotes adjoint matrix of F. Here, $\mathbf{i}$ stands for the imaginary unit.

Let us now denote the general form of the CBF preconditioning matrix M, for the matrix A, by

$$M_{CBF} = tridiag(C_{i,i-1}, C_{i,i}, C_{i,i+1}) \qquad i = 1, 2, \ldots n$$

Here, $C_{i,j} = Block - Circulant(A_{i,j})$ is block-circulant approximation of the corresponding block $A_{i,j}$ [3], [4]. Note that the approach to defining block-circulant approximations can be interpreted as simultaneous averaging of the matrix coefficients, and changing the Dirichlet boundary conditions to the periodic ones.

Each PCG iteration consists of one solution of the linear system with the preconditioner. The CBF preconditioner can be written in the form

$$M_{CBF} = (I \otimes F \otimes F)(\Lambda \otimes I \otimes I)(I \otimes F^* \otimes F^*)$$

and the solution of the linear system with M_{CBF} requires one forward 2D Discrete Fourier Transform (DFT), solution of the tridiagonal linear systems, and one backward 2D DFT.

The details of the sequential and parallel realizations, of the CBF preconditioner, have been described in [5], [6], which should be consulted for the remaining details.

IV. Numerical Tests – Experimental Setup

Conducted experiments have been selected to illustrate the convergence rate, as well as the parallel performance of the developed algorithms for the 3D elliptic problems. Specifically, test problems, with variable coefficients in the form

$$\begin{aligned} \frac{\partial}{\partial x_1}\left[\left(1 + \frac{\epsilon}{2}\sin\left(2\pi\left(x_1 + x_3\right)\right)\right)\frac{\partial u}{\partial x_1}\right] &\quad + \qquad (2) \\ \frac{\partial}{\partial x_2}\left[\left(1 + \frac{\epsilon}{2}\sin\left(2\pi\left(x_1 + x_2\right)\right)\right)\frac{\partial u}{\partial x_2}\right] &\quad + \\ \frac{\partial}{\partial x_3}\left[\left(1 + \epsilon e^{x_1 + x_2 + x_3}\right)\frac{\partial u}{\partial x_3}\right] &\quad = \quad f\left(x_1, x_2, x_3\right) \end{aligned}$$

where $\epsilon \in [0, 1]$ is a parameter have been considered. It is well known that the circulant preconditioners are competitive with the incomplete LU factorization for moderately varying coefficients. This reflects the averaging of the coefficients, used in the block-circulant approximations.

The right hand side f, is chosen in such a way that the problem (2) has solution

$$u\left(x_1, x_2, x_3\right) = \sin 2\pi x_1 \sin 2\pi x_2 \sin 2\pi x_3.$$

All computations are done in double precision. The standard iteration stopping criterion is $||\mathbf{r}^{N_{it}}||_{M^{-1}} / ||\mathbf{r}^0||_{M^{-1}} < 10^{-6}$, where $\mathbf{r}^j$ stands for the residual at the jth iteration step of the preconditioned conjugate gradient method. The code has been implemented in C. For the implementation of the preconditioning, Fast Fourier Transform (FFT) was used, and functions fftw_init_threads, fftw_plan_with_nthreads, fftw_plan_many_dft, and fftw_execute from the FFTW (the Fastest Fourier Transform in the West) library were used. A hybrid parallel code, based on joint application of MPI and OpenMP-based parallelizations has been developed [7], [8], [9], [10], [11].

In this contribution, the parallel code has been tested on cluster computer system Avitohol, at the Advanced Computing and Data Centre of the Institute of Information and Communication Technologies of the Bulgarian Academy of Sciences. The Avitohol consists of HP Cluster Platform SL250S GEN8. It has 150 servers, and two 8-core Intel Xeon E5-2650 v2 8C processors and two Intel Xeon Phi 7120P coprocessors per node. Each processor runs at 2.6 GHz. Processors within each node share 64 GB of memory. Each Intel Xeon Phi has 61 cores, runs at 1.238 GHz, and has 16 GB of memory. Nodes are interconnected with a high-speed InfiniBand FDR network (see, also http://www.hpc.acad.bg/).

For the experiments, Intel C compiler has been used and the code was compiled using the following options: "-O3 -qopenmp -L$(MKLROOT)/lib/intel64 -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lmkl_rt -lm" for the processors, and "-O3 -qopenmp -mmic -L$(MKLROOT)/lib/mic -lmkl_intel_lp64 -lmkl_intel_thread -lmkl_core -lpthread -lmkl_rt -lm" for the coprocessors. Intel MPI was used to execute the code on the Avitohol computer system.

TABLE I
USED MEMORY, NUMBER OF ITERATIONS AND TIME (IN SECONDS) FOR THE EXECUTION OF THE PARALLEL ALGORITHM ON ONE NODE USING ONE MPI PROCESS AND VARYING THE NUMBER OF THREADS.

n	memory	N_{it}	Error	threads					
				1	2	4	8	16	32
120	707 Mb	57	7.7 E-3	8.53	7.68	4.80	4.13	3.81	7.42
240	3325 Mb	96	3.8 E-3	160.32	141.73	88.64	58.88	46.68	84.79
360	6631 Mb	132	2.6 E-3	771.61	732.17	423.83	276.96	218.15	372.30
480	15629 Mb	166	1.9 E-3	2425.11	2307.68	1302.72	832.37	671.93	1070.61
600	30458 Mb	200	1.5 E-3	6403.09	6336.52	3533.77	2155.12	1566.76	2598.90
720	52542 Mb	231	1.3 E-3	14192.50	13565.10	7753.98	4656.22	3346.86	5197.23

V. EXPERIMENTAL RESULTS AND ANALYSIS

The first series of experiments established the "baseline" performance. In Tables I and II results obtained using processors of a single node of Avitohol are presented. Table I shows the used memory, the number of iterations, the maximal error of the obtained solution, and the total execution time. The results have been obtained using only shared memory parallelism: i.e. there was one MPI process and up to 32 OpenMP threads.

The first observation concerns memory use. For n increasing from 120 to 720, memory consumption grows from 707 Mb to 52542 Mb. This means that the limit of size of the problem that can be solved on a single node has been reached. This was checked experimentally, and increasing n to 840 resulted in an "out of memory" error.

Second, let us note that for $n = 120$ there is no performance gain with the number of used threads. Clearly, problem is too small. However, for $n = 720$ speedup of order of 5 has been reached for 16 threads. This was also the "best result". Moving to 32 threads resulted in speedup decreasing to approximately 3. Interestingly, almost no performance improvement was observed when moving form 1 to 2 threads.

Approaching the performance from the"completely opposite" perspective, Table II shows the execution time when using only distributed memory parallelism; i.e. one OpenMP thread and up to 32 MPI processes. It should be noted that in the current (prototype) implementation, the algorithm works correctly only if the number of mesh points in each coordinate direction (n) is divisible by the number of processes. In the Table, the best execution time, for each size of the problem, is marked in bold.

It is easy to note that, for $n = 120, 240, \ldots 720$ the algorithm works much faster when using only MPI parallelism and utilizing multiple threads (in comparison to using OpenMP based multi-threading). For the largest case ($n = 720$), the solver that is using MPI and 30 threads is more than 2 times faster than using OpenMP and 16 threads (the fastest result from Table I). Overall, for the smallest problem ($n = 120$) the fastest execution is obtained when using 15 MPI processes. For bigger problems, on the other hand, the best results have been reached when using 30 processes.

TABLE II
EXECUTION TIME (IN SECONDS) FOR THE EXECUTION OF THE PARALLEL ALGORITHM ON ONE NODE USING ONE OPENMP THREAD.

n	processes			
	15	16	30	32
120	**1.68**		2.07	
240	26.02	24.50	**19.66**	
360	167.61		**83.46**	
480	532.95	589.49	**286.14**	306.16
600	1200.36		**709.44**	
720	2535.59	2385.89	**1834.67**	

The second series of experiments concerned use of individual processors. Specifically, Table III shows the execution time when using only processors from multiple nodes (from 2 to 8). Here, results for 1 OpenMP thread, as well as 16 and 32 threads are reported. Note that, results for $n = 960$ are also reported. This was possible due to the fact that the problem was "split" into at least two nodes and, therefore, it "fit in" (did not generate out of memory errors). Again, the best execution time, for each problem size, is marked in bold.

The algorithm runs the fastest when using 16 threads in just few cases, i.e. for $n = 120$ on 3 and 8 nodes, for $n = 240$ on 5 nodes, and for $n = 960$ on 2 nodes. In the remaining cases, the execution using one thread is the fastest. Considering the largest case ($n = 960$), for a single OpenMP thread, speedup of order 6 can be observed when moving from 2 to 8 nodes. Moreover, when comparing results with those reported in Table II, for $n = 720$, the best result on 8 nodes is more than 6 times faster.

In the next series of experiments, the performance of the co-processors has been evaluated [12]. Specifically, Tables IV and V present times collected on the Avitohol using only Intel Xeon Phi co-processors (processors have not been used for solving the computational problem). Here, Table IV shows the execution time on one co-processor for $n = 120, 240, 360$. Again, the best execution time is marked in bold.

Here, the positive effect of combining OpenMP and MPI based parallelism can be observed. For the largest problem that fit in the memory of the co-processor ($n = 320$), use of 4 OpenMP threads improved the performance by more than 4 times. This could be interpreted as a case of super-linear speedup. However, delving into this point is out of scope of this contribution.

Next, Table V presents execution times obtained when solving the problem on co-processors (only), but using up to 8 nodes. Note that, since the code is a prototype, case of 7 nodes had to be excluded. Here, again, it was possible, for

TABLE III
TIME (IN SECONDS) FOR THE EXECUTION OF THE PARALLEL ALGORITHM ON UP TO 8 NODES.

n	nodes											
	2		3		4		5		6		8	
	p_c	time	p_c	time	p_c	time	p_c	time	p_c	time	p_c	time
	1 OpenMP thread											
120	60	**1.21**	60	1.21	60	**0.52**	60	**0.58**	60	**0.47**	120	**0.43**
240	60	**10.81**	48	**10.49**	60	**8.92**	80	12.23	120	**5.80**	120	**4.68**
360	60	**44.53**	90	**43.79**	60	**37.17**	60	**33.91**	90	**35.54**	120	**32.14**
480	60	**138.40**	96	**115.16**	120	**104.98**	80	**102.40**	96	**94.00**	120	**80.36**
600	60	**336.15**	75	**264.43**	120	**213.35**	60	**229.31**	75	**193.97**	120	**149.82**
720	60	**766.71**	96	**865.68**	120	**389.33**	120	**453.28**	90	**405.31**	120	**292.54**
960	32	6552.93	96	**1985.21**	120	**1545.64**	160	**1597.70**	96	**1490.35**	96	**1047.86**
	16 OpenMP threads											
120	2	1.78	3	**1.08**	4	0.79	5	0.68	6	0.58	8	**0.43**
240	2	27.16	3	18.86	4	14.58	5	**11.85**	6	10.10	8	7.22
360	2	125.11	3	87.83	4	66.78	5	54.80	6	48.17	8	36.38
480	2	367.13	3	254.32	4	203.64	5	161.31	6	143.73	8	107.52
600	2	859.94	3	589.26	4	456.51	5	382.11	6	334.35	8	254.41
720	2	1643.03	3	1167.26	4	907.63	5	740.11	6	641.02	8	494.79
960	2	**6299.66**	3	3803.46	4	2957.54	5	7235.53	6	2081.50	8	1557.30
	32 OpenMP threads											
120	2	3.54	3	1.83	4	2.31	5	1.76	6	1.95	8	1.31
240	2	50.26	3	24.03	4	32.37	5	23.75	6	21.58	8	16.99
360	2	201.49	3	96.97	4	115.48	5	96.14	6	88.27	8	69.45
480	2	579.54	3	409.70	4	318.41	5	259.65	6	225.49	8	175.30
600	2	1371.68	3	948.50	4	722.42	5	585.15	6	501.13	8	385.27
720	2	2746.62	3	1893.08	4	1461.16	5	1181.41	6	1007.01	8	784.46
960	2	9611.23	3	5979.38	4	4526.58	5	9967.10	6	3053.44	8	2298.87

TABLE IV
EXECUTION TIME (IN SECONDS) FOR SOLVING OF 3D PROBLEM USING ONLY ONE CO-PROCESSOR OF THE AVITOHOL.

using one MPI process

n	threads				
	60	120	200	240	244
120	8.55	7.24	7.26	7.35	7.31
240	144.06	99.49	85.05	81.44	79.93
360	792.95	538.19	417.85	389.99	374.33

using p_m MPI processes and
q_m OpenMP threads

n	p_m	q_m	time	p_m	q_m	time
120	120	1	2.86	120	2	**2.25**
240	120	1	104.63	60	4	**18.29**
360	120	1	170.97	60	4	**73.90**

larger number of nodes, to solve the problem for $n = 960$. In the Table, the best execution time is marked in bold.

It can be seen that the algorithm runs faster using 2 threads for $n = 120$ and 4 threads for $n = 240, 360$. In this context, it should be recalled that the memory of one co-processor is only 16 GB. This memory limit is the reason that the code could have been run only for small size problems. In particular, for problems with $n = 480$ at least 2 co-processors were needed, while for $n = 960$ the code could have been executed starting from 12 co-processors.

In the final series of experiments, processors and co-processors have been jointly used. Specifically, Table VI shows the best execution times collected on the Avitohol using Intel Xeon processors working together with the Intel Xeon Phi co-processors. Here, the code was executed using: on processors — p_c MPI processes and every process runs q_c OpenMP threads; on co-processors — p_m MPI processes and every process runs q_m OpenMP threads. In each case, the optimal combination of the number of MPI processes and the number of threads has been used. These combinations have been established experimentally. The memory limitation resulted in not being able to run experiments, for $n = 960$, for less than 3 nodes. For the reasons explained above, there are no results for 7 nodes.

As can be seen, due to the, above stated, memory limitations on co-processors, the largest problem size $n = 960$ required at least 3 nodes to be solved. Considering problem of size $n = 720$, use of 8 nodes turned out to be ineffective, as solution time increased, as compared to the use of 6 nodes. For 6 nodes an almost perfect speedup (larger than 5), has been obtained. Interestingly, for all problem sizes, use of 8 nodes resulted in performance that was inferior to 6 nodes. We do not have an explanation of this fact, other than possibility that in this case operations not related to the solution of the problem had to run “somewhere” and their execution interfered with execution of the solver. For the largest problem, when comparing the performance obtained on 3 and on 6 nodes, a speedup of almost 6 was recorded. This shows that if ample resources are provided, the proposed approach behaves as expected and parallelizes well, when applying hybrid approach to algorithm parallelization.

To better visualize the relationship between execution times, they have been visualized also in Figure 1. Here, the execution time of the hybrid code, on up to 8 nodes for

TABLE V
TIME (IN SECONDS) FOR THE EXECUTION OF THE PARALLEL ALGORITHM ON UP TO 8 NODES USING ONLY CO-PROCESSORS.

n	nodes											
	1			2			3			4		
	p_m	q_m	time	p_m	q_m	time	p_m	q_m	time	p_m	q_m	time
120	120	2	2.25	4	120	8.69	120	6	4.96	120	8	4.03
240	60	4	18.29	4	240	104.58	240	6	58.24	240	8	42.14
360	60	4	73.90	4	240	481.40	6	244	360.95	8	240	291.30
480	2	240	1615.20	4	240	1443.04	6	244	1077.12	8	200	870.12
600				4	200	3373.44	120	12	1809.61	120	16	1265.60
720				4	240	6746.71	120	6	3554.11	120	16	2673.83

	nodes								
	5			6			8		
	p_m	q_m	time	p_m	q_m	time	p_m	q_m	time
120	120	10	3.44	120	24	3.78	15	120	3.96
240	240	10	37.85	240	12	31.69	240	15	27.22
360	10	200	253.00	12	240	225.09	15	200	187.94
480	10	200	741.41	12	200	659.75	16	200	524.95
600	120	10	1098.67	120	24	980.98	120	30	936.67
720	10	200	3472.91	120	24	2018.62	120	30	1694.33
960				120	24	6137.27	120	30	5234.65

TABLE VI
EXECUTION TIME (IN SECONDS) FOR SOLVING OF 3D PROBLEM USING OPTIMAL COMBINATIONS OF PROCESSORS AND CO-PROCESSORS OF THE AVITOHOL.

n	nodes														
	1					2					3				
	p_c	q_c	p_m	q_m	time	p_c	q_c	p_m	q_m	time	p_c	q_c	p_m	q_m	time
120	2	8	2	120	7.15	32	1	28	34	3.62	48	1	12	60	2.05
240	2	8	2	244	79.71	32	1	28	34	54.16	6	8	6	120	43.82
360	30	1	30	17	305.37	32	1	14	17	210.11	6	8	6	244	198.51
480	30	1	30	17	828.02	32	1	28	34	620.84	48	1	48	30	466.77
600	30	1	30	17	1919.31	64	1	56	17	1378.84	48	1	27	24	943.17
720	2	8	2	244	5273.09	64	1	56	17	2630.10	48	1	42	34	2110.96
960						4	4	1	244	7702.80	48	1	48	30	6662.39
	4					6					8				
	p_c	q_c	p_m	q_m	time	p_c	q_c	p_m	q_m	time	p_c	q_c	p_m	q_m	time
120	8	8	7	240	2.57	96	1	24	60	1.55	64	2	56	30	3.03
240	128	1	112	8	30.49	192	1	28	30	23.40	128	1	112	17	16.36
360	8	8	7	244	171.41	12	8	12	244	115.17	15	8	15	120	95.26
480	8	8	8	244	477.08	12	8	12	244	341.25	16	8	16	240	268.53
600	64	1	56	34	810.39	96	1	54	48	525.54	15	8	15	240	648.94
720	64	1	56	34	1643.04	96	1	84	34	1193.33	16	16	16	244	1859.61
960	64	1	56	34	5230.96	96	1	96	30	3630.39	16	16	16	244	3725.99

$n = 240, 480, 960$, for CPU-only, co-processor only and when both CPU and co-processor were used. For each of these cases, results are represented using the same color and marking.

It can be seen that use of multiple nodes allows one to solve large problems. Nevertheless, the speedup, resulting from adding nodes is not overwhelming.

VI. CONCLUDING REMARKS

The aim of this contribution was to experimentally explore relationship between (1) 3D elliptic solver, based on preconditioned conjugate gradient, (2) its hybrid parallelization consisting of applying shared memory OpenMP threads and distributed memory MPI approach, and (3) complex supercomputer architecture, based on nodes, processors and co-processors. It has been established that memory availability is one of the key issues that strongly influences parallel performance. In this context it is difficult to apply standard performance measures, such as speedup, since largest problems require large number of nodes to be executed. However, even if a code can be executed on different number of nodes, adding more nodes may not result in performance gains. There is a "sweet spot" where the problem is executed the fastest and adding more resources does not help. This also means that potential for standard speedup is somewhat limited.

All these observations can be linked to complex interplay between hybrid parallelization and hybrid computer architecture. This may be also a warning sign that potential gains from hybrid approaches may be outweighed by losses caused by complexity of interactions between various "components".

ACKNOWLEDGMENTS

We acknowledge the provided access to the e-infrastructure of the National Centre for High Performance and Distributed Computing. This work has been accomplished with the partial support by the Grant No BG05M2OP001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014-2020) and co-financed by the European Union through the European structural and Investment

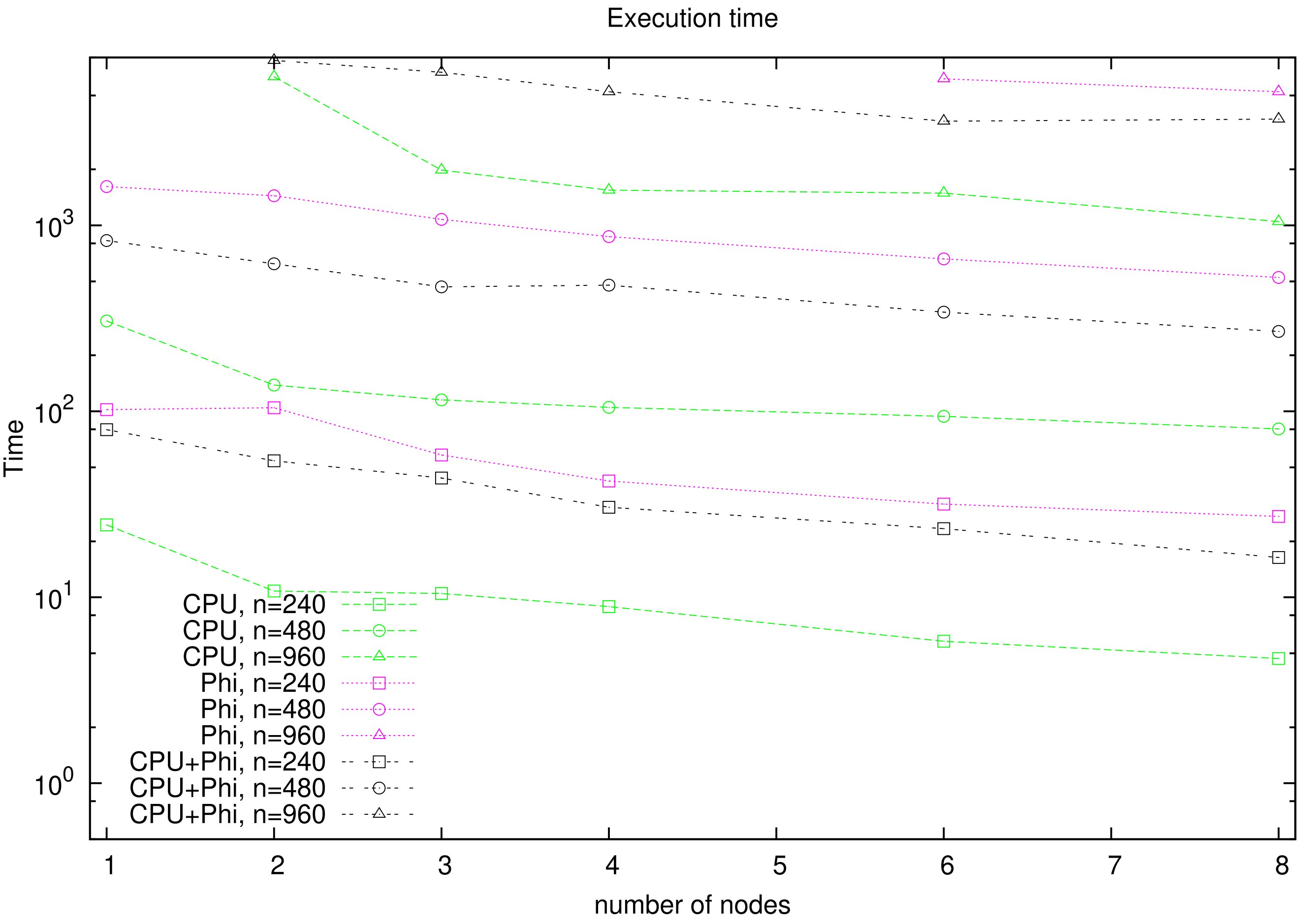


Fig. 1. Execution time for $n = 240, 480, 960$.

funds. Work presented here is a part of the collaborative grant between Polish Academy of Sciences and Bulgarian Academy of Sciences, IC-PL/01/2022-2023, “Practical aspects of scientific computing”.

References

[1] A. Axelsson and M. Neytcheva, *Supercomputers and numerical linear algebra*. Nijmegen: KUN, 1997.

[2] B. Bylina, J. Bylina, P. Stpiczyński, and D. Szałkowski, “Performance analysis of multicore and multinodal implementation of SpMV operation,” in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 2. IEEE, 2014, pp. 569–576.

[3] I. Lirkov and Y. Vutov, “Parallel performance of a 3D elliptic solver,” in *Proceedings of the International Multiconference on Computer Science and Information Technology*, M. Ganzha, M. Paprzycki, J. Wachowicz, and K. Węcel, Eds., vol. 1, 2006, pp. 579–590.

[4] ——, “The convergence rate and parallel performance of a 3D elliptic solver,” *System Science*, vol. 32, no. 4, pp. 73–81, 2007.

[5] I. Lirkov and S. Margenov, “Parallel complexity of conjugate gradient method with circulant block-factorization preconditioners for 3D elliptic problems,” in *Recent Advances in Numerical Methods and Applications*, O. Iliev, M. Kaschiev, B. Sendov, and P. Vassilevski, Eds. Singapore: World Scientific, 1999, pp. 482–490.

[6] I. Lirkov, S. Margenov, and M. Paprzycki, “Parallel performance of a 3d elliptic solver,” in *Numerical Analysis and Its Applications II*, ser. Lecture Notes in Computer Science, L. Vulkov, J. Waśniewski, and P. Yalamov, Eds., vol. 1988. Springer, 2001, pp. 535–543.

[7] R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan Kaufmann, 2000.

[8] B. Chapman, G. Jost, and R. Van Der Pas, *Using OpenMP: portable shared memory parallel programming*, ser. Scientific and engineering computation series. MIT press, 2008, vol. 10.

[9] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. The MIT Press, 2014.

[10] M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: The Complete Reference*, ser. Scientific and engineering computation series. Cambridge, Massachusetts: The MIT Press, 1997, second printing.

[11] D. Walker and J. Dongarra, “MPI: a standard Message Passing Interface,” *Supercomputer*, vol. 63, pp. 56–68, 1996.

[12] F. Kružel and K. Banaś, “Finite element numerical integration on Xeon Phi coprocessor,” in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 2. IEEE, 2014, pp. 603–612.

DOI: 10.15439/2023F895
ISSN 2300-5963 ACSIS, Vol. 35

Controllability for English-Ukrainian Machine Translation by Using Style Transfer Techniques

Daniil Maksymenko,
Nataliia Saichyshyna,
Oleksii Turuta
0000-0003-3223-5130
0000-0002-5145-0015
0000-0002-0970-8617
Kharkiv National University of Radio Electrics
Nauky Ave. 14,
61165 Kharkiv, Ukraine
Email: {daniil.maksymenko, nataliia.saichyshyna, oleksii.turuta}@nure.ua

Marcin Paprzycki
0000-0002-8069-2152
Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw
Email: marcin.paprzycki@ibspan.waw.pl

Mirela Alhasani
0000-0002-9110-394X
EPOKA University,
Tirana, Albania
Email: malhasani@epoka.edu.al

Maria Ganzha
0000-0001-7714-4844
Faculty of Mathematics and Information Science
Warsaw University of Technology
Koszykowa 75,
Warszawa, Poland
Email: m.ganzha@mini.pw.edu.pl

***Abstract*—While straightforward machine translation got significant improvements in the last 10 years with the arrival of encoder-decoder neural networks and transformers architecture, controllable machine translation still remains a difficult task, which requires lots of research. Existing methods like tagging provide very limited control over model results or they require to support multiple models at once, like domain fine-tuning approach.**

In this paper, we propose a method to control translation results style by transferring features from a set of texts with target structure and wording. Our solution consists of new modifications for the encoder-decoder networks, where we can add feature descriptors to each token embedding to decode input text into the translation with the proposed domain. In conducted experiments with English-Ukrainian translation and a set of 4 domains our proposed model gives more options to influence the result than some existing approaches to solve the controllability model.

***Index Terms*— Machine Translation, Controllability, NLG, Style Transfer.**

I. Introduction

LAST 10 years became very prolific for machine translation solutions as they finally achieved quality, which can be compared to a human processed results in many cases. The first significant step was the usage of recurrent encoder-decoder models, however, they were significantly overperformed by new attention-based transformer networks [1], which compare each part of the input sequence to multiple parts of input simultaneously. Their architecture allowed to reduce training time and made parallelization of the process easier compared to RNNs as they always need the n-1 state to compute the nth one. Pretrained models like mBART[2] gave the ability to capture low-resource languages much better than ever before.

However, these models usually do not give any methods to influence their results. We can get multiple options out of them or interpret their decisions by using SHAP [3] and similar frameworks, but we can't easily change the way these decisions are made. The easiest way to modify model behavior would be to finetune it using a small specialized corpus, but we need to support a whole model zoo for each separate domain or style to implement this approach, which can become expensive and difficult to manage.

The cheapest method to change at least some words in translation would be by applying usage dictionaries and finding another possible translation for a certain word or phrase, which could correspond better. This method will not allow us to modify translation according to a certain external context, we would just search for another option among popular ones.

Another approach proposes adding tags with style or other necessary features to influence the model. It should work well with both recurrent and bidirectional encoders as such tags are usually added at the beginning of the text, so their embeddings can further influence every step of translation generation [4]. This solution is not flexible enough as we can't encode all necessary features into a set of special markers and we do not know how the model will act if we change their order. Also, even a slight change in this marking would require us to completely retrain the model, which would be time-consuming and expensive.

Some new papers propose concatenating vectors with certain features like length, sentiment, officialness, or politeness to text embedding. However, we need to mark each translation with necessary feature values before training and any change in this marking or addition of a new feature would require a full-on retraining of such a model.

Modern generative models like GPT3 [5] and its next versions can conduct translation once they are given some examples even if they were not originally trained specifically for any other language than English. They still can generate some transliterations instead of translations or confuse related languages. Such models can be controlled via prompts, so we can try to add some statements like "make it more serious" to change the features of the translation. So in order to control such a model we need to pass some examples of the desired behavior and know the desired result to some extent.

In this research, we present an architecture based on transformer encoder-decoder models, which should conduct style transfer of a certain domain during translation by concatenating token embeddings with a text descriptor vector before decoding embeddings into the target translation. We provide explanations for this approach, comparisons with other available methods by both token and embedding metrics, and example translations generated by the proposed model.

II. *Datasets*

As our main aim was to increase machine translation models controllability by using transfer learning techniques we needed to gather some domain-specific and datasets, which contain texts of a certain style and structure. Styles should be distinct to enrich the model with knowledge in as many different types of texts as possible. We prepared 4 small specialized datasets with English-Ukrainian pairs with the following styles:

- general texts, which contain photo descriptions from the Multi30k dataset [6], which was translated by our team, and the results were presented in our previous paper;
- official texts, which consist of laws translations gathered from the Verkhovna Rada of Ukraine website [7];
- scientific texts use abstracts from Ukrainian papers gathered from Google Scholar service;
- programming documentation sentences, which were gathered from the official Vue framework website.

More information on these datasets is available in our previous papers. They describe mining, transformations, and cleaning for those specialized corpora.

We targeted sentence granularity for all text pairs, however, pairs in some domains contain one compound Ukrainian sentence and some simple English corresponding ones. Such behavior was spotted mostly in the scientific domain (abstracts from Ukrainian papers). We left them as they were written without splitting them. Other pairs with multiple sentences, which could be split into multiple ones, were transformed into 2 or more pairs of sentences.

Another large set of texts we used was gathered from multiple OPUS corpora [8]. They contain book reviews, subtitles, TED talk transcriptions, etc. They contain lots of messy data, which can even harm the model performance. As an example, there are a lot of texts with incorrect translations or translations, which can be understood only in a full original text context. Some texts contain some scrapping leftovers like tags or links. Many entries propose translations not in Ukrainian but in other similar languages, but it can be useful to learn some similar grammar or words, especially when the target language is a low-resource one. Another group of task is NER in low-resource languages [9].

There are around 60 million text pairs for the English-Ukrainian language set in OPUS, but we used only 2,247,528 texts. Cases like links or tags were cleaned using Python libraries, but we still needed to clean some incorrect translations. We could not check even this small chunk of OPUS manually, so we needed to automatize meaning comparisons of original and translated texts. Siamese XLM-R [10] for the semantic search was used to accomplish this as it supports both English and Ukrainian. We encoded each text into a vector with 512 elements and calculated the cosine similarity to its Ukrainian counterpart. The model was initialized from the **distiluse-base-multilingual-cased-v2** [11] checkpoint from the huggingface hub. This checkpoint was trained by using the Knowledge Distillation method. After checking multiple pairs and their cosine similarities we decided to use 0.4 as a threshold value. Pairs, which have lower similarity scores, are considered to be incorrect.

Some examples which have a value lower than 0.4 were examined. Most of them were really bad translations or missed some crucial part of the original text to understand why they should be translated this way. However, one corpus had lots of phraseologies, which were scored as errors by the XLM-R model as it tried to understand them in their literal sense. As an example, the phrase "murder will out" was translated as "правда вспливе". This is a correct translation, but the score is less than 0.4 as the model does not understand figurative sense. Such cases were not deleted from datasets and were used during model training as such cases can be really useful and hard to learn correctly.

The removal of texts with only links, tags, or empty lines and the removal of incorrect pairs reduced the dataset from 2,247,529 texts to 1,642,849 ones. Table 1 shows the number of pairs in each corpus and assigns a certain domain to each one except OPUS sets.

It is worth noting that OPUS corpora were used only in one step of the conducted experiment. Other specialized sets were used at each step of the experiment. Also, we split 25% of gathered specialized corpora into a test subset, which contains 9,625 text pairs with all 4 mentioned domains.

III. *Proposed Solution*

We propose a further development of the previously mentioned technique with the concatenation of the vector of target features to the input text tokens embeddings. As it was described before this method used vectors with a fixed set of features like length, sentiment, etc. We propose to use semantic descriptors of text, which can be combined with de-

TABLE I.
DATASETS OVERVIEW

Dataset name	Domain	Number of text pairs
Subset of OPUS corpora	-	1 642 849
Laws translations from Verkhovna Rada of Ukraine website	Official	4 000
Scientific articles abstracts from Google Scholar	Scientific	2 000
Vue framework documentation	Documen tation	2 500
Photo descriptions from Multi30k	General	30 000
Total	**-**	**1 681 349**

scriptors of domains to conduct style transfer from a certain domain to the input text translation.

Architecture would use a pretrained encoder and decoder and all the changes would happen after the creation of token embeddings and before their decoding into the target language translation. This way we can use an already established model as a foundation for our new solution instead of training the MT model from scratch, which would require millions of text pairs, lots of computational resources, and time.

Semantic descriptors of texts can be obtained from an external pretrained model trained for semantic search task. It would return a vector descriptor of the input text, which would allow us to place it into a certain embedding space and compare input to other texts the model has previously seen. Similarity to other texts can point the model towards the usage of certain words and styles as it can find suitable examples of target translations in this embedding space. It means that the descriptor does not carry any information about the style or features of translation output, but it shows the model text pairs which can be used as examples of necessary behavior as their descriptors are similar to the input descriptor. Fig. 1 shows 2D projections of semantic descriptors obtained from the semantic search model (in this case it was siamese BERT).

X and Y here are values generated by TSNE to reduce vectors from 384 elements to just 2, which we can easily visualize as a scatterplot. Texts from all 4 domains form multiple clusters and even subclusters based on their meaning, usage of words, sentiment, and tone. That is exactly what we need as further we can point the model toward one of these subclusters to gather translation features out of it and pass them to the decoder.

We need not only the input text descriptor to control the translation process but also domain descriptors. We will consider the average vector descriptor of all texts in a certain domain as a descriptor of this domain. In the next formula, we show the calculation of each element of the domain descriptor.

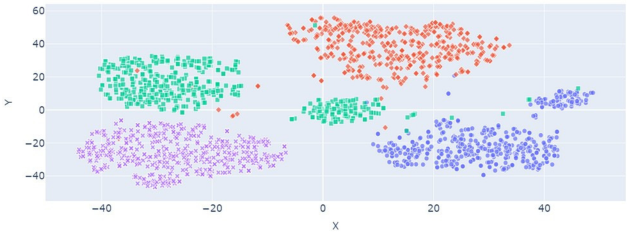


Fig 1. 2D projections of text semantic embeddings

$$V_{mean\,domain_j} = \frac{\sum_{i=0}^{N} V_{i,j}}{N} \quad (1)$$

So in this approach, we need to combine original text descriptor and this domain descriptor. We propose to do it by conducting a linear combination of the original text descriptor and vector of difference between text and domain. It is shown in the following formula, where is a transformation power coefficient, $V_{original}$ is an embedding vector of original input text, $V_{mean\,domain}$ is a mean embedding vector of texts in certain domain and descriptor is the final vector, which provides context on the way the text should be translated:

$$difference = V_{original} - V_{mean\,domain} \quad (2)$$

$$descriptor = V_{original} - \alpha * difference \quad (3)$$

Our hypothesis is that usage of semantic search embeddings should provide more control over the way the encoder-decoder model translates a text by showing it the desired domain and putting the text among ones with similar features. The transformation power $coefficient\,\alpha$ should indicate the power of changes, which we want to make and how much should descriptor be shifted into a certain embedding subspace.

However, the concatenation of the vector to each row of the token embeddings matrix will change its form. Let's say that matrix of token embeddings has the form NxM, where N is the number of tokens and M is the dimensionality of the embedding. Let's mark the size of the semantic descriptor as K. After concatenation our token embedding matrix will have the form Nx(M+K). Such a matrix would be impossible to pass into the original decoder as it still expects just an NxM matrix. We either need to create our own decoder and train it from scratch or create a dimension reduction layer, which would reduce the new concatenated matrix to its original size, so we can use a pretrained decoder. However, even the second option still requires some tuning as we add a new raw layer, which will make values in the matrix different from the initial ones. We would lose the connection between the encoder and decoder, so it has to be restored by tuning a new dimension reduction layer, so it would make new em-

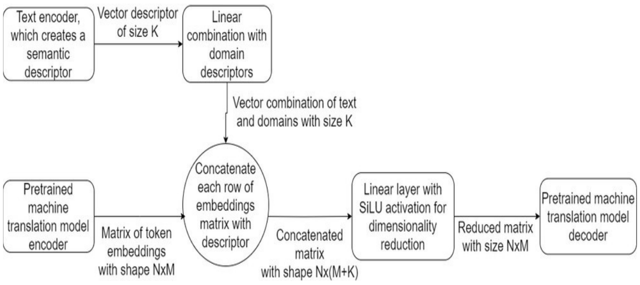

Text encoder, which creates a semantic descriptor
Vector descriptor of size K
Linear combination with domain descriptors
Vector combination of text and domains with size K
Pretrained machine translation model encoder
Matrix of token embeddings with shape NxM
Concatenate each row of embeddings matrix with descriptor
Concatenated matrix with shape Nx(M+K)
Linear layer with SiLU activation for dimensionality reduction
Reduced matrix with size NxM
Pretrained machine translation model decoder

Fig 2. Diagram of proposed machine translation model

beddings closer to the original ones and incorporate new knowledge obtained from the semantic search model.

Fig. 2 shows the architecture of a proposed machine translation model with style transfer abilities.

Currently we implemented dimension reduction as a linear layer with the number of output features equal to M. We used SiLU activation with batch normalization after the linear unit. We see the usage of Variable Selection Networks as another good option for this task, which we want to explore in our further research. This layer proved its effectiveness for classification tasks and time series feature selection, so we would like to see if it preserves such quality in the case of embedding dimensionality reduction to pass only the most significant values to the decoder for each token.

We want to use transformed semantic search embeddings of texts (combined with a certain domain embedding) concatenated to each token embedding as an additional mapping to give the decoder a hint of the necessary translation style, words domain, and tone, which we want to get. These vectors will not carry the style features themselves and they will not be hardcoded in there with a certain allowed range like in other similar approaches. They should be generated by an external model and linearly combined with a domain descriptor vector to shift features into the necessary cluster and make the overall vector closer to the texts with the desired style in the feature space. Semantic embeddings should only point toward texts with a translation style similar to the one we would like to achieve.

This way we can use any pretrained encoder-decoder machine translation model as a foundation for this architecture. Then we need to choose an external model to obtain sentence embeddings for texts and add a concatenation step for each row of the token embedding matrix to add values from sentence embedding at the end of each row. The final modification step would be to add a dimensionality reduction layer, which would restore the original dimension of token matrix rows, so we can use the original pretrained decoder. This process is shown in Fig.3 with an example where we have 512 feature embeddings for tokens and 384 number sentence embeddings.

The modified model will still need some fit as a new layer will not be trained at all, which would cause wrong translations due to the decoder getting previously unseen values. The only change for the train and validation datasets would be the need to calculate sentence embeddings.

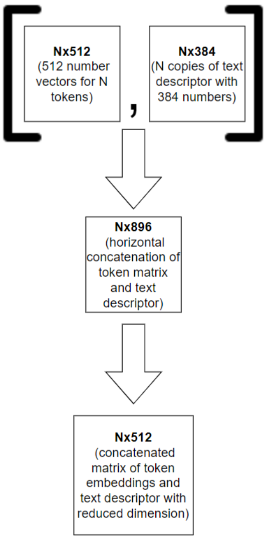

Nx512
(512 number vectors for N tokens)
,
Nx384
(N copies of text descriptor with 384 numbers)
Nx896
(horizontal concatenation of token matrix and text descriptor)
Nx512
(concatenated matrix of token embeddings and text descriptor with reduced dimension)

Fig 3. Semantic search embeddings injection process

The perfect case for this architecture would be to transfer style from a single provided example, but we should check this hypothesis. The primary use case would still be to combine text and domain descriptors to modify translation generation.

IV. *Metrics*

We use both token and embedding metrics in this research to measure the performance of obtained solutions. BLEU [12] was chosen as a default machine translation token metric and METEOR [13] was chosen as it works better with morphologically rich target languages due to the usage of stemming and synonyms dictionaries during scoring.

As for the embedding metrics we decided to use BERT Score [14] as it proposes a method to measure text generation quality by measuring semantic similarities of token embeddings obtained from the BERT model. So this way we would be able to compare texts by both their structure using token metrics and meanings by embeddings.

Generated examples of controllable translation [15] should be scored as well, however, we do not have references for all possible modifications of final translations, so mentioned token and embedding metrics will not have any chance to measure the quality of results as they need benchmarks in the target language. We will use a siamese XLM-R trained for multilingual semantic search to measure cosine similarity of original English text to each new generated Ukrainian translation. We will use a siamese XLM-R trained for multilingual semantic search to measure cosine similarity of original English text to each new generated Ukrainian translation. Model will be initialized from clip-ViT-B-32-

multilingual-v1 [11] checkpoint, because it was trained to replicate image domain embeddings, which should be useful for model, so it can gather knowledge not only from text information but also from pictures.

V. *Model Zoo*

Our main model for experiments will be MarianMT [16] pretrained for English-Ukrainian translation by Helsinki University on OPUS corpora. We used implementation from huggingface transformers library, which uses the BART interface. The model has 6 layers bidirectional encoder, like in BERT models, and 6 layers autoregressive decoder, like in GPT models. In this research we use this model with the following modifications to achieve controllability:

1. MarianMT fine-tuned with a small specialized corpus to capture its style, structure, and common words. We create a separate version of the model for each domain;

2. MarianMT tuned with all gathered specialized corpora to check how cross-domain knowledge can help the model learn the language better and if it still would be able to distinguish styles;

3. MarianMT with the addition of a special token-marker of necessary style at the beginning of the input text without fine-tuning (for example: "[official] Ukraine is a sovereign and independent, democratic, social, law-based state");

4. MarianMT with the addition of a special token-marker of necessary style at the beginning of the input text tuned on full specialized corpora to use all the advantages of bidirectional encoding and autoregressive decoding to better distinguish provided domains;

5. MarianMT modified with our proposed solution (concatenation of text-descriptor on each token embedding and dimensionality reduction for the obtained matrix to pass it into the original decoder). As it was mentioned before we would need to train the dimensionality reduction layer to restore the connection between the original encoder and decoder, so that is where we are going to use cleaned OPUS corpora. We will train this model using both our specialized datasets and OPUS data.

So we train variants 1, 2, and 4 only on 4 small specialized datasets. Version 3 will not be fine-tuned at all and version 5 gets trained with both OPUS and our datasets, as it needs to teach a new layer from scratch and learn how to use text descriptors for domain adaptation. We train each model with a fixed budget of 36 hours on Nvidia T4 GPU and then compare them by whole test dataset results and on separate domains in it to measure the controllability of obtained solutions.

Text descriptors will be gathered from siamese BERT for semantic search initialized from all-MiniLM-L6-v2 [17] checkpoint trained by the sentence-transformers team. It returns vectors with 384 elements, while MarianMT encodes each token in a vector with 512 elements. So after the concatenation of the token and text descriptor, we will have vectors with 896 elements. It means that the dimensionality reduction layer should reduce the size from 896 back to 512 elements, so we can pass the results to the original, pretrained decoder.

VI. *Experiments. Comparing models on full test dataset*

First of all we will train separate models for each domain and one, which would receive all gathered, specialized corpora. All these models should be scored on the full version of the test dataset. Table 2 shows the results of the training.

The best score on full test dataset was achieved with the model, which got all specialized corpora, which is expected as this model saw every style features. The second place is occupied by the model, which was trained with official texts (laws translation), which can be explained by the difficulty of this specific domain. Sentences there contain a lot of specific, uncommon words and phrases or even common words with new senses. The structure is strict and differs significantly from other styles.

We expected higher results from the scientific domain as it can also provide some unique knowledge to a model, which would not be possible to retrieve from other groups of texts. However, it gets lower scores in all 3 metrics than the official domain model. As we mentioned earlier this domain contains some difficult cases, where English text consists of multiple simple sentences and its Ukrainian counterpart has just one big compound sentence. It can confuse the model and also makes it difficult to calculate token metrics correctly.

Table II.
Separate models for each domain

№	Model variant	BLEU	METE OR	BERT F1 Score
1	Original OPUS MT MarianMT	11.20	0.2807	0.8115
2	MarianMT tuned with general texts	12.70	0.3034	0.8380
3	MarianMT tuned with official texts	25.34	0.3861	0.8630
4	MarianMT tuned with scientific texts	18.80	0.3347	0.8448
5	**MarianMT tuned with all special corpora**	**34.16**	**0.4754**	**0.8983**

Original OPUS MT MarianMT and version tuned with general texts have the lowest and quite similar scores. Photo descriptions from Multi30k did not give any new insights as they mostly consist of simple sentences with just a subject, an action, and sometimes a brief description of the environment. It does not differ from the original OPUS corpora, which contain lots of general domain texts too.

The next step is to check how a special token marker would affect the model performance. Table 3 shows scores for the model, which gets such markers without any fine-tuning, and for the fine-tuned version, where each text is marked with a style tag.

Special token without any fine-tuning does not give any significant boost to the original model scores and even makes the BERT Score worse. However, a tuned version of MarianMT, which learned how to use such tokens, overperforms even the previous best result obtained with MarianMT tuned with all specialized corpora without any tags. As we said before this special tag can influence every other token embedding and the decoder behavior due to their bidirectional and autoregressive natures respectively. The model distinguishes styles better and still learns information from all of them simultaneously.

The final step would be to teach our proposed model. We were able to teach it for 5 epochs with our set budget. Fig. 4 shows the metrics plot for each epoch and compares them to the previous best solution (MarianMT tuned with all specialized corpora with style tags).

We scaled BLEU to the 0-1 range in this plot to place all plots in the same subspace. The model completely loses the ability to translate after modification of the addition of concatenation with semantic descriptor and dimensionality reduction layer. The new embedding matrix has the same shape as the original one, but the values are not matched with what the decoder was getting earlier. Token metrics show it really well as they become almost equal to 0. However, BERT Score still gives average scores, which can be a huge misdirect if we do not calculate token metrics simultaneously. The model just generates random Ukrainian texts without any connection to the original English one. For example, our input is “Laws have been around for over 4000 years”. Generated translation without any fine-tuning was “Це означає, що ми маємо право вирішувати, що робити, а що ні.”. The model completely lost the ability to translate and embedding metrics were not able to capture it properly. The results of this experiment proved that BERT Score can not be used as a single metric to measure translation quality as it should be accompanied by some classic token approaches. It can be explained by the usage of multilingual BERT as an encoding model, as we compared it to other models for the Ukrainian language in previous papers. It completely loses to ones like XLM-R, so the default implementation of the BERT Score does not work well as a benchmark for Ukrainian language text generation tasks.

Table III.
Special token-marker models

№	Model variant	BLEU	METEOR	BERT F1 Score
1	Original OPUS MT MarianMT	11.20	0.2807	0.8115
2	MarianMT tuned with all special corpora	34.16	0.4754	0.8983
3	MarianMT with special token without tuning	11.72	0.3086	0.8085
4	**MarianMT with special token tuned**	**37.08**	**0.4923**	**0.9019**

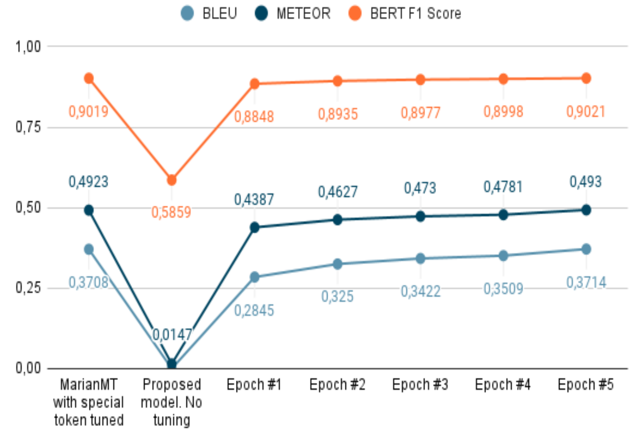


Fig 4. Training plot of proposed model

5 epochs of training with both OPUS corpora and our specialized sets were enough to restore the connection between the encoder and decoder of MarianMT and incorporate new knowledge obtained from semantic embedding space. Our model was able to overcome the previous best results scored with MarianMT tuned on all specialized corpora with special tokens. It achieved BLEU equal to 37.14, METEOR 0.4930, and BERT F1 Score 0.9021 on the full test dataset.

This first part of the experiment proves that our model can generate translations on the same level as some established controllable translation solutions. Now we need to check models on different domains included in the test set to check how all models distinguish different styles and to check if our model is able to beat separate specialized models for each domain.

VII. *Experiments. Measuring controllability*

We measured each metric for 3 domains individually for each model to compare their performance and to understand how well the proposed model distinguishes domains. The desired result for our proposed model would be to perform on par with specialized models for each domain or at least get a close score. We will start with official texts. Scores can be found in Table 4.

The proposed model gives better results for all 3 metrics in comparison to all other models and most importantly it overcomes model trained only for the official style translations. It surpasses 50 by BLEU, which indicates that it is capable to provide fluent law translations. As it was said a few times before this style contains lots of difficult cases like uncommon words or strict structure of the sentence. Metrics show that model with provided semantic descriptors was able to capture these cases well enough.

Table IV.
Comparison by official texts domain

Model variant	BLEU	METE OR	BERT F1 Score
MarianMT tuned with official texts	49.48	0.6044	0.9247
MarianMT tuned with all special corpora	48.60	0.5987	0.9239
Original OPUS MT MarianMT	08.06	0.2444	0.7778
MarianMT with special token without tuning	9.10	0.2764	0.7862
MarianMT with special token tuned	51.93	0.6141	0.9285
Modified MarianMT with semantic descriptors	**53.25**	**0.6473**	**0.9303**

Measurements on general texts retain high quality too, as the proposed model still gets BLEU higher than 50. It overcomes all other models here and gives a significant boost to the translation quality. Also, it is interesting to see how a special token without tuning made results only worse for the initial model here as it was already trained to deal with general texts and here it gets an unknown entity, which only creates more errors in comparison to the target translation. Both models trained on all specialized corpora lost to the model trained only on general texts. Even style marker did not help the model distinguish other styles from image descriptions well enough to beat the specific model.

The results for general texts are represented in Table 5 (in this case image descriptions from Multi30k).

The last ones are abstracts from scientific articles. Results are represented in Table 6.

The proposed model works better than other ones for this domain too, however, the translation quality is still low. Such BLEU indicates that it still makes significant errors. METEOR and BERT Score show that probably the model still tries to replace original constructions with similar, synonymous ones. Such a decline in performance was probably provoked by the mentioned difference in the structure of English and Ukrainian counterparts. Also, it is interesting how the model tuned with all texts overcomes the one with abstracts only, which can be explained by a high diversity of abstracts in terms of their topics. Additional texts provide the model with more knowledge of some less-represented subdomains. It can be seen in Fig. 1, where a subset of scientific texts divides into 2 categories. We clustered these texts additionally and obtained 3 large clusters, which mainly can be described as articles about laws in spheres of economics and education, mechanics articles, and ones about biology and chemistry. The number of clusters was found via the silhouette method. Fig. 5 shows 2D projections of sentence embeddings of scientific texts clustered into 3 categories (where blues points are laws articles, green ones are about mechanics, and red ones are about chemistry and biology). So laws domain should help with the first

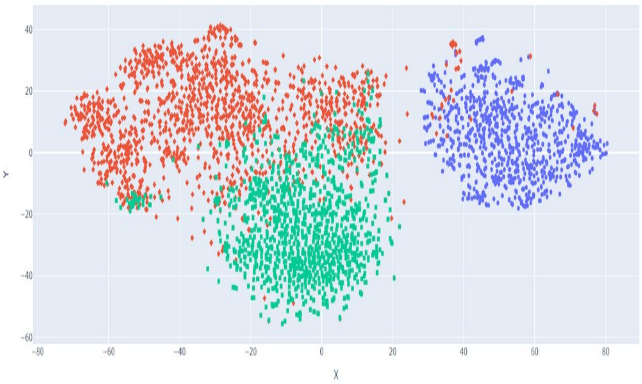


Fig 5. Scientific articles clusters

Table V.
Comparison by general texts domain

Model variant	BLEU	METE OR	BERT F1 Score
MarianMT tuned with general texts	42.40	0.4083	0.9181
MarianMT tuned with all special corpora	40.06	0.3948	0.9128
Original OPUS MT MarianMT	22.90	0.3264	0.8599
MarianMT with special token without tuning	22.022 3	0.3730	0.8428
MarianMT with special token tuned	40.89	0.4029	0.9164
Modified MarianMT with semantic descriptors	**53.46**	**0.5301**	**0.9264**

Table VI.
Comparison by scientific texts domain

Model variant	BLEU	METE OR	BERT F1 Score
MarianMT tuned with scientific texts	21.93	0.4127	0.8495
MarianMT tuned with all special corpora	23.42	0.4291	0.8548
Original OPUS MT MarianMT	10.94	0.2710	0.7956
MarianMT with special token without tuning	10.89	0.2736	0.7952
MarianMT with special token tuned	25.22	0.4523	0.8648
Modified MarianMT with semantic descriptors	**26.64**	**0.4686**	**0.8618**

TABLE VII.
EXAMPLES OF CONTROLLABLE TRANSLATION

№	Original	Translation	Modifications of text descriptor	Cosine similarity score (original to translation)
1	acquire ownership of intellectual property rights	набуття права інтелектуальної власності	Absent	0.9714
	This translation is close to the original meaning, but does not fully stick to the law language style. The issue here is that both “ownership” and “rights” can be translated as “власність” here, which would be a more correct translation. “право” and “власність” can be used as a more general style options.			
2	acquire ownership of intellectual property rights	набути право власності на інтелектуальну власність	Linear combination with official domain (=3.5)	0.9748
	Here we get “ownership” translated as “право власності”, which would be a more correct translation of this sentence according to laws terms. So model correctly used one word to translate two English ones in the same sentence and preserved the official style.			
3	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почала просити студентів скласти кілька перекладів одного вірша для презентації класу.	Absent	0.9810
	Just a correct translation with general style words used.			
4	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почав просити самих студентів скомпілювати кілька перекладів одного вірша для презентації класу.	Linear combination with a subset of a casual domain where the subject is male (=4.5)	0.9849
	Same style, but model captured the change of subject gender in provided examples and changed words forms (like “почав” instead of “почала”).			
5	I began asking the students themselves to compile multiple translations of a single poem for class presentation.	Я почав вимагати від студентів складання декількох перекладів єдиного вірша для презентації класу.	Previous modification + official domain with =3.5	0.9854
	Official style made the request “began asking” sound more as a requirement (“почав просити” became “почав вимагати”). English-like word “скомпілювати” got replaced by Ukrainian original word “складання” and the form became closer to English passive voice.			
6	In case you broke something you must pay for this	Якщо ти щось зламав, ти повинен заплатити за це	Absent	0.9812
	Just a normal translation with correct meaning			
7	In case you broke something you must pay for this	Якщо ви розбили що-небудь, ви повинні відповісти	Linear combination with official domain (=7)	0.9683
	Combination with official domain, which mostly consists of juridical documents and laws, made “pay for this” translation sound more like “carry responsibility” instead of paying money, which could be used one of more interpretations			
8	why don’t you come sit down with me?	чому б тобі не присісти зі мною?	Absent	0.9766
	Straightforward, correct translation			
9	why don’t you come sit down with me?	чому ви не приєднаєтеся до мене?	Official domain with coefficient 5.5	0.9137
	“you” gets translated as a more formal address and the sentence gets interpreted as “why don’t you join me”, which could be one of the possible translations depending on a larger context			
10	Do you want to hear a dirty joke? Ok. A white horse fell in the mud.	Ви хочете почути брудний жарт? Гаразд. Білий кінь впав у грязюку.	Absent	0.9692
	-			
11	Do you want to hear a dirty joke? Ok. A white horse fell in the mud.	Хочете почути грязну анекдоту? Гаразд. У грязюку впав білий кінь.	Old literature domain with coefficient value 4.5	0.9720
	By combining the source text with old literature we change the word “жарт” into “анекдота”, which would be an outdated way to translate “joke”. This word can still be used, but more as a joke genre name.			
12	Excuse me. Do you know the way to the zoo?	Вибачте, ви знаєте шлях до зоопарку?	Absent	0.9715
	This translation is correct and would be understood by a native Ukrainian speaker, but it copies the structure of the English source instead of adapting it.			
13	Excuse me. Do you know the way to the zoo?	Вибачте, ви знаєте, як пройти до зоопарку?	Casualness domain with coefficient 5.5	0.9702
	Here we get a more correct adaptation of “Do you know the way to the zoo” phrase, which would be a more common way to build this phrase in Ukrainian.			

cluster and technical documentation can provide more insight into terms from the second cluster, which can also be seen in Fig. 1, where a subset of scientific articles gets placed right between laws and documentation.

These measurements proved that our proposed model is able to translate 3 different domains with high quality and it distinguishes their features well by using information obtained from semantic descriptors. So these vectors can really help the model find enough examples of necessary translations among learned examples and they can be used to control translation style.

VIII. *Controllability examples*

We created 4 domain descriptors to test the proposed model. Each one of them was calculated as a mean embedding vector of texts corresponding to each domain. We encoded only English input texts, so the model searches for pairs close to the descriptor vector and uses their features to decode embeddings with the proposed style.

- Casual domain was calculated from 1000 image descriptions from the Multi30k dataset;
- Official domain was calculated from 1000 laws sentences;
- Instruction domain was calculated from 1000 documentation sentences;
- Old literature domain was calculated from 1000 sentences gathered from English literature from Project Gutenberg.

So now we have 4 domain descriptors with 384 elements each. We conducted some experiments on controllability to find optimal values of the transformation coefficient . Values below 3.5 usually do not change output text at all or change it slightly (as an example the only change can be the form of a single word). However, values lower than 3.5 can be used when we use multiple domains at once. Values higher than 7 shift descriptor values too much, so most of them become more than 1 or less than -1. It breaks the decoding process, so we get just a single word repeated as many times as the maximum number of output tokens allows or we just get some random symbols.

We show some examples of controllable translation in Table 7.

The model still can make some significant errors during style transfer. For example, we caught some errors with high coefficient values for the official domain. If we set it to 6.5 or higher it transforms some texts too much and literally changes their meaning. The input text was “Excuse me. Do you know the way to the zoo?”. Translation with official domain and coefficient equal to 7.0 was “Вибачте, чи знаєте ви шляхи до участі у виборчому окрузі”. In our opinion, it could be solved by using just one example of the desired style, so transfer could happen without setting of transformation coefficient.

We tested mentioned style transfer without transformation coefficient or creation of domain descriptors. We just pass another text as an example of the desired style and create its descriptor. It is then passed to the model instead of the input translation text descriptor. However, currently, the model does not make any changes based on just one example. It translates the text as if nothing was passed at all. In our opinion model needs more tuning to start working in a one-shot learning mode and transfer style from just one example instead of a whole set.

So the model can be used for controllable machine translation task but needs some precalculated domain descriptors to transfer the style of certain text set.

IX. *Conclusion*

In this research we proposed a solution to increase the controllability of machine translation models by using style transfer. We proposed a modified encoder-decoder architecture, which concatenates text semantic descriptor to each token embedding before decoding it into the target translation. This way we can point the model towards texts with necessary features, which we want to transfer into the final translation. The proposed solution was compared to established approaches like domain fine-tuning and the addition of a style marker by token and embedding metrics. Models were compared on a full multi-style test dataset and on each style separately. Examples of style transfer from a set of references were provided and a hypothesis for working in a one-shot learning mode was tested. Currently model needs more tuning to transfer style from just one example.

During our experiments, we tested the proposed solution only for 3 domains for English-Ukrainian translation. Also, we chose the optimal values range for the transformation coefficient by checking the changes after tweaking its value. The proposed model can be tuned further to learn new domains better. This solution can be scaled to a larger number of languages by changing the external model, which generates semantic descriptors.

As a further development, we propose to tune the model enough to finally run it in a one-shot learning mode. Also, we would like to interpret semantic descriptors in more detail to provide better control over text features and get a better understanding of each value influence. It can be done by using the sparse embeddings approach. Also, we would like to further modify the proposed architecture by trying other semantic encoders or changing the structure of the dimensionality reduction module.

Acknowledgment

This article is based upon work and STSM from COST Action Multi3Generation (CA18231), , supported by COST (European Cooperation in Science and Technology).

References

[1] A. Gillioz, J. Casas, E. Mugellini and O. A. Khaled, "Overview of the Transformer-based Models for NLP Tasks," 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 2020, pp. 179-183, doi: 10.15439/2020F20.

[2] M. Lewis, ‘BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension’. arXiv, 2019.

[3] S. M. Lundberg and S.-I. Lee, ‘A Unified Approach to Interpreting Model Predictions’, Advances in Neural Information Processing Systems, 2017, p. 30.

[4] Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning Erdem, Kuyu, Yagcioglu, Frank, Parcalabescu, Plank, Babii, Turuta et al. Journal of Artificial Intelligence Research 73 (2022) 1131-1207. https://doi.org/10.1613/jair.1.12918

[5] T. B. Brown et al., ‘Language Models are Few-Shot Learners’, arXiv [cs.CL]. 2020.

[6] Saichyshyna N., Maksymenko D., Turuta O., Yerokhin A., Babii A. and Turuta O. 2023. Extension Multi30K: Multimodal Dataset for Integrated Vision and Language Research in Ukrainian. In Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP), pages 54–61, Dubrovnik, Croatia. Association for Computational Linguistics.

[7] https://zakon.rada.gov.ua/rada/main/en/llenglaws

[8] R. Hanslo, "Deep Learning Transformer Architecture for Named-Entity Recognition on Low-Resourced Languages: State of the art results," 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS), Sofia, Bulgaria, 2022, pp. 53-60, doi: 10.15439/2022F53.

[9] J. Tiedemann, S. Thottingal, ‘OPUS-MT -- Building open translation services for the World’, Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, 2020, pp. 479–480.

[10] A. Conneau *, K. Khandelwal *, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzman, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov Unsupervised Cross-lingual Representation Learning at Scale

[11] N. Reimers and I. Gurevych, ‘Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks’, in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 11 2019.

[12] M. Post, ‘A Call for Clarity in Reporting BLEU Scores’, Proceedings of the Third Conference on Machine Translation: Research Papers, 2018, pp. 186–191.

[13] S. Banerjee, A. Lavie, ‘METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments’, Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005, pp. 65–72.

[14] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, ‘BERTScore: Evaluating Text Generation with BERT’. ArXiv, 2019.

[15] D.Maksymenko, N.Saichyshyna, O.Turuta, O.Turuta, A.Yerokhin, and A. Babii. 2022. Improving the machine translation model in specific domains for the ukrainian language. In 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), pages 123–129.

[16] M. Junczys-Dowmunt, ‘Marian: Fast Neural Machine Translation in C++’. arXiv, 2018.

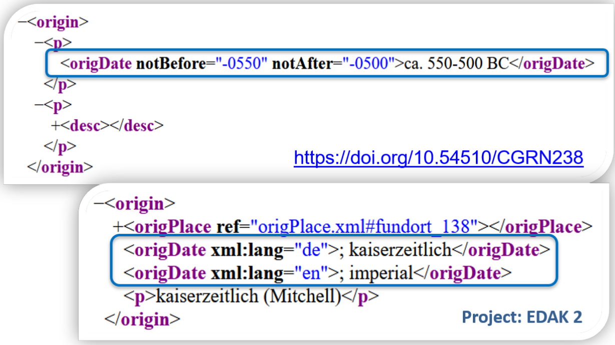
[17] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

DOI: 10.15439/2023F1515
ISSN 2300-5963 ACSIS, Vol. 35

EpiDoc Data Matching for Federated Information Retrieval in the Humanities

Sylvia Melzer*†, Meike Klettke‡, Franziska Weise*, Kaja Harter-Uibopuu* and Ralf Möller†
*Universität Hamburg
Centre for the Study of Manuscript Cultures, Warburgstraße 26, 20354 Hamburg, Germany
Email: {sylvia.melzer, franziska.weise, kaja.harter}@uni-hamburg.de
†University of Luebeck
Institute of Information Systems, Ratzeburger Allee 160, 23562 Luebeck, Germany
Email: moeller@uni-luebeck.de
‡University of Regensburg
Faculty for Computer Science and Data Science, University Street 31, 93053 Regensburg, Germany
Email: meike.klettke@ur.de

Abstract—**The importance of federated information retrieval (FIR) is growing in humanities research. Unlike traditional centralized information retrieval methods, where searches are conducted within a logically centralised collection of documents, FIR treats each information system as an independent source with its own unique characteristics. Searching these systems together as a centralised source results in lower precision in humanities research, even when the research data itself is structured and stored according to standardised guidelines such as EpiDoc, and requires the need to be able to trace the origin of records to avoid incorrect historical conclusions. Matching of queries against all data sets in each source is proving less effective. A global search index that enables traceable matching of key values deemed relevant would provide a more robust solution here. In this article, we propose a solution that introduces a novel EpiDoc data matching procedure, facilitating traceable FIR across distinct epigraphic sources.**



```
−<origin>
  −<p>
    <origDate notBefore="-0550" notAfter="-0500">ca. 550-500 BC</origDate>
   </p>
  −<p>
    +<desc></desc>
   </p>
 </origin>
```

```
−<origin>
   +<origPlace ref="origPlace.xml#fundort_138"></origPlace>
    <origDate xml:lang="de">; kaiserzeitlich</origDate>
    <origDate xml:lang="en">; imperial</origDate>
    <p>kaiserzeitlich (Mitchell)</p>
  </origin>
```

Fig. 1. Different “date” representations in EpiDoc. Top: EpiDoc file from the CGRN project. Bottom: EpiDoc file from the EDAK project.

I. Introduction

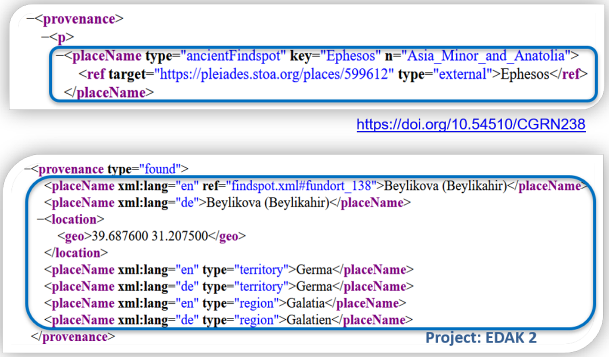
IN THE field of humanities, the need for federated information retrieval (FIR) is becoming increasingly important [5], [13], [20], [21]. FIR refers to the process of searching for relevant information across distributed and autonomous information systems within a database federation. A database federation provides a logical centralisation of data without the need to change the physical implementation of databases and maintain the identity of autonomous developed databases.

Information systems have emerged in some humanities projects, e.g., the epigraphy projects “Epigraphische Datenbank zum antiken Kleinasien” (EDAK) [23] and “Collection of Greek Ritual Norms” (CGRN) [3], they are not collaboratively searchable because these information systems run in heterogeneous hardware and software environments, and have different data models. Although both projects use an epigraphy-specific XML format called EpiDoc [7], a customized version of TEI (Text Encoding Initiative) [22], is additionally provided to exchange research data. The purpose of the EpiDoc format is to enhance machine-readability, and effectively searching for specific information within EpiDoc files relies on correctly matching the research data extracted from these files across different sources.

Although an EpiDoc schema was developed in each of the two projects, in Fig. 1 it is shown that the general structure is similar, however, in detail the XML tags are applied differently. In an example, according to the TEI guidelines, the XML tag “origDate” is used to represent dates. In practice, concrete date specifications for “origDate” vary. In CGRN a date presents a century and in EDAK an epoch (see Fig. 1). When specifying the place, mapping from both sources becomes even more difficult because the semantics of the place terms are different. While in CGRN the place name is specified in the XML tag “ref”, in EDAK the tag “placeName” is used (see Fig. 2). Only an expert in this field can say exactly how the places can be mapped onto each other. For an efficient FIR, a correct matching of data sets from different sources must be defined in advance to provide precise IR results.

In the humanities, mapping of data from different sources is still done manually. The manual procedure is necessary

The research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany´s Excellence Strategy – EXC 2176 ‘Understanding Written Artefacts: Material, Interaction and Transmission in Manuscript Cultures’, project no. 390893796.



```
–<provenance>
  –<p>
    –<placeName type="ancientFindspot" key="Ephesos" n="Asia_Minor_and_Anatolia">
        <ref target="https://pleiades.stoa.org/places/599612" type="external">Ephesos</ref>
     </placeName>
```

https://doi.org/10.54510/CGRN238

```
–<provenance type="found">
    <placeName xml:lang="en" ref="findspot.xml#fundort_138">Beylikova (Beylikahir)</placeName>
    <placeName xml:lang="de">Beylikova (Beylikahir)</placeName>
  –<location>
      <geo>39.687600 31.207500</geo>
    </location>
    <placeName xml:lang="en" type="territory">Germa</placeName>
    <placeName xml:lang="de" type="territory">Germa</placeName>
    <placeName xml:lang="en" type="region">Galatia</placeName>
    <placeName xml:lang="de" type="region">Galatien</placeName>
  </provenance>
```

Project: EDAK 2

Fig. 2. Two different representations of "placeName" in EpiDoc. Top: EpiDoc file from the CGRN repository. Buttom: EpiDoc file from the EDAK repository.

due to the need to achieve a high degree of precision in semantic evaluation, which is crucial to avoid making false statements about cultural artefacts. The current challenge is to automatically match the data, which is stored in various EpiDoc formats. Searching through hundreds upon thousands of EpiDoc files can be challenging, so it can be advantageous to transfer data represented in EpiDoc to a relational database. The advantages of transforming data from EpiDoc into a relational database are:

- (i) higher query answering performance because relational databases are optimized for query answering,
- (ii) scalability which makes relational databases a good choice for large data sets, and
- (iii) opportunity to use pre-defined data types and to define relationships between data elements.

In the CGRN and EDAK projects, relational mappings exist beside the XML representations. We use both, the existing relational mappings and the XML tags, to compute relevant key values and use these values for our new indexing approach for a precise and performant EpiDoc data matching. If the relational mappings are missing, it is possible to use large language models such as Generative Pre-trained Transformer 2 (GPT-2) [24] or other versions to generate the required relational mappings. Computing the relational mappings is done by entering the specific EpiDoc schemas into the language model, which can then generate the required mappings.

This article presents a novel method to overcome the difficulties of automatically matching epigraphic data from different sources with different EpiDoc schemas while ensuring semantic precision. The proposed method offers potential benefits for humanities scholars by maintaining precision while minimising heterogeneity caused by differences in data semantics. Furthermore, the proposed EpiDoc data matching approach can be applied to an FIR to offer scholars in the humanities access to a wider range of information from distributed and autonomous sources.

The remaining article is structured as follows: Section II gives an overview of some work on the representation of epigraphic data, as well as selected approaches to match these data. Section III describes a new matching process for epigraphic research data to enhance correct semantic mapping. This process provides the basis for enabling FIR, which is described in Section IV. Section V concludes this article and gives an outlook.

II. Related Work

Studies emphasise the need for careful selection of repositories in federated search to avoid describing objects that are not searchable. A prototype federated search engine [17] or cross-domain information system [11] has been developed to address this problem by integrating selected repositories. However, manual mapping is usually required to link different content with high precision. This article proposes an automatic data mapping that eliminates the need for manual mapping.

Data in EpiDoc.XML is syntactically represented as XML documents. Existing work on XML matching can be applied for schema matching and mapping. These methods typically begin with element-level similarity assessment [1] and then extend to data set level comparison [10]. For comparing EpiDoc data sets, this article suggests adopting similarity functions used for XML elements, such as Levenshtein distance [14] or the Soundex algorithm [9]. The overall similarity between data sets can be evaluated using metrics like Jaccard distance [8].

Applications processing semi-structured data often require schema matching for tasks like schema integration and schema clustering. Previous research, such as [10] and [18], has defined similarity functions for semi-structured data (DTDs) and schema fragments (from XSD) to address these needs. In our approach, we can focus specifically on matching different variants of EpiDoc, assuming that all input data are in this format. This eliminates the need to rely on general schema matching algorithms and allows us to treat the different EpiDoc variants as dialects of the same language.

III. Matching of Epigraphic Research Data

Matching data sets involves the process of comparing two or more data sets to identify similar elements. In the following the general process of matching data and the process of matching EpiDoc data are presented (cf. [4] and see Fig. 3).

A. General Process of Matching Data

a) Data pre-processing: The initial step in data pre-processing involves preparing the data sets for matching. This involves cleaning, formatting, and standardizing the data sets to ensure compatibility and effective comparison. This may also include removing duplicates and identifying missing data.

b) Indexing: The complexity of matching records increases with the number of records to be matched. Indexing is a strategy to pre-select potential matches and leads to a reduction in the number of matches. Indexing usually involves identifying the key variables that will be used to match the data sets. These variables may include unique identifiers, such as the titles of editions, dates, or places. A traditional indexing

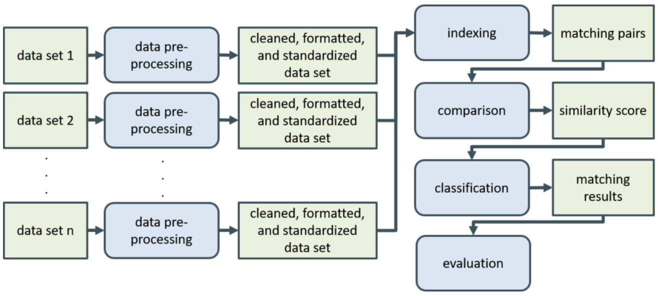


Fig. 3. The general process of matching n data sets. Based on [4] (extended)

method has the name blocking [2]. The method compares only those records that are based on the same so-called blocking criteria. In this article, we present an advanced blocking method that considers both the XML and relational structure of databases created based on the XML files to determine the matching criteria and then identify the candidate pairs for comparison.

c) Comparison: In this step, the data sets are compared to identify matches. For this purpose, the similarity between the key values used during indexing, but also additional other values, will be calculated.

d) Classification: Based on the data set comparison, the matching records are classified as match or non-match.

e) Evaluation: The final step involves validating the matched data sets and reviewing the results for accuracy and completeness. This may involve checking for errors, inconsistencies, or missing data and making any necessary adjustments.

B. Process of Matching EpiDoc Data

1) Data Pre-processing: As a starting point, we use the EpiDoc files from the projects EDAK and CGRN. The EpiDoc data are transformed into a relational database using the so-called databasing on demand (DBoD) approach [12], [19]. The DBoD approach is used for building project-specific information systems on demand in a few hours and with few resources. The DBoD process consists of the following steps:

1) Transformation of all EpiDoc files into one CSV file. The mapping of the EpiDoc XML elements into a canonical mapping was carried out via the widely used EpiDoc XSLT stylesheets [6] as defined by epigraphers (cf. data representations on the websites: [3], [23]).
2) Insert all the data from the CSV file in a database instance.

From the website it could be inferred that both projects have different relational representations. The EDAK project has the column names “Edition”, “Inscription type”, “Obejct type”, “Region”, “Place”, “Date (epoch)”, “Text”, . . ., while CGRN has the column names “Edition”, “Provenance”, “Place”, “Date (century)”, “Text”, ...

To ensure comparability between centuries and epochs, we have transformed them into a starting time (notBefore) and an ending time (notAfter). For instance, the “imperial” epoch was converted to notBefore=1 and notAfter=300. This conversion allows for a standardized representation of time periods, facilitating analysis and comparison across different historical eras.

In [4] it is suggested to standardize tables, so that the relational models consist of the same entity types, and thus facilitates the data matching process. Since in our case the mapping rules are given (defined by epigraphers), it is not necessary to use the same database models for all databases, especially since in practice this form of database entry would not be accepted by the users due to the diverse requirements. In the indexing step, it is shown that mapping rules, rather than standardisation of databases, are sufficient to successfully perform data matching. If the structure of the relational database is already known, mapping EpiDoc data sets directly to a database instance using XSLT or a similar transformation language may be easier and more efficient. However, if these mapping rules are missing, our hypothesis is that large language models can be useful to define such mapping rules if the input data are the EpiDoc schemas and the EpiDoc guidelines. We tested our hypothesis with the use of ChatGPT [16], which has statistical knowledge, acquired through its extensive training on large data sets. We gave ChatGPT the EpiDoc guidelines as input and the two EpiDoc schemas, and asked it to provide us with the mapping for transforming EpiDoc data into a relational database. As output, we received tables that correspond to the hierarchical structure of the XML file. In total, ChatGPT delivered more tables than needed, but all contents remain taken into consideration.

2) Indexing: Indexing includes identifying the key variables for an efficient schema matching process. For existing relational databases, it can be assumed that the column names belong to the key variables and are used for their project-specific analysis. Therefore, the column names are regarded as key variables. Since matching all key variables is inefficient, a blocking procedure is traditionally used [4]. This reduces the number of comparisons and improves performance. This article employs an alternative approach to existing blocking methods by utilizing the XML schema to identify matching candidates. For seen data (mapping rules known), XSLT is used, while ChatGPT is used for unseen data (mapping rules are not known). The identification of matching candidates is described in the next paragraph.

If two sets A and B of XML tags, where the sets A and B are from different EpiDoc schemes, are mapped to the same element, then that element is a matching candidate to be added to the matching candidate set C. Let $A = \{a_1, \dots a_i\}$ and $B = \{b_1, \dots b_j\}$ be sets of XML tags, and let f be a function which represents a mapping from A to B: $f : A \rightarrow B$, then the matching candidates C are given by:

$$C = \{b \in B : \exists a \in A \text{ with } f(a) = b\} \qquad (1)$$

In the given example, the EpiDoc schema “EDAK” belongs to set A and “CGRN” to set B. Table I displays the column names used in the respective projects and the

TABLE I
OVERVIEW OF MATCHING CANDIDATES DERIVED FROM THE EPIDOC SCHEMES AND RELATIONAL REPRESENTATION OF THE EPIDOC CONTENT

EpiDoc Schema	Column name	XML tag	matching candidate C
EDAK	Edition	title	no
EDAK	Inscription type	term	no
EDAK	Object type	objectType	no
EDAK	Region	placeName	yes
EDAK	Place	placeName	yes
EDAK	Date (epoch)	origDate	yes
EDAK	Text	div	yes
CGRN	Edition	idno	no
CGRN	Provenance	placeName	yes
CGRN	Date (century)	origDate	yes
CGRN	Text	div	yes

TABLE II
MATCHING DATA OF EDAK AND CGRN

project	ID	placeName	Sndx-PN
EDAK	a_1	Pisidia	P230
EDAK	a_1	Antiochia	A532
EDAK	a_2	Ephesus	E120
CGRN	b_1	Ephesos	E120
CGRN	b_2	Tomis	T520
CGRN	b_3	Athens	A352

corresponding XML tags. The matching candidates are $C = \{\text{placeName}, \text{origDate}, \text{div}\}$.

a) Matching: The data also includes words that sound similar and can also be judged semantically similar, e.g. “Ephesus” and “Ephehos.” In the simple comparison, the terms are evaluated as different, even though they are the same place.

The commonly used phonetic coding algorithm Soundex is employed to find a match despite minor differences. Each word is coded into a letter and a three-digit number sequence, words with the same coding are scored as similar. That means for our example that “Ephesus” coded as E120 and “Ephehos” coded as E120 are semantically similar. For more details of the Soundex algorithm is given in [15].

For data represented with the data type “Text” or “Date”, the Soundex procedure is not applied as it maps to strings and numeric values. Although “Date” is also a numeric value, it represents a period of time, and the comparison of time periods differs from that of strings and numeric values. This is not considered in the indexing procedure, but in our comparison step.

Using the matching criteria Soundex for the matching candidate “placeName,” then the indices and record pairs are presented as shown in Table II. The matching key values P230, A532, and E120 are identified. The only record pair that was identified is (a_2, b_1) for E120.

Formally, the set of matching pairs are computed as follows:

$$f_{\text{Sndx}} : C_{A_{\text{Sndx}}} \to C_{B_{\text{Sndx}}}$$

$$\begin{aligned} P = &\{(a \in C_{A_{\text{Sndx}}}, b \in C_{B_{\text{Sndx}}}) : \exists a \in C_{A_{\text{Sndx}}} \\ &\text{with } f_{\text{Sndx}}(a) = b\} \end{aligned} \tag{2}$$

When comparing the EDAK data set with the CGRN data set, there is only the overlap with one region. This result was to be expected, as it is common in the humanities for research to be conducted in a very specialised area, and it can be assumed that there is little overlap. Nevertheless, in the humanities one is interested in finding other interpretations of texts or even the same data sets. Our next application example shows that our algorithm can also handle larger data sets and that more similar data sets are to be expected in the context of a humanities project. We split one part of the EDAK data set into two so that we have 199 entries in one data set and 201 in the other. Of these, 159 matching candidates were identified based on the same region. The two comparisons (CGRN + EDAK; EDAK part 1 + EDAK part 2) provide the results, as expected, with a high degree of precision.

If comparison is made with the tables or columns provided by ChatGPT, the difference is that there is a larger number of tables or table columns that would have to be compared with each others. As a result, as the number of tables grows, more key candidates emerge and more matches are made than would be necessary. However, if there are no mapping rules, this approach is still helpful because precise results can still be shown.

3) Comparison: The comparison process in schema matching indicates the degree of similarity between two record pairs to determine whether they are a match or not.

The comparison function $c(a_i, b_j)$ maps the matching pairs values of a_i and b_j as well as the pairs with the data type “Text” to a similarity score in the range $[0, 1]$, where 0 indicates no similarity and 1 indicates a perfect match. The comparison function can be defined using different similarity metrics, such as the Jaccard coefficient, cosine similarity, or edit distance, depending on the characteristics of the schema elements and the matching criteria.

In this article, the Jaccard similarity is used to compare the sets of matching terms from P associated with a_i and b_j defined as follows:

$$c(a_i, b_j) = \left| \frac{P(a_i) \cap P(b_j)}{P(a_i) \cup P(b_j)} \right| \tag{3}$$

where $|.|$ denotes the cardinality of a set. For dates, the similarity is computed in the following way. Assuming d_1 and d_2 represent the time periods d_{a_1} to d_{a_2} and d_{b_1} to d_{b_2}, the date similarity is given by:

$$sim_{\text{date}}(a_1, b_1) = \begin{cases} 1 & d_{a_1} \leq d_{b_2} \text{ and } d_{b_1} \leq d_{a_2} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

If there is a temporal overlap, the similarity score is 1, otherwise 0.

For “placeName”, the similarity score sim_{place} is also 1 if there is a match, otherwise 0.

TABLE III
COMPARISON

ID	placeName	Date	div	sim_{all}
a_1	Ephesus	0001 - 0300	[text]	
b_2	Ephesos	-550 - 500	[text]	
	1	1	0.5	2.5

The comparison function can be used to rank the candidate matches based on their similarity scores:

$$\begin{aligned}\text{sim}_{\text{all}}(a_i, b_j) = {} & sim_{\text{place}}(a_i, b_j) \\ & + sim_{\text{date}}(a_i, b_j) \\ & + c(a_i, b_j) \end{aligned} \tag{5}$$

and to select the best match(es) according to a given threshold or ranking criteria.

The comparison between the two EDAK data sets revealed a similarity score ranging between 2.00 and 2.25. The determination of whether this score indicates a match is explained in the following step.

4) Classification: Classifying the compared record pairs based on their summed similarities is a two-class (binary) classification task. Each compared record pair is classified to be either a match (1) or a non-match (0) depending on a threshold value θ.

The classification of each compared record pair can be based on either the full comparison vectors or on the summed similarities. Based on the summed similarity score, a match is defined as:

$$\text{match} = \begin{cases} 1 & \text{sim} \geq \theta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

In the context of the project, a good value for θ is between the "number of attributes" divided by 2 and the total "number of attributes" to achieve matching results between approximately 50% and 100%. Formally:

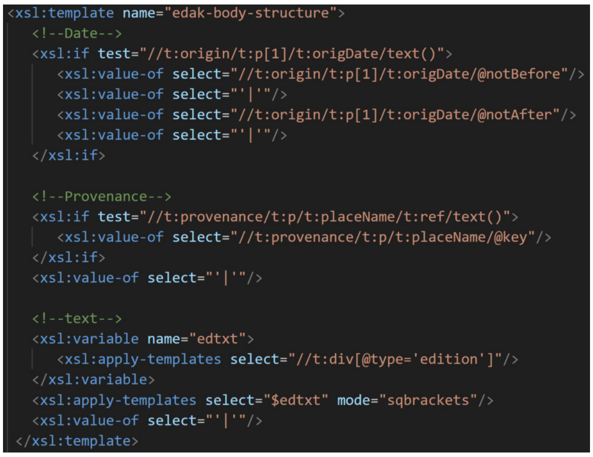
$$\frac{\text{number of attributes}}{2} \leq \theta \leq \text{number of attributes}. \tag{7}$$

If $\theta =$ "number of attributes" (100% similarity), then it could indicate a duplicate. It is important to note that, in general, the optimal value for θ may depend on the specific characteristics of the data sets being compared and the desired level of similarity between them. Now the matching results can be evaluated by experts.

5) Evaluation: In the EDAK project some dates are marked as "Unknown". In this case, the date should have been cleaned up in the pre-processing step or should have been taken into account in the algorithm with another θ value. After we adjusted θ, the similarity score increased.

IV. Federated Information Retrieval

FIR is an approach used in information retrieval (IR) systems where multiple autonomous and distributed information sources are integrated to provide a unified and comprehensive search experience. Unlike traditional centralized IR methods, which rely on a single collection of documents, FIR treats each information system as an independent source with its own unique characteristics.



```
<xsl:template name="edak-body-structure">
   <!--Date-->
   <xsl:if test="//t:origin/t:p[1]/t:origDate/text()">
      <xsl:value-of select="//t:origin/t:p[1]/t:origDate/@notBefore"/>
      <xsl:value-of select="'|'"/>
      <xsl:value-of select="//t:origin/t:p[1]/t:origDate/@notAfter"/>
      <xsl:value-of select="'|'"/>
   </xsl:if>

   <!--Provenance-->
   <xsl:if test="//t:provenance/t:p/t:placeName/t:ref/text()">
      <xsl:value-of select="//t:provenance/t:p/t:placeName/@key"/>
   </xsl:if>
   <xsl:value-of select="'|'"/>

   <!--text-->
   <xsl:variable name="edtxt">
      <xsl:apply-templates select="//t:div[@type='edition']"/>
   </xsl:variable>
   <xsl:apply-templates select="$edtxt" mode="sqbrackets"/>
   <xsl:value-of select="'|'"/>
 </xsl:template>
```

Fig. 4. XSLT code shows that the EpiDoc data is read from the project CGRN

A federation of database systems is called federated database system (FDBS) and integrates multiple autonomous database systems into a single database system. However, the identity of the individual databases is not lost in the merging process. In general, the constituent the physically decentralized databases are interconnected via computer networks.

We have implemented a prototype to integrate the new indexing procedure into an FDBS and thus enable FIR. For this purpose, we have written a Python script combined with the XSLT stylesheets for each project (EDAK and CGRN) that first transforms the EpiDoc data into a relational database model. To do this, it was necessary to adapt the existing XSLT stylesheets to transfer the project-specific mappings from XML to a relational database. Fig. 4 shows the representative XSLT source code representing how the EpiDoc data (date, provenance, and text) is read from the CGRN project.

A further Python script was written for the presented matching process of epigraphic research data to compute the similarity score for the three selected attributes: "Place", "Date" (splitted into "notBefore" and "notAfter" to represent a period), "Text". The result of our script is a tabular listing of all selected attributes that are ranked according to the similarity score. CGRN and EDAK did not provide any results for the area for the place name "Ephesus" and the period 1-300 (notBefore-notAfter). This result was almost to be expected, since research in the field of the humanities is designed in such a way that the projects are usually distinct in terms of content and detail.

As evidence of the applicability of the new data matching process, we generated two data sets from the EDAK project and compared them with each other. In Table IV the top 10 results with the highest similarity are presented. In sum, we have an overall similarity of 39.75% within the result set, which was to be expected.

TABLE IV
TOP 10 RESULTS OF THE EPIDOC DATA MATCHING PROCESS

Edak 1	Edak 2	notBefore	notAfter	Sndx-PN	Sim_Score
TAM V 2, 1152	TAM V 2, 868	1	300	L356	2,25
TAM V 2, 1152	TAM V 2, 1151	1	300	L356	2,21
TAM V 2, 1075	TAM V 2, 1151	1	300	L356	2,20
TAM V 2, 1024	TAM V 2, 987	1	300	G430	2,19
MAMA VII, Nr. 67	Robinson, TAPhA 57 (1926) , Nr. 7	1	300	L250	2,19
Laminger-Pascher, Inschriften Lykaoniens (1992) , Nr. 303	Robinson, TAPhA 57 (1926) , Nr. 7	1	300	L250	2,19
TAM V 2, 1075	TAM V 2, 868	1	300	L356	2,19
TAM V 2, 1026	TAM V 2, 987	1	300	G430	2,17
TAM V 2, 1085	TAM V 2, 1061	1	300	G430	2,17
TAM V 2, 1024	TAM V 2, 1061	1	300	G430	2,17
MAMA VIII, Nr. 315	Robinson, TAPhA 57 (1926) , Nr. 7	1	300	L250	2,17
TAM V 2, 1182	TAM V 2, 868	1	300	L356	2,16

The current implementation so far only supports a search by the three selected categories (attributes). We have also prototypically implemented our new EpiDoc data matching method as an FIR in such a way that a user can enter the attributes “placeName”, “notBefore”, “notAfter”, and “text.” As a result, the user receives a list with the matching candidates and the similarity score such as presented in Table IV.

V. SUMMARY AND OUTLOOK

This article is about the increasing importance of federated information retrieval (FIR) in the field of humanities. An FIR, unlike traditional centralised information retrieval methods, treats each information system as an autonomous resource with unique properties. The article proposes a novel EpiDoc data matching procedure which uses the XML schema representations and relational representations to identify the matching candidates, so that on the one hand the number of data matches is reduced and on the other hand the precision is maintained. The new procedure was successfully implemented as a prototype and will be evaluated in the future by transferring it to the productive system at the Universität Hamburg.

REFERENCES

[1] Algergawy, A., Nayak, R., Saake, G.: Element similarity measures in XML schema matching. Inf. Sci. **180**(24), 4975–4998 (2010)

[2] Baxter, R., Christen, P., Churches, T.: A Comparison of Fast Blocking Methods for Record Linkage. Workshop on Data Cleaning, Record Linkage and Object Consolidation at the Ninth ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington DC (2003)

[3] Carbon, J.M., Peels-Matthey, S., Pirenne-Delforge, V.: Collection of Greek Ritual Norms (CGRN) (2017-, consulted on 10/05/2023). https://doi.org/https://doi.org/10.54510/CGRN0, http://cgrn.ulg.ac.be

[4] Christen, P.: Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Springer Publishing Company, Incorporated (2012)

[5] Demeester, T., Nguyen, D., Trieschnigg, D., Develder, C., Hiemstra, D.: Snippet-Based Relevance Predictions for Federated Web Search. In: Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S., Agichtein, E., Segalovich, I., Yilmaz, E. (eds.) Advances in Information Retrieval. pp. 697–700. Springer Berlin Heidelberg (2013)

[6] Elliott, T., Au, Z., Bodard, G., Cayless, H., Lanz, C., Lawrence, F., Vanderbilt, S., Viglianti, R., et al.: EpiDoc Reference Stylesheets (version 9). Available: https://sourceforge.net/p/epidoc/wiki/Stylesheets/ ((2008-2017)), accessed January 22, 2022

[7] Elliott, T., Bodard, G., Mylonas, E., Stoyanova, S., Tupman, C., Vanderbilt, S., et al.: EpiDoc Guidelines: Ancient documents in TEI XML (Version 9). Available: https://epidoc.stoa.org/gl/latest/. ((2007-2022)), accessed January 22, 2022

[8] Jaccard: The distribution of the flora of the alpine zone. In: New Phytologist. vol. 11, pp. 37–50 (1912)

[9] Jacobs, J.: Finding words that sound alike. The SOUNDEX algorithm. Byte 7 pp. 473–474 (1982)

[10] Lee, M.L., Yang, L.H., Hsu, W., Yang, X.: Xclust: clustering xml schemas for effective integration. In: Proceedings of the eleventh international conference on Information and knowledge management. pp. 292–299 (2002)

[11] Melzer, S., Peukert, H., Wang, H., Thiemann, S.: Model-based Development of a Federated Database Infrastructure to support the Usability of Cross-Domain Information Systems. In: IEEE International Systems Conference (SysCon 2022), Montreal, Canada. IEEE (2022)

[12] Melzer, S., Schiff, S., Weise, F., Harter, K., Möller, R.: Databasing on demand for research data repositories explained with a large epidoc dataset. CENTERIS (2022)

[13] Melzer, S., Thiemann, S., Möller, R.: Modeling and simulating federated databases for early validation of federated searches using the broker-based sysml toolbox. In: IEEE International Systems Conference, SysCon 2021, Vancouver, BC, Canada, April 15 - May 15, 2021. pp. 1–6. IEEE (2021)

[14] Miller, F.P., Vandome, A.F., McBrewster, J.: Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau-Levenshtein Distance, Spell Checker, Hamming Distance. Alpha Press (2009)

[15] Odell, M.K., Russell, R.: Patent numbers 1261167 (1918) and 1435663 (1922). Washington, DC: US Patent Office (1918)

[16] OpenAI: ChatGPT (Vers. 3.5). https://openai.com (2021)

[17] Pergantis, M., Varlamis, I., Giannakoulopoulos, A.: User Evaluation and Metrics Analysis of a Prototype Web-Based Federated Search Engine for Art and Cultural Heritage. Information **13**(6), 285 (Jun 2022)

[18] Rahm, E., Do, H.H., Massmann, S.: Matching large xml schemas. ACM SIGMOD Record **33**(4), 26–31 (2004)

[19] Schiff, S., Melzer, S., Wilden, E., Möller, R.: TEI-Based Interactive Critical Editions. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems. pp. 230–244. Springer International Publishing, Cham (2022)

[20] Shokouhi, M., Baillie, M., Azzopardi, L.: Updating Collection Representations for Federated Search. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 511–518. SIGIR ’07, Association for Computing Machinery, New York, NY, USA (2007)

[21] Shokouhi, M., Si, L.: Federated Search. Foundations and Trends® in Information Retrieval **5**(1), 1–102 (2011)

[22] Text Encoding Initiative: P5: Guidelines for Electronic Text Encoding and Interchange, Version 4.0.0. https://tei-c.org/Vault/P5/4.0.0/doc/tei-p5-doc/en/html/ (2020), accessed 29 June 2022

[23] Universität Hamburg: Epigraphische Datenbank zum antiken Kleinasien (2013-2016), https://www.epigraphik.uni-hamburg.de/content/index.xml

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is All you Need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1075–1080

DOI: 10.15439/2023F1151
ISSN 2300-5963 ACSIS, Vol. 35

VICRA: Variance-Invariance-Covariance Regularization for Attack Prediction

Aditya Srinivas Menon
0009-0000-9326-028X
Student, Computer Science
Indian Institute of Information Technology Kottayam
Kerala, India
Email: adityasrinivas20bcs8@iiitkottayam.ac.in

Gouri Nair
0009-0007-5793-9966
Student, Electronics and Communication
Indian Institute of Information Technology Kottayam
Kerala, India
Email: gourinair20bec7@iiitkottayam.ac.in

***Abstract*—In cybersecurity, accurate and timely prediction of attacks plays a crucial role in mitigating the risks and impacts of cyber threats. However, traditional attack prediction methods that rely on training Machine Learning (ML) algorithms directly on raw data often suffer from high false alarm rates and low detection rates, leading to inaccurate and unreliable results. To overcome these limitations, this paper presents a novel approach that integrates attack prediction with self-supervision using variance-invariance-covariance regularization (VICReg). The proposed method harnesses VICReg to enhance raw data and generate representations while leveraging self-supervision to learn meaningful features without supervision. Training classic ML algorithms on these refined representations improves prediction accuracy and enhances the robustness of the learning process. We provide a comprehensive description of the proposed method and present an evaluation of its performance on several benchmark datasets. The experimental results demonstrate the superiority of the proposed method over classic ML algorithms.**

***Index Terms*—Self-supervised learning, Deep Learning, Structured Data, Attack Prediction, Wireshark**

I. Introduction and Related Work

CYBERSECURITY is a major concern for businesses, governments, and individuals, as the damage caused by worldwide cybercrime is expected to reach $10.5 trillion annually by 2025. The global cybersecurity workforce is projected to be short 1.8 million people by 2022, with 66% of respondents reporting that they don't have enough capacity to address current threats. Predictive analysis has the potential to give organizations an advantage by allowing them to allocate their defence resources more effectively and automate the process of attack forecasting and prediction.

Some of the most actively studied problems include Network Risk Scoring (NRS) [1], Threat Detection and Classification (TDC) [2], Attack Prediction [3], phishing detection [4], web shell classification and automating security pipelines.

A. Attack Prediction

The number and sophistication of cyberattacks are constantly increasing, making it increasingly difficult for organizations to protect themselves against all likely threats. By predicting and preparing for potential attacks, risks and losses can be minimized. Attack prediction refers to the process of identifying and forecasting potential security threats or vulnerabilities in a system or network. This is a critical aspect of cybersecurity, as it helps organizations to proactively protect themselves against future attacks and to minimize the impact of any breaches that do occur.

TABLE I
Example of Wireshark captured data in table format

No.	Time	Source	Protocol	Length	Info
1	0	192.168.1.2	HTTP	98	GET Tindex.html
2	0.05	192.168.1.1	HTTP	145	HTTP/ 1.1 200 0K
3	0.06	192.168.1.2	HTTP	98	GET /css/style.css
4	0.09	192.168.1.1	HTTP	756	HTTP/ 1.1 200 0K
5	0.1	192.168.1.2	HTTP	98	GET /js/script.js
6	0.14	192.168.1.1	HTTP	903	HTTP/I.1 200 0K

B. Prediction Logic

One of the key tools in addressing cybersecurity threats is the use of network packet analyzers. These tools are designed to capture, analyze, and interpret network traffic, to identify potential security breaches and malicious activities. Among these tools, Wireshark [5] is a free and open-source (GNU General Public License) platform independent tool that serves as a packet analyzer. It is used for network issue resolution, examination, the development of communication protocols and educational purposes. Wireshark intercepts packets and presents them in a table format, with each row representing a single packet and each column displaying various details about the packet.

The captured packets can be filtered and sorted using various criteria, such as the protocol used, the source and destination addresses, or the specific type of data being transmitted. Table I-B shows an example of Wireshark data displaying six packets in table format. The columns include the time, source IP address, protocol, packet length, and a brief description of the packet

C. Self Supervised Learning

Self-supervised learning (SSL) [6] is a machine learning approach that seeks to acquire data representations without explicit supervision, thereby eliminating the need for labeled data. Through this method, the model autonomously learns valuable features and representations, which can be utilized for downstream tasks. SSL holds significant potential for enhancing the efficiency and effectiveness of learning algorithms in scenarios where labeled data is limited or costly to obtain.

One of the earliest works in SSL was the autoencoder [7], a neural network architecture that learns to reconstruct its input by training on an unlabeled dataset. Another popular SSL technique is contrastive learning [8] which is a method of training a model to distinguish between different representations of the same data.

SSL has been applied to a wide range of tasks such as computer vision [9], natural language processing [10] and speech recognition. SSL is still an active area of research and many questions remain open. For example, there is currently no consensus on the best way to evaluate the quality of the representations learned by SSL methods [11]. Additionally, the effectiveness of SSL for certain tasks or domains is still being explored. The issue of collapsing problem [12] in learning architecture is often mitigated by the presence of hidden biases, which may not have a transparent explanation or interpretation. This ensures that the learning process remains stable and effective. However, the underlying reasons or justifications for these biases may not always be readily apparent or easily interpretable.

D. VICReg

VICReg [13] a study by Meta Research introduced an approach that explicitly addresses the collapse problem by incorporating a straightforward regularization term on the variance of the embeddings along each dimension independently. VICReg, combines this variance term with a decorrelation technique that focuses on reducing redundancy and covariance regularization. By integrating these strategies, VICReg achieves state-of-the-art results on a range of downstream tasks, effectively overcoming the collapse problem and enhancing the quality and diversity of the learned embeddings.

While Self-Supervised Learning (SSL) has garnered substantial interest and recognition in the domains of computer vision and natural language processing (NLP), where large-scale datasets of unlabeled images are readily available (e.g. ImageNet), there has been very less research behind the adoption of SSL to tabular data. We apply self-supervision to predict attacks from tabular data using VICReg in this paper. Following are some of the significant observations:

1) Self supervision using VICReg on tabular data before applying Machine Learning (ML) algorithms helps in improving prediction accuracy.
2) When it comes to attack prediction, swap noise, a complementary approach to existing augmentation techniques in the tabular data setting, proved to be effective.
3) VICRA improves attack prediction accuracy compared to traditional Machine Learning (ML) methods.

Our key contributions can be summarized as follows:

1) We address the problem of Attack Prediction on wireshark features as a Machine Learning (ML) problem. We present the problem as an anomaly detection task for tabular data.
2) We propose a novel technique called VICRA (Variance-Invariance-Covariance Regularization for Attack Prediction) which uses self-supervision to enhance the tabular embeddings using swap noise and show significant increase in performance.
3) By leveraging the inherent structure of data and regularizing the learning process, the method is able to improve prediction accuracy and robustness.
4) We present a pipeline to train attack prediction models on wireshark data using VICRA.
5) We investigate the performance of VICRA attack prediction on popular datasets by comparing it with the current ML approaches.
6) Our VICRA technique improves the accuracy by over 2.48% for NSL KDD, 0.90% for UNSW NB15 and 7.17% for AWID2 than traditional ML approaches.

The rest of the paper is organized as follows. Our proposed approach is described in Section 2. Performance evaluation and findings of the work are shown in Section 3, Section 4 concludes the finding of the work.

II. Proposed Approach

In this section, we formally introduce our proposed VICRA system and highlight the specific areas of the problem that we aim to solve. The architectural overview of the proposed system is shown in Figure II.

The system takes Wireshark features as input and predicts if it's an attack or not. The overall procedure includes the following steps: (1) data preparation, (2) self-supervised learning, (3) embedding cloud generation, and (4) attack prediction. The primary focus of this research is to use self-supervision to enhance the feature embeddings while improving metrics for attack prediction. The processes are then thoroughly explained.

A. Data Preparation

As seen in Figure 1 the dataset is in the form of raw Wireshark features. To obtain useful information from the raw features we clean the data using standard data preparations methods which are mentioned below.

1) Missing Values: The data collected might have a lot of missing features. There are many proposed approaches on handling missing data in Wireshark data. For our approach, we perform List-wise Deletion [14] on categorical and binary features followed by Simple Mutation on continuous features. In List-wise Deletion, every case that has one or more missing values is removed whereas in Simple Mutation the missing value is replaced by the mean of the values in that feature.

2) Feature Selection: The Wireshark data that was recorded includes incorrect fields and extra information. In this step, feature selection methods reduce the number of features. In the process of attribute selection, information gain and gain

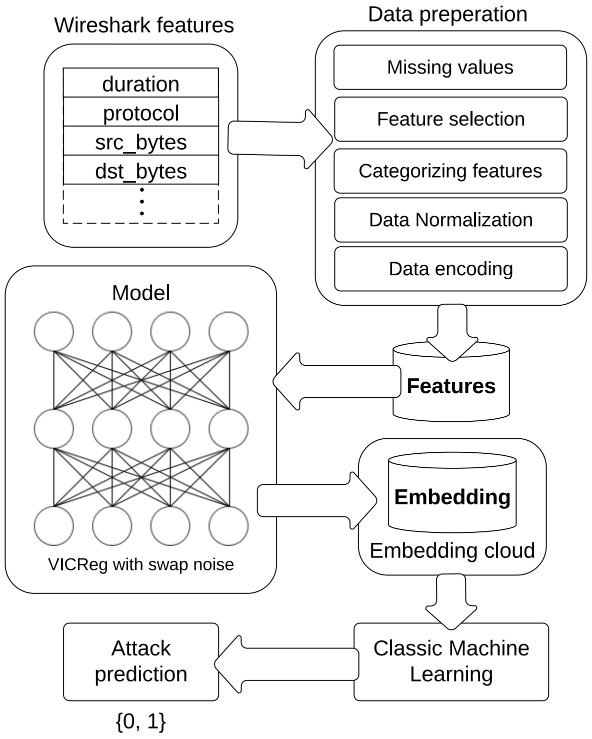


Fig. 1. System architecture of the VICRA System

ratio are commonly employed techniques for assessing the relevance of variables with respect to the target variable [15] [16]. Some columns like IPv4/IPv6 addresses are also removed since the model tends to overfit on these features.

3) Categorizing Features: The features are now categorized into continuous, categorical, discrete and binary for further pre-processing. Categorizing features is a crucial step in pre-processing data for attack prediction. Features are categorized into continuous (e.g., time duration), categorical (e.g., protocol type), discrete (e.g., number of packets), and binary (e.g., event occurred or not) types. By categorizing features, appropriate pre-processing techniques can be chosen for each type.

4) Data Normalization: Continuous features are either normalized or standardized. Log-transformation is performed on skewed data. Log transformation helps mitigate the effect of skewness by reducing the variability in the data and bringing it closer to a normal distribution.

5) Data encoding: Categorical features are one-hot encoded, and discrete features are treated as categorical or binned into ranges. Binary features require no pre-processing.

The output of this phase (given T) is clean and structured data which is fed into the VICReg model for self supervision.

B. Self Supervision

Figure II-B provides an illustration of the VICReg architecture, which encompasses variance, invariance, and covariance regularization. The process begins with a batch of features T obtained from the Data preparation step. From this, two sets of noisy features X and X' are generated and encoded into representations Y and Y'. These representations are then passed through an expander, resulting in the production of embeddings Z and Z'.

To ensure the effectiveness of the embeddings, several regularization techniques are applied. Firstly, the distance between embeddings from the same feature is minimized. Additionally, the variance of each embedding variable within a batch is maintained above a specified threshold. Furthermore, the covariance between pairs of embedding variables over a batch is attracted to zero, promoting decorrelation between the variables. It is worth noting that the two branches in the architecture do not necessarily share the same architecture or weights, although in most experiments, they consist of shared weight Feed Forward Layers (FFL).

To generate the noisy features X and X', swap noise is introduced to the original features, a process that is elaborated upon in Section III-C. After training, the model is then utilized for inference on the features obtained in the previous step. The resulting embeddings are generated and subsequently stored in the embedding cloud for further analysis or downstream tasks.

C. Embedding Cloud

The embeddings generated from the self-supervised inference are combined to form an embedding cloud as mentioned in [17]. The embedding cloud is a permanent storage of preprocessed embeddings which are used while training. Once the embedding cloud is generated and saved, we can proceed to train the model on the embeddings for attack prediction.

D. Attack Prediction

The embeddings stored in the embedding cloud, along with the ground truth labels, are utilized to train the Machine Learning (ML) model instead of training the model on the raw features. The self supervision performed regularizes the learning process and leverages the inherent structure of the data. The proposed method is found to yield improved prediction accuracy and greater robustness compared to traditional feature-based approaches. The use of self-supervised learning for generating embeddings has been demonstrated to be a promising approach for training ML models in a variety of applications. Our results suggest that this approach has the potential to be a useful tool in the field of cybersecurity for predicting and mitigating cyber attacks. While the proposed approach shows promising results for attack prediction, further research is required to fully explore its potential and assess its applicability to various types of attack prediction tasks.

III. Evaluation

A. Dataset

For our evaluation, we used three benchmark datasets commonly used in the field of cybersecurity: AWID 2 [18], NSL KDD [19], and UNSW NB15 [20] [21] [22]. These datasets provide a diverse range of attack scenarios and network traffic patterns, allowing us to assess the performance of our proposed approach across different contexts. The AWID 2

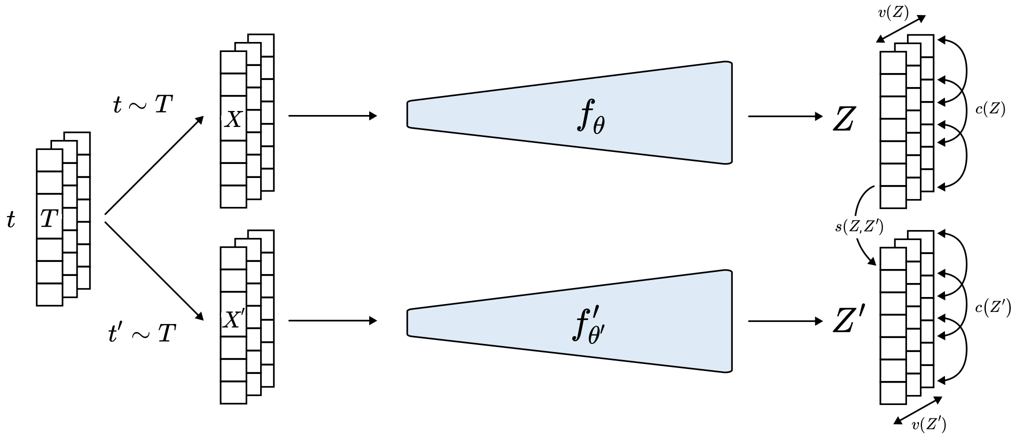


Fig. 2. VICReg architeture chosen for the wireshark data.

dataset contains wireless intrusion detection system (WIDS) data, the NSL KDD dataset is derived from the KDD Cup 1999 dataset, and the UNSW NB15 dataset includes network traffic data with various attack types. Table III-A provides an overview of the dataset distribution for three different datasets: AWID2, NSL KDD, and UNSW NB15. It is also clear from the distribution that in AWID2 dataset close to 97.15% of the data is from the “Normal” label, whereas in NSK KDD and UNSW NB15 datasets, the “Normal” label is only 53.46% and 31.94% respectively.

In order to evaluate the proposed approach, a binary classification scenario was created for each dataset. In this scenario, the "Normal" label was assigned the binary value of 0, while all other attack labels were grouped together and assigned the binary value of 1. This binary classification setup allows for the examination of the model’s performance in distinguishing between normal instances and instances associated with various types of attacks. By treating normal instances as the negative class (0) and attacks as the positive class (1), the model can be trained and tested to assess its ability to correctly classify instances as either normal or attack-related. This approach simplifies the problem by focusing on differentiating between normal behavior and malicious activities, enabling the evaluation of the model’s effectiveness in detecting and classifying attacks within the given datasets.

1) Data preparation: Prior to applying self supervision and learning algorithms, the dataset is cleaned using the techniques mentioned in Section 2.1. This includes handling missing values, selecting and categorizing features, data normalization for numerical features and data encoding for categorical features. The post processed data (T) is fed in batches to the self supervised VICReg model after adding noise.

2) Swap noise: Our proposed framework offers a complementary approach to existing augmentation techniques employed in the tabular data setting. As such, we conducted experiments involving the introduction of noise to randomly selected entries within each subset. This was achieved by overwriting the value of a chosen entry with another value randomly sampled from the same column. This augmentation technique is referred to as ’swap-noise’. In a previous study conducted by Michael Jahrer (MJ) [23], a noise creation method known as ’swap noise’ was introduced. This method involves randomly swapping a small portion of columns between two samples in order to generate noisy samples for training purposes. In the following section, we present our implementation of the swap noise technique, based on MJ’s original approach.

B. Baseline

To evaluate the performance of VICRA, we compared it against several methods commonly used in attack prediction tasks. These methods include traditional machine learning algorithms such as logistic regression, decision trees, and support vector machines, as well as deep learning models such as feed-forward neural networks. Additionally, we implemented our own baseline model that directly trained on the raw features without the self-supervised learning step.

C. Experiment Setup

To conduct the experiment, we first preprocess the AWID2, NSL KDD and UNSW NB15 datasets using the approach mentioned in Section 2.1. The features are then run through the VICReg model for self supervision. The VICReg model is a multi-layer perceptron architecture with stacked layers of linear transformations, batch normalization, and ReLU activation functions. The model consists of an expander module that is responsible for expanding the input features. It takes in features and applies a linear transformation followed by batch normalization and ReLU activation. This process is repeated n times in the expander module. The model is trained for 50 epochs and the representations are generated for each wireshark feature in the dataset. The generated representation

TABLE II
OVERVIEW OF THE DATASETS

AWID2			NSL KDD			UNSW NB15	
Class	Label Count		Class	Label Count		Class	Label Count
Normal	157,749,037		Normal	67,343		Normal	56,000
Impersonation	1,884,378		DoS	45,927		Attack	119,341
Injection	1,530,373		Probe	11,656			
Flooding	1,211,459		R2L	995			
			U2R	52			
Total	**162,375,247**		**Total**	**125,973**		**Total**	**175,341**

TABLE III
RESULTS FOR VICRA ON AWID2, NSL KDD, AND UNSW NB15 ACROSS FOUR DIFFERENT APPROACHES, DECISION TREE, LOGISTIC REGRESSION, MLP AND SVC

		Decision Tree		**Logistic Regression**		**MLP**		**SVC**	
		w VICReg	w/o VICReg	**w VICReg**	w/o VICReg	w VICReg	w/o VICReg	**w VICReg**	w/o VICReg
NSL KDD	Precision ↑	**95.07%**	90.72%	**92.06%**	91.38%	**97.47%**	95.58%	**96.59%**	96.53%
	Recall ↑	**70.79%**	69.36%	**68.23%**	66.33%	69.16%	62.45%	67.70%	65.19%
	F1 Score ↑	**81.16%**	78.62%	**78.37%**	76.86%	**80.91%**	75.54%	**79.60%**	77.82%
	FPR ↓	**0.0485**	0.0937	**0.0777**	0.0827	**0.0237**	0.0382	0.0316	0.0310
	FNR ↓	**0.2921**	0.3064	**0.3177**	0.3367	**0.3084**	0.3755	**0.3230**	0.3481
	Accuracy ↑	**81.29%**	78.52%	**78.57%**	77.27%	**81.42%**	76.98%	**80.25%**	78.85%
UNSW NB15	Precision ↑	67.98%	74.53%	88.87%	90.85%	90.82%	78.36%	89.76%	75.66%
	Recall ↑	85.43%	96.79%	**83.05%**	61.08%	81.34%	92.45%	80.53%	96.67%
	F1 Score ↑	75.71%	84.21%	**85.86%**	73.05%	**85.82%**	84.82%	**84.89%**	84.88%
	FPR ↓	0.1888	0.1552	0.0488	0.0289	0.0386	0.1198	**0.0431**	0.1459
	FNR ↓	0.1457	0.0321	**0.1695**	0.3892	0.1866	0.0755	0.1947	0.0333
	Accuracy ↑	82.50%	88.41%	**91.27%**	85.61%	**91.42%**	89.43%	**90.85%**	89.00%
AWID2	Precision ↑	**89.76%**	87.57%	**73.21%**	67.71%	**73.21%**	57.44%	**64.38%**	58.24%
	Recall ↑	**92.58%**	24.62%	**86.51%**	72.98%	86.51%	92.70%	72.98%	79.69%
	F1 Score ↑	**91.15%**	38.44%	**79.30%**	70.24%	**79.30%**	70.93%	**68.41%**	67.30%
	FPR ↓	0.1077	0.0043	**0.3226**	0.3547	**0.3226**	0.6999	**0.4114**	0.5823
	FNR ↓	**0.0742**	0.7538	**0.1349**	0.2702	0.1349	0.0730	0.2702	0.2031
	Accuracy ↑	90.92%	91.31%	**77.21%**	68.80%	**77.21%**	61.65%	**65.98%**	60.91%

is stored in the embedding cloud as a json object before using it for attack prediction.

As seen in [18] we choose four Machine Learning approaches, Decision Tree, Logistic Regression, Multi Layer Perceptron (MLP) and Support Vector Machines (SVM) for attack prediction. As a way to demonstrate the importance of self supervision and test whether it works, we train attack prediction models both on raw features T and representations Z. For each dataset four such models are trained on raw features and the self supervised representations and the metrics are logged for comparison.

D. Evaluation Metrics

The most commonly deployed performance metrics for validating the performance of ML and DL methods for attack prediction are Accuracy, F1 Score, Precision and Recall.

- Precision is defined as the ratio of total number of correctly predicted packets by total number of predicted packets.
- Recall is defined as the ratio of total number of correctly predicted packets by the sum of correctly predicted packets and the number of missed packets.
- F1-score: Given precision and recall, F-score is defined as the Harmonic mean of precision and recall
- Accuracy is defined as the ratio of the total number of correctly predicted packets to the total number of packets in the dataset.

E. Results

The results are shown in Table III. It can be observed that the models trained on self supervised VICReg embeddings perform better in the given metrics compared to the models trained on raw features. The experiment was done using four different approaches for attack prediction to show that self supervision helps improve prediction metrics regardless of the choice of the model. On an average across the four methods, self supervision improves the accuracy by over 2.48% for NSL KDD, 0.90% for UNSW NB15 and 7.17% for AWID2 than training the models on raw features. It is also to be noted that for datasets like AWID2 with over 97.15% data labeled as normal the improvement in accuracy is significantly higher compared to datasets like NSL KDD and UNSW NB15 where the percentage of data labeled as normal is 53.46% and 15.97% respectively.

IV. Conclusion

We tackle the challenge of Attack Prediction on wireshark features as a Machine Learning (ML) problem by framing it as an anomaly detection task for tabular data. To address this, we introduce a novel technique called VICRA (Variance-Invariance-Covariance Regularization for Attack Prediction). VICRA leverages self-supervision to enhance tabular embeddings using swap noise, resulting in a significant performance boost. By incorporating the underlying data structure and applying regularization during the learning process, VICRA improves prediction accuracy and robustness. We present a comprehensive pipeline for training attack prediction models on wireshark data using VICRA. To evaluate the effectiveness of VICRA, we conduct extensive experiments on popular datasets and compare its performance with existing ML approaches. Our results demonstrate that VICRA achieves substantial accuracy improvements, surpassing traditional ML approaches by over 2.48% for NSL KDD, 0.90% for UNSW NB15, and 7.17% for AWID2 datasets. Overall, VICRA offers a promising solution for enhancing attack prediction capabilities in the context of Wireshark data analysis.

References

[1] N. Paltrinieri, L. Comfort, and G. Reniers, “Learning about risk: Machine learning for risk assessment,” *Safety Science*, vol. 118, pp. 475–486, 2019. doi: https://doi.org/10.1016/j.ssci.2019.06.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753518311184

[2] F. Ullah, H. Naeem, S. Jabbar, S. Khalid, M. A. Latif, F. Al-turjman, and L. Mostarda, “Cyber security threats detection in internet of things using deep learning approach,” *IEEE Access*, vol. 7, pp. 124 379–124 389, 2019. doi: 10.1109/ACCESS.2019.2937347

[3] X. Fang, M. Xu, S. Xu, and P. Zhao, “A deep learning framework for predicting cyber attacks rates,” *EURASIP Journal on Information Security*, vol. 2019, no. 1, p. 5, May 2019. doi: 10.1186/s13635-019-0090-6. [Online]. Available: https://doi.org/10.1186/s13635-019-0090-6

[4] O. A. Akanbi, I. S. Amiri, and E. Fazeldehkordi, “Chapter 1 - introduction,” in *A Machine-Learning Approach to Phishing Detection and Defense*, O. A. Akanbi, I. S. Amiri, and E. Fazeldehkordi, Eds. Boston: Syngress, 2015, pp. 1–8. ISBN 978-0-12-802927-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128029275000010

[5] [Online]. Available: https://www.wireshark.org/

[6] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, “A cookbook of self-supervised learning,” 2023.

[7] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006. doi: 10.1126/science.1127647. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1127647

[8] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, 2006. doi: 10.1109/CVPR.2006.100 pp. 1735–1742.

[9] D. Wang, Y. Zhang, K. Zhang, and L. Wang, “Focalmix: Semi-supervised learning for 3d medical image detection,” 06 2020. doi: 10.1109/CVPR42600.2020.00401 pp. 3950–3959.

[10] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, “Improving language understanding by generative pre-training,” 2018.

[11] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” *arXiv preprint arXiv:2003.04297*, 2020. doi: https://doi.org/10.48550/arXiv.2003.04297

[12] C. Zhang, K. Zhang, C. Zhang, T. X. Pham, C. D. Yoo, and I. S. Kweon, “How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning,” in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=bwq6O4Cwdl

[13] A. Bardes, J. Ponce, and Y. LeCun, “VICReg: Variance-invariance-covariance regularization for self-supervised learning,” in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=xm6YD62D1Ub

[14] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, “A survey on missing data in machine learning,” *Journal of Big Data*, vol. 8, no. 1, p. 140, Oct 2021. doi: 10.1186/s40537-021-00516-9. [Online]. Available: https://doi.org/10.1186/s40537-021-00516-9

[15] W. I. D. Mining, “Data mining: Concepts and techniques,” *Morgan Kaufinann*, vol. 10, pp. 559–569, 2006.

[16] M. Asaduzzaman, M. S. Majib, and M. M. Rahman, “Wi-fi frame classification and feature selection analysis in detecting evil twin attack,” *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 1704–1707, 2020. doi: 10.1109/TENSYMP50017.2020.9231042s

[17] A. S. Menon and K. Anand, “X-abi: Toward parameter-efficient multilingual adapter-based inference for cross-lingual transfer,” in *Data Management, Analytics and Innovation*, N. Sharma, A. Goje, A. Chakrabarti, and A. M. Bruckstein, Eds. Singapore: Springer Nature Singapore, 2023, pp. 303–317.

[18] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, “Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the kdd cup 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009. doi: 10.1109/CISDA.2009.5356528 pp. 1–6.

[20] N. Moustafa, “Designing an online and reliable statistical anomaly detection framework for dealing with large high-speed network traffic,” Ph.D. dissertation, 2017. [Online]. Available: http://hdl.handle.net/1959.4/58748

[21] N. Moustafa and J. Slay, “Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set),” in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015. doi: 10.1109/MilCIS.2015.7348942 pp. 1–6.

[22] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, “NetFlow datasets for machine learning-based network intrusion detection systems,” in *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, 2021, pp. 117–135. [Online]. Available: https://doi.org/10.1007%2F978-3-030-72802-1_9

[23] M. Jahrer, “Porto seguro’s safe driver prediction solution.” [Online]. Available: https://www.kaggle.com/c/porto-seguro-safe-driver-prediction/discussion/44629#250927

DOI: 10.15439/2023F7053
ISSN 2300-5963 ACSIS, Vol. 35

Segmentation Methods Evaluation on Grapevine Leaf Diseases

Szilárd Molnár
Automation Department
Technical University of Cluj-Napoca
Romania
Email: Szilard.Molnar@aut.utcluj.ro

Levente Tamás
Automation Department
Technical University of Cluj-Napoca
Romania
Email: Levente.Tamas@aut.utcluj.ro

***Abstract*—The problem of vine disease detection (VDD) was addressed in a number of research papers, however, a generic solution is not yet available for this task in the community. The region of interest segmentation and object detection tasks are often complementary. A similar situation is encountered in VDD applications as well, in which crop or leaf detection can be done via instance segmentation techniques as well. The focus of this work is to validate the most suitable methods from the main literature on vine leaf segmentation and disease detection on a custom dataset containing leaves both from the laboratory environment and cropped from images in the field. We tested five promising methods including the *Otsu's thresholding*, *Mask R-CNN*, *MobileNet*, *SegNet*, and *Feature Pyramid Network* variants. The results of the comparison are available in Table I summarizing the accuracy and runtime of different methods.**

I. Introduction

VINE DISEASE DETECTION plays an important role in the overall vineyard management allowing the loss reduction and the overcome of the pesticide overuse. The early stage VDD allows the degree of contamination reduction, which implicitly implies a positive economic impact as well.

Remote sensing plays an important role in precision agriculture, allowing the detection of different diseases, estimation of yield, or the computation of the fertilizer rates [29]. With the widespread of Unmanned Aerial Vehicles (UAV) in agriculture as well, close-range remote sensing expanded the range of applications for precision agriculture. The classical image processing algorithms were replaced by deep learning-based variants also for segmentation and object detection.

The most currently available solutions based on convolutional neural networks (CNN) are based on a sliding window approach, which allows the operations on smaller-sized image patches in favor of computational speed. However, for segmentation and detection purposes the whole image view could improve the segmentation boundaries and the accuracy of the detection.

In this paper, we propose to compare the already existing segmentation methods for masking diseased spots on grapevine leaves. For this, we create a mixture of datasets, which contains images from a laboratory environment as well as leaves cropped from images captured in the field from proprietary and publicly available datasets. Our proprietary dataset is captured with a mid-range commercial drone at low-altitude flight using a high-resolution (4K) camera.

The main contribution of this paper is the overview of the existing methods for this particular scenario with close-range remote sensing and the conclusions of the experimental finding in challenging datasets from various vineyards. The paper is organized as follows: the state of the art is presented in Section II, the dataset and method in Section III, and the comparison of the methods in Section IV.

II. State of the Art

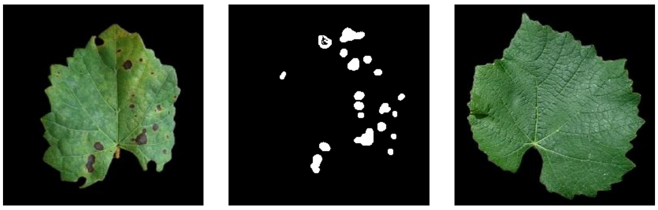
Being an important aspect of precision viticulture, disease detection has a wide range of solutions in the literature. Many researchers seek a new way to stop the spread of diseases as early as possible, to reduce the chances of plant disposal and decreased quality. As far as the domain, multiple approaches exist. The first way to compare these approaches is to specify if the used images are from a laboratory environment or from the field. The approaches focusing on field image processing can be further split into proximal sensing, mainly using a conventional RGB camera, and remote sensing, using a variety of different mediums, such as RGB, multispectral, or hyperspectral. In this section, we provide a brief overview of the existing disease detection methods.

Cruz et al. [8] use transfer learning to detect grapevine yellow disease on single-leaf images while comparing multiple architectures. They experiment with numerous architectures, to conclude that *ResNet-50* [26] has the best *accuracy-to-complexity* ratio.

Similarly, Liu et al. [16] detect grapevine diseases using images of grapevine leaves. The images are either from a laboratory or from the field, however, an image contains only one leaf in both cases. The different leaf sizes are resolved using *dense inception convolutional neural network* from *GoogLeNet* [26] and *asymmetric factorization approach* [27].

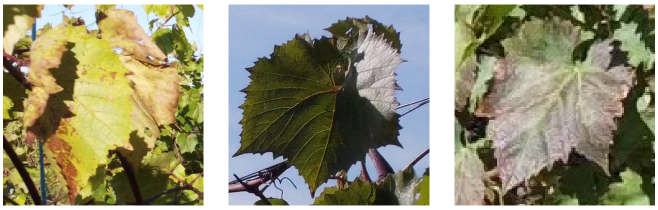
While Gutiérrez et al. [10] capture their images in the field, they manually segment their images, to contain only one leaf, which either represents *downy mildew* and *spider mite*

[1] The authors are thankful for the support of Analog Devices Romania, for the equipment list and Nvidia for graphic cards offered as support to this work. This work was financially supported by the Romanian National Authority for Scientific Research, project nr. PN-III-P2-2.1-PED-2021-3120. The authors are also thankful to the Domus Foundation for their support.



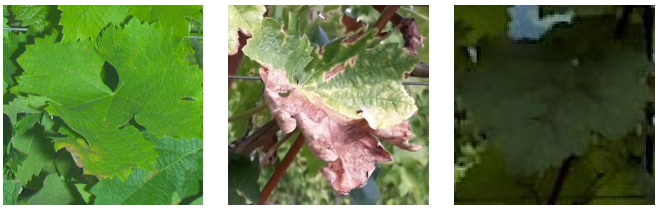
(a) *PV_data-disease* (b) Binary mask (c) *PV_data-healthy*

Fig. 1: Samples from the *PV_data* dataset.



(a) *Cj_data*-disease (b) *Cj_data*-healthy (c) *Ap_data*

Fig. 2: Samples from our dataset.



(a) *Ab_data* (b) *Al_data* (c) *S3_data*

Fig. 3: Samples from datasets: *Ab_data*, *Al_data*, *S3_data*

symptoms. The RGB data is converted into HSV color space. The authors claim this color space change ensures robustness for their *hue thresholding*-based method.

Morellos et al. [20] detect (*esca* and *powdery mildew* using transfer learning. Comparing multiple architectures, *Inception v3* [27] provides the overall best classification accuracy.

Mousavi and Farahani [21] base their work on the mixture of *VGG16* [25] and *Faster R-CNN* [23]. This method captures images of grapevines using a drone, however, the leaves are individually segmented before disease detection and localization.

Although all of these methods detect diseases, they do not create a binary mask to segment the diseased spots on the leaves. One example of this can be the work of Abdelghafour et al. [2], who detect *downy mildew* by capturing the images using a high-power flashlight, similar to Liu et al. [17], which causes an instantaneous segmentation, then converting the images into L*a*b color space. *Local structure tensor* [14] is used to extract geometric features

III. Material and Methods

In this section, we provide a brief description of the used datasets, and methods.

A. Datasets

Data is a highly valuable asset in computer vision. It is used to calibrate and evaluate the model, therefore we need a dataset with high variability. In this section, we describe the used datasets.

As the primary dataset, we use the *PlantVillage* dataset created by Hughes et al. [13], with the codename: *PV_data*. Other versions of this dataset also exist, for example by Cruz et al. [8], however, ultimately we chose the one available on GitHub[1], because in this case the background of the images is already blackened, Figure 1, unlike other versions, where the background is a gray table surface.

Additionally, we create an *infield-dataset*, which contains cropped images from vineyards from various locations. This ensures a wide variety of camera angles and lighting conditions. The first two such datasets are the ones we have access to, each of them located in Romania, courtesy of the University of Agricultural Sciences and Veterinary Medicine. Our main vineyard is located in Cluj-Napoca (codename: *Cj_data*), then less data is from Apoldu de Sus (codename: *Ap_data*). These images are captured using a DJI Mini 2 drone, using the onboard 4K camera.

The next dataset is from Abdelghafour et al. [1] (codename: *Ab_data*). This is a vineyard near Bordeaux. The uniqueness of this dataset is that while the images are captured from a camera mounted on a tractor, the creators use a high-power flashlight, Figure 3a. The result is a highly detailed canopy, with a dark, almost invisible background, all this with consistency, independently from weather or time of day.

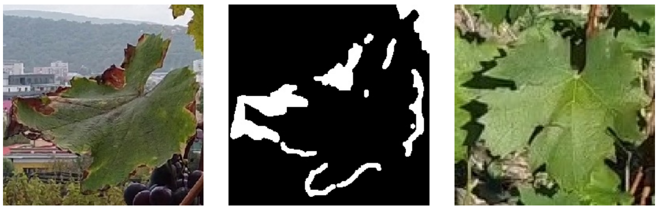
The fifth dataset is created by Alessandrini et al. [4] (codename: *Al_data*), using an Italian vineyard, focusing on leaves with *esca* disease, from different distances and angles, Figure 3b.

The last dataset is created by Casado et al. [7], named S3CavVineyardDataset (codename: *S3_data*), based on a swiss vineyard, Figure 3c. The images are perpendicular to the vines, captured from a tractor.

1) Data Organization: Since the task in this work is disease detection on single-leaf images, we need to have a ground truth mask for each image, which is created by us manually using GIMP [28].

From the dataset, we use 648 images of leaves with some sort of disease (*black rot*, *esca*, and *grapevine yellow*, or *dry leaf*), and 433 images of healthy leaves. Additionally, we crop leaves from other datasets: *Cj_data*, *Al_data*, *Ab_data*, *Ap_data*, and *S3_data*. We call this latter group *infield* images, hence their background is not black, but the real environment, Figure 4. In the *infield* group, 118 diseased leaves, and 100 healthy leaves are included. The task is disease detection, hence in the case of healthy images, the mask is just a black image, meaning that no diseased parts are present. The *PV_data* images are considered as *group1*, with an 80-20 train-test image ratio. The *infield* images are considered *group2* with

[1] https://github.com/shreyansh-kothari/Grapes-Leaf-Disease-detection



(a) Diseased sample. (b) Binary mask. (c) Healthy sample.

Fig. 4: Samples cropping *infield* leaves for disease detection.

a 20-80 train-test image ratio. We plan three test cases. In the first case, we train only on the images from *group1* and test only on *group1*. In the second test, we train only on the images from *group1* and test on *group2*. In the third test, we train on images from *group1* and *group2* and test on *group2*. All of these images are sized 255×255 pixels.

B. Methods

The disease detection task segments a region of interest, for this we choose both neural network-based methods, as well as a classical method to analyze their performance. We choose different architectures, to provide a wider analysis.

1) Mask R-CNN: The first machine learning algorithm that we include is the *Mask R-CNN* [11], which is used for precision viticulture by many researchers, for example, Ghiani et al. [9] and Santos et al. [24]. This is a well-known method, together with its other variants, such as *Faster R-CNN* ([23]). The base for implementing this method can be found at the link[2].

2) MobileNetV3: The idea of using *MobileNetV3* [12] comes from Aghi et al. [3], who use it for canopy segmentation and row detection. The base for implementing this method is available[3]. The main advantage of this model is its simplicity and lightness, making it more suitable for running on embeded devices.

3) Feature Pyramid Network: The Feature Pyramid Network FPN [15] architecture stands as a middle-ground between the lightness of *MobileNetV3* and the accuracy of *Mask R-CNN*. We have seen the FPNs perform decently in surface normal estimation application [18], and canopy segmentation [19], since different support sizes are analogous on some levels to vine leaves. The base for implementing this method can be found at the link[4].

4) SegNet: As the name suggests, *SegNet* [5] is a neural network designed for segmentation. Similarly to *Mask R-CNN*, *SegNet* is also well-known and widely used. Since it is based on an encoder-decoder architecture, the latent space could be helpful in encoding the diseased parts. The base for implementing this method is available[5].

5) Otsu's thresholding: Otsu's thresholding [22] is a dynamic thresholding application, meaning that instead of choosing a static value, and masking the image according to this value, *Otsu's thresholding* analyses each image, and chooses a thresholding value that is more decisive. The main drawback of this method is that despite the RGB color space using 3 channels, *Otsu's thresholding* only works with monochromatic images. One solution would be to mimic the work of Abdelghafour et al. [2], who convert the input signal into HSV color space and apply *Otsu's thresholding* only on the hue channel. However, because RGB does not have a hue value, we conduct a series of tests, to define the best solution. This phase is similar to the training phase in the case of a neural network since we use the training data for estimating an optimal set of parameters, which are later applied to the test data.

We run the thresholding method for each channel, which results in 3 binary masks. Then we combine these masks with each other, achieving a total of 7 masks. Then we do the same thing, but this time inverting the binary masks, since it is possible, that the region of interest might fall into the lower end of the thresholding. We compare the binary masks with the ground truth masks to determine the combination which gives us the best accuracy. Additionally, we create another set of estimation masks, where each individual channel is either inverted or not, depending on the previous results, and then combine these masks to determine the best combination. The ideal combination is noted for each case, and this parameter is used at the time of evaluation. Rather interestingly, from these initial tests, the optimal combination is between the red and blue channels, while the green channel results in slightly worse accuracy. The base for implementing this method is the OpenCV library [6].

IV. Evaluation

In this section, we show the results of the conducted tests. For each task, the accuracy is calculated on the percentage of the pixels correctly estimated, compared to the ground truth. At first sight, this task might seem trivial, because of the small images, yet, the shade difference and the varying spot shapes add a layer of complexity to it. As we described previously, we conduct 3 tests: 1) train on *PV_data* (864 images), test on *PV_data* (217 images); 2) train on *PV_data* (864 images), test on *infield* images (174 images); 3) train on *PV_data* with added infield images (908 images), test on *infield* images (174 images). The last test case is to see how much the accuracy rises by adding 5% more images from the test domain. Accuracy can be seen in Table I, and the range of false positives and false negatives in Table II.

From our tests, we can see that *SegNet* is not suitable for understanding healthy leaves, where it should not extract any region of interest, yet it does, which pulls back the performance by at least 20%. Furthermore, *Otsu's thresholding* is extremely unstable. On the other hand, both of these methods are the fastest. On the first test, *Mask R-CNN* performs the best, although, it is the slowest, while *MobileNetV3* and FPN

[2] https://github.com/matterport/Mask_RCNN

[3] https://github.com/MrD1360/deep_segmentation_vineyards_navigation

[4] https://github.com/molnarszilard/canopy_segmentation

[5] https://github.com/say4n/pytorch-segnet

TABLE I: Accuracy of the various methods for disease segmentation, including the runtime.

Method	Test1[%]	Test2[%]	Test3[%]	Time[s]
Otsu	62.4	46.7	46.7	**0.0004**
Mask R-CNN	**93.64**	61.04	*86.48*	0.160
MobileNetV3	83.76	**81.89**	82.97	0.088
FPN	90.52	50.24	85.57	0.015
SegNet	63.3	59.12	65.1	0.007

TABLE II: The approximate percentage of false positives and false negatives for the various methods for disease segmentation in the three test cases

Method	FP_{t1}	FN_{t1}	FP_{t2}	FN_{t2}	FP_{t3}	FN_{t3}
Otsu	35	2	48	6	48	6
Mask R-CNN	3	3	38	1	10	3
MobileNetV3	0	16	11	8	3	14
FPN	8	1	48	2	4	12
SegNet	35	2	40	0	31	4

perform relatively well, in a much shorter time, which can be important for real-time applications on the field.

Another aspect that we want to check is the amount of increase in accuracy if a few *infield* images are added to the training. In the case of *Otsu's method*, we find virtually no difference, while for the other methods, we see an increase in accuracy between 10-20%, which is significant for such little data. This test is an indication, that it is worth pretraining a model with general images, from various grape leaves, and then training a few epochs with a few additional images from the domain of application. However, we think that in the case of *MobileNetV3* we see an anomaly in the second test because the result is too accurate.

Additionally, we also observed, that on average the number of false positives is higher for *Otsu's method*, *Mask R-CNN*, FPN, and *SegNet*, while for *MobileNetV3* the false negatives are higher. We generally prefer false positives, because in VDD an image flagged as infected should be further investigated by a specialist, therefore, be corrected, however, an infected leaf that is not flagged is unnoticed.

V. Conclusion

In this work, we compared the performance of existing segmentation algorithms from the state of the art for vine disease leaf segmentation and detection. Overall, the CNN-based methods performed well except for *SegNet*, while the *Otsu's thresholding* gave poor results, even if it is the fastest method. We also proved, that adding just a few images from the target domain to the general dataset, yields significantly better performance. While *Mask R-CNN* provides relatively good accuracy, the FPN-based method offers much faster execution without an increased loss of accuracy and an overall smaller memory footprint for the model. The latter aspect is relevant for the embedded implementation of the methods.

For future work, we would like to experiment with different color spaces, as the color spaces can affect the performance of the CNN methods. Although, our raw data is in RGB, a neural network could be capable of optimizing the data in a latent layer better than a simple color conversion.

The disease segmentation can be extrapolated on entire grapevine canopies, which removes the necessity for individual leaf extraction. Additionally, further tests should be done using more variable datasets, including synthetic datasets, and more vine species captured from different angles from different vineyards.

References

[1] Florent Abdelghafour, Barna Keresztes, Aymeric Deshayes, Christian Germain, and Jean-Pierre Da Costa. "An annotated image dataset of downy mildew symptoms on Merlot grape variety". In: *Data in Brief* 37 (2021), page 107250. DOI: https://doi.org/10.1016/j.dib.2021.107250.

[2] Florent Abdelghafour, Barna Keresztes, Christian Germain, and Jean-Pierre Da Costa. "In Field Detection of Downy Mildew Symptoms with Proximal Colour Imaging". In: *Sensors* 20.16 (2020), page 4380. DOI: 10.3390/s20164380.

[3] Diego Aghi, Simone Cerrato, Vittorio Mazzia, and Marcello Chiaberge. "Deep Semantic Segmentation at the Edge for Autonomous Navigation in Vineyard Rows". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - October 1, 2021*. IEEE, 2021, pages 3421–3428. DOI: 10.1109/IROS51168.2021.9635969.

[4] M. Alessandrini, R. Calero Fuentes Rivera, L. Falaschetti, D. Pau, V. Tomaselli, et al. "A grapevine leaves dataset for early detection and classification of esca disease in vineyards through machine learning". In: *Data in Brief* 35 (2021), page 106809. DOI: https://doi.org/10.1016/j.dib.2021.106809.

[5] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pages 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] A. Casado-García, J. Heras, A. Milella, and R. Marani. "Semi-supervised deep learning and low-cost cameras for the semantic segmentation of natural images in viticulture". In: *Precision Agriculture* (2022), pages 1–26. DOI: 10.1007/s11119-022-09929-9.

[8] Alberto Cruz, Yiannis Ampatzidis, Roberto Pierro, Alberto Materazzi, Alessandra Panattoni, et al. "Detection of grapevine yellows symptoms in *Vitis vinifera* L. with artificial intelligence". In: *Computers and Electronics in Agriculture* 157 (2019), pages 63–76. DOI: 10.1016/j.compag.2018.12.028.

[9] Luca Ghiani, Alberto Sassu, Francesca Palumbo, Luca Mercenaro, and Filippo Gambella. "In-Field Automatic Detection of Grape Bunches under a Totally Uncontrolled Environment". In: *Sensors* 21.11 (2021), page 3908. DOI: 10.3390/s21113908.

[10] Salvador Gutiérrez, Inés Hernández, Sara Ceballos, Ignacio Barrio, Ana M. Díez-Navajas, et al. "Deep learning for the differentiation of downy mildew and spider mite in grapevine under field conditions". In: *Computers and Electronics in Agriculture* 182 (2021), page 105991. DOI: 10.1016/j.compag.2021.105991.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN". In: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pages 2980–2988. DOI: 10.1109/ICCV.2017.322.

[12] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, et al. "Searching for MobileNetV3". In: *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pages 1314–1324. DOI: 10.1109/ICCV.2019.00140.

[13] David P. Hughes and Marcel Salathé. "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing". In: *Computing Research Repository* abs/1511.08060 (2015).

[14] Hans Knutsson, Carl-Fredrik Westin, and Mats T. Andersson. "Representing Local Structure Using Tensors II". In: *Image Analysis - 17th Scandinavian Conference, SCIA 2011, Ystad, Sweden, May 2011. Proceedings*. Edited by Anders Heyden and Fredrik Kahl. Volume 6688. Lecture Notes in Computer Science. Springer, 2011, pages 545–556. DOI: 10.1007/978-3-642-21227-7_51.

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, et al. "Feature Pyramid Networks for Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pages 2117–2125. DOI: 10.1109/CVPR.2017.106.

[16] Bin Liu, Zefeng Ding, Liangliang Tian, Dongjian He, Shuqin Li, et al. "Grape Leaf Disease Identification Using Improved Deep Convolutional Neural Networks". In: *Frontiers in Plant Science* 11 (2020), page 1082. DOI: 10.3389/fpls.2020.01082.

[17] Ertai Liu, Kaitlin M. Gold, David Combs, Lance Cadle-Davidson, and Yu Jiang. "Deep semantic segmentation for the quantification of grape foliar diseases in the vineyard". In: *Frontiers in Plant Science* 13 (2022), page 978761. DOI: 10.3389/fpls.2022.978761.

[18] Szilárd Molnár, Benjamin Kelényi, and Levente Tamás. "Feature Pyramid Network Based Efficient Normal Estimation and Filtering for Time-of-Flight Depth Cameras". In: *Sensors* 21.18 (2021), page 6257. DOI: 10.3390/s21186257.

[19] Szilárd Molnár, Barna Keresztes, and Levente Tamás. "Feature Pyramid Network based Proximal Vine Canopy Segmentation". In: *IFAC-PapersOnLine* (2023).

[20] Antonios Morellos, Xanthoula Eirini Pantazi, Charalampos Paraskevas, and Dimitrios Moshou. "Comparison of Deep Neural Networks in Detecting Field Grapevine Diseases Using Transfer Learning". In: *Remote Sensing* 14.18 (2022), page 4648. DOI: 10.3390/rs14184648.

[21] Seyed Amirhossein Mousavi and Gholamreza Farahani. "A Novel Enhanced VGG16 Model to Tackle Grapevine Leaves Diseases With Automatic Method". In: *IEEE Access* 10 (2022), pages 111564–111578. DOI: 10.1109/ACCESS.2022.3215639.

[22] Nobuyuki Otsu. "A Threshold Selection Method from Gray-Level Histograms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 9.1 (1979), pages 62–66. DOI: 10.1109/TSMC.1979.4310076.

[23] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (2017), pages 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

[24] Thiago T. Santos, Leonardo L. de Souza, Andreza A. dos Santos, and Sandra Avila. "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association". In: *Computers and Electronics in Agriculture* 170 (2020), page 105247. DOI: 10.1016/j.compag.2020.105247.

[25] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Edited by Yoshua Bengio and Yann LeCun. Association for Computing Machinery, 2015.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, et al. "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pages 1–9. DOI: 10.1109/CVPR.2015.7298594.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pages 2818–2826. DOI: 10.1109/CVPR.2016.308.

[28] The GIMP Development Team. *GIMP*. Version 2.10.12. June 12, 2019. URL: https://www.gimp.org.

[29] M. Weiss, F. Jacob, and G. Duveiller. "Remote sensing for agricultural applications: A meta-review". In: *Remote Sensing of Environment* 236 (2020), page 111402. DOI: https://doi.org/10.1016/j.rse.2019.111402.

DOI: 10.15439/2023F2483
ISSN 2300-5963 ACSIS, Vol. 35

Automatic speaker's age classification in the Common Voice database

Adam Nowakowski, Włodzimierz Kasprzak
0000-0000-0000-0000, 0000-0002-4840-8860
Warsaw University of Technology, Institute of Control and Computation Eng.
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
Email: wlodzimierz.kasprzak@pw.edu.pl

***Abstract*—An approach to speaker's age classification using deep neural networks is described. Preliminary signal features are extracted, based on mel-frequency cepstral coefficients (MFCC). For gender classification an MLP network appears to be a satisfactory lightweight solution. For the age modelling and classification problem, two network types, ResNet34 and x-vectors, were tested and compared. The impact of signal processing parameters and gender information (both theoretic perfect realistic imperfect) onto the age classification performance was experimentally studied. The neural networks were trained and verified on the large "Common Voice" dataset of English speech recordings.**

I. Introduction

ALTHOUGH work on speaker age recognition dates back to the 1950s [1], this problem is still difficult to solve in practice. Several reasons for this state of affairs can be listed. The speaker's perceived age and his/her chronological age can differ significantly. To train the age classifier well, a very large database of recordings labeled with the age of the speakers will be required. By its nature, the sound of the same speaker's speech depends on many factors independent of age, such as gender, weight, temperament, mood, ethnicity. The first systems with relatively good efficiency in estimating the age of the speaker were developed some 20 years ago [2], [3]. Currently developed systems for estimating the age of the speaker are based on acoustic modeling used in the speaker recognition (speaker identification and verification) systems [4]. The basic classic machine learning methodologies used for this problem are UBM-GMM (Universal Background Model - Gaussian Mixture Model) and "i-vectors" [5], [6], [7]. An early solution based on deep neural networks is the "x-vector" [8], [9], [10] network. Other deep network architectures, such as LSTM [11] or ResNet [12], were also proposed for this purpose.

In this paper, we propose a gender-informed approach to speaker's age classification using the large "Common Voice" database [13] for neural network training and testing. In section 2, this database is introduced in more details. The implemented system SAR (speaker's age recognition) is presented in section 3. Here, we already present results of initial experiments, aimed to find optimal settings of two signal processing parameters. The main experiments and a summary of age classification performance, follows in section 4. At the end, in section 5, we conclude the work with a summary of results.

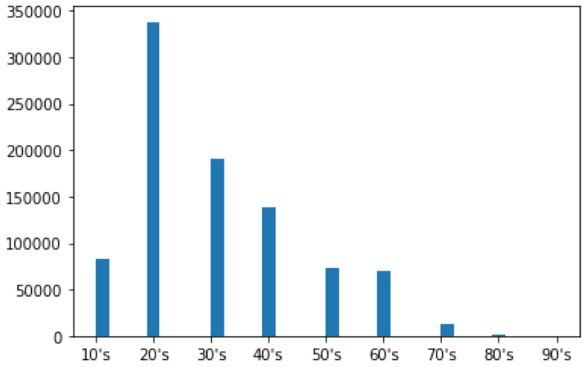


Fig. 1. Statistics of the English language subset of "Common Voice" recordings according to age groups

II. Datasets

A. "Common Voice"

Two large databases of tagged recordings were analyzed for the speaker's age recognition (SAR) system. The first one is "Age-VOX-Celeb" [14], which contains age tags of celebrity recordings, downloaded from "YouTube". The second base is "Common Voice" [13], a Mozilla project, dedicated to record the speech of ordinary Internet user. Everyone can register and record his voice. Other users can listen to the recordings and evaluate their correctness. The tags included speaker's accent, age and gender. The content of the database is growing every day – it contains over 37 million audio files with speech samples in many languages. The "Common Voice" database was chosen here, due to its easy accessibility and a large containment of almost 900,000 recordings in English from over 18,000 people (Fig. 1).

The database contains recordings of voices ranging from teenagers to people in their nineties, but the distribution of age groups is significantly uneven. There are only few participants over the age of seventy. Therefore, for the first series of experiments we decided to limit the age classification to the

This work was supported by "Narodowe Centrum Badań i Rozwoju", Warszawa, Poland, grant No. CYBERSECIDENT/455132/III/NCBR/2020. The publication was funded by Warsaw University of Technology.

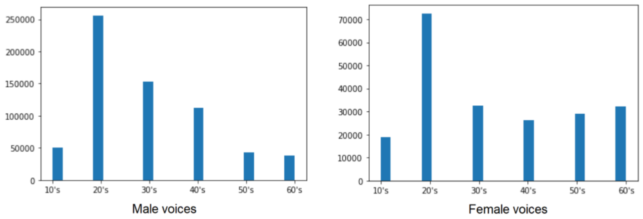


Fig. 2. Distribution of the English subset of "Common Voice" recordings according to speaker's gender and age

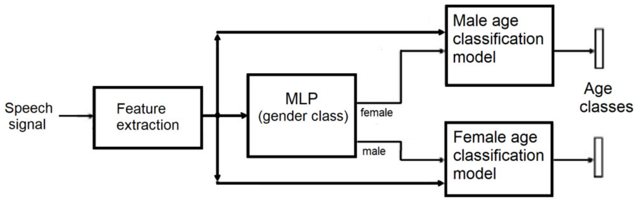


Fig. 3. Structure of SAR

first six age groups. The number of recordings from females is around 200,000 and are much lower than the number of recordings from males, which are over 650,000 (Fig. 2). This leads to our separate treating of age classes for women and men.

This large number of speakers supports extensive speaker classification studies but it also requires extensive computational resources for training and testing neural network models. As a tradeoff solution, we decided to select a training subset of 25,000 recordings of men from each age group and 15,000 recordings of women from each age group. The test set consisted of thousand recordings for each age group, regardless of gender. In total, 12 classes were distinguished, which resulted from taking into account the 6 age groups and 2 gender of the speaker.

III. Solution SAR

The overall structure of the speaker's age recognition (SAR) system is shown in Figure 3. There are three processing stages: feature extraction, gender classification and age group classification.

A. *Feature extraction*

Standard speech features were chosen based on ”mel frequency cepstral coefficents” (MFCC), delivered by popular audio processing library - the LibRosa library [15]. A feature vector is generated for every signal frame. It consists of 70 coefficients, i.e. 3×23 MFCC-based (i.e., MFCCs, delta MFCCs and delta-delta MFCCs) plus 1 energy coefficient.

Nevertheless the standard MFCC-based signal parametrization, some signal segmentation parameters still need to be set optimally. We experimented with different settings of two parameters: the window length (n_fft) and the delay between consecutive windows (hop_ length).

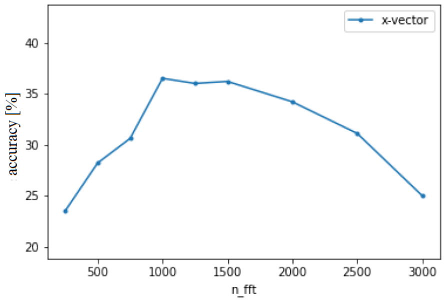


Fig. 4. Experiments with different window lengths

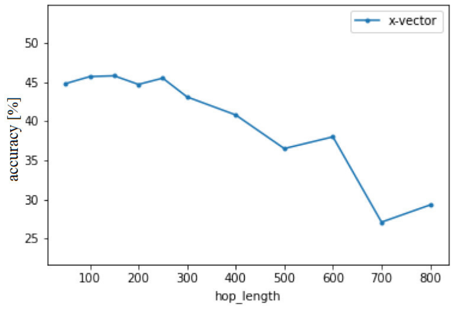


Fig. 5. Experiments with different hop lengths

In the first type of experiments various window lengths, have been set, the x-vector network was trained and its test performance was evaluated. In this test series, the hop_length was set to half of n_fft. The classification performance as a function of window length is shown in Figure 4. The highest quality of results is observed for n_fft in the range between 1000 and 1500 signal samples. (i.e., 45 – 68 ms). Both for smaller and larger windows, the performance is deteriorated. Hence, we selected a window size of 1000 samples, using a FFT of order 1024.

In experiments with different hop lengths, the window size was fixed. The delay parameter was studied in the range from 50 to 800 (Figure 5). As expected, the results show a tendency of increased classification accuracy with decreased hop length. The best performance was stable obtained for delays between 50 and 250 samples (i.e., ca 2 – 11 ms). Among them, we have chosen 250 samples for reasons of computational efficiency. The accuracy of the x-vectors network was 45.5%, only slightly worse than 45.7% and 45.8% for 100 and 150 samples, respectively. Please note, that already with this optimal setting of the hop length parameter, the multi-class classification accuracy of the x-vectors network has been increased by 9% (from 36.5% to 45.5%).

B. *Gender classification*

For the purpose of gender classification, the feature vectors of all the frames of given recording, are combined into a single vector. This constitutes the input of a multi-layer perceptron (MLP) with two hidden layers. The network is trained on samples annotated with gender information. We evaluated the

Layer	Layer context	Total context	Input x output
Conv1D	[t-2,t+2]	5	210x512
Conv1D	{t-2,t,t+2}	9	1536x512
Conv1D	{t-3,t,t+3}	15	1536x512
Conv1D	{t}	15	512x512
Conv1D	{t}	15	512x1500
stat pooling	[0,T]	T	1500Tx3000
linear	{0}	T	3000x512
linear	{0}	T	512x512
softmax	{0}	T	512xN

Fig. 6. The x-vectors network [8]

trained model on a small database, containing recordings of 24 speakers (the RAVDESS database), and its accuracy was 98.7%.

C. x-vectors

X-vectors is a very popular speech recognition network, recently proposed in [8] and already ighly cited in the literature. The architecture of the x-vector network is summarized in Figure 6. The discussed architecture uses a one-dimensional convolution layer - the filter kernel operates along the time domain, while each feature is treated separately. The first five layers operate in this way. Statistical data (mean and deviation) are extracted from the last convolutional layer for each output feature (each channel). This operation is to ensure a fixed length of the output vector, which will later be processed by two dense layers. Finally, there is a softmax layer that maps their outputs to 12 age classes.

D. ResNet34

The second, much more complicated neural network used in our work, is ResNet34 (Figure 7). This architecture also enjoys popularity [16], [17]. It is based on convolutional networks and residual connections. It starts with a convolutional layer with 64 output channels and a 3×3 reception area. Then, there is a ResNet block of 3 layers, each one composed of 2 convolutional layers, with a 3×3 area and 64 outputs. Next, three more ResNet blocks follow, with same kernel size but growing number of outputs. An average pooling layer, a dense layer and softmax complete the network.

We conclude, that the main difference between the two considered architectures is the use of different convolutional layers. X-vectors is using a lightweight 1-D convolution along time axis, whereas ResNet34 applies a true 2-D convolution along time and feature indices.

IV. Results

The technological stack of the system implementation consists of: the Python 3.9.6 language, Jupyter Notebook 8 interactive code editor Librosa 9 library in Python for audio signal processing. PyTorch 10 library for neural network tools and utility libraries, like NumPy and Pandas.

Layer	Layer context
Conv2D	3x3, 64
ResNetBlock1	$\begin{bmatrix}3x3,64\\3x3,64\end{bmatrix}$ x3
ResNetBlock2	$\begin{bmatrix}3x3,128\\3x3,128\end{bmatrix}$ x4
ResNetBlock3	$\begin{bmatrix}3x3,256\\3x3,256\end{bmatrix}$ x6
ResNetBlock4	$\begin{bmatrix}3x3,512\\3x3,512\end{bmatrix}$ x3
stat pooling	avarage pooling
linear	512x12
softmax	-

Fig. 7. The ResNet34 network [16]

	x-vectors	ResNet34
6 classes - no gender information	25,5%	32,7%
12 classes (2 x 6) - single model - gender information	53,8%	67,1%
6 classes x two separate models - gender information	59,6%	70,8%
3 classes x two separate models - gender information	77,3%	86,1%

Fig. 8. Summary of age classification results obtained on the "Common Voice" subset

A. Age classification

In Figure 8, we give results of several training and test series of the two considered network architectures for the class-balanced subset of the „Common Voice" English dataset. Both x-vectors and ResNet34 were applied in the same way - they were trained in 15 epochs with the feature set. The second model is more complex than x-vectors (one needs ca. 10 times more parameters to train) but it shows a better performance than the first one in all experiments.

B. Age classification without gender information

Consider first a 6-class problem, when there is one model for both male and female speakers and no gender information controls the classification process. Both networks perform poorly (25,5% x-vectors, 32,7% ResNet34). If perfect gender information is available to the system, one can expect better results.

C. Age classification with perfect gender information

Consider first a single network model created for a 12-class problem (2 gender $\times 6$ age groups), with the additional information about gender, that allows to select the most likely output from the proper 6-class subset. The ResNet34 shows a better total accuracy of 67,1% in this classification case, compared to an accuracy of 53,8% of the x-vectors.

Now, remember our proposed architecture, shown in Figure 3, with two separate models, trained separately on the two gender samples. Each of the two models is classifying a

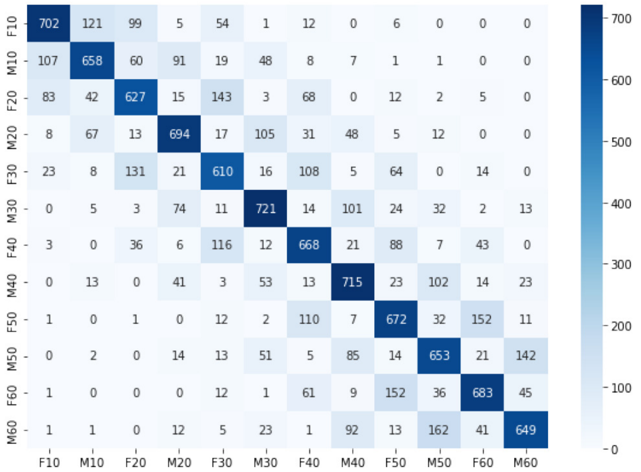


Fig. 9. Confusion matrix of age classification into 12 classes under known gender information, obtained on the "Common Voice" dataset using the ResNet34 network

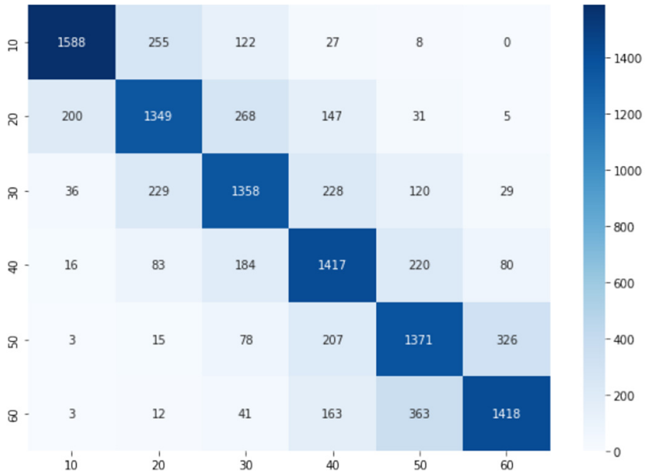


Fig. 10. Confusion matrix of age classification for separate networks, each created for 6 classes, under known gender information (presenting combined results of the two networks), obtained on the "Common Voice" dataset using the ResNet34 network

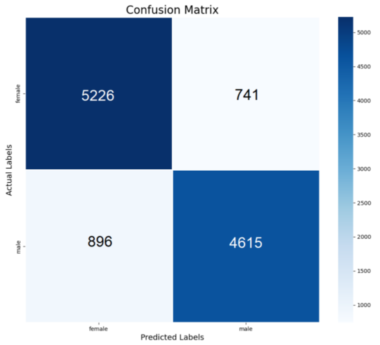


Fig. 11. Confusion matrix of gender classification in the gender-and-age experiment

speaker into one of 6 age classes. In this solution, the performance of both model types is again increased - ResNet34 achieves 70,8% and x-vectors achieves 59,6%.

D. Grouping of age classes

An obvious way further to improve the results is to decrease the number of classes, by grouping difficult-to-distinguish classes. In many applications, the speaker's age classification problem can be reduced to three age classes: teenagers (class "10"), adults (classes "20", "30", "40") and senior adults (classes "50", "60"). In this case, the x-vector-based solution has shown increased accuracy from 59.6% (for 6 classes) to 77.3% (for 3 classes), and the ResNet34 - from 70.8% (6 classes) to 86,1% (3 classes).

E. Confusion matrices

The above results can be justified, when confusion matrices are studied. Such a matrix for results on 12 classes, obtained with ResNet34, is presented in Figure 9. Already a general view leads to the conclusion, that main errors happen between neighbour age groups, as the "errors" (represented by big numbers outside the diagonal) concentrate in the direct neighborhood of the diagonal axis. For example, a misclassification of a teenager as a 60+ senior is practically excluded. Similar observation comes from an error matrix created for a 6-class problem, with gender information, were the results for the same ages classes of man and women are combined (Figure 10).

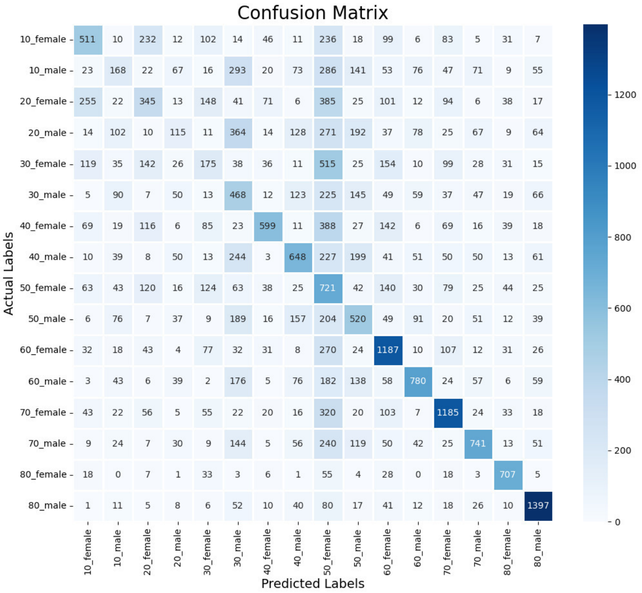
F. Real gender and age classification

In the second series of experiments, we simulated the realistic case of imperfect gender information. We divided 25000 recordings into three sets: a) for training the age model (now for 8 age classes), b) for training the gender model, c) for coupling age and gender into 16 age/gender classes, i.e., class 10_male, 20_male, ... , 80_male, 10_female, 20_female, ... , 80_female. The results on the test subsets, obtained for models created after 30 learning epochs, are presented by two confusion matrices, given in Figures 11 and 12.

The gender success rate (Recall) for females is 87.58% and for males – 83.74%. Thus, the weighted gender-average success rate is 85.74%. The overall age success rate (average Recall) of the combined gender–and–age classifier is 41%, which applies to a 16–classes problem. Interestingly, an increasing age is correlated with growing success rate. The oldest age groups of people 70+ and 80+ are recognized very well.

V. Conclusions

The proposed three-stage approach to speaker's age classification has been trained and tested on a large dataset containing recordings of a high number of speakers. Presented results have shown the positive impact of perfect gender information onto age classification performance. We have also received realistic performance scores under non-perfect gender information. The "heavy-weight" ResNet34 network models



	10_female	10_male	20_female	20_male	30_female	30_male	40_female	40_male	50_female	50_male	60_female	60_male	70_female	70_male	80_female	80_male
10_female	511	10	232	12	102	14	46	11	236	18	99	6	83	5	31	7
10_male	23	168	22	67	16	293	20	73	286	141	53	76	47	71	9	55
20_female	255	22	345	13	148	41	71	6	385	25	101	12	94	6	38	17
20_male	14	102	10	115	11	364	14	128	271	192	37	78	25	67	9	64
30_female	119	35	142	26	175	38	36	11	515	25	154	10	99	28	31	15
30_male	5	90	7	50	13	468	12	123	225	145	49	59	37	47	19	66
40_female	69	19	116	6	85	23	599	11	388	27	142	6	69	16	39	18
40_male	10	39	8	50	13	244	3	648	227	199	41	51	50	50	13	61
50_female	63	43	120	16	124	63	38	25	721	42	140	30	79	25	44	25
50_male	6	76	7	37	9	189	16	157	204	520	49	91	20	51	12	39
60_female	32	18	43	4	77	32	31	8	270	24	1187	10	107	12	31	26
60_male	3	43	6	39	2	176	5	76	182	138	58	780	24	57	6	59
70_female	43	22	56	5	55	22	20	16	320	20	103	7	1185	24	33	18
70_male	9	24	7	30	9	144	5	56	240	119	50	42	25	741	13	51
80_female	18	0	7	1	33	3	6	1	55	4	28	0	18	3	707	5
80_male	1	11	5	8	6	52	10	40	80	17	41	12	18	26	10	1397

Fig. 12. Confusion matrix of age classification into 16 classes (8 age groups × 2 gender) with imperfect, realistic gender information

has clearly outperformed the ”x-vectors” model, which is a popular DNN approach to speaker recognition.

REFERENCES

[1] E. D.Mysak and T. Hanley, ”Aging processes in speech: Pitch and duration characteristics”, *Journal of Gerontology,* vol. 13, 1958, no. 3, pp. 309–313, https://doi.org/10.1093/GERONJ/13.3.309.

[2] N. Minematsu, M. Sekiguchi and K. Hirose, ”Automatic estimation of one's age with his/her speech based upon acoustic modeling techniques of speakers”, *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP),* vol. 1, 2002, pp. 137–140, https://doi.org/10.1109/ICASSP.2002.5743673.

[3] C. Müller, F. Wittig and J. Baus, ”Exploiting speech for recognizing elderly users to respond to their special needs”, *Interspeech, Proc. 8th Eur. Conf. Speech Commun. Technol.,* 2003, pp. 1305–1308, https://doi.org/10.21437/Eurospeech.2003-413.

[4] U. Kamath, J. Liu and J. Whitaker, *Deep Learning for NLP and Speech Recognition,* Springer Nature Switzerland AG, Cham, 2019, https://doi.org/10.1007/978-3-030-14596-5.

[5] P. G. Shivakumar, M. Li, V. Dhandhania and S. S. Narayanan, ”Simplified and supervised i-vector modeling for speaker age regression”, *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2014, pp. 4833–4837, https://doi.org/10.1109/ICASSP.2014.6854520.

[6] M. H. Bahari, M. McLaren, H. Van Hamme and D. A. van Leeuwen, ”Speaker age estimation using i-vectors”, *Engineering Applications of Artificial Intelligence,* vol. 34, 2014, pp. 99–108, https://doi.org/10.1016/j.engappai.2014.05.003.

[7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel and P. Ouellet, ”Front-End Factor Analysis for Speaker Verification”, *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 19, 2011, no. 4, pp. 788–798, https://doi.org/10.1109/TASL.2010.2064307.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, ”X-Vectors: Robust DNN Embeddings for Speaker Recognition”, *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2018, pp. 5329–5333. https://doi.org/10.1109/ICASSP.2018.8461375.

[9] B. Gu, W. Guo, L. Dai and J. Du, ”An Adaptive X-vector Model for Text-independent Speaker Verification”, 2020. https://doi.org/10.48550/ARXIV.2002.06049.

[10] L. Zhou, M.Wang, Y. Qian, H. Luo, H. Li and X. Lin, ”Text-independent Speaker Recognition Based on X-vector”, *2022 7th International Conference on Signal and Image Processing (ICSIP),* 2022, pp. 121–125. https://doi.org/10.1109/ICSIP55141.2022.9887021.

[11] R. Zazo, P. Sankar Nidadavolu, N. Chen, J. Gonzalez-Rodriguez and N. Dehak, ”Age Estimation in Short Speech Utterances Based on LSTM Recurrent Neural Networks”, *IEEE Access*, vol. 6, pp. 22524–22530, 2018, https://doi.org/10.1109/ACCESS.2018.2816163.

[12] A. I. Mansour and S. S. Abu-Naser, ”Classification of Age and Gender Using Resnet - Deep Learning”, *International Journal of Academic Engineering Research (IJAER),* vol. 6, 2022, no. 8, 20–29, https://philpapers.org/rec/MANCOA-4/.

[13] R. Ardila, M. Branson, K. Davis et al., ”Common Voice: A Massively-Multilingual Speech Corpus”, *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020),* 2020, pp. 4218–4222, https://aclanthology.org/2020.lrec-1.520/.

[14] N. Tawara, A. Ogawa, Y. Kitagishi and H. Kamiyama, “Age-VOX-Celeb: Multi-Modal Corpus for Facial and Speech Estimation”, *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2021, pp. 6963–6967, https://doi.org/10.1109/ICASSP39728.2021.9414272.

[15] LibRosa, ”Audio and music processing in Python”, https://librosa.org/

[16] C. Li, X.Ma, B. Jiang et al., *Deep Speaker: an End-to-End Neural Speaker Embedding System*, May 2017. arXiv:1705.02304 [cs.CL] (or arXiv:1705.02304v1 [cs.CL]) https://doi.org/10.48550/arXiv.1705.02304.

[17] S. Hourri and J. Kharroubi, “A deep learning approach for speaker recognition”, *International Journal of Speech Technology*, vol. 23, 2020, pp. 123–131, https://doi.org/10.1007/s10772-019-09665-y.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1093–1100 DOI: 10.15439/2023F5063 ISSN 2300-5963 ACSIS, Vol. 35

Sensitivity Study of a Large-scale Air Pollution Model on the Bulgarian Petascale Supercomputer Discoverer

Tzvetan Ostromsky, Ivan Dimov, Rayna Georgieva
Institute of Information and
Communication Technologies,
Bulgarian Academy of Sciences (IICT-BAS),
Acad. G. Bonchev Str., Block 25-A,
1113 Sofia, Bulgaria
Email: ceco@parallel.bas.bg,
ivdimov@bas.bg, rayna@parallel.bas.bg

Venelin Todorov
Department of Information Modelling,
Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences (IMI-BAS),
Acad. G. Bonchev Str., Block 8,
1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg

***Abstract*—The focus of this study is on the optimal use of high performance computing in the area of environmental security (air pollution transport, in particular). Contemporary mathematical models of air pollution transport should include a fairly large set of chemical and photochemical reactions to be established as a reliable simulation tool. The investigations and the numerical results reported in this paper have been obtained by using a large-scale mathematical model called the Danish Eulerian Model (DEM).**

For optimization of some applications of the Danish Eulerian Model in various important scientific, social and economic areas, it is of great importance to simplify the model as much as possible, preserving the high reliability of its output results. A careful sensitivity analysis is needed in order to decide how to do such simplifications. On the other hand, it is important to analyze the influence of variations of the initial conditions, the boundary conditions, the rates of some chemical reactions, etc. on the model results in order to make right assumptions about the possible simplifications, which could be done. The sensitivity analysis version of the Danish Eulerian Model was created for these purposes. Its complexity is of higher order, a real challenge for the top performance supercomputers nowadays. The sensitivity analysis version of DEM (SA-DEM) has been implemented on the new Bulgarian petascale supercomputer DISCOVERER. It is a part of the European High Performance Computing Joint Undertaking (EuroHPC), which is building a network of 8 powerful supercomputers across the European Union (3 pre-exascale and 5 petascale).

The results of some scalability experiments with SA-DEM on the new Bulgarian petascale supercomputer DISCOVERER are presented here. They are compared with similar experiments performed on the Mare Nostrum III supercomputer at Barcelona Supercomputing Centre – the most powerful supercomputer in Spain by that time, upgraded currently to the pre-exascale Mare Nostrum V, also part of the EuroHPC JU infrastructure.

Keywords: sensitivity analysis, air pollution, numerical model, supercomputer, parallel algorithm, scalability

I. Introduction

Environmental security is rapidly becoming a significant topic of present interest all over the world. It is necessary to carry out many comprehensive scientific studies and to analyze carefully the most important physical and chemical processes during the transport, and transformations under the transport of air pollutants. An effective performance of such complicated procedures requires a joined research and collaboration between experts in the field of environmental modeling, numerical analysis and scientific computing.

The aim of the present work is to propose a new mechanism for investigation the sensitivity of the calculated concentration levels of important pollutants (like nitrogen dioxide NO_2 and especially ozone O_3) due to variation of rates of the involved chemical reactions in a real-life scenario of air pollution transport over Europe with the Unified Danish Eulerian Model (UNI-DEM).

In investigation of various highly complex engineering, physical, environmental, social, and economic systems it is important to measure relations that describe the effect on the output results when the conditions for the input change.

Sensitivity **a**nalysis (SA) is the study of how uncertainty in the output of a model can be apportioned to different sources of uncertainty in the model input [20]. Two classes in sensitivity analysis are considered in the existing literature: local SA and global SA. Local SA studies how some small variations of inputs around a given value change the value of the output. Global SA takes into account all the variation range of the inputs, and apportions the output uncertainty to the uncertainty in the input factors.

In general, several sensitivity analysis techniques are available [20]. Most existing methods for providing SA rely heavily on special assumptions connected to the behavior of the model (such as linearity, monotonicity and additivity of the relationship between input factor and model output). Among quantitative methods, variance-based methods are the most often used [19]. The main idea of these methods is to evaluate how the variance of an input or a group of inputs contributes into the variance of model output.

Computational tasks arising in the treatment of large-scale air pollution models are enormous. It is highly desirable to simplify as much as possible the model, keeping high level of reliability of models' results. Sensitivity analysis is rather helpful in order to decide where and how simplifications can be made. On the other hand, it is important to analyze the influence of variations of the initial conditions, the boundary conditions and/or the chemical rates on the model results in order to make right assumptions about the simplifications which have to be implemented. Such an analysis can give valuable information about the performance of reliable and reasonable simplifications or to identify parameters and mechanisms the accuracy of which should be improved, because the model results are very sensitive to variations of these parameters and mechanisms. Thus, the goal could be: (i) improving the model, (ii) increasing the reliability of the results, and (iii) identifying processes that must be studied more carefully.

The rest of the paper is organized as follows: In Section II the general concept of global sensitivity analysis is introduced in terms of ANOVA high-dimensional model representation. In Section III the Danish Eulerian Model is described, including its high-performance parallel code UNI-DEM and its special sensitivity analysis version SA-DEM. In Section IV numerical results from some scalability experiments with SA-DEM on two of the most powerful supercomputers in Europe are given. Finaly, some conclusins are drawn.

II. Sensitivity Analysis Concept

A. Global Sensitivity Indices

When the sensitivity of the concentrations calculated by UNI-DEM (or any other deterministic mathematical model) is studied, it is convenient to introduce some stochastic variables and equations.

It is assumed that the mathematical model can be presented as a model function

$$\mathrm{u} = f(\mathrm{x}), \qquad \mathrm{x} = (x_1, x_2, \ldots, x_d) \in U^d \equiv [0;1]^d \tag{1}$$

is a vector of input parameters with a joint **p**robability **d**ensity **f**unction (p.d.f.) $p(\mathrm{x}) = p(x_1, \ldots, x_d)$. In general, real problems are characterized by multiple outputs. Here it is assumed that a scalar output is given. It is also assumed that input variables are independent (non-correlated input variables) and the density function $p(\mathrm{x}) = p(x_1, x_2, \ldots, x_d)$ is known, even if x_i are not actually random variables. This implies that the output u is also a random variable, as it is a function of the random vector x, with its own p.d.f.

It is reasonable to introduce an indicator that measures the importance of the influence of a given input parameter onto the output. The main indicator referred to a given input parameter $x_i,\ i = 1, \ldots, d$ (normalised between 0 and 1) is defined as

$$\frac{\mathbf{D}[\mathbf{E}[\mathrm{u}|x_i]]}{\mathbf{D}_\mathrm{u}}, \tag{2}$$

where $\mathbf{D}[\mathbf{E}[\mathrm{u}|x_i]]$ is the variance of the conditional expectation of u with respect to x_i and $\mathbf{D}_\mathrm{u}$ is the total variance according to u. This indicator is named *first-order sensitivity index* by Sobol' [23] or *correlation ratio* by McKay [13]. A brief review of measures of importance used in variance-based methods for sensitivity analysis is given in [3].

The *total sensitivity index* [10] provides a measure of the total effect of a given parameter, including all the possible joint terms between that parameter and all the others. The **t**otal **s**ensitivity **i**ndex (TSI) of input parameter $x_i,\ \ i \in \{1, \ldots, d\}$ is defined in the following way [10], [23]:

$$S_{x_i}^{tot} = S_i + \sum_{l_1 \neq i} S_{il_1} + \sum_{l_1, l_2 \neq i, l_1 < l_2} S_{il_1 l_2} + \ldots + S_{il_1 \ldots l_{d-1}} \tag{3}$$

where S_i is called *the main effect (first-order sensitivity index)* of x_i and $S_{il_1 \ldots l_{j-1}}$ is the j-th order sensitivity index (respectively *two-way interactions* for $j = 2$, *three-way interactions* for $j = 3$ and so on) for parameter x_i $(2 \leq j \leq d)$. The higher-order terms describe the interaction effects between the unknown input parameters $x_{i_1}, \ldots, x_{i_\nu},\ \nu \in \{2, \ldots, d\}$ on the output variance. Usually for practical computations the set of input parameters is classified according their TSI [3]: *very important* if $0.8 < S_{x_i}^{tot}$, *important* if $0.5 < S_{x_i}^{tot} < 0.8$, *unimportant* if $0.3 < S_{x_i}^{tot} < 0.5$, and *irrelevant* if $S_{x_i}^{tot} < 0.3$. In subsection II-B we will show how sensitivity indices $S_{l_1\ \ldots\ l_\nu}$ are defined via the variances of conditional expectations $\mathbf{D}_{l_1} = \mathbf{D}[f_{l_1}(x_{l_1})] = \mathbf{D}[\mathbf{E}(\mathrm{u}|x_{l_1})]$, $\mathbf{D}_{l_1\ \ldots\ l_\nu}, 2 \leq \nu \leq d$ (see, equation (8)). It is often reasonable to assume (see [12], [17]) that relatively small subsets of input variables in high-dimensional models have the main impact on the output. The high dimensional sums can be neglected when many practical problems are studied. This means that one can use low-order indices preferably, but should be able to control the contribution of higher-order terms.

B. The Sobol' Approach

The Sobol' method is one of the most often used variance-based methods. To our best knowledge the Sobol' sensitivity measure [23] was first published in [22]. An important advantage of this method is that it allows to compute not only the first-order indices, but also indices of a higher-order in a way similar to the computation of the main effects. The total sensitivity index can be calculated with just one Monte Carlo integral per factor.

The method for global SA applied here is based on a decomposition of an integrable model function f in the d-dimensional factor space into terms of increasing dimensionality:

$$f(\mathrm{x}) = f_0 + \sum_{\nu=1}^{d} \sum_{l_1 < \ldots < l_\nu} f_{l_1 \ldots l_\nu}(x_{l_1}, x_{l_2}, \ldots, x_{l_\nu}) \tag{4}$$

where f_0 is a constant. The total number of summands in equation (4) is 2^d (see [25]) and, in general, this so called high dimensional model representation [23] is non-unique. But, if each term is chosen to satisfy the following condition

$$\int_0^1 f_{l_1 \ldots l_\nu}(x_{l_1}, x_{l_2}, \ldots, x_{l_\nu}) \underset{\cdot}{\mathrm{x}}_{l_k} = 0, \quad 1 \leq k \leq \nu \leq d \tag{5}$$

then (4) is unique. The representation (4) is called **ANOVA-representation** of the model function $f(\mathrm{x})$ [24]. Here and hereafter the variables of integration are markeed by a dot below in the integration formulae.

The functional decomposition of $[0;1]^d$ ANOVA (meaning: **ANalysis Of VAriance**) -representation has been studied by many authors [2], [9], [21], [26]. Sobol' has proven [22] that the decomposition (4) is unique on the assumption (5) and the functions of the right-hand side can be defined in a unique way by multidimensional integrals [24]:

- $f_0 = \displaystyle\int_{U^d} f(\mathrm{x})\underset{\cdot}{\mathrm{x}}$;
- $f_{l_1}(x_{l_1}) = \displaystyle\int_{U^{d-1}} f(\mathrm{x}) \prod_{k\neq l_1} \underset{\cdot}{\mathrm{x}}_k - f_0,\ l_1 \in \{1,2,\ldots,d\}$;
- $f_{l_1 l_2}(x_{l_1},x_{l_2}) = \displaystyle\int_{U^{d-2}} f(\mathrm{x}) \prod_{k\neq l_1,l_2} \underset{\cdot}{\mathrm{x}}_k - f_0 - f_{l_1}(x_{l_1}) - f_{l_2}(x_{l_2}), l_1, l_2 \in \{1,\ldots,d\}$.

An additional essential property of the terms in the ANOVA-representation is their mutual orthogonality:

$$\int_{U^d} f_{i_1\ldots i_\mu}(\mathrm{x}) f_{j_1\ldots j_\nu}(\mathrm{x})\ \underset{\cdot}{\mathrm{x}} = 0,$$
$$(i_1,\ldots,i_\mu) \neq (j_1,\ldots,j_\nu), \quad \mu,\nu \in \{1,\ldots,d\}.$$

It follows from the assumption that the above subsets of indices differ from one another at least one element and the corresponding integral vanishes for this index due to (5).

The quantities

$$\mathbf{D} = \int_{U^d} f^2(\mathrm{x})\underset{\cdot}{\mathrm{x}} - f_0^2 \qquad (6)$$
$$\mathbf{D}_{l_1\ \ldots\ l_\nu} = \int f^2_{l_1\ \ldots\ l_\nu}(\mathrm{x})\underset{\cdot}{\mathrm{x}}_{l_1}\cdots\underset{\cdot}{\mathrm{x}}_{l_\nu}$$

are called variances (total and partial variances, respectively) and have been obtained after squaring and integrating over U^d the equality (4) on the assumption that $f(\mathrm{x})$ is a square integrable function (thus all terms in (4) are also square integrable functions). Therefore, the total variance of the model output is partitioned into partial variances [22] in the analogous way as the model function, that is the unique ANOVA-decomposition:

$$\mathbf{D} = \sum_{\nu=1}^{d} \sum_{l_1<\ldots<l_\nu} \mathbf{D}_{l_1\ldots l_\nu}. \qquad (7)$$

It is obvious that the use of terms of probability theory is based on the following interpretation: in general, the input parameters are random variables distributed in U^d that defines $f_{l_1\ \ldots\ l_\nu}(x_{l_1}, x_{l_2},\ldots,x_{l_\nu})$ also as random variables with variances (6). For example f_{l_1} is presented by a conditional expectation:

$$f_{l_1}(x_{l_1}) = \mathbf{E}(\mathrm{u}|x_{l_1}) - f_0$$

and respectively

$$\mathbf{D}_{l_1} = \mathbf{D}[f_{l_1}(x_{l_1})] = \mathbf{D}[\mathbf{E}(\mathrm{u}|x_{l_1})].$$

Based on the above assumptions about the model function and the output variance, the following quantities

$$S_{l_1\ \ldots\ l_\nu} = \frac{\mathbf{D}_{l_1\ \ldots\ l_\nu}}{\mathbf{D}}, \quad \nu \in \{1,\ldots,d\} \qquad (8)$$

are called Sobol' global sensitivity indices [22], [24]. This formula coincides for $\nu = 1$ with (2) and the so defined measures correspond to the main effect of input parameters as well as the interactions effect. Using the definition of these measures as ratios of variances and dividing (7) by $\mathbf{D}$, it is easy to show that the following properties hold for the Sobol' global sensitivity indices: $S_{l_1\ \ldots\ l_\nu} \geq 0$, and

$$\sum_{\nu=1}^{d} \sum_{l_1<\ldots<l_\nu}^{d} S_{l_1\ \ldots\ l_\nu} = 1. \qquad (9)$$

Based on the results discussed above it is clear that the mathematical treatment of the problem of providing global sensitivity analysis consists in evaluating total sensitivity indices (3) and in particular Sobol' global sensitivity indices (8) of corresponding order. And that leads to computing of multidimensional integrals: $I = \int_\Omega g(\mathrm{x})p(\mathrm{x})\,\underset{\cdot}{\mathrm{x}},\ \Omega \subset \mathbf{R}^d$, where $g(\mathrm{x})$ is a square integrable function in Ω and $p(\mathrm{x}) \geq 0$ is a probability density function, such that $\int_\Omega p(\mathrm{x})\,\underset{\cdot}{\mathrm{x}} = 1$. This means that in general case one needs to compute 2^d integrals of type (6) to obtain $S^{tot}_{x_i}$. As we discussed earlier the basic assumption underlying representation (4) is that the basic features of the model functions (1) describing typical real-life problems can be presented by low-order subsets of input variables [12], [17], that are constants, terms of first and second order. Thus, the high-dimensional sums (referred to higher-order interactions effects) in (4) can normally be neglected. Therefore, based on this assumption, one can assume that the dimension of the initial problem can be reduced.

Nevertheless, the calculating of the integrals defined by formulas (6) requires integration of different integrands that is not effective according to the computational cost. The procedure for computing global sensitivity indices measuring effect (main or otherwise) of the input parameters that is overcoming this disadvantage has been proposed by Sobol' [24]. Consider an arbitrary set of m variables ($1 \leq m \leq d-1$): $\mathrm{y} = (x_{k_1},\ldots,x_{k_m}),\ 1 \leq k_1 < \ldots < k_m \leq d,$ and let z be the set of $d-m$ complementary variables. Thus $\mathrm{x} = (\mathrm{y},\mathrm{z})$. Let $K = (k_1,\ldots,k_m)$.

The variances corresponding to the subsets y and z can be defined as

$$\mathbf{D}_\mathrm{y} = \sum_{n=1}^{m} \sum_{(i_1<\ldots<i_n)\in K} \mathbf{D}_{i_1\ \ldots\ i_n}, \qquad (10)$$
$$\mathbf{D}_\mathrm{z} = \sum_{n=1}^{d-m} \sum_{(j_1<\ldots<j_n)\in \bar{K}} \mathbf{D}_{j_1\ \ldots\ j_n},$$

where the complement of the subset K in the set of all parameter indices is denoted by $\bar{K}$. The first sum in (10) is extended over all subsets $(i_1,\ldots,i_n)$, where all indices $i_1,\ldots,i_n$ belong to K. Then the total variance corresponding to the subset y is $\mathbf{D}^{tot}_\mathrm{y} = \mathbf{D} - \mathbf{D}_\mathrm{z}$ and it is extended over all subsets $(i_1,\ldots,i_\nu),\ 1 \leq \nu \leq d$, where at least one $i_l \in K,\ 1 \leq l \leq \nu$.

The procedure for computation of global sensitivity indices is based on the following representation of the variance $\mathbf{D}_\mathrm{y} =$

$\int f(\mathrm{x})\ f(\mathrm{y},\mathrm{z}')\d{\mathrm{x}}\d{\mathrm{z}}' - f_0^2$ (see [24]). The last equality allows to construct a Monte Carlo algorithm for evaluating $f_0, \mathbf{D}$ and $\mathbf{D}_{\mathrm{y}}$, where $\xi = (\eta, \zeta)$:

$$\frac{1}{N}\sum_{j=1}^{N} f(\xi_j) \xrightarrow{P} f_0,$$
$$\frac{1}{N}\sum_{j=1}^{N} f(\xi_j)\ f(\eta_j, \zeta_j') \xrightarrow{P} \mathbf{D}_{\mathrm{y}} + f_0^2,$$
$$\frac{1}{N}\sum_{j=1}^{N} f^2(\xi_j) \xrightarrow{P} \mathbf{D} + f_0^2,$$
$$\tfrac{1}{N}\textstyle\sum_{j=1}^{N}\ f(\xi_j)\ f(\eta_j', \zeta_j) \xrightarrow{P} \mathbf{D}_{\mathrm{z}} + f_0^2.$$

For example, for $m = 1, \mathrm{y} = \{x_{l_1}\}, l_1 \in \{1, \ldots, d\}$ and $\mathrm{z} = \{1, \ldots, d\}\backslash l_1$:

$$S_{l_1} = S_{(l_1)} = \mathbf{D}_{(l_1)}/\mathbf{D},\ S_{l_1}^{tot} = \mathbf{D}_{l_1}^{tot}/\mathbf{D} = 1 - S_{\mathrm{z}}.$$

It is important to estimate the computational cost for computing the sensitivity indices in order to be able to compare this approach with other existing approaches. The computational cost of estimating all first-order ($m = 1$) and total sensitivity indices via the scheme proposed by Sobol' can be defined as $N(2d + 1)$ model function evaluations (N model runs for f_0, dN model runs for the first-order terms, and dN model runs for the total effect terms), where N is the sample size and d is the number of input parameters. It should be noted that the most frequently used variance-based methods as Sobol' method and FAST (Fourier Amplitude Sensitivity Test) (and their improved versions) have a computational cost proportional to dN of estimating all main and total effects of input parameters (see [18]).

The computing of higher-order interactions effect can be performed by an iterative process. For example,

$$S_{(l_1 l_2)} = \mathbf{D}_{(l_1 l_2)}/\mathbf{D} = S_{l_1} + S_{l_2} + S_{l_1 l_2},$$

and $S_{l_1 l_2}$ can be obtained assuming that the corresponding first-order sensitivity indices have already been computed.

III. Description and Parallel Implementations of the Danish Eulerian Model (DEM)

DEM is a powerful large scale air pollution model, with more than 30-year development history [28], [15], [16], [29]. Over the years it was successfully applied in different long-term environmental studies in various areas. processes in the atmosphere should be taken into account, which are mathematically represented by a complex PDE system. To simplify it a proper splitting procedure is applied. As a result the initial system is replaced by several simpler systems (submodels), connected with the main physical and chemical processes. These systems should be calculated in a large spatial domain, as the pollutants migrate quickly on long distances, driven by the atmosphere dynamics, especially on high altitude. Here they are exposed to temperature, light and other condition changes in extremely wide range, so does the speed of most chemical reactions. One of the major sources of difficulty is the dynamics of the atmospheric processes, which require small time-step to be used (at least, for the chemistry submodel) in order to get a stable numerical solution of the corresponding system. All this makes the treatment of large-scale air pollution models a tuff and heavy computational task. It has always been a serious challenge, even for the fastest and most powerful state-of-the-art supercomputers. [7], [29].

The Danish Eulerian Model (DEM) [27], [28] is mathematically represented by the following system of partial differential equations:

$$\begin{aligned}\frac{\partial c_s}{\partial t} = &-\frac{\partial(uc_s)}{\partial x} - \frac{\partial(vc_s)}{\partial y} - \frac{\partial(wc_s)}{\partial z} + \\ &+\frac{\partial}{\partial x}\left(K_x\frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y\frac{\partial c_s}{\partial y}\right) + \\ &+\frac{\partial}{\partial z}\left(K_z\frac{\partial c_s}{\partial z}\right) + \\ &+E_s + Q_s(c_1, c_2, \ldots, c_q) - (k_{1s} + k_{2s})c_s\ ; \\ &\qquad s = 1, 2, \ldots, q\ ;\end{aligned} \tag{11}$$

where the following notation is used:

q - number of equations = number of chemical species,
c_s - concentrations of the chemical species considered,
u, v, w - components of the wind along the coordinate axes,
K_x, K_y, K_z - diffusion coefficients,
E_s - emissions in the space domain,
k_{1s}, k_{2s} - coefficients of dry and wet deposition respectively ($s = 1, \ldots, q$),
$Q_s(c_1, c_2, \ldots, c_q)$ - non-linear functions that describe the chemical reactions between the species.

A. Splitting into submodels and domain decomposition

The above rather complex system is split into three subsystems (submodels), according to the major physical and chemical processes as well as the numerical methods applied in their solution (marked by different colors in the right-hand-side of the system). These are (i) the horizontal advection and diffusion, (ii) chemistry, emissions and deposition and (iii) vertical exchange submodels, respectively. The discretization of the spatial derivatives in the right-hand-sides of these submodels results in forming three large systems of ordinary differential equations.

Chemical reactions play a significant role in the model. Moreover, both non-linearity and stiffness of the equations are mainly introduced by the chemistry (see [30]). On the other hand, this is one of the models of atmospheric chemistry, where the chemical processes are described with great detail in a very accurate way. The chemical scheme used in the model is the well-known condensed CBM-IV (Carbon Bond Mechanism; the scheme was proposed in [8], but some enhancements have been obtained in [28] by adding several reactions for handling the ammonia-ammonium transformations in the atmosphere). It includes 35 pollutants and 116 chemical reactions, where 69 are time dependent and the rest 47 are time-independent. The scheme is suitable and adequate to study cases of high concentrations of chemical species.

Another crucial point on the way towards efficient numerical solution of the sub-models is the domain decomposition technique. This is a natural way to achieve distributed memory parallelization of any numerical problem over a large spatial domain. In some cases, however, like the advection-diffusion equations in particular, there is always certain overhead due to the boundary conditions treatment. Minimizing this overhead is a key point towards efficient optimization. On the other hand, optimization should not restrict the portability of the parallel implementation, as the intensive development in the computer technology inevitably leads to regular updates or complete replacement of the outdated hardware. Standard parallel programming tools as MPI and OpenMP (for distributed / shared memory models) are used in order to preserve portability of the code. An important issue towards efficient parallel optimization is also the load-balance. Sometimes the MPI barriers, used to force synchronization between the processes in data transfer commands, do not allow good load-balance. This obstacle can be avoided to some extent by using non-blocking communication routines from the MPI standard library.

More details about the numerical methods, applied to solve these systems, can be found in [1], [11], [28].

B. Parallelization strategy

The MPI standard library is used as a main parallelization tool. The MPI (Message Passing Interface) was initially developed as a standard communication library for distributed memory computers. Later, proving to be efficient, portable and easy to use, it became one of the most popular parallelization tools for application programming. Now it can be used on much wider class of parallel systems, including shared-memory computers and clustered systems (each node of the cluster being a separate shared-memory machine). Thus it provides high level of portability of the code.

In the case of DEM, MPI parallelization is based on the space domain partitioning [15], [16]. The space domain is divided into sub-domains (the number of the sub-domains is equal to the number of MPI tasks). Each MPI task works on its own sub-domain. On each time step there is no data dependency between the MPI tasks on both the chemistry and the vertical exchange stages. This is not so with the advection-diffusion stage. Spatial grid partitioning between the MPI tasks requires overlapping of the inner boundaries and exchange of certain boundary values on the neighboring subgrids for proper treatment of the boundary conditions. The subdomains are usually too large to fit into the fastest cache memory of the corresponding CPU. In order to achieve good data locality, the smaller calculation tasks are grouped in chunks (if appropriate) for more efficient cache utilization. An input parameter CHUNKSIZE is provided, which controls the amount of short-term reusable data in order to reduce the transfer between the cache and the main (slower access) memory. It should be tuned with respect to the cache size of the target machine.

More detailed description of the main computational stages of DEM and the parallelization techniques used in each of them can be found in [1], [4], [15], [16], [28], [29], [30].

IV. Numerical experiments with SA-DEM on the petascale supercomputer IBM MareNostrum III in Barcelona, Spain and Discoverer EuroHPC in Sofia, Bulgaria

Results of scalability experiments with the 2-D fine-resolution grid version of SA-DEM on two of the most powerful supercomputers in Europe are shown in Tables I and II in this section. Some values of the user-defined parameters of SA-DEM used in the experiments on both machines are as follows:

- Grid-version: $(480 \times 480 \times 1)$;
- Time period of modelling: 1 year;
- Time step: 90 sec. (both in advection and chemistry stages);
- Cache utilization parameter: `NSIZE` = 32 .

A. Numerical experiments on the IBM MareNostrum III supercomputer at BSC - Barcelona, Spain

Characteristics of the system IBM MareNostrum III

- 3028 nodes IBM dx360 M4, 16-core, 32 GB RAM per node;
- 48488 cores in total (Intel SandyBridge-EP E5-2670, 2600 MHz);
- Total RAM $>$ 94 TB; Disk storage 1,9 PB;
- Interconnection networks: Infiniband / Gigabit Ethernet;
- Theoretical peak performance $\sim$ 1 PFLOPS.

B. Numerical experiments on the EuroHPC JU supercomputer DISCOVERER in Bulgaria

Below are described some of the most important technical characteristics of the DISCOVERER supercomputer, installed 2 years ago in Sofia Tech Park. by Atos company. The machine is part of a new network of 8 powerful supercomputers in the the European Union, build up and governed by the European High Performance Computing Joint Undertaking (EuroHPC JU).

System properties:

- System model type: ATOS BullSequana XH2000;
- The system consists of 12 racks, 376 blades, 1128 nodes (18 of them – Fat nodes), 2 login nodes (for public access to the system);
- There are 2256 processors and 144384 cores in total;
- Total RAM: 302592 GB (128 GB per node); total disk storage: $\sim$ 12 PB;
- Interconnection: Dragonfly+ with 200 Gbps (IB HDR) bandwidth per link;
- Sustained max. performance: 4.518 PFLOPS (on Linpack standard benchmark tests);

TABLE I
TIME (T) IN SECONDS, SPEED-UP (SP) AND PARALLEL EFFICIENCY (E) OF SA-DEM (FINEST GRID) ON THE SPANISH SUPERCOMPUTER IBM MARENOSTRUM III AT BSC, BARCELONA

Time and speed-up of SA-DEM (MPI + OpenMP) on IBM MareNostrum III
$(480 \times 480 \times 1)$ **grid, 35 species, CHUNKSIZE=32**

# CPU	#	Advection			Chemistry			TOTAL		
/threads	NODES	T [s]	(*Sp*)	*E* [%]	T [s]	(*Sp*)	*E* [%]	T [s]	(*Sp*)	*E* [%]
10	1	80606	(*10*)	100 %	73426	(*10*)	100 %	165006	(*10*)	100 %
40	3	18760	(*43*)	108 %	15938	(*46*)	115 %	38775	(*43*)	106 %
80	5	9837	(*82*)	103 %	8551	(*86*)	107 %	21728	(*76*)	95 %
160	10	5130	(*157*)	98 %	4332	(*169*)	106 %	12525	(*132*)	82 %
320	20	2870	(*281*)	88 %	2292	(*320*)	100 %	8097	(*204*)	64 %
640	40	1511	(*534*)	83 %	1192	(*616*)	96 %	5299	(*311*)	49 %
960	60	1206	(*669*)	70 %	790	(*929*)	97 %	4034	(*409*)	43 %
1600	100	869	(*927*)	58 %	486	(*1510*)	94 %	3269	(*505*)	32 %
2400 /2	150	728	(*1107*)	46 %	407	(*1804*)	75 %	2415	(*683*)	28 %
4800 /4	300	265	(*3040*)	63 %	156	(*4712*)	98 %	1482	(*1113*)	23 %
15360/16	960	105	(*7698*)	50 %	48	(*15170*)	99 %	509	(*3239*)	21 %

TABLE II
TIME (T) IN SECONDS, SPEED-UP (*Sp*) AND THE TOTAL EFFICIENCY (E) OF SA-DEM ON THE EUROHPC JU SUPERCOMPUTER DISCOVERER IN SOFIA, BULGARIA

Time (T) in seconds and speed-up (*Sp*)
of SA-DEM on DISCOVERER
$(480 \times 480 \times 1)$ grid, 35 species, CHUNKSIZE=32

NP	#	Advection		Chemistry		TOTAL			
(MPI)	NODES	T [s]	(*Sp*)	T [s]	(*Sp*)	T [s]	(*Sp*)	*E* [%]	
10	1	72142	(*10.0*)	64726	(*10.0*)	146335	(*10*)	100 %	%
20	2	36175	(*19.9*)	30027	(*21.6*)	71129	(*21*)	103 %	%
40	3	18297	(*39.4*)	14295	(*45.3*)	36619	(*40*)	100 %	%
80	5	9523	(*75.8*)	7839	(*82.6*)	20383	(*72*)	90 %	%
160	10	4781	(*150.9*)	3925	(*164*)	11769	(*124*)	78 %	%
320	20	2525	(*285.7*)	2037	(*317*)	6861	(*213*)	67 %	%
640	40	1332	(*541.7*)	1034	(*626*)	4852	(*302*)	47 %	%
960	60	1017	(*709.7*)	697	(*929*)	3472	(*421*)	44 %	%
1600	100	787	(*916.7*)	463	(*1398*)	2822	(*519*)	32 %	%

– Theoretical peak performance: 6 PFLOPS, ratio (max to peak): 0.753;
– TOP500 ranking: # 91 in the world, # 27 in EU by the time of instalation (June 2021).

Computing node design:

– CPU type: AMD EPYC 7H12 (code name Rome), 64-core, frequency 2.6 GHz, power consumption 280W;
– CPU sockets per node: 2, CPU Cores per node: 128;
– Main memory per node : 256GB (Each of the 18x Fat nodes has 1024GB Memory);
– Memory type and frequency : 16GB DDR4 RDIMM 3200MT/s DR, (The fat nodes are equipped with 64-GB DDR4 RDIMM 3200MT/s DR);
– Node DP TeraFlop/s peak performance: 5.325TFlops;
– Node sustained performance on Linpack tests: 3.940TFlops;
– DP ratio TeraFlop/s – peak vs Linpack: 0,74 ;
– Linpack node power consumption: 665.1 W per 256 GB compute node; 747.0 W per Fat compute node (Cooling subsystem power consumption excluded);
– Number and bandwidth of network interfaces : 1x 200Gbps HDR.

High performance network properties:

– Interconnection family: IB HDR;
– Interconnection bandwidth per link: 200 Gbps (IB HDR);
– Expected latency (worst case for a 1 kB message): 520ns;
– Interconnection topology: Dragonfly+ ;
– Number of compute nodes per isle (2 Racks): 192;
– Blocking factor within isle : 2:1;
– Number of links to I/O partition: 120;
– Performance: 40 X HDR 200Gb/s ports in a 1U switch, 80 X HDR100 100Gb/s ports in a 1U switch;
– Aggregate switch throughput: 16 Tb/s;
– Up to 15.8 billion messages-per-second;
– Switch latency: 130 ns.

Management network properties:

– Network family: Ethernet;
– Network bandwidth : 10GbE/1GbE.

Storage system and I/O capacity:

– Total net capacity of data: 2031.89 TB;
– Total net capacity for metadata storage + home/apps – 15.25 TB; User home folders and application binaries will be in Data drive; The useable capacity of the filesystem will be 1 to 2
– Aggregated performance: 20 GB/s;
– Number of data modules: 164 HDD;

– Number of metadata modules: 11 SSD;
– Data module details: Net capacity provided (PB): 2.03 PB, Performance provided: 20 GB/s, Number and type of storage elements: 164 HDD + 11 SSD, Size per storage element: 6 TB + 1.92 TB, CPU Cores per server: 10, Main memory per server: 150 GB, Memory type and frequency: DDR4 2666 MT/s, Number and bandwidth interfaces to control data network: 4 x GigE RJ45 for OS access and hardware management, Number and bandwidth interfaces to bulk data network (RDMA): 4x HDR100 IB / 100GbE ports (same ports as for metadata).

V. Conclusions

Sensitivity analysis and particularly the results, reported in this work, have an important twofold role: for mathematical models verification and/or improvement, and/or on the other hand, for a reliable interpretation of experts of main effect, interaction and higher-order interaction effect of input parameters on model output. Variance-based analysis is an useful tool for an advanced investigation of relationships between model parameters, output results and internal mechanisms regulating the system under consideration. Specifying the most important chemical reactions for the model output the specialists from various applied fields (chemistry, physics) may obtain valuable information for an improvement of the model and thus it will lead to an increase of reliability and robustness of predictions.

The results of numerical experiments performed show that:

- The parallel MPI implementation of SA-DEM is well balanced, portable and runs efficiently on some of the most powerful supercomputers in Europe, including the Bulgarian Petascale supercomputer Discoverer, part of the EuroHPC JU network.
- The efficiency and speed-up is higher in the computationally-intensive stages. In particular, the chemistry stage (which does not need any communication between the tasks) has almost linear overall speed-up. The advection stage scales pretty well too, taking into account that there is some unavoidable computational overhead due to overlapping boundaries of the partitioning.
- The time for the computationally-intensive stages is additionally reduced in relation with the number of threads in the hybrid MPI-OpenMP code with the OpenMP lower level of parallelism switched on, which can be exploited on core level within a node.
- Further attention should be payed on the optimization of the I/O processes in order to reduce the slowdown of the execution on large number of nodes.

Acknowledgments

This research was supported in part by the Bulgarian NSF project "Efficient methods for modelling, optimization and decision making" (contract # KP-06-N52/5), and by the European High-Performance Computing Joint Undertaking via the Discoverer Fast Track & Benchmark Access project "Installing, benchmarking, performance and scalability analysis of the UNI-DEM package for air pollution modelling".

References

[1] V. Alexandrov, A. Sameh, Y. Siddique and Z. Zlatev, Numerical integration of chemical ODE problems arising in air pollution models, *Env. Modeling and Assessment*, 2 (1997) 365–377.

[2] G. Archer, A. Saltelli, I. Sobol', Sensitivity measures, ANOVA-like techniques and the use of bootstrap, *J. Stat. Comput. Simul.* **58** (1997), 99–120.

[3] K. Chan, A. Saltelli, S. Tarantola, Sensitivity analysis of model output: variance-based methods make the difference, In: S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, eds., *Proceedings of the 1997 Winter Simulation Conference* (1997), 261–268.

[4] I. T. Dimov, *Monte Carlo Methods For Applied Scientists*, World Scientific (2007).

[5] I. Dimov, Z. Zlatev, Testing the sensitivity of air pollution levels to variations of some chemical rate constants, *Notes on Numerical Fluid Mechanics* **62** (1997), 167–175.

[6] I. Dimov, R. Georgieva, S. Ivanovska, Tz. Ostromsky, Z. Zlatev, Studying the sensitivity of pollutants' concentrations caused by variations of chemical rates, *J. Comput. Appl. Math., Vol. 235 (2010), pp. 391–402.*

[7] I. Dimov, K. Georgiev, Tz. Ostromsky, Z. Zlatev, Computational challenges in the numerical treatment of large air pollution models, Ecological Modelling ***179*** *(2004), pp. 187–203.*

[8] M. Gery, G. Whitten, J. Killus, M. Dodge. A photochemical kinetics mechanism for urban and regional scale computer modelling, J. Geophys Res. **94** *(D10) (1989), 12925–12956.*

[9] W. Hoeffding, A class of statistics with asymptotically normal distribution, Ann. Math. Statist. **19** *(3) (1948), 293–325.*

[10] T. Homma, A. Saltelli, Importance measures in global sensitivity analysis of nonlinear models, Reliability Engineering and System Safety **52** *(1996), 1–17.*

[11] Ø. Hov, Z. Zlatev, R. Berkowicz, A. Eliassen and L. P. Prahm, Comparison of numerical techniques for use in air pollution models with nonlinear chemical reactions, Atmospheric Environment, Vol. 23 (1988), pp. 967–983.

[12] J. Jacques, C. Lavergne, N. Devictor, Sensitivity analysis in presence of model uncertainty and correlated inputs, Reliability Engineering and System Safety **91** *(2006), 1126–1134.*

[13] M. McKay, Evaluating prediction uncertainty, Technical Report NUREG/CR-6311/, *US Nuclear Regulatory Commission and Los Alamos National Laboratory (1995).*

[14] Tz. Ostromsky, I. Dimov, R. Georgieva, Z. Zlatev, Air pollution modelling, sensitivity analysis and parallel implementation, Int. Journal of Environment and Pollution, Vol. 46 (1-2), (2011), pp. 83–96.

[15] Tz. Ostromsky, Z. Zlatev, Parallel Implementation of a Large-scale 3-D Air Pollution Model, in: Large Scale Scientific Computing (S. Margenov, J. Wasniewski, P. Yalamov, Eds.), LNCS-2179, Springer, 2001, pp. 309–316.

[16] Tz. Ostromsky, Z. Zlatev, Flexible Two-level Parallel Implementations of a Large Air Pollution Model, in: Numerical Methods and Applications (I.Dimov, I.Lirkov, S. Margenov, Z. Zlatev - eds.), LNCS-2542, Springer (2002), pp. 545–554.

[17] H. Rabitz, O. Alis, J. Shorter, K. Shim, Efficient input-output model representations, Computer Phys. Comm. **117** *(1999), pp. 11–20.*

[18] A. Saltelli, Making best use of model valuations to compute sensitivity indices, Computer Physics Communications **145** *(2002), pp. 280–297.*

[19] A. Saltelli, K. Chan, M. Scott, Sensitivity Analysis, *John Wiley & Sons publishers, Probability and Statistics series (2000).*

[20] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, *Halsted Press (New York, 2004).*

[21] I. Sobol', Multidimensional quadrature formulas and Haar functions, *Nauka, Moscow, 1969 (in Russian).*

[22] I. M. Sobol', Sensitivity estimates for nonlinear mathematical models, Matem. Modelirovanie **2** *(1) (1990), pp. 112–118.*

[23] I. M. Sobol', Sensitivity estimates for nonlinear mathematical models. Mathematical Modeling and Computational Experiment **1** *(1993), pp. 407–414.*

[24] I. M. Sobol', Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Mathematics and Computers in Simulation, **55** *(1-3) (2001), pp. 271–280.*

[25] I. M. Sobol', Theorem and examples on high dimensional model representation, Reliability Engineering and System Safety **79** *(2003), pp. 187–193.*

[26] *A. Takemura, Tensor analysis of ANOVA decomposition,* J. Amer. Statist. Assoc. **78** *(1983), pp. 894–900.*

[27] *WEB-site of the Danish Eulerian Model, available at:* http://www.dmu.dk/AtmosphericEnvironment/DEM

[28] *Z. Zlatev, Computer treatment of large air pollution models, Kluwer (1995).*

[29] *Z. Zlatev, I. Dimov, Computational and Numerical Challenges in Environmental Modelling, Elsevier, Amsterdam (2006).*

[30] *Z. Zlatev, I. Dimov, K. Georgiev,* Modeling the long-range transport of air pollutants, IEEE Computational Science & Engineering, **1** *(3) (1994), pp. 45–52.*

DOI: 10.15439/2023F7504
ISSN 2300-5963 ACSIS, Vol. 35

Inscrutability versus Privacy and Automation versus Labor in Human-Centered AI: Approaching Ethical Paradoxes and Directions for Research

Hagen Peukert
0000-0002-3228-316X
Universität Hamburg
Email: hagen.peukert@uni-hamburg.de

***Abstract*—From an analysis of ethical paradoxes based on the inscrutability feature of AI algorithms and resulting from recent advances in this field, this paper emphasizes the pressingness of dedicating research to the potential consequences on societal organization and interactions. With reference to Critical Theory that needs to be recombined with other socio-technical theories, new perspectives on future research is offered and discussed in light of privacy and labor market, their mutual influence as well as limitations.**

I. Introduction

[6] OUTLINE possible challenges of managing human-centered Artificial Intelligence (AI). In the proposed model, the "frontiers of AI" (pp. 5, 14) are placed in a two dimensional space of scope and performance, which set the boundaries of three identified facets: autonomy, learning, and inscrutability. Moreover, the authors suggest further challenges likely to be relevant in the future; among them a reuniting frontier, i.e. ethical issues. In a wealth of literature, the later are referred to as the prevalent concern of the digital age and a major research gap that needs to be occupied on a much larger scale [8], [19], [14]. More specifically, violation of privacy, deep fakes, and accountability are burning issues evolving with AI and at the same time revealing different views on AI technology as well as conflicting at least partly with the prevalent value system established in the non-digital world.

In the digital world, the right for anonymity and the right of forgetting can no longer be guaranteed because algorithms are enabled to establish chains of correlations between pieces of information that can neither be foreseen nor fully understood by a human brain. Consequently, effective barriers to predict and protect these rights cannot be set up including the idea that AI is also employed to discover privacy violations itself. Similar to a trapdoor-one-way function in mathematics, the integrity of data can be manipulated by neural nets in such a way that it becomes impossible to clearly identify fake video clips and pictures from true material. Both, deep fakes and backtracking to a specific person, is realized but not prevented with AI. In other words, the issue at stake is the inscrutability condition as outlined in [6], which creates a paradox. On the one hand, humans have found a method that really helps them in getting work done; yet they are not capable of understanding how it really works. This inscrutability feeds back on the privacy of users. On the other hand, the lack of understanding AI methods also accounts for undermining fundamental rights mainly due to the inscrutability property, which is, by the same token, accountable for more job efficiency elsewhere. Thus, deep neural nets produce advantages and disadvantages that directly conflict each other. In analogy to nuclear fission, today it is still unclear if the advantages of AI outweigh the disadvantages. And this applies to the digital as well as to the analogous world.

The aforementioned paradox really consists of several ethical issues. Due to the inscrutability feature of AI systems, violated privacy has at least a direct and an indirect consequence. The direct consequence consists in massive collections on personality traits and unique identification vectors unknown to the users and future use. The indirect consequence is a substitution effect of human labor. AI algorithms collect data from users granting insights into processes and best practices on their work expertise while pretending help. The gained knowledge can then be used to oust human labor.

In this study, a critical perspective on these paradoxes will be provided. First the background of the theoretical framework, in which the identified AI paradoxes can be embedded, is given. Second, the nature of the paradoxes is analyzed and discussed. Last, the study closes with possible limitations and a short summary of the outcome.

II. Theoretical Framework for Analysis

The aim of the Frankfurt School was to scrutinize and challenge the existing power relations in questioning their underlying (often not explicitly known, but subconsciously assumed) preconditions, how these evolved over time and if these are still valid under the actual conditions that persist at present [24], [2], [17], [15]. Now, we find ourselves in a constantly readjusting societal value system. Socio-economic elites of a society have the power to influence belief and values in their favor and interests. Although elites cannot predict the outcome of their actions with certainty and must not consciously do so, the degree of impact is higher than for other societal groups. Put briefly, the area of influence is not absolute, yet it is higher, especially if considering the number of its members. These sociological findings form the theoretical backbone of what is referred to as Critical

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

Theory under the additional constraint that technology itself establishes and strengthens power relations, i.e. it is tantamount to the means of production, capital goods, and financial assets in classical Marxian thinking. The two ethical paradoxes to be elaborated on in section III are model cases for this theoretical set up since technology is abused to (again) substitute labor. The substitution alone must not be negative. What is disadvantageous is how the socio-economic elite finds a way to keep the productivity gain from the substitution for themselves. From a sociological point of view, the substitution of white collar workers is especially demanding because it may reveal a difference to the first wave of machine automation, which stroke blue collar workers only.

At this point, it is necessary to extend Critical Theory by yet another explanatory variable, that is, proximity, i.e. the degree of similarity to the interests of the socio-economic elite. The hypothesis here would be: The closer a socio-economic group is to the socio-economic elite; the more concessions are made. Within this approach, one would have to observe that the substitution of white collar workers by AI technology comes along with political action that pays more tribute to the well-being of e.g. office workers compared to what the industrial workers experienced in the first wave of machine automation in the last decades of the last century. For the German case, one may speculate how the “Hartz IV” acts differ from the more recent legislation called “Bürgergeld”, which may have construed towards possible future firings coming along with AI engagement in public administrations. At the European level, new AI legislation (AI act) is in preparation, which also includes concessions in this direction [30], [21]. This alone may not be evidence enough for including proximity in Critical Theory, but it gives the incentive to carry out research in this direction.

A different perspective on the challenges revealed by the proposed paradoxes is laid out in Socio-Technical Theories [25], [31], [32], [33], [1]. Here the focus is on the organization and the question of how socio-technical change is analyzed towards the optimal way taking into account employees and their values. There are some more theories tying into the ethics of AI and society, which should be considered and be integrated into future research in this direction. These include *Privacy Calculus Theory* [23], [7], [20] to validate the assumptions of the preset theoretical material. This would offer an IS perspective on privacy issues if extended by findings of *Technological Network Analysis*. In addition, *Information Asymmetry Theory* [3] and certainly also as a cascade of several very influential theories, the *Unified Theory of Acceptance and Use of Technology* [40], are needed to further add on explanatory potential. One of its inputs can be contrasted to the simpler *Technology Acceptance Model* [12], [13], that in a similar manner modifies its predecessor. By the same token these theories could be enriched by or contrasted to old classics such as *Rational Choice Theory* or *Prospect Theory*, but also by more recent findings prevalent in *Organizational Culture Theory*, *Psychological Ownership Theory*, and *Work Systems Theory*.

III. Analyzing AI Ethics Paradoxes

This study is supposed to reveal a contrastive view on two seemingly ethical paradoxes that are selected purposefully for their differing contents, assumptions, and consequences for the society, but at the same time make interconnectedness visible, that is, the inscrutability paradox provokes the "‘AI versus labor"’ dilemma. A stepwise comparison will also reveal the methodological set up of analyzing the two paradoxes. Roughly, the dynamic equilibrium model [37] serves as an orientation. Yet the differentiation of paradox and dilemma applies to the economic context that appreciates logical paradoxes, but not to ethics per se for that moral dilemmas and paradoxes are synonymously used. An ethical paradox is recognized in situations, in which important values or norms are violated no matter how an individual behaves [11].

While the questions on “AI versus labor” need a quantitative analysis of secondary sources, the “inscrutability paradox” is for the most part a qualitative investigation of network designs and algorithms as well as a collection of results in the literature. For the former, to check the assumptions implicitly made by the claim AI substituted labor, data on the branches of unemployment and correlations with branches of dismissal are available by the federal Statistical Office and the Census Bureau. Additionally, data of recruiting branches, expatriates and qualified immigration needs to be aggregated (cp e.g. [22]). From there, it becomes clear for which qualifications organizations aim at. Last, programs of work creation schemes are considered. More particularly, the duration and number of programs that are aligned to AI technology can here be determined. And to evaluate the quality of these programs, they are compared to educational tracks typically present at university curricula.

The second method addressing the “inscrutability paradox” is qualitative. As a starting point, a short collection of acknowledged results of big data analyses could be laid out (e.g. [26]). Some popular findings include results that happen to be technically correct, but are neither causal nor plausible, yet valid with respect to the algorithmic short cut through the data (e.g. of the form: people with green shirts, long hair, and . . . have an 80 percent chance of getting a heart attack). If made public, some of these findings may feedback into future analysis and change the final result with unfair consequences for independent parties (e.g. sellers of green shirts). By use of these examples, an analysis of the algorithmic construct of a limited number of networks used for big data analysis will reveal that the inscrutability feature rather erodes privacy than enhancing it. Now, the same is done for net privacy issues (integrity, authentication, anonymity) and contrasted with e.g. profiling or fingerprinting. Last, privacy and inscrutability is brought together by showing what the application of the respective other algorithmic set up would mean for inscrutability and privacy.

The interrelatedness of privacy and the labor market is revealed when looking at the forecasted consequences on the white collar labor market. Now knowledge jobs, sales agents, law consultants, and the like are affected on a large scale and much will depend on how fast respective AI technologies will be introduced in public and private organizations and if labor forces have sufficient time to shift to other areas of high-quality jobs. The average education time for these vocations are estimated to be five years. It is likely to depend on the age of the learner, too. This line of argumentation hints at the supposition of the paradox: the advantages of human-centered AI technologies in the service sector that are supposed to assist and add value to employed staff are traded off against the cost of their labor, that is, human workforce is potentially substituted by machine power whereas the profit of this substitution is roughly the cost of the employed. But different from the first wave of the machine revolution, which massively reached out to blue collar workers, the ethical paradox could be seen again in the inscrutability of deep learning. Without consciously knowing, service workers grant worldwide petabytes of data on processes, best practices, behaviors. Now, AI algorithms use this naturally grown knowledge, derive new patterns and routines from it. In fact, optimization functions produce even better results than their human counterpart would do. Put briefly, the human employee helps the machine to learn the specifics and secrets of its job. By doing so employees downsize and cut their own jobs while AI tools pretend to assist them, which they do, but in the background collect valuable data to get rid of the human workforce in the long run, which is prevented from transparency by the asynchronous inscrutability property. The knowledge gathered here is most of the time not even explicitly accessible to any human understanding (cf [35]).

This shows how human-centered AI technology in the digital sphere produces ethical paradoxes that further replicate serious consequences, even if unintentionally brought forward, for the interaction in the analogous world. First, it brings together the characteristics of inscrutability acknowledged as inherently preconditioned in all deep net's AI technology, on the one hand, and privacy issues, deep fakes, and accountability, on the other hand. Second, an ethical problem may also be seen in promoting the advantages of AI for the existing workforce while really exploiting their knowledge and preparing its substitution without making it transparent. This is not, as one may think at first glance, an inscrutability issue per se, but mediated by privacy violations since the inscrutability of the inner workings of a deep net are not what is hidden from the public, but the fact that more or less sensitive knowledge is gained from its aid. As a first working hypothesis it looks as though the first paradox seems to be inbuilt as the property of inscrutability that produces a dilemma no matter how it is framed. Inscrutability fosters anonymity, but conflicts with accountability and, at the same time, it discloses privacy (big data correlations), but facilitates disguise (deep fakes). The second paradox, however, is concerned with the social context, for which it is possible to make additional assumptions and find additional factors of influence. As such these assumptions and factors of influence should be modifiable towards the value system of our society. AI may disguise spying on labor skills while pretending support, but it is still the decision of the members of the society to accept its consequences. In other words, if a machine substitutes human labor, we may decide that the substituted workforce continues to receive the full payment and may have more leisure time.

To validly evaluate possible consequences and developmental trends, it is to be further examined if additional assumptions could be made. They have to stand the test of plausibility and if possible have to be derived from the theoretical basis or should be included as a given fact. The latter is the case when the exact employment figures from Census of federal statistical offices, respective branches, and significant workforce shifts are taken into consideration. An example for a theoretical finding that could counterbalance an ethical evaluation is that in times of a high labor demand, the dismissed workforce is even better off if employed in newly created and emerging AI technology branches. This is the case in a Schumpeterean understanding. Yet, what could not be foreseen in the 1930ies, not even up until the turn of the century, was the velocity, in which these changes take place, and its unpredictable consequences for the societal set up (labor, wealth, values). As we have learned from the last burst of the Internet bubble in the early 2000th, interest rate reversals, or the financial crises with a 10-year recession of the American employment market later on that public institutions such as educational systems do not keep pace with the necessary requirements dictated by a digital world economy.

As a consequence, for an ethical evaluation, it is not enough to show one positive path out of dilemmas, but to take into account all (thinkable) possible scenarios and a plausible estimation of their occurrence probability. Indeed, a scenario is more likely if a theoretical claim or a claim derived from a theoretical basis has proven right by past events, which harbor similar assumptions. Paraphrased as a research question, one could ask what are the ethical paradoxes that follow from the assumptions given in the literature, i.e. inscrutability and privacy, task automation and labor market? What is the nature of these paradoxes? Under which costs and assumptions could they be resolved? The answer of an ethical question needs to be contextualized in a social context, in which a set of values prevail. Typically, these values neither are of equal importance ("speaking about the dead" vs. murder) nor do they stay constant over time (adultery, piousness). Some values may change rather drastically if power relations or other dynamics overcome a critical limit (euthanasia, right of succession, role of men and women), others are more rigid and seem to be static (theft, right of possession, piety).

Societies (as groups of people of different sizes [38]) happen to converge on values and its members show an intuitive understanding [27], [18], [16]. Values are key to social cohesion. It implies bottom-up learning processes that emerge over long time spans and they consolidate subconsciously in the collective memory of a community. Also, an intuitive and entrenched understanding of values is necessary to rank values [36], [5], [10], [4], [28], [29]. It is this ranking together with changes in the understanding of values that leads to conflicts and ethical paradoxes whereas changes are often driven by technological innovations (such as AI) or major scientific findings (evolution, solar system, relativity).

Hence to understand ethical paradoxes, it is essential to guarantee social cohesion and thus this research directly offers its practical usefulness. Adjustments in the value system is an ongoing process in all societies and if made transparent, it becomes more robust. The humble scientific value comprises two aspects. First, it consists of the systematic proof of the underlying assumptions and balancing the ranking of values, i.e. how can AI as a new driver of technological innovation and its consequences be embedded into the existing value system. Second the correct derivation of action alternatives from the AI placement must be considered.

There are no clear cut solutions for ethical problems, otherwise they would not be any. Solutions are sketched as appraisals of all alternatives, a careful balancing of all known pros and cons while taking into account the value system prevalent at the time of evaluation. This modus operandi is common in Ethics in general and in Ethics in Information Science in particular. And it has been done for several Topics in AI research [39]. However, along the lines of inscrutability in the condensed form of the frontiers model [6], there is no systematic ethical investigation deriving privacy violations from the inherently given inscrutability feature and arguing for a causal relation to effects on the labor market.

IV. Limitations

When viewed through the lens of Critical Theory, the "Labor versus Automation" paradox could turn out to be not a real ethical paradox because there seem to be legislative solutions that do not contradict to ethical values. Yet the entire chain of arguments cannot be overlooked from a superficial assessment. So the legislature in favor of white collar workers could have negative effects for others, e.g. blue collar workers. The assessment could turn out to be different if considering Socio-technical theories that focus on organizations. Other theoretical groundwork may come forward with ideas not yet considered. All of them need to be accounted for the final evaluation.

For the case of the "inscrutability versus privacy" paradox, there exist at least no obvious solution from which a major drawback on the values could be denied. So no matter what is done, it will always be ethically questionable. Both, inscrutability and privacy, reinforces the weaknesses of the respective other. As a preliminary thought experiment: Increasing transparency, i.e. decreasing inscrutability, also raises the negative effects on privacy. Strengthening privacy (e.g. Thor browsing) enhances inscrutability even further. In addition, it would further lead to a bias towards users unfamiliar with the technology. As identified in the above argument, technology is seen on the same analytical level as financial assets or means of production. So it feeds back into supporting distorted power relations.

A possible limitation for the examination of labor and automation could be seen in the general economic situation on the labor market. Throughout all service branches, organizations report a high deficit on qualified employees. If this situation continues as some outlooks suggest, the labor-automation dilemma would lose its ethical grasp. In this case, the net effect of automation is very likely to be positive with respect to labor substitution. And it would only be relevant for Critical Theory for other aspects such as the traditional thinking in this field, that is, alienation of the human being from nature, but no longer the machine human substitution as the source of societal unfairness.

Concerning the limitations of inscrutability and privacy issues, it is not possible to take into consideration all prevalent architectures of neural nets for a technical analysis. Due to lack of evidence, it is still open if the inscrutability feature of neural nets could also be used to enhance privacy without paying off on accountability (e.g. by widely establishing cryptographic solutions [9]), so that inscrutability is not misused as a data collector, but as a data protector. However, there are no references in the relevant literature that really makes this theoretical possibility plausible.

V. Summary

The paper at hand set out to identify two relevant ethical paradoxes that come along with human-centered AI technology. It turns out that the inscrutability property of AI technology as produced by deep neural nets and by an increasing blurring of the ground truth input, invokes privacy violations. Privacy violations, on the other side, make it possible to collect huge amounts of data, which enable a machine view on services and processes that is not accessible to a human brain. These views are exploited as shortcuts bearing large efficiency gains for carrying out these services and thus making human work obsolete with seemingly dramatic consequences for the labor market. From this situation new paradoxes emerge from the very moment that humans allow machine exploitation without being able to grasp the full account of such decision. The consequences cannot be understood for the inscrutability argument; privacy stretches out over observations as to what is clicked, when and where. In fact, we see some kind of a cascade following from inscrutability over privacy to the engagement of labor. In addition to identifying the paradoxes, a specific theoretical context, in which these phenomena could best be studied is given and some obvious limitations to these approaches are also outlined.

References

[1] Adman, P., and Warren, L. 2000. "Participatory sociotechnical design of organizations and information systems - an adaptation of ethics methodology," Journal of Information Technology (15:1), pp. 39-51.

[2] Adorno, T. W. 1972. „Zur Logik der Sozialwissenschaften,“ in Gesammelte Schriften. Band 8: Soziologische Schriften I, Suhrkamp: Frankfurt am Main, pp. 547–565.

[3] Akerlof, G. A. 1970. "The market for 'lemons': Quality uncertainty and the market mechanism", The Quarterly Journal of Economics, pp. 488-500.

[4] Allen, V. L. and Levine, J: M. 1971. "Social support and conformity: The role of independent assessment of reality," Journal of Experimental Social Psychology (7), pp. 48-58.

[5] Asch, S. E. 1951. “Effects of group pressure upon the modification and distortion of judgments,” in Groups, leadership and men, H. Guetzkow (ed.), Pittsburgh: Carnegie, pp.177-190.

[6] Berente, N., Gu, B., and Recker, J. 2021. “Managing Artificial Intelligence,” MIS Quarterly (45:3), pp. 1-41.

[7] Blatchford, C. 2000. “Information security, business and the internet — Part 1,” Network Security 2000(1), pp. 8–12.

[8] Bechmann, A. and Bowker, C. G. 2019. “Unsupervised by any other name: Hidden layers of knowledge production in artificial intelligence on social meda,” Big Data & Society, pp. 1-11.

[9] Chaum, D. 1992. “Achieving Electronic Privacy, Scientific American,” pp. 66-101, URL: https://chaum.com/wp-content/uploads/2021/12/ScientificAmerican-AEP.pdf.

[10] Cialdini, R. B. and Trost, M. R. 1998. “Social influence: Social norms, conformity, and compliance,” in Handbook of social psychology D. T. Gilbert, S. T. Fiske and G. Lindzey (eds), Boston: McGraw-Hill, pp. 151-192.

[11] Cohen, B. 1967. “An Ethical Paradox,” Mind (76:302), pp. 250–259.

[12] Davis, F. D. (1989). “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” MIS Quarterly 13(3), 319-339.

[13] Davis, F. D., Bagozzi, R. P. and Warshaw, P. R. (1989). “User acceptance of computer technology: A comparison of two theoretical models,” Management Science (35:8), pp. 982-1003.

[14] DCC (Digital Curation Center). 2020. “The Role of Data in AI: Report for the Data Governance Working Group of the Global Partnership of AI,” School of Informatics, University of Edinburgh.

[15] Habermas, J. 1981. Der philosophische Diskurs der Moderne. Zwölf Vorlesungen, Frankfurt am Main: Suhrkamp.

[16] Hogg, M. A., Sherman, D. K., Dierselhuis, J., Maitner, A. T. and Moffitt, G. 2007. “Uncertainty, entitativity, and group identification,” Journal of Experimental Psychology (43), pp. 135-142.

[17] Horkheimer, M. 1988. „Traditionelle und kritische Theorie,“ in Gesammelte Schriften. Band 4: Schriften 1936–1941. Frankfurt am Main: Fischer.

[18] Janis, I. L. 1982. Groupthink, Boston: Hougthon Mifflin.

[19] Jaton, F. (2021): Assessing biases, relaxing moralism: On ground-truthing practices in machine learning design and application. Big Data & Society, pp. 1-15.

[20] Jozani, M., Ayaburi, E., Ko, M., and Choo, K. K. R. (2020). Privacy concerns and benefits of engagement with social media-enabled apps: A privacy calculus perspective. Computers in Human Behavior (107), 106260.

[21] Kerkmann, C. 2022. Künstliche Intelligenz: Wirtschaft warnt vor „massiven Einschränkungen“ durch AI Act. URL am 23. Januar 2023: https://www.handelsblatt.com/technik/it-internet/eu-regulierung-kuenstliche-intelligenz-wirtschaft-warntvor-massiven-einschraenkungen-durch-ai-act/28850684.html.

[22] Kropp, P., Theuer, S. and Fritzsche, B. 2018. „Immer mehr Tätigkeiten werden durch Digitalisierung ersetzbar: Aktualisierte Substituierbarkeitspotenziale in Thüringen, IAB-Regional. IAB Sachsen-Anhalt-Thüringen,“ No. 02/2018, Institut für Arbeitsmarkt und Berufsforschung (IAB), Nürnberg.

[23] Laufer, R. S. and Wolfe, M. 1977. „Privacy as a Concept and a Social Issue: A Multidimensional Developmental Theory,” Journal of Social Issues (33), pp. 22-42.

[24] Marcuse, H. 1965. „Philosophie und kritische Theorie,“ in Kultur und Gesellschaft I, Frankfurt am Main: Suhrkamp, pp. 102–127.

[25] Markus, L. 1983. “Power, politics, and MIS implementation,” Communications of the ACM (26:6), pp. 430-444.

[26] McFarland, D. A., and McFarland, H. R. 2015. „Big Data and the danger of being precisely inaccurate,”. Big Data & Society (2:2). https://doi.org/10.1177/2053951715602495 .

[27] Milgram, S. 1974. Obedience to authority: An experimental view, New York: Harper & Row.

[28] Moscovici, S. 1976. Social Influence and social change, London: Academic Press.

[29] Moscovici, S. 1980. “Toward a theory of conversion behavior,” in Advances in experimental social psychology (13), L. Berkowitz (ed.), pp. 209-234.

[30] Müller, A. 2022. „Der Artificial Intelligence Act der EU: Ein risikobasierter Ansatz zur Regulierung von Künstlicher Intelligenz.“ Zeitschrift für Europarecht (1).

[31] Mumford, E. 1983. Designing human systems, the ETHICS approach. Manchester Business School, Manchester, U.K.

[32] Mumford, E. 2000, “Socio-technical design: An Unfulfilled Promise or a future Opportunity”. in Organizational and Social Perspectives on Information Technology, Baskerville, R., Stage, J., and DeGross, J.I. (eds), Boston: Kluwer academic Publications, pp 33-46.

[33] Mumford, E., (2003), Redesigning Human Systems, Hershey: IRM Press.

[34] Mumford, E. and Weir, M., (1979). Computer Systems and work Design: The ETHICS Method, New York: Wiley & Sons.

[35] Paaß, G. and Hecker, D. 2020. Künstliche Intelligenz – Was steckt hinter der Technologie der Zukunft?, Springer Vieweg.

[36] Sherif, M. 1936. The psychology of social norms, New York: Harper & Row.

[37] Smith, W. and Lewis, M. (2011). "Toward a theory of paradox: A dynamic equilibrium model of organizing". Academy of Management Review, (36:2), pp. 381-403.

[38] Turner, J. C. 1991. Social Influence, Buckingham: Open University Press.

[39] Vallor, S. 2016. Technology and the Virtues: A Philosophical Guide to a Future Worth Wanting. Oxford: University Press.

[40] Venkatesh,V., Morris, M. G.;Davis, G. B.;Davis, F. D. 2003. “User acceptance of information technology: Toward a unified view”, MIS Quarterly (27:3), 425-478.

DOI: 10.15439/2023F7072
ISSN 2300-5963 ACSIS, Vol. 35

New measures of algorithms quality for permutation flow-shop scheduling problem

Radosław Puka
AGH University,
ul. Gramatyka 10, 30-067 Kraków, Poland
Email: rpuka@agh.edu.pl

Iwona Skalna, Tomasz Derlecki
AGH University,
ul. Gramatyka 10, 30-067 Kraków, Poland
Email: {skalna, derlecki}@agh.edu.pl

***Abstract*—The permutation flow-shop scheduling problem (PFSP) is an important problem in production industry. The problem has been a subject of many research and various algorithms to solve PFSP have been developed over the years. The newly developed algorithms are usually tested on Taillard and VRF benchmarks and their results are compared using various measures that assess the size of error made by an algorithm and the computation time. In this paper, we propose two new measures to assess the quality of results of algorithms for solving PFSP with the makespan criterion. The first ARD.NEH measure gives similar results as the well known ARPD measure but is robust to updates of the best known solutions of benchmark problems. The second ARID measure is an interval-based measure which is able to assess whether the good quality of an algorithm results stems from its good behavior of this algorithm for a few instances or from its good behavior for most instances. The computational experiments confirm the usefulness of the proposed quality measures.**

I. Introduction

The permutation flow-shop scheduling problem (PFSP) is one of the most studied combinatorial optimization problems, rooted in the manufacturing industry. It can be defined as follows: given a finite set of m machines $\{M_1, \ldots, M_m\}$ and a finite set of n jobs $\{J_1, \ldots, J_n\}$, each of which should go through all the m machines in the same order, the goal is order the jobs so as to minimize the assumed optimization criterion (e.g., makespan, total tardiness, flow time, cost, energy consumption).

The PFSP with makespan criterion, commonly referred to as $Fm|prmu|C_{\max}$ [1], is undoubtedly the most frequently investigated scheduling problem. Garey and Johnson [2] proved that $Fm|prmu|C_{\max}$ is NP-hard if $m \geqslant 3$. Therefore, various heuristics have been developed to solve this problem in a reasonable amount of time. Among them, the Navaz, Enscore and Ham (NEH) construction heuristic [3] plays an important role; for a long time NEH has been regarded as the best heuristic for solving $Fm|prmu|C_{\max}$.

Since optimal solutions are generally not known for some instances, the only way to asses the results of new methods is to compare them with the best solutions known so far. The well-known measure of solution quality, initially referred to as the increase over optimum (IOO) [4] and later as the relative percentage deviation (RPD) [5], is defined as:

$$\mathrm{RPD} = \frac{S - Best}{Best} \times 100\%, \tag{1}$$

where S is the solution of the evaluated algorithm and $Best$ is the best solution known so far for a given instance of the problem. For a group of instances, a synthetic solution quality measure, called the average relative percentage deviation (ARPD), is calculated as:

$$\mathrm{ARPD} = \frac{1}{I}\sum_{i=1}^{I} \frac{S_i - Best_i}{Best_i}, \tag{2}$$

where I is the number of instances, S_i is the solution of the evaluated algorithm on the instance i of a given size, and $Best_i$ is the best solution known so far for this instance.

The quality of solutions is obviously not the only aspect of algorithms evaluation – the running time is also an important feature (we often face the trade-off between the quality of results and computational time). Literature research shows that the computational time is often reported in time units (usually in milliseconds) [6], sometimes, especially in case of simpler algorithms, the computational complexity is provided. Given several algorithms to be compared and various instances, the computational effort is usually measured by using the average CPU time (ACPU) computed as follows:

$$\mathrm{ACPU}_j = \frac{1}{I}\sum_{i=1}^{I} \mathrm{CPU}_{i,j}, \tag{3}$$

where $\mathrm{CPU}_{i,j}$ is the CPU time consumed by algorithm j on instance i. However, the running time scheduling algorithms strongly depends on the size of the problem instance, therefore Fernandez-Viagas and Framinan [7] proposed to measure the average relative percentage time (ARPT) consumed by algorithm j:

$$\mathrm{ARPT}'_j = \frac{1}{I}\sum_{i=1}^{I} \mathrm{RPT}_{i,j}, \tag{4}$$

where $\mathrm{RPT}_{i,j}$ (relative percentage computation time of algorithm j for instance i) is computed as:

$$\mathrm{RPT}_{i,j} = \frac{\mathrm{CPU}_{i,j} - \mathrm{ACT}_i}{\mathrm{ACT}_i}, \tag{5}$$

This study was conducted under a research project funded by a statutory grant of the AGH University for maintaining research potential.

and ACT_i (average computational time for instance i) is computed as:

$$\mathrm{ACT}_i = \frac{\sum_{j=1}^{J} \mathrm{CPU}_{i,j}}{J}. \tag{6}$$

Since ARPT'_j can yield negative values ($\mathrm{ARPT}'_j > -1$), Fernandez-Viagas and Framinan [8] proposed to compute $\mathrm{ARPT} = \mathrm{ARPT}' + 1$, which allows the graphics to be shown in logarithmic scale.

The above described features of ACPU and ARPT make these two measure not very authoritative and quite cumbersome in practice. In [9], we have proposed the ART.NEH (the Average Relative Time over NEH) indicator defined by the following formula:

$$\mathrm{ART.NEH} = \frac{\sum_{i=1}^{I} \frac{\mathrm{CPU}_i}{\mathrm{CPU}_{i,\mathrm{NEH}}}}{I}, \tag{7}$$

where I is the number of considered instances, CPU_i is the CPU time of a considered algorithm for the instance i, and $\mathrm{CPU}_{i,\mathrm{NEH}}$ is the CPU time of NEH for the instance i. ART.NEH indicates how many times, on average, the evaluated algorithm is faster (ART.NEH<1) or slower (ART.NEH>1) than the classical NEH. Following this idea, we propose in Section II several new measures to compare the quality of results produced by algorithms for solving PFSP with the makespan criterion. Numerical experiments showing the usefulness of the proposed measures are described in Section III. The paper ends with concluding remarks.

II. New measures of algorithms efficiency

New algorithms are expected to be better than existing ones, but a fair comparison of algorithms is quite difficult (due to implementation issues and hardware used). However, most of papers on solving PFSP with the makespan criterion provide the results produced by NEH. So, it seems quite natural to use this well-known heuristic as a computational benchmark.

The ARPD indicator given by formula (2) is by far the most popular measure for assessing the quality of scheduling algorithms taking into account the size of the error. It has, however, some drawbacks which led to the development of alternative measures. An important drawback, we want to emphasize, is that the value of ARPD can change when new better solutions are found for an analyzed instance. In this regard, the ARPD value of an algorithm can change significantly over the years. A good example can be the most known PFSP benchmark – Taillard's benchmark [10] published in 1993. Though it is now 30 years since its publication, better solutions are still found for various instances [11]. Thus, since the ARPD factors change, the whole measure change as well. In that case, it is difficult to compare the new results with existing (published) ones due to different reference values (the results of such a comparison may not be reliable). To get rid of this drawback, in this paper we propose a new measure ARD.NEH (Average Relative Deviation over NEH) which will not change in time thanks to the use of the NEH results as reference results. The proposed ARD.NEH measure is computed from the following formula:

$$\mathrm{ARD.NEH} = \frac{1}{I}\sum_{i=1}^{I} \frac{\mathrm{NEH}_i - S_i}{\mathrm{NEH}_i}, \tag{8}$$

where I is the number of instances, S_i is the solution of the evaluated algorithm on the instance i, and NEH_i is the solution obtained using the NEH algorithm for this instance.

The main reason for developing this measure was to make it easier to compare the results produced by new algorithms with the results available in the literature. The advantage of ARD.NEH over ARPD is that it does not change over time. This particular feature of ARD.NEH is due to the fact that ARD.NEH does not depend on the best solutions known so far, but on the results of NEH. So, the measure is especially useful to deal with those problems for which the optimal solution is not know yet. Since ARD.NEH indicates how far the results of an algorithm are from the results of NEH, the greater is ARD.NEH the better.

Another new measure to assess the quality of the results, we propose in this paper, is the $\mathrm{ARID}(\inf, \sup)$ (Average Relative Interval Deviation) measure. $\mathrm{ARID}(\inf, \sup)$ is different from existing quality measures in that it is based on the interval $[\inf, \sup]$ (it is assumed that the interval $[\inf, \sup]$ can be improper) instead of a single value (reference point). By taking different intervals, we can obtain various quality measures. The concept behind this measure is to equalize the impact of each benchmark instance on the final value of the evaluation measure. The value of $\mathrm{ARID}(\inf, \sup)$ is computed from the following formula:

$$\mathrm{ARID}(\inf, \sup) = \frac{1}{I}\sum_{i=1}^{I} \frac{\max(\inf, \sup) - S_i}{\sup - \inf} \tag{9}$$

where S_i is the solution for the instance i.

Proposition 1: ARPD and ARD.NEH measures are a special case of the ARID measure.

Proof: Let I be the set of instances, and Best_i, NEH_i, and S_i the best known solution, the solution produced by NEH, and the solution for the instance i, respectively. It holds that

$$\mathrm{ARID}(\mathit{Best}, 0) = \frac{1}{I}\sum_{i=1}^{I} \frac{\max(\mathit{Best}_i, 0) - S_i}{0 - \mathit{Best}_i} =$$

$$= \frac{1}{I}\sum_{i=1}^{I} \frac{\mathit{Best}_i - S_i}{-\mathit{Best}_i} = \frac{1}{I}\sum_{i=1}^{I} \frac{S_i - \mathit{Best}_i}{\mathit{Best}_i} = \mathrm{ARPD}$$

$$\mathrm{ARID}(0, \mathrm{NEH}) = \frac{1}{I}\sum_{i=1}^{I} \frac{\max(\mathrm{NEH}_i, 0) - S_i}{\mathrm{NEH}_i - 0} =$$

$$= \frac{1}{I}\sum_{i=1}^{I} \frac{\mathrm{NEH}_i - S_i}{\mathrm{NEH}_i} = \mathrm{ARD.NEH}$$

■

In what follows, we set $\inf = \mathit{Best}$, $\sup = \mathrm{NEH}$, where *Best* means that we use the best solutions (makespans) known so far for benchmark instances, and NEH means that we use the solutions produced by NEH for the respective instances. Then, ARID(*Best*,NEH) (further referred to as simply ARID), similarly as ARPD, uses the best know solutions, so ARID is recommended to be used for problems with $Best = Opt$. ARID allows to equalize the impact of different instances on the final result. For example, the ARPD value for Taillard benchmark is the most influenced by the instances having the best solutions far from the optimum and the less influenced by the instances having the best solution close to the optimum. Making each instance to have comparable impact on the final evaluation of an algorithm, allows to compare different algorithms in terms of the stability of their results in relation to the dynamically determined value, which in this case is the result of NEH. The result of NEH can therefore be considered as a kind of assessment of the difficulty of a given instance. The stability of an algorithm should be understood here as a possibility to obtain better results than NEH for as many instances as possible. Let us note that the value of ARID, similarly as the value of ARD.NEH, should be maximized. The next section presents the experiments that aim to show the usefulness of the proposed measure of the algorithms quality.

III. Computational experiment

The measures proposed in Section II were used to assess the results of various algorithms for Taillard benchmark [10] and VRF Large benchmark instances [12]. Best solutions provided by the authors of the benchmarks are updated with the recent results presented in [11] (Taillard) and [13] (VRF Large).

Tables I and II show the values of the ARPD, ARD.NEH and ARID measures obtained for, respectively, Taillard and VRF Large benchmarks by using selected deterministic algorithms for solving PFSP (cf., [14], [15], [16], [17], [18], [7], [8], [19], [20], [21], [22], [9]). As can be seen from the tables, only two algorithms (RAER and RAER-di) achieved negative values of ARD.NEH measure, which means that their average results were worse than the average result of NEH. It can be seen as well that only FRB and N-list technique-based algorithms (the latter will be further referred to as N-algorithms) achieved the results that are better than NEH results by more than 1 percent, for both benchmarks. As for the ARID measure, only FRB algorithms and N-algorithms achieved the values greater than 15%. Moreover, only 3 algorithms (for Taillard benchmark) and 2 algorithms (for VRF Large benchmark instances) achieved the results greater than 50%, which means that only 3 algorithms were able to improve the results of NEH by, on average, more than a half distance between the best solution produced by NEH and the best solution known so far for a given instance.

TABLE I
ARPD, ARD.NEH and ARID values for Taillard benchmark

Algorithm	ARPD	ARD.NEH	ARID
RAER	3.94	-0.56	-56.99
RAER-di	3.57	-0.20	-40.97
NEH	3.37	0.00	0.00
NEMR	3.21	0.15	-6.91
NEHKK1-di	3.20	0.17	-0.16
KKER	3.19	0.17	-3.25
NEH1-di	3.15	0.21	4.73
NEHKK2	3.14	0.22	7.54
NEHR	3.10	0.26	2.52
NEH-di	3.08	0.28	9.37
vN-NEH+(2)	3.02	0.34	9.69
NEMR-di	3.01	0.34	4.81
N-NEH+(2)	2.99	0.36	10.27
NEHFF	2.95	0.41	1.41
KKER-di	2.91	0.44	13.87
NEHR-di	2.90	0.46	13.48
NEHD-di	2.88	0.47	6.00
vN-NEH+(3)	2.82	0.52	15.83
N-NEH+(3)	2.74	0.60	18.64
SP+(0.3)N+(2)	2.70	0.64	18.25
vN-NEH+(4)	2.67	0.67	20.42
N-NEH+(4)	2.60	0.74	22.09
FRB4$_2$	2.37	0.95	31.04
N-NEH+(8)	2.36	0.96	29.48
vN-NEH+(8)	2.28	1.04	32.77
SP+(0.3)N+(4)	2.27	1.05	32.85
SMα+(8)N+(2)	2.26	1.06	35.14
N-NEH+(16)	2.24	1.08	33.15
FRB4$_4$	2.17	1.15	34.93
vN-NEH+(16)	2.07	1.25	42.25
SP+(0.3)N+(8)	2.03	1.29	40.02
SMα+(8)N+(4)	2.01	1.31	43.51
FRB4$_8$	1.99	1.32	40.22
FRB2	1.98	1.33	32.81
FRB4$_6$	1.96	1.35	40.88
FRB4$_{10}$	1.92	1.39	42.49
SP+(0.3)N+(16)	1.89	1.41	44.51
SMα+(8)N+(8)	1.86	1.44	48.47
FRB4$_{12}$	1.84	1.46	45.01
SMα+(8)N+(16)	1.75	1.55	51.70
FRB3	1.66	1.64	50.06
FRB5	1.53	1.77	55.79

Figures 1 and 2 show the rank (y-axis) of each algorithm with respect to the specific quality measure. As we can see, the ranks with respect to ARD.NEH and ARPD coincide for all algorithms. This means that the ARPD measure can be successfully replaced with the ARD.NEH measure. If we take a look at the ARID measure, we can see that this measure ranks the algorithms in a different manner than the other two measures. Those algorithms that are ranked below the line determined by the ARPD and ARD.NEH measures can be considered as more stable. The results produced by these algorithms are less due to the fact of significant improvements of NEH results for single instances, and more due to improvements of NEH results for more instances. Due to the design of the measure, large improvements for single instances are less promoted than frequent but less significant improvements. Hence the deterioration of the results of individual algorithms,

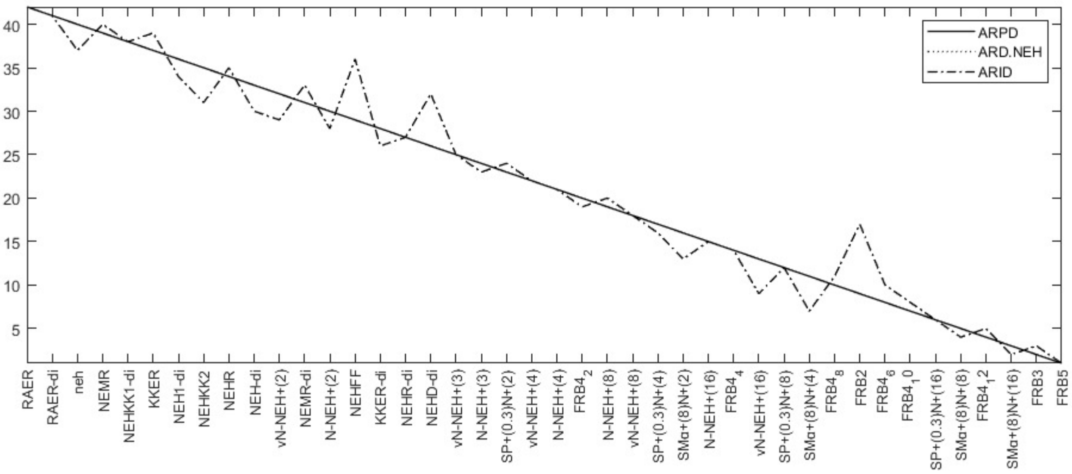


Fig. 1. Ranking of algorithms based on ARPD, ARD.NEH and ARID values for Taillard benchmark

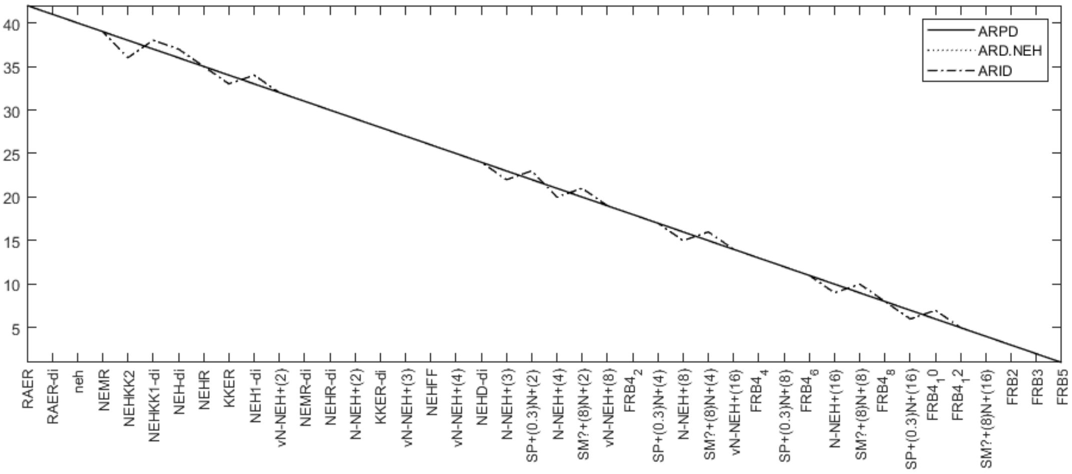


Fig. 2. Ranking of algorithms based on ARPD, ARD.NEH and ARID values for VRF Large instances

for which the position for the ARID measure deviates upwards from the line ARPD/ART.NEH.

IV. Conclusion

This work proposes two new measures for assessing the quality of results produced by algorithms for solving permutation flow-shop problems with the makespan criterion. The first ARD.NEH measure has the very useful feature of elimination of the dependency of the quality assessment from the best known results which, as shown by the performed analysis, change over time, and therefore the comparison of new results with the older one might be cumbersome. The second ARID measure is to our best knowledge the first interval-based measure. It is worth to underline that the ARID measure with properly selected intervals is equivalent to ARPD or ARD.NEH measures. The proposed new measure have been tested on 42 selected deterministic algorithms for solving PFSP run on Taillard and VRF Large benchmarks. Based on the obtained results it can be concluded that ARPD and ARD.NEH measures coincide, i.e., they rank the algorithms in a very similar manner. The ARID measure, in turn, is useful in assessing the stability of the algorithms, i.e., it indicates whether a good (average) quality of results stems from good results for a few instances of from good results for most instances. The numerical experiments show that the proposed measures are very useful for more reliable comparison of algorithms for solving PSFP with the makespan criterion.

References

[1] R. Graham, E. Lawler, J. Lenstra, and A. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," in *Discrete Optimization II*, ser. Annals of Discrete Mathematics, P. Hammer, E. Johnson, and B. Korte, Eds. Elsevier, 1979, vol. 5, pp. 287–326. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016750600870356X

TABLE II
ARPD, ARD.NEH AND ARID VALUES FOR VRF LARGE INSTANCES

Algorithm	**ARPD**	**ARD.NEH**	**ARID**
RAER	3.54	-0.13	-5.29
RAER-di	3.41	-0.01	-1.37
NEH	3.41	0.00	0.00
NEMR	3.30	0.10	3.00
NEHKK2	3.29	0.12	4.07
NEHKK1-di	3.27	0.13	3.70
NEH-di	3.27	0.14	3.78
NEHR	3.24	0.16	4.86
KKER	3.23	0.17	5.44
NEH1-di	3.23	0.17	5.09
vN-NEH+(2)	3.21	0.19	5.65
NEMR-di	3.14	0.26	8.00
NEHR-di	3.10	0.29	9.15
N-NEH+(2)	3.08	0.32	9.80
KKER-di	3.08	0.32	9.92
vN-NEH+(3)	3.07	0.33	9.93
NEHFF	3.03	0.37	12.33
vN-NEH+(4)	2.96	0.44	12.82
NEHD-di	2.94	0.45	14.88
N-NEH+(3)	2.88	0.51	15.79
SP+(0.3)N+(2)	2.88	0.52	15.59
N-NEH+(4)	2.75	0.64	19.82
SMα+(8)N+(2)	2.74	0.65	19.43
vN-NEH+(8)	2.68	0.71	21.50
FRB4$_2$	2.65	0.73	22.35
SP+(0.3)N+(4)	2.57	0.82	25.03
N-NEH+(8)	2.47	0.91	28.54
SMα+(8)N+(4)	2.47	0.91	27.92
vN-NEH+(16)	2.40	0.98	29.99
FRB4$_4$	2.39	0.98	30.27
SP+(0.3)N+(8)	2.30	1.07	33.42
FRB4$_6$	2.25	1.12	34.45
N-NEH+(16)	2.22	1.15	36.09
SMα+(8)N+(8)	2.22	1.15	35.76
FRB4$_8$	2.15	1.22	37.65
SP+(0.3)N+(16)	2.06	1.31	40.77
FRB4$_{10}$	2.05	1.31	40.28
FRB4$_{12}$	2.02	1.34	41.15
SMα+(8)N+(16)	2.02	1.34	42.08
FRB2	1.82	1.53	47.38
FRB3	1.40	1.94	59.77
FRB5	1.12	2.21	68.37

[2] M. Garey and D. Johnson, *Computers and intractability*. San Francisco: W.H. Freeman, 1979, vol. 174.

[3] M. Nawaz, E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0305048383900889

[4] R. Ruiz and C. Maroto, "A comprehensive review and evaluation of permutation flowshop heuristics," *European Journal of Operational Research*, vol. 165, no. 2, pp. 479–494, 2005.

[5] B. Naderi and R. Ruiz, "The distributed permutation flowshop scheduling problem," *Computers & Operations Research*, vol. 37, pp. 754–768, 04 2010.

[6] E. Taillard, "Some efficient heuristic methods for the flow shop sequencing problem," *European Journal of Operational Research*, vol. 47, no. 1, pp. 65–74, 1990.

[7] V. Fernandez-Viagas and J. Framinan, "On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem," *Computers & Operations Research*, vol. 45, pp. 60–67, 2014.

[8] V. Fernandez-Viagas, R. Ruiz, and J. Framinan, "A new vision of approximate methods for the permutation flowshop to minimise makespan: State-of-the-art and computational evaluation," *European Journal of Operational Research*, vol. 257, no. 3, pp. 707–721, 2017.

[9] R. Puka, I. Skalna, J. Duda, and A. Stawowy, "*vN*-neh+ algorithm with modified n-list technique to solve the permutation flow shop problem with makespan criterion," 2022, http://dx.doi.org/10.2139/ssrn.4239708.

[10] E. Taillard, "Benchmarks for basic scheduling problems," *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993, project Management anf Scheduling. [Online]. Available: https://www.sciencedirect.com/science/article/pii/037722179390182M

[11] J. Gmys, "Exactly solving hard permutation flowshop scheduling problems on peta-scale gpu-accelerated supercomputers," *INFORMS Journal on Computing*, vol. 34, no. 5, pp. 2502–2522, 2022.

[12] E. Vallada, R. Ruiz, and J. Framinan, "New hard benchmark for flowshop scheduling problems minimising makespan," *European Journal of Operational Research*, vol. 240, no. 3, pp. 666–677, 2015.

[13] J.Gmys, M. Mezmaz, N. Melab, and D. Tuyttens, "A computationally efficient branch-and-bound algorithm for the permutation flow-shop scheduling problem," *European Journal of Operational Research*, vol. 284, no. 3, pp. 814–833, 2020.

[14] X. Dong, H. Huang, and P. Chen, "An improved NEH-based heuristic for the permutation flowshop problem," *Computers & Operations Research*, vol. 35, no. 12, pp. 3962–3968, 2008, part Special Issue: Telecommunications Network Engineering.

[15] P. Kalczynski and J. Kamburowski, "An improved NEH heuristic to minimize makespan in permutation flow shops," *Computers & Operations Research*, vol. 35, no. 9, pp. 3001–3008, 2008, part Special Issue: Bio-inspired Methods in Combinatorial Optimization.

[16] ——, "An empirical analysis of the optimality rate of flow shop heuristics," *European Journal of Operational Research*, vol. 198, no. 1, pp. 93–101, 2009.

[17] S. Rad, R. Ruiz, and N. Boroojerdian, "New high performing heuristics for minimizing makespan in permutation flowshops," *Omega*, vol. 37, no. 2, pp. 331–345, 2009.

[18] I. Ribas, R. Companys, and X. Tort-Martorell, "Comparing three-step heuristics for the permutation flow shop problem," *Computers & Operations Research*, vol. 37, no. 12, pp. 2062–2070, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030505481000050X

[19] K.-C. Ying and S.-W. Lin, "A high-performing constructive heuristic for minimizing makespan in permutation flowshops," *Journal of Industrial and Production Engineering*, vol. 30, no. 6, pp. 355–362, 2013. [Online]. Available: https://doi.org/10.1080/21681015.2013.843597

[20] R. Puka, J. Duda, A. Stawowy, and I. Skalna, "N-NEH+ algorithm for solving permutation flow shop problem," *Computers & Operations Research*, vol. 132, p. 105296, 2021.

[21] R. Puka, B. Łamasz, and I. Skalna, "Improving n-neh+ algorithm by using starting point method," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022, pp. 357–361.

[22] R. Puka, I. Skalna, and B. Łamasz, "Swap method to improve n-neh+ algorithm," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 2022, pp. 1–6.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1113–1117
DOI: 10.15439/2023F669
ISSN 2300-5963 ACSIS, Vol. 35

PSE for Analysis of 3D Tomographic Images in Materials Science

Paulo Quaresma [*], Pedro Medeiros [*], Adriano Lopes [†] and Alexandre Velhinho [‡]
[*]NOVA LINCS, Informatics Department, Faculty of Science and Technology, Universidade NOVA de Lisboa,
2829-516 Caparica, Portugal
Email: {pjq, pdm}@fct.unl.pt
[†] ISTAR-IUL, Information Science and Technology Department, Instituto Universitário de Lisboa (ISCTE-IUL),
1649-026 Lisboa, Portugal
Email: adriano.lopes@iscte-iul.pt
[‡]CENIMAT/i3N, Material Sciences Department, Faculty of Science and Technology, Universidade NOVA de Lisboa,
2829-516 Caparica, Portugal
Email: ajv@fct.unl.pt

***Abstract*—In the field of Materials Science, tomographic images play an important role in the analysis of composite materials. We present a computational environment that helps specialists in the field to carry out analysis and evaluation of samples of composite materials. This environment takes the form of a tailored Problem Solving Environment (PSE) and builds upon the SCiRun PSE. Its implementation is driven primarily by four major attributes: modularity, flexibility, interactivity and performance. Users can easily assemble networks of modules, with some of the modules being specifically designed for materials science analysis. These modules are flexible in terms of configuration, so yielding more flexibility to the setup of the networks, as well as in relation to the user interaction upon them once running. The implementation of data processing algorithms supporting critical modules rely on parallel programming. Furthermore, the quality of tomographic images under analysis is an issue of concern.**

I. Introduction

RESEARCHERS in the field of materials science use X-ray micro/nanotomography (mCT) for studying composite materials, namely for 3D geometrical characterization of the material's constituent phases. The tomographic image that is reconstructed using specialized software corresponds to a 3D matrix, where each voxel in space is usually represented by an integer corresponding to its grey-level.

On that basis, it is important to provide materials science specialists with proper software to accomplish the research goals set. In particular, researchers are mainly interested on:

- Visualizing data in 3D;
- To perform different image processing operations in order to remove any artifacts present in the image;
- To exclude irrelevant objects to the ongoing analysis;
- To perform image processing operations that label each of the distinct objects under consideration;
- To obtain geometric information that establishes a statistical description of the entire population of objects under consideration.

In this article we describe a framework for building flexible environments for the analysis of tomographic images of composite materials. Besides the normal operations we may expect to use in tools of this category, this framework is mostly concerned with usability and data quality issues that materials science specialists might face. They are:

- Modularity and flexibility, as the system must support an easy way of specifying the processing steps, and should allow to easily perform testing and reconfiguration tasks;
- Interactivity, in the sense that individual processing and visualization operations should be carried out faster since specialists want to see the outcome of those operations as quickly as possible, and also to allow a smooth steering of the computations;
- Tomographic image quality, since it should not be taken for granted that all images will show high contrast.

The organization of the paper is as follows: Section II presents related work that has been developed in the area of computational environments for analysing scientific data. Then, in Section III, we introduce a framework alongside guidelines to build a computational environment to process and analyse scientific data, followed in Section IV by an implementation with focus on data collected from material science experiments. In order to validate our proposal, we discuss a case-study in Section V, in particular concerning tomographic images with low contrast, so difficult to process, and finally Section VI wraps up with conclusions.

II. Related Work

Computational environments to process tomographic images can broadly be split into two major categories: environments that allow users to apply processing algorithms to tomographic images on a one-o-one basis, that is, with the simple paradigm *read-transform-visualize* in sight, and the so-called visual programming environments, more friendly but complex, which allow users to set up a network of processing modules via graphical deployment in a canvas, with related computations following the data-flow model [1].

This work was partially funded by National Funds through FCT – Portuguese Foundation for Science and Technology, Reference UID/CTM/50025/2019 and NOVA LINCS (UIDB/04516/2020) with the financial support of FCT.IP.

As inferred from above, visual programming environments allow specialists to specify a sequence of processing steps by choosing a set of modules available in a menu, including obviously reading tomographic images, and interconnects them. An example of those is the commercial software Avizo/Amira [2].

On the other hand, one can prefer to use the other category of environments, like the image processing and visualization software ImageJ/Fiji [3] or Paraview [4]. Worth pointing out that we can always follow the route of developing dedicated software, like the case of spam mentioned in [5].

In respect to visual toolkits mentioned above that rely on the data-flow model, sometimes referred to as PSEs due to its usability in various scientific areas, it is clear that a major advantage they present is that they can be tailored to the specific needs of a particular scientific area, yielding to dedicated environments. One example is SCIRun [6] from the University of Utah, USA. The toolkit has been very successful regarding the development of dedicated PSEs. For example, it is the case of BioPSE [7], which is specifically tailored for running bio-electric field simulations on top of SCIRun.

III. Framework

Given the information provided by practitioners in the materials science field, our understanding is that we should have an integrated software solution, embracing both image processing algorithms and visualization capabilities, but underlying a clean and easy-to-use approach. On that basis, the proposed solution will address primarily the following requirements:

- Open-source desktop solution but providing users interactivity and computational steering;
- Processing of tomographic images, including the ones with low contrast;
- Availability of various processing algorithms, even the complex ones requiring higher computational resources;
- Providing adequate data formats in accordance to the processing operations of concern.

We borrow the idea from the concept of PSE, sustained by the data-flow model [1], upon which SciRun is a prominent example. Hence, our solution provides four distinct categories of modules: data readers, filters, mappers and renders. Fig. 1 depicts the processing model we advocate.

Specialists will have at their disposal such modules to create networks, that ultimately will solve their problems. The networks will be managed by the specialists themselves. This includes the setting of control parameters and of both data and images to/from modules via input/output ports.

IV. Implementation

All the developed modules were built on top on SciRun. Notice that the interactivity and steering requirement is delivered by SciRun. Next, we will introduce new implemented modules, yielding to an open-source solution that works on desktops.

A major concern is that tomographic images showing up low-contrast between matrix and particles are challenging to identify and characterize objects – let us focus hereafter on particles. Common approaches sometimes fail to to so and some take too much time to deliver results. That is why we have taken a careful approach while designing those modules that are related the most. For example, modules belonging to the category *Filter* have been implement using OpenMP or CUDA, so a parallelization approach targeting both CPUs and GPUs. Operations that do occur at voxel level will take advantage of data parallelization.

In respect to visualization functionalities, we take advantage of native SciRun visualization modules, mostly for general 3D visualization. But for specific purposes, like visualizing and analysing particle features, specialists are able (i) to automatically launch external viewers and, importantly, (ii) to use a new 2D visualization module to check features on a image plane basis, regardless of its orientation in the 3D space. Also, for a better understanding, specialists can playback the outcomes of image operations that were applied, in sequence.

In relation to image operations, and among the various modules that have been implement, there are some operations that deserve to be singled out. They are: edge detection, segmentation, erode and dilate, and crucially particle identification and subsequent characterization.

Edge detection. This operation basically creates conditions to correctly identify particles. Examples of filters that relate to this task are *Unsharp* (mask to unsharping to enhance high-frequencies like boundaries), *Gradient* (first derivatives), *Laplacian* (second derivatives to enhance tiny boundaries), *Sobel* (Sobel derivatives to give direction of intensity variations) and *ZeroCrossings* (location based on 2nd derivatives).

Segmentation. The goal in this operation is to highlight particles within the raw image. This is carried out via the *Thresholding* filter: Assuming that we have a raw image defined in a gray scale, by applying the filter we get a black-and-white image. Also, we can chose to apply bi-segmentation, meaning that we end up with black, or white, or unchanged voxels. Crucially, this operation requires setting critical cut-off levels, usually inferred with the help of image histograms.

Erode and Dilate. These two operations working together help to achieve particle separation. It happens when somehow the boundaries of particles touch to each other, that is, it seems there are contiguous voxels but belonging to different particles. Notice that the given image is already in black-and-white. *Dilate* implies enlarging the border of a particle (white to black), whereas *Erode* is the converse operation. When there is a sequence *Erode* then *Dilate*, it is called *Open* operation. The reverse sequence is called *Close* operation.

Particle Identification. This crucial task implies the use of various filters supporting a bi-segmentation process, which is even more critical when the raw tomographic images show low contrast between matrix and reinforcements. It also relies on two modules: *ParticleLabelling* and *PoissonReconstruction*. The first one is based on a labelling algorithm [8] but implemented with OpenMP or CUDA, (there are two versions available, meaning it is up to the specialist to decide which version is going to be used) yielding to a fine-tuned parallel implementation to reduce execution times. The second module uses

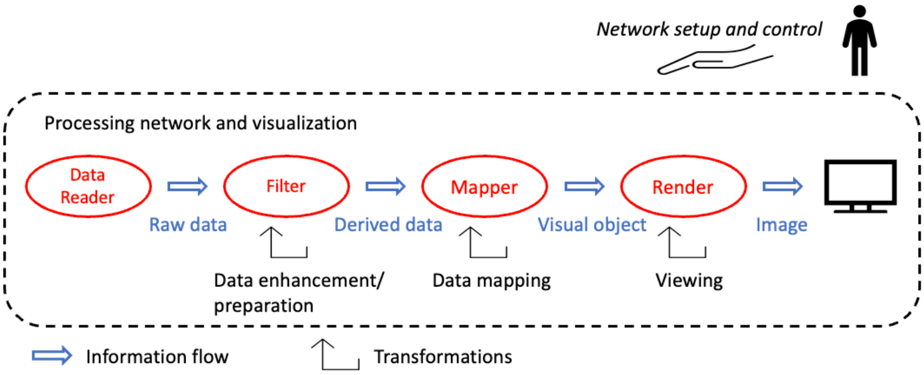


Fig. 1. Processing model based on the data-flow visualization paradigm.

the Possion surface reconstruction algorithm [9] available in the Point Cloud Library (PCL) library. (https://pointclouds.org) While the first module produces sets of interconnected voxels, (parts of potential particles) the second module takes a cloud of points as input (those voxels) and properly reconstructs the surfaces of the particles. Then, as output, polygonal meshes are generated to depict the reconstructed surfaces.

Particle Characterization. Once particles have been geometrically identified, specialists still have to further evaluate the outcome as it is presented to them. Notice that we may end up with clusters of particles or even *fake* particles. But in the end, the final decision about accepting or rejecting a particular particle rests on the specialists. The outcome will be a set of particles of interest and their characterization – the location within the sample, the geometric profile such as volume, area and bounding box, among other similar concepts.

Worth pointing out that the way data is organized affects the performance of applied algorithms. That is why it has been also implemented a set of rapid but robust data formats converters. Then, in a particular situation, specialists will decide which ones to use in order to achieve better efficiency and performance. Also, because the visual appearance of particles is helpful in the characterization process, specialists have at their disposal various visualization functionalities.

V. Evaluation

In order to validate the extended PSE, we discuss now a case-study concerning samples of aluminum as the base material (matrix) and tungsten carbide as reinforcements but showing low contrast between them. That is, if we were to draw an histogram of densities, it will not show two clear peaks – one corresponding to the matrix and another one to the reinforcements – as we were expecting to obtain in a clear bi-segmentation process.

The samples were collected at the European Synchrotron Radiation Facility in Grenoble and were defined in a regular grid. For evaluation purposes, we use a subset corresponding to a uniform 3D cube lattice of dimension $[512\times512\times432]$, with each voxel corresponding to one μm^3 approximately [10].

Looking at the raw tomographic images, they show low contrast between the base material and reinforcements, and contain various porous. (See Fig. 2) Furthermore, information gathered during the collecting process hinted that the particles were showing a cone-shaped, convex geometry, they could be broken, and the average size is about 35 μm.

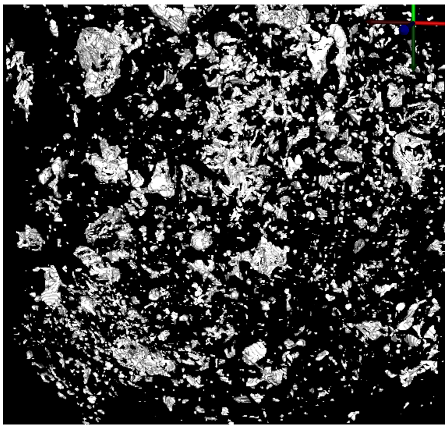


Fig. 2. Glimpse of a raw tomographic image prior to any processing.

Overall, the goal is to identify particles inside the sample an then to characterize the ones of interest. The sequence of operations works as follows:

1) Removal of porous;
2) Increasing contrast between particles and base material;
3) Particles labelling;
4) Particles detection;
5) Particles characterization.

The initial two operations are mostly supportive of the particles labelling process. Hence, and given the modules available, a typical workflow to accomplish the tasks mentioned can be split into three sequential stages: particles labelling, particles detection and finally particles characterization. In the following we will provide further details about these three stages.

A. Particles labelling

The goal here is to figure out potential locations of particles in the volumetric sample. It starts by smoothing the raw data, that is, reducing the noise in the tomographic image and then, in sequence, applying band pass filters, labelling the image, followed again with enhancement using pass filters.

For example, Fig. 3 shows a circular air porous that is going to be removed by first painting its interior with the colour of the matrix so once bi-segmentation is applied later on, it will be converted into matrix. At this point we are able to get an initial identification of reinforcements.

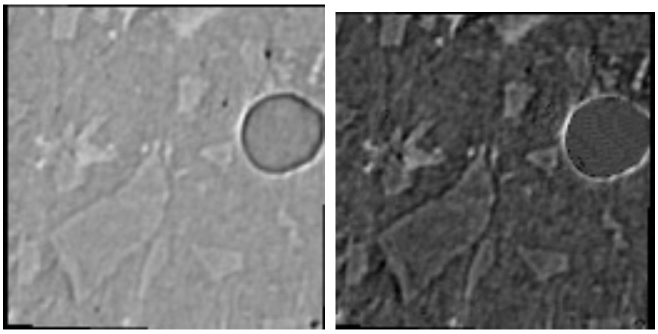


Fig. 3. Circular air porous (left) that will be removed once bi-segmentation is applied, but only after pre-painting its interior as matrix (right).

Then, it follows a bi-segmentation process using high and low pass filters, alongside operations to erode/dilate the outcome. The outcome will be regions of connected voxels that in the end may be considered as particles.

In this experiment we have identified 1 239 connected voxel regions at this stage, that is, 1 239 particle candidates.

Fig. 4 shows a network of modules to support the labelling process. Notice that some modules, like those related to pass filters, also output information to visualization modules so specialists can figure out the results of intermediate operations. This includes drawing histograms.

B. Particles detection

At this stage the goal is to figure out the proper boundaries of the real particles. It implies carrying out careful analysis in relation to potential regions of particles that have been considered in the previous stage, so we will end up with particles of potential interest, with proper closed boundaries.

As shown in Fig. 5, particles boundaries are not continuous at the beginning. Therefore, first it is required a reconstruction of the boundaries, which is done using the Poisson surface reconstruction algorithm. Only then we can compute the exact number of particles and respective size.

In this case-study, some of the 1 239 regions of connected voxels originated in the previous stage can still be considered as noise. Therefore, we have used a module to discard those *fake* particles. The cut-off size value set was 100 voxels, which is a value somehow derived from the pre-understanding and knowledge of the specialist about the sample. As result, there were 202 particles with acceptable size, that were then submitted to the Poisson surface reconstruction algorithm. The final outcome was a set of particles, described via a set of *Ply* files and representing polygonal meshes that can be visualized.

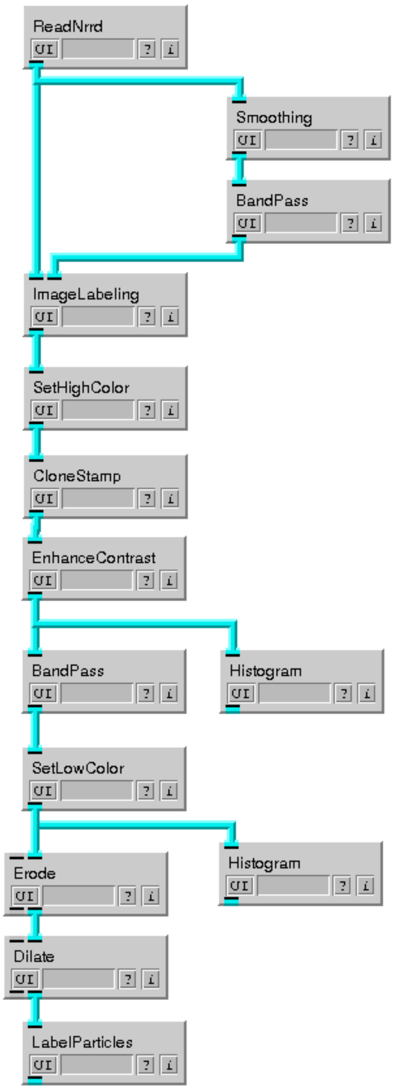


Fig. 4. Network of modules set by a specialist with the purpose of supporting particle labelling.

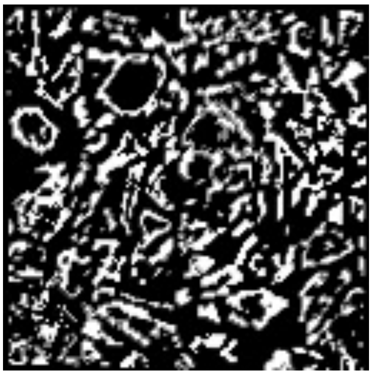


Fig. 5. Initially, particles boundaries are not continuous so proper identification and reconstruction is required.

C. Particles characterization

At this final stage, the goal is to deliver the set of particles the specialist is interested on, and with detailed characterization, mostly based on the geometric profile. It matters not only the selection itself but the quality of characterization.

As depicted in Fig. 6, given the detected particles from the previous stage (Ply files), which may not be entirely correct from a semantic point of view, we enter into an iterative process where, at each iteration, the specialist can accept a particular particle, or reject it, or else submit it to a enhancement process, likewise in the previous stage.

The decisions made are also supported by the viewing of particles, as highlighted in Fig. 7.

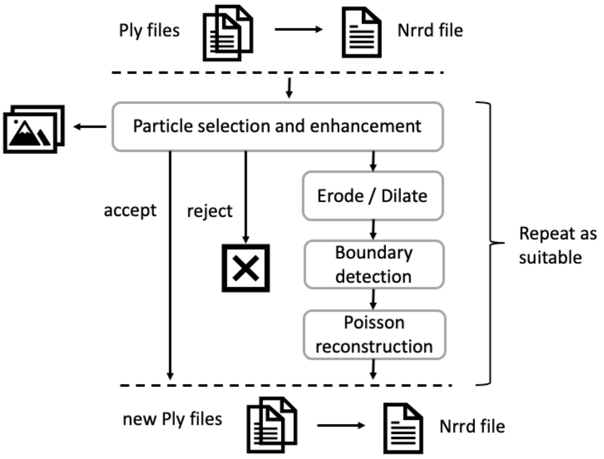


Fig. 6. Iterative process to select and enhance particles of interest for further characterization.

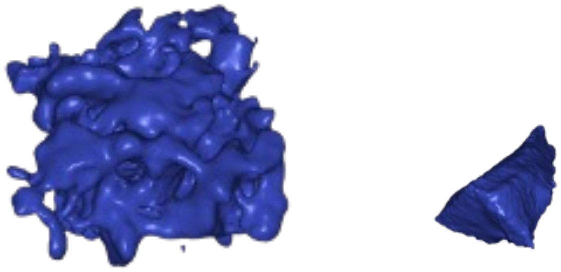


Fig. 7. Visualization to help selecting and enhancing particles of interest: From cluster requiring further processing (left) to accepted particle (right).

Also, the geometric profiles of the particles of interest are computed and stored in a SQLite database. Among other features, it includes the location within the sample, surface area, volume, bounding box, etc. It is worth noting that specialists have general pre-understanding about the samples they are working with, namely in relation to shape and average particle size.

The information stored in the database can be exported to files for further usage with other tools of convenience. Nonetheless, the implemented *FeatureVisualizer* module is tailor-made for the purpose of visualizing particle features within the context of the PSE itself. The underlying thinking is that it is important to provide extra flexibility to specialists.

VI. Conclusions

We have presented a dedicated PSE to help material science specialists to carry tasks of analysing tomographic images. The proposed solution fits into the list of requirements set by specialists, who were keen on having an environment where (i) they could easily use a wide range of algorithmic strategies to carry out their experiments, (ii) the build up of the processing network was done in a flexible manner and (iii) they should experience human-computer interactivity as much as possible, alongside fast and effective visualizations.

As highlighted in the discussion about particles identification in Section V, tomographic images with low contrast pose additional demands as far as type and number of operations that have to be included in the processing network. In the case-study introduced, the system delivered the results specialists were looking for from a scientific perspective. Also, the network of modules was relatively easy to set up and its steering afterwards was effective.

As a final note, once particles are properly identified and stored in a database, and having various data formats at disposal, a specialist can also use external tools to further analyse the outcome of the experiment, all but in a cohesive working environment.

References

[1] R. Haber and D. McNabb, *Visualization idioms: A conceptual model for scientific visualization systems*. IEEE, 1990, p. 74–93.

[2] D. Stalling, M. Westerhoff, and H.-C. Hege, “Amira – a highly interactive system for visual data analysis,” in *The Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds., 2005, pp. 749–767.

[3] W. Burger and M. Burge, *Digital Image Processing in Java*. Berlin, Heidelberg: Springer-Verlag, 2007. ISBN 1846283795

[4] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. Clifton Park, NY, USA: Kitware, Inc., 2015. ISBN 1930934300

[5] O. Stamati, E. Andò, E. Roubin, R. Cailletaud, M. Wiebicke, G. Pinzón, C. Couture, R. Hurley, R. Caulk, D. Caillerie, T. Matsushima, P. Bésuelle, F. Bertoni, T. Arnaud, A. Ortega Laborin, R. Rorato, S. Yue, A. Tengattini, O. Okubadejo, and G. Birmpilis, “spam: Software for practical analysis of materials,” *Journal of Open Source Software*, vol. 5, p. 2286, 07 2020. doi: 10.21105/joss.02286

[6] D. M. Weinstein, S. G. Parker, J. Simpson, K. Zimmerman, and G. M. Jones, “Visualization in the scirun problem-solving environment,” in *The Visualization Handbook*, C. D. Hansen and C. R. Johnson, Eds. Academic Press/Elsevier, 2005, pp. 615–632. ISBN 978-0-12-387582-2

[7] S. Computing and I. I. (SCI), “Biopse: Problem solving environment for modeling, simulation, image processing, and visualization for biomedical computing applications,” 2014. [Online]. Available: http://www.scirun.org

[8] B. Preto, F. Birra, A. Lopes, and P. Medeiros, “Object identification in binary tomographic images using gpgpus,” *International Journal of Creative Interfaces and Computer Graphics*, vol. 4, no. 2, p. 40–56, 2013.

[9] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP ’06. Goslar, DEU: Eurographics Association, 2006. ISBN 3905673363 p. 61–70.

[10] A. Velhinho, P. Sequeira, F. B. Fernandes, J. D. Botas, and L. A. Rocha, “Al/sicp functionally graded metal-matrix composites produced by centrifugal casting: effect of particle grain size on reinforcement distribution,” in *Materials Science Forum*, vol. 423, 2003, pp. 257–262.

DOI: 10.15439/2023F592
ISSN 2300-5963 ACSIS, Vol. 35

Path Length-Driven Hypergraph Partitioning: An Integer Programming Approach

Julien Rodriguez[1,2], François Galea[1], François Pellegrini[2], Lilia Zaourar[1]
0000-0003-3583-0859 - 0000-0002-1594-152X - 0000-0003-3983-6289 - 0000-0002-6660-4347
Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France[1],
Université de Bordeaux, LaBRI et INRIA[2]
name.surname@cea.fr[1]; francois.pellegrini@u-bordeaux.fr

***Abstract*—Circuit prototyping on multi-FPGA (Field Programmable Gate Arrays) platforms is a widely used technique in the VLSI (Very-Large-Scale Integration) context. Due to the ever-increasing size of circuits, it is necessary to use partitioning algorithms to place them on multi-FPGA platforms. Existing partitioning algorithms focus on minimizing the cut size but do not consider the critical path length, which can be degraded when mapping long paths to multiple FPGAs. However, recent studies try to consider the degradation of the critical path and the target topology but these works still use cutting minimization algorithms. In this work, we propose a mathematical model as an integer program (IP) based on the Red-Black Hypergraph model that considers the minimization of the critical path degradation and the target topology. We compare our partitioning results with KHMETIS, a min-cut algorithm, and show a better critical path for many circuit instances.**

I. INTRODUCTION

OUR work concerns practical improvements of the electronic circuit design chain. The typical hardware design flow includes different steps, such as floor planning, placement, and routing, that may concern very large logic circuits. To deal with such large circuits, the methods involved may benefit from divide-and-conquer approaches that allow for working locally on separate parts of the circuit, greatly reducing the work on the global circuit. Such a divide-and-conquer approach also enables circuit prototyping on a multi-FPGA platform, where the circuit is too large (in terms of resource consumption) to be implemented on a single FPGA. In such cases, a strong constraint is to mitigate a possible increase in the signal propagation delay of the longest combinatorial path, known as the *critical path*. Indeed, in synchronous circuits, the critical path length determines the maximum frequency at which the circuit may operate; mapping long paths across several FPGAs is likely to degrade the critical path.

Circuit partitioning is both an essential step in the design flow of electronic circuits, and a challenging multi-constraint optimization problem. It must address both the multi-resource issue (i.e., capacity limits on each FPGA and their interconnection links) and the minimization of the critical path degradation.

Traditional partitioning tools use the now classic multi-level scheme (see Fig. 1) consisting of three phases: *coarsening*, *initial partitioning*, and *refinement* [1]. The coarsening phase uses a recursive clustering method to transform the circuit model, a hypergraph, into a smaller one. During the second phase, an initial partitioning is computed on the smallest coarsened hypergraph. Finally, for each coarsening level, the solution for the coarser level is extended to the finer level and then refined using a local refinement algorithm. The initial partitioning algorithm presented in this paper concerns the first step of the multilevel framework described above.

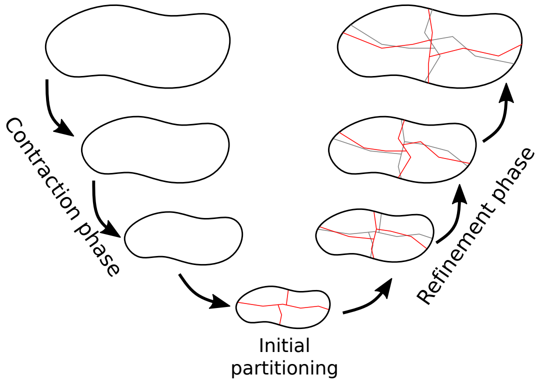


Fig. 1. Multi-level scheme

Our work focuses on balanced hypergraph partitioning, in which our objective functions are both *path-cost* minimization and the classical *min-cut* objective that is still relevant to us. The hypergraph model we consider in our research context consists of a union of directed acyclic hypergraphs (DAH) [2]. The global hypergraph is assumed to be connected; otherwise, its disconnected components are processed independently. The source and sink vertices of each DAH (which represent registers and I/O ports) are labeled red, while other vertices are black. Red vertices can be shared by multiple DAHs, which makes the global hypergraph connected. A *path-cost* function models the impact of a cut on the red-to-red paths during partitioning. Each partition of a hypergraph will result in cuts along some paths, inducing additional traversal costs. Our aim is to find a partition of minimum path cost, such that the size of the cut is also minimized. Our research context only considers the paths between two red vertices and a non-uniform cut cost between parts.

The classical approach is to model this problem with a hypergraph, using cost functions that minimize cut size. However, it has been shown in [3] that the cut size does not address the path cost efficiently during the hypergraph partitioning procedure. This is why several authors proposed pre- and/or post-processing steps in order to reduce the degradation of cut paths [3], [4], [5]. In this paper, we devise a dedicated integer programming model that minimizes path cost degradation during partitioning, based on the red-black hypergraph structure, which can be used as an initial partitioning method in a multilevel framework.

The remainder of the paper is organized as follows: Section 2 presents a reminder of our red-black hypergraph structure as well as previous works. Section 3 describes our coarsening scheme before the initial partitioning and the integer programming model. Our experiments are outlined in Section 4. We conclude and give perspectives in Section 5.

II. Preliminaries

In this part, we define the notations and definitions used in this work.

A. Definitions and Notations

Let $\mathcal{H} \overset{\text{def}}{=} (\mathcal{V}, \mathcal{A}, \mathcal{W}_v, \mathcal{W}_a)$ be a directed hypergraph, defined by a set of vertices $\mathcal{V}$ and a set of hyperarcs $\mathcal{A}$, with a vertex weight function $\mathcal{W}_v : \mathcal{V} \to \mathbb{R}^+$ and a hyperarc weight function $\mathcal{W}_a : \mathcal{A} \to \mathbb{R}^+$. Every hyperarc $a \in \mathcal{A}$ is a subset of vertex set $\mathcal{V}$: $a \subseteq \mathcal{V}$. Let $s^+(a)$ be the source vertex set of hyperarc a, and $s^-(a)$ its sink (destination) vertex set. We consider here, without loss of generality, that each hyperarc has a single source, so $\forall a, |s^+(a)| = 1$. As hyperarcs connect vertices, let $\Gamma(v)$ be the set of neighbor vertices of vertex v, and $\Gamma^-(v) \subseteq \Gamma(v)$ and $\Gamma^+(v) \subseteq \Gamma(v)$ the sets of its inbound and outbound neighbors, respectively.

In the model we propose, hypergraphs that model circuits are be represented as sets of interconnected DAHs, according to a red-black vertex coloring scheme. Red vertices correspond to I/O (Inputs/Outputs) ports and registers, and black vertices to combinatorial circuit components. Let $\mathcal{V}^R \subset \mathcal{V}$ and $\mathcal{V}^B \subset \mathcal{V}$ be the red and black vertex subsets of $\mathcal{V}$, such that $\mathcal{V}^R \cap \mathcal{V}^B = \emptyset$ and $\mathcal{V}^R \cup \mathcal{V}^B = \mathcal{V}$. A hypergraph or sub-hypergraph $\mathcal{H}$ is a DAH iff its red vertices $v_R \in \mathcal{V}^R$ are either only sources or sinks (*i.e.*, $\Gamma^-(v_R) = \emptyset$ or $\Gamma^+(v_R) = \emptyset$), and no cycle path connects a vertex to itself.

Using this definition, we can represent circuit hypergraphs as *red-black hypergraphs*, *i.e.*, sets of DAHs that share some of their red vertices. Let $\mathbf{H}(\mathbf{V}, \mathbf{A}) \overset{\text{def}}{=} \{\mathcal{H}_i, i \in \{1 \ldots n\}\}$ be a red-black hypergraph, such that every $\mathcal{H}_i$ is a DAH and an edge-induced sub-hypergraph of $\mathbf{H}$. Consequently, $\mathbf{V} = \bigcup_i \mathcal{V}_i$, $\mathbf{A} = \bigcup_i \mathcal{A}_i$, $\mathbf{V}^R = \bigcup_i \mathcal{V}_i^R$, and $\mathbf{V}^B = \bigcup_i \mathcal{V}_i^B$. Moreover, $\forall i, j$ with $i \neq j$, if $\mathcal{V}_{i,j} = \mathcal{V}_i \cap \mathcal{V}_j \neq \emptyset$, then $\mathcal{H}_i$ and $\mathcal{H}_j$ share source and/or sink vertices, *i.e.*, $\mathcal{V}_{i,j} \subset \mathbf{V}^R$.

In this model, the paths in $\mathbf{H}$ to consider when addressing the objective of minimizing path-cost degradation during partitioning are only the paths interconnecting red vertices, as these red-red paths represent register-to-register paths in combinatorial circuits. Since only red vertices are shared between DAHs in $\mathbf{H}$, red-red paths only exist within a single DAH and can never span across several DAHs.

Let us define $\mathbf{P}$ as the set of red-red paths in $\mathbf{H}$, such that $\mathbf{P} \overset{\text{def}}{=} \{p | p \text{ is a path in } \mathcal{H} \in \mathbf{H}\}$. From these paths and a function $d_{\max}(u, v)$, which computes the maximum distance between vertices u and v of some DAH $\mathcal{H}$, we can define the longest path distance for $\mathcal{H}$ as: $d_{\max}(\mathcal{H}) \overset{\text{def}}{=} \max(d_{\max}(u, v) | u, v \in \mathcal{H})$ and, by extension, for $\mathbf{H}$, as: $d_{\max}(\mathbf{H}) \overset{\text{def}}{=} \max(d_{\max}(\mathcal{H}) | \mathcal{H} \in \mathbf{H})$.

A partition Π of $\mathbf{H}$ is a splitting of $\mathbf{V}$ into vertex subsets π_i, called parts, such that:

(i) all parts π_i, given a capacity bound M, respect the capacity constraint:

$$\sum_{v \in \pi_i} \mathcal{W}_v(v) \leq M$$

(ii) all parts are pairwise disjoint:

$$\forall i \neq j, \pi_i \cap \pi_j = \emptyset$$

(iii) the union of all parts is equal to $\mathbf{V}$:

$$\bigcup_i \pi_i = \mathbf{V}$$

Consequently, in our model, the distance between two vertices u and v may increase during partitioning due to the additional cost of routing paths between two (or more) parts. Let $D_{kk'}$ be the penalty associated with parts k and k' such that if u is in part k and v is in part k', then:

$$d_{\max}^{\Pi}(u, v) \geq d_{\max}(u, v) + D_{kk'} \tag{1}$$

For a given partition Π of $\mathbf{H}$, the *path-cost* is defined by the function: $f_p(\mathbf{H}^{\Pi}) = \max(d_{\max}(\mathcal{H}^{\Pi}) | \mathcal{H} \in \mathbf{H})$.

Let a red-black hypergraph $\mathbf{H}$ and a partition Π, the connectivity $\lambda_{\Pi}(a)$ of some hyperarc $a \in \mathcal{A}$ is the number of parts connected by a. If $\lambda_{\Pi}(a) > 1$, then a is said to be cut; otherwise, it is entirely contained within a single part and is not cut. The cut of partition Π is the set $\omega(\Pi)$ of cut hyperarcs, *i.e.*, $\omega(\Pi) \overset{\text{def}}{=} \{a \in \mathcal{A}, \lambda_{\Pi}(a) > 1\}$. The cut size is defined as $f_c \overset{\text{def}}{=} \sum_{a \in \omega(\Pi)} \mathcal{W}_a(a)$. If all hyperarcs have the same weight (equal to 1), the cut size is equal to $|\omega(\Pi)|$. Another cut metric used by some partitioning tools to measure the quality of partitioning is called *connectivity-minus-one* [6]. The connectivity-minus-one cost function f_λ of some partitioned hypergraph $\mathbf{H}^{\Pi}$ is defined as: $f_\lambda = \sum_{a \in \mathbf{A}} (\lambda_{\Pi}(a) - 1) \times \mathcal{W}_a(a)$.

B. Previous work

Several approaches in the literature have been attempted to improve the performance of circuit partitioning. We present some recent work on circuit partitioning for rapid prototyping that considers performance constraints. Many of these works attempt to tweak existing *min-cut* partitioning tools, which are used as black boxes, to consider additional constraints. For example, [3] presents a multi-objective approach based on hMetis. The authors compute the K most critical paths at each partitioning step, using a metric cost that considers the critical path length, the cut number along critical paths, and the weight of the hyperarcs associated with the critical paths. Reference [4] compares a classical method using hMetis for partitioning followed by a placement algorithm with a derived approach consisting of placement and routing during the partitioning step. The results show better critical path values compared to the two-step approach. More recently, [5] performs some pre- and post-processing on the hypergraph to capture the critical path minimization objective within the cut-size metric, using hMetis as the partitioning tool. Reference [7], presents an IP model to address the hypergraph partitioning problem. The model is not dedicated to mapping and critical path minimization but to minimize the cut cost.

III. Contributions

We now present our core contribution. The first part consists of a coarsening algorithm to reduce the size of the hypergraph. In the second part, we present our IP model used as initial partitioning.

A. Coarsening method

The heavy-edge matching (HEM) approach for graph coarsening presented in [8] is widely used in hypergraph and graph partitioning tools [9], [10] and yields efficient results in many cases. Our coarsening algorithm is based on a heavy-edge matching approach. It consists of reducing the instance's size while minimizing the merged vertices' weight differences as much as possible. The risk in merging vertices is to end up with disproportionate weights of vertices, which may prevent the initial partitioning from exploring different solutions. However, in the context of critical path minimization, it may be interesting to merge all vertices along the critical path into one large vertex. This method is not necessarily interesting when the circuit contains many critical or semi-critical paths. It is, therefore, necessary to find a compromise between creating a large vertex by securing the cut along the critical path and balancing the fusion to allow a more practical exploration search during the initial partitioning phase. The vertex criticality model the value of the longest path traversing the vertex. Our algorithm groups vertices by criticality to favor the grouping of critical paths. Vertices with a smaller weight are selected to favor balanced coarsening.

TABLE I
Indices and set definitions

Set	Definition
$\mathbf{V}$	set of vertices
$\mathbf{E}$	set of hyperedges
J	set of jobs
O_l	ordered set of operations of job l, ($i \in O_l$), where O_{l1} and $O_{ln'}$ are the first and the last elements of O_l
i,i'	vertices/operation index (i, $i' \in \mathbf{V}$)
j, j'	hyperedges index ($j, j' \in \mathbf{E}$)
l, l'	job index ($l, l' \in J$)
k	part index

TABLE II
Parameters definitions

Parameter	Definition
n	number of vertices
m	number of hyperedges
h_{ij}	1 if vertex i is connected to hyperedge j, 0 otherwise
c_{kr}	capacity of part k for resource r
q_{ir}	quantity of resource r, required by i
d_i	propagation time of vertices (operation) i
$D_{k,k'}$	delay between part k and k'
$\mathcal{W}_v$	vertex weight
$\mathcal{W}_a$	hyperedge weight

B. Integer Program

The objective of the IP model is to minimize the degradation of the critical path, so we need to calculate the maximum degradation among all possible degradations. We also need to model the target topology to consider the different delays between each part. Cut minimization tools do not address these two aspects: path length and topology. Cut minimization tools only limit the connections between parts. As this objective is still essential in practice, we add a second objective to our model: minimizing the connectivity minus one. As the paths between two red vertices do not contain cycle, it is possible to see the chain of black vertices in a path as a sequence of operations/tasks i associated with a job l. In our model, we consider scheduling constraints to minimize the impact of partitioning on the critical path. Given a path (job) $p = v_0, v_1, v_2$, the critical time associated with the path equals $\sum_{v \in p} d_v$. If vertices (tasks) belonging to p are placed in different parts, then a time penalty must be added to the total time of p. A summary of the integer model can be found in Table I, the parameters in Table II, and the variables in Table III. Below is the integer program with two objectives 2a for critical path minimization and 2b for connectivity cost minimization:

$$\min z_{\max} \tag{2a}$$

$$\min \sum_j \mathcal{W}_a^j \left(\sum_k y_{jk} - 1 \right) \tag{2b}$$

$$\text{subject to} : \sum_k x_{ik} = 1, \quad \forall i \tag{2c}$$

$$h_{ij} x_{ik} \leq y_{jk}, \quad \forall i, j, k \tag{2d}$$

$$\sum_i q_{ir} x_{ik} \leq c_{kr}, \quad \forall k, r \tag{2e}$$

$$\sum_{i,i' \in O_l} d_i + x_{ik} x_{i'k'} D_{kk'} \leq z_l, \quad \forall k, k', l \tag{2f}$$

$$z_l \leq z_{\max}, \quad \forall l \tag{2g}$$

$$x_{ik}, y_{jk} \in \{0, 1\}, z_l \in \mathbb{N} \quad \forall i, j, k, l \tag{2h}$$

Constraint 2c states that each vertex is mapped onto one part. Constraint 2d guarantees that y_{jk} equals the connectivity cost

TABLE III
VARIABLES DEFINITIONS

Variable	Definition
x_{ik}	1 iff the vertex i is mapped onto part k, 0 otherwise
y_{jk}	1 iff the hyperedge j has a vertex placed on part k
z_l	completion time of job l
$z_{\max}$	maximum completion time of jobs

associated with hyperedge j. The constraint 2e ensures the capacity constraint is respected. The constraints 2f and 2g determine the value of the delay of the job (path) and the maximum delay (*critical path*). The constraint 2h are the non-negativity and integrity conditions on the variable.

There are symmetries in the solution space in hypergraph partitioning for cut size minimization. Indeed, if there are ω hyperedges between parts, ω remains unchanged regardless of the labels of the parts. On the other hand, in our problem, we are trying to minimize the path cost, which is degraded by routing paths between parts that are not always fully connected. There are models for partitioning graphs and hypergraphs with symmetry-breaking constraints [11]. However, these constraints are too restrictive for the solution space associated with path cost. In our problem, the target topology defines a time penalty associated with path routing. As a result, we cannot consider all partitions with the same subset of vertices but different labels, identical, from a routing point of view. An example can be found in Figure 2. Note that some symmetries exist, for example: if we take the partition a, shown in Figure 2. It is possible to create a partition a' by swapping the vertices of π_0 and π_3 and of π_1 and π_2. Future work will involve improving the model to remove these symmetries.

IV. EXPERIMENTAL RESULTS

To validate our models and algorithms, we have performed experiments on benchmarks [12] of logic circuits. These circuits consist of acyclic combinatorial blocks, bounded by their input and output registers. Every combinatorial block can therefore be modeled as a DAH. Their computation time is conditioned by their critical path, defined as the longest path between two registers (*i.e.*, two red vertices). Our work aims at minimizing the degradation of the critical path during partitioning according to the target topology. For each instance, we use topology data to define a traversal cost $d(v)$ for each vertex v, corresponding to the traversal time of a logic element. As the degradation between the parts can be non-homogeneous, we have defined several architecture topologies composed of four elements. The test architecture is a chain $\pi_0, \pi_1, \pi_2, \pi_3$. We did not consider the fully connected topology to highlight the advantage of our topology-aware algorithms over regular partitioners like KHMETIS. To solve the initial partitioning problem, we use Gurobi Optimiser version 9.1.2 with a time limit set to $600s$. During the refinement phase, we use the DKFM [2], a local search algorithm dedicated to minimizing path length. This algorithm is inspired by FM [13], a local search algorithm for minimising the number of hyperedges between two parts. We use KHMETIS rather than HMETIS because HMETIS is based on recursive bipartitioning methods, which often do not respect the balance constraint. We use the maximum criticality as a weight for the hyper-edges as [2] to guide KHMETIS to minimise the number of cuts along the critical path as much as possible.

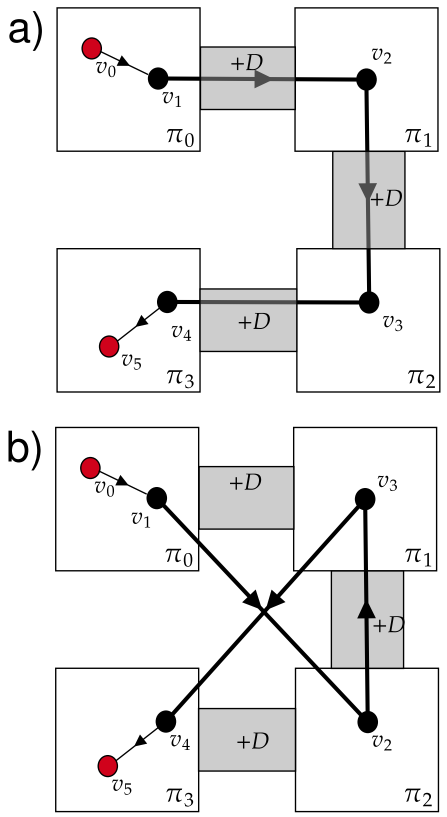


Fig. 2. In this example, the path $p = v_0, v_1, v_2, v_3, v_4, v_5$ is partitioned into 4 parts. In partition a, the path admits a routing penalty of $3D$, where D is the traversal time between parts. In partition b, the routing penalty is $5D$. Since there is no route between π_0 and π_2, we must necessarily pass through π_1 to get there, which gives a cost of $2D$ to get from π_0 to π_2. The same goes for π_1 to π_3. From the point of view of the size of the cut, partition a allows a cost of 3 cut edges, as does partition b. Partitions a and b are identical and symmetrical for cut minimization.

A. Results

Table IV shows that our approach gives better results. Indeed, the first coarsening step allows the grouping of the most critical vertices while maintaining a balance in the reduced hypergraph. Finally, since the initial partitioning considers the topology, it allows for finding an appropriate placement before the refinement phase. For instances B14 and B17, the time limit is not sufficient for Gurobi to find a good solution. A method needs to be found to better reduce the size of the instance while retaining sufficient criticality information for the integer program. Table V shows us a better performance

TABLE IV
RESULTS FOR PATH-COST f_p IN NANO-SECONDE (NS)

Instance	KHMETIS (ns)	Multilevel+IP+DKFM (ns)
b01	60	**50**
b02	**30**	**30**
b03	50	**40**
b04	**60**	**60**
b05	**50**	**50**
b06	40	**30**
b07	90	**60**
b08	90	**70**
b09	**40**	**40**
b10	**80**	**80**
b11	80	**70**
b12	**40**	**40**
b13	30	**20**
b14	**40**	100
b17	**200**	215.52

of KHMETIS for the function f_λ. Note that our approach sometimes allows a better solution for both f_p and f_λ.

TABLE V
RESULTS FOR CONNECTIVITY f_λ

Instance	KHMETIS	Multilevel+IP+DKFM
b01	**15604**	24288
b03	13903	**10922**
b03	**21297**	29962
b04	**38320**	103659
b05	**30753**	70329
b06	20526	**19043**
b07	**30372**	114329
b08	**28170**	44511
b09	24830	**24445**
b10	**32989**	41600
b11	**49883**	57329
b12	**30743**	99448
b13	**4567**	6000
b14	**214772**	1578740
b17	**846531**	3149961

V. CONCLUSION

In this paper, we present a multilevel approach to the problem of red-black hypergraph (circuit) partitioning on not fully connected topologies. Our approach consists of exploiting the vertices' criticality to group the critical paths in the same part during the coarsening phase. Finally, we propose a mathematical model considering the two objectives: f_p and f_λ for the initial partitioning. For the refinement, we use the DKFM algorithm. Our results show that our approach is better at minimizing f_p than a min-cut partitioning tool, even if it is oriented towards the criticality of hyperarcs. It may be interesting to test our approach on other more extensive benchmarks, as well as to test other coarsening algorithms to improve the results of initial partitioning based on our IP.

REFERENCES

[1] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: applications in VLSI domain," *IEEE Transactions on VLSI Systems*, vol. 7, no. 1, pp. 69–79, 1999.

[2] J. Rodriguez, F. Galea, F. Pellegrini, and L. Zaourar, "A hypergraph model and associated optimization strategies for path length-driven netlist partitioning," in *International Conference on Computational Science*. Springer, 2023, pp. 652–660.

[3] C. Ababei, S. Navaratnasothie, K. Bazargan, and G. Karypis, "Multi-objective circuit partitioning for cutsize and path-based delay minimization," in *IEEE/ACM ICCAD 2002.*, 2002, pp. 181–185.

[4] M.-H. Chen, Y.-W. Chang, and J.-J. Wang, "Performance-driven simultaneous partitioning and routing for multi-fpga systems," in *2021 58th ACM/IEEE DAC*, 2021.

[5] S.-H. Liou, S. Liu, R. Sun, and H.-M. Chen, *Timing Driven Partition for Multi-FPGA Systems with TDM Awareness*. New York, NY, USA: Ass. Comp. Mach., 2020, p. 111–118. [Online]. Available: https://doi.org/10.1145/3372780.3375558

[6] U. Çatalyürek, K. Devine, M. Faraj, L. Gottesbüren, T. Heuer, H. Meyerhenke, P. Sanders, S. Schlag, C. Schulz, D. Seemaier, and D. Wagner, "More recent advances in (hyper)graph partitioning," *ACM Computing Surveys*, vol. 55, no. 12, mar 2023. [Online]. Available: https://doi.org/10.1145/3571808

[7] D. Kucar, S. Areibi, and A. Vannelli, "Hypergraph partitioning techniques," *DYNAMICS OF CONTINUOUS DISCRETE AND IMPULSIVE SYSTEMS SERIES A*, vol. 11, pp. 339–368, 2004.

[8] G. Karypis and V. Kumar, "Analysis of multilevel graph partitioning," in *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing*, ser. Supercomputing '95. New York, NY, USA: Association for Computing Machinery, 1995, p. 29–es. [Online]. Available: https://doi.org/10.1145/224170.224229

[9] K. George and K. Vipin, "Hmetis: a hypergraph partitioning package," *ACM Transactions on Architecture and Code Optimization*, 1998.

[10] F. Pellegrini, "Scotch and PT-Scotch Graph Partitioning Software: An Overview," in *Combinatorial Scientific Computing*, O. S. Uwe Naumann, Ed. Chapman and Hall/CRC, 2012, pp. 373–406. [Online]. Available: https://hal.inria.fr/hal-00770422

[11] P. Bonami, V. H. Nguyen, M. Klein, and M. Minoux, "On the solution of a graph partitioning problem under capacity constraints," in *Combinatorial Optimization: Second International Symposium, ISCO 2012, Athens, Greece, April 19-21, 2012, Revised Selected Papers 2*. Springer, 2012, pp. 285–296.

[12] F. Corno, M. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and first ATPG results," *IEEE Design & Test of Computers*, vol. 17, no. 3, pp. 44–53, 2000.

[13] C. Fiduccia and R. Mattheyses, "A linear-time heuristic for improving network partitions," in *19th DAC*, 1982.

DOI: 10.15439/2023F2510
ISSN 2300-5963 ACSIS, Vol. 35

Laundry Cluster Management Using Cloud

Mateusz Salach*, Bartosz Trybus†, Bartosz Pawłowicz‡, Marcin Hubacz†
*Department of Complex Systems
0000-0002-9199-3460
Rzeszow University of Technology
al. Powstancow Warszawy 12, 35-959 Rzeszow
Email: m.salach@prz.edu.pl
†Department of Computer and Control Engineering
0000-0002-4588-3973
0000-0002-2748-1145
Email: btrybus@prz.edu.pl, m.hubacz@prz.edu.pl
‡Department of Electronic and Telecommunications Systems
0000-0001-9469-2754
Email: barpaw@prz.edu.pl

***Abstract*—Electronic devices in the 21st century have numerous network components, including wireless or wired Internet access modules. Connecting devices to networks and cloud services enables them to access new functionalities and unlock system updates and device security enhancements. The article presents the concept of an intelligent laundry management system based on RFID and cloud computing. The Internet connection not only unlocks additional features of the washing machine, such as different washing modes, but also allows for selecting the appropriate detergent level and washing parameters based on the textile material being washed. Additionally, the paper presents the solution and measurement studies on the accuracy of textile identification.**

I. Introduction

Technological development can be observed in many aspects of human life. Modern solutions can be seen all around, for example, controlling home facilities from a smartphone [1], shopping online using a phone, computer, or even autonomously by a refrigerator. The use of technology facilitates work, life, and provides more opportunities for self-focus. Solutions based on the Internet of Things (IoT) are gaining significant popularity, as well as the increasingly popular idea of the Internet of Everything (IoE) [2]. The concept of IoT devices is to establish a stable connection to both home and public computer networks, as well as the cloud, which enables remote administration of the device, whether it's a refrigerator, car, or a home automation system. The wide availability of services and possibilities can influence living conditions if one knows how to use them physically and safely. However, in many cases, the solutions proposed by manufacturers are unsafe [3], [4]. IoT security measures are often minimal, limited to simple mechanisms. Such solutions provide significant opportunities for hackers who can, for example, gain access to a building by attacking a home appliance, thus gaining entry to the main home network. The analysis of IoT devices from a cybersecurity perspective is particularly emphasized in the era of a very large number of IoT devices in residential buildings. Research allows for the detection of new methods of authentication and security measures used in various IoT device [5],[6].

IoT devices are most commonly equipped with WiFi modules or, in the case of specialized devices, Ethernet modules, which, when connected to the global network, unlock additional functionalities. These devices can be based on less advanced controllers such as NodeMCU [7],[8], ESP [9],[10], Raspberry Pi [11], as well as more advanced modules like PLC controllers [12]. Such solutions can utilize artificial intelligence algorithms for efficient management of electricity consumption [13],[14] and [15], or heating systems [16],[17],[18], not only in individual buildings but also in a cluster of buildings managed from the cloud [19],[20] and [21]. Due to the availability of components and a substantial base of online tutorials, it is possible to build custom Smart Home solutions using popular microcontrollers such as Arduino, ESP, etc. However, a key element is securing the entire system against external attacks.

As mentioned, IoT devices unlock most of their functionality through connection to the cloud computing. Devices equipped with a series of cameras can analyze their content and react accordingly, for example, by notifying the user that it's time to go shopping. Cars equipped with temperature sensors can inform the user through a push notification on their phone, via a dedicated application, that the car is heated or cooled and ready to drive. The amount of data required for processing and the memory needed to store the database is often so large that it is impossible to store all the requested information locally. Thus, in the case of many devices working together, the communication requires a connection to the cloud computing in order to properly manage the data exchange.

An example of such a solution can be a system of intelligent washing devices currently under development, as presented in this paper. It is assumed here, that the washing appliances have additional modules that allow for management and adjustment of washing parameters for textile materials using an RFID system. The concept of integrating washing devices with the

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

cloud computing is demonstrated using the example of the Microsoft Azure IoT Hub service and its services, RFID system, and proprietary laundry management software. Research on information reading in washing devices is also presented. The last section summarizes the research results and presents potential paths for the project's development.

II. Smart Laundry in the Era of IoT and Cloud Computing

Despite the large access to household washing machines, laundries are still popular and used not only for traditional everyday clothes, but especially for textiles that require specialized treatments. Each fabric requires appropriate washing agents to be used without damaging the material. Some clothes require more specific measures, while for others they can be more universal. The idea of an intelligent laundry system is to adapt the chemical agents, their dosage, and grouping based on the fabric inserted into the washing machine.

Considering the wide range of laundry detergents available and the constant emergence of new liquids and powders, storing their information along with the clothing would be an outdated solution. This means that whenever new products appear, the information for each garment would require updating. A much flexible and future-proof solution is to store the data in the cloud, where the database can be updated at any time. This allows for the utilization of products that have recently entered the market in a very short time.

The idea of operation of an intelligent cluster of washing devices is presented in Fig. 1

As seen, each device is equipped with an Ethernet module or a WiFi module for communication with the network. When a washing machine connects to the Internet, it immediately unlocks the connection to the cloud by retrieving necessary data, such as pending software updates, new washing programs. The device may also send data to the manufacturer regarding the current status of the machine or informing the operator about the current washing process. An offline mode must be provided in case of a network connection problem. However, when appropriate fabrics are detected, the offline data is overwritten with information obtained from the cloud.

The RFID (Radio Frequency IDentification) system was used to recognize the textiles. They contain RFID tags with information about the product [22].Typical data is shown in Fig. 2.

The RFID identifier has security keys and a write-lock feature, preventing the introduction of incorrect materials/components into the washing device, thus avoiding damage. An important element is the `Cloth_code` - a special string of characters assigned, for example, by manufacturers during production, confirming that it is a suitable material for washing. This key is verified against a database to confirm the material after reading its content. The UUID serves as a unique product identification number. It can be checked in the database for existence and based on it, the ideal washing conditions can be selected, taking into account the manufacturer's recommendations available in the cloud database. In

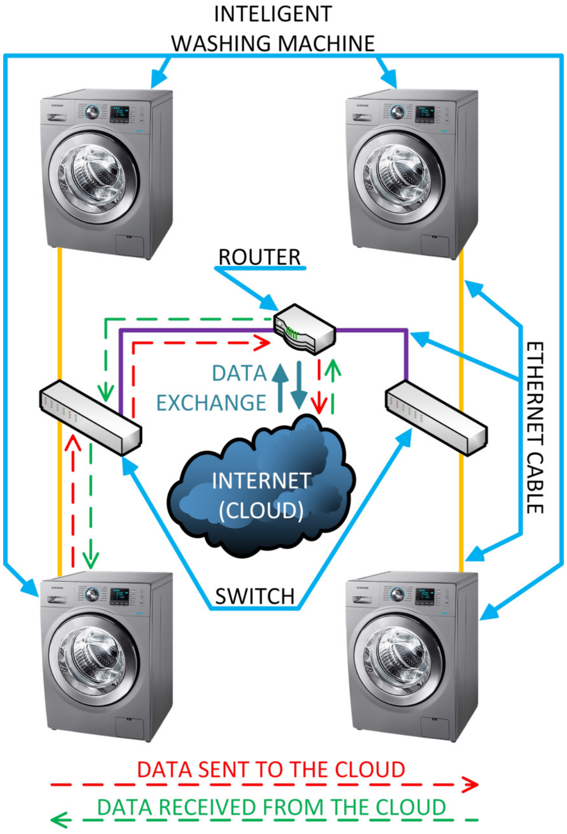


Fig. 1. A cluster of intelligent washing devices

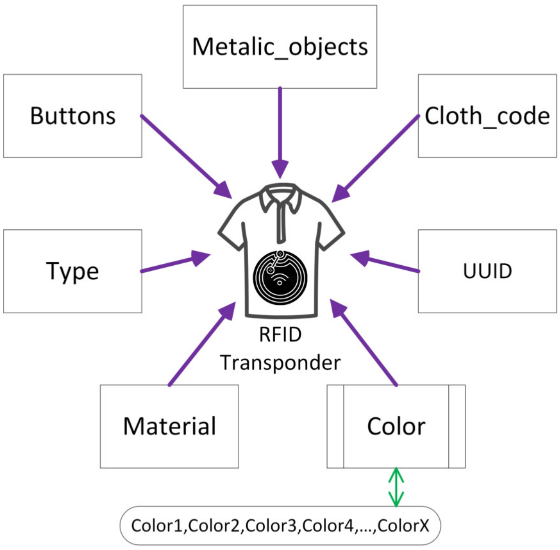


Fig. 2. Data structure in the memory of the RFID identifier

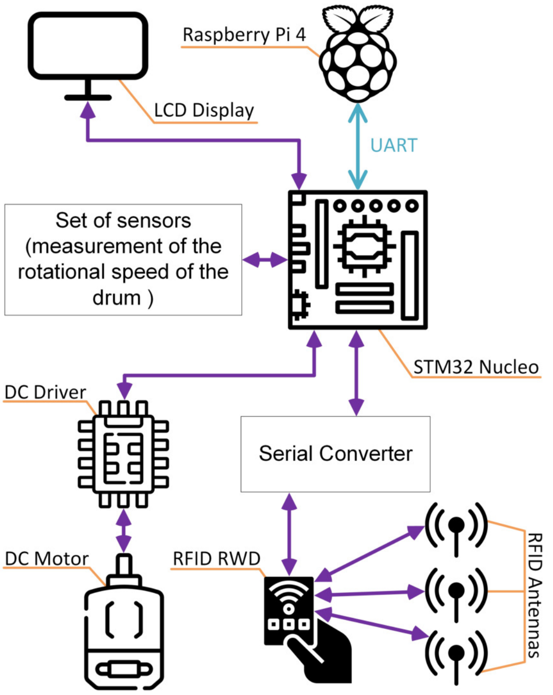


Fig. 3. Diagram of connections and communication between electronic components

the absence of records associated with a particular material, other parameters such as type, fabric, and color are taken into consideration.

To conduct the tests, a device was developed that contains 3 FEIG RFID read/write readers (RWDs). The entire device is controlled using an STM32 microcontroller embedded in the Nucleo board, which features a 32-bit M4 processor with 512KB flash memory and 128KB RAM. The Nucleo board is connected to a touch LCD display, FEIG reading/writing devices, and DC motors for drum rotation control. Since the Nucleo board does not have wireless or wired network modules for internet communication, the Raspberry Pi 4 microcomputer was used. It runs Raspberry Pi OS Lite, which does not have a graphical interface. The Raspberry Pi is connected to the STM microcontroller via the UART interface. The functional diagram is presented in Fig. 3.

After placing clothes equipped with RFID identifiers, the drum rotates to read the data from memory using RWD. In the next step, the information is retrieved and aggregated by a dedicated Python script running on the Raspberry Pi. The data is adjusted accordingly and transmitted to the Azure cloud using Raspberry Pi's wireless (WiFi) or wired (Ethernet) modules.

The project utilized the Azure IoT Hub service, which provides the necessary sub-services to fully leverage the potential of the intelligent cluster of washing machines. In the cloud, the uploaded data is compared with the records in the database, and after successful verification, the device receives instructions on the specific washing parameters for the given fabric. Each incoming data originates from a dedicated device identified by a unique identifier assigned by Azure. This enables the aggregation and processing of a large volume of data from multiple devices simultaneously. When sending data to the cloud, the device must authenticate itself in the service before transmitting any information to the database. To authenticate a device in the cloud, it is necessary to include it in the Azure IoT Hub settings, which will generate the required connection string along with login credentials. Each device is provided with access keys and individual character string, allowing for its identification in the cloud. All steps are presented in the diagram shown in Fig. 4.

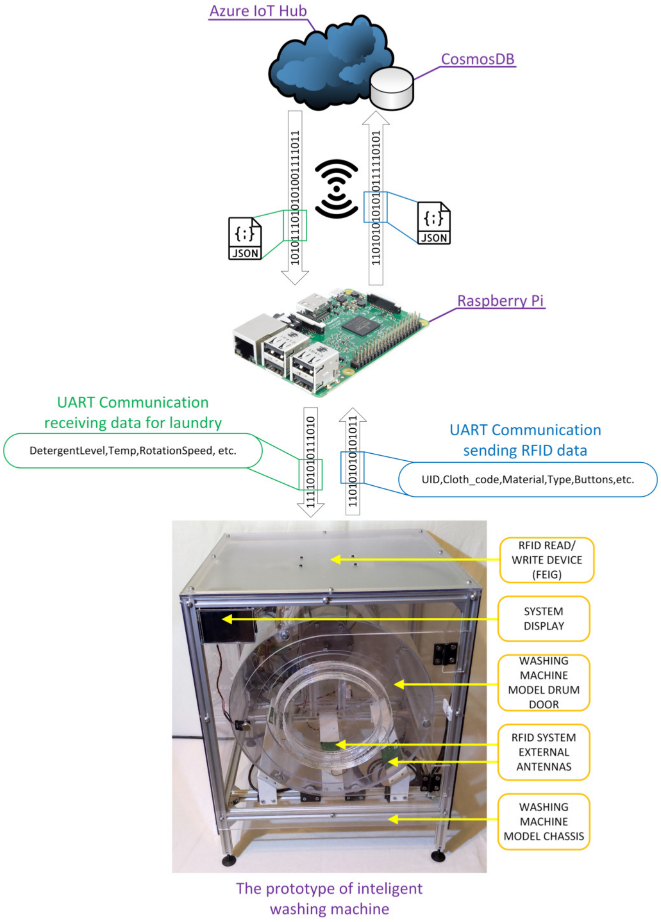

Azure IoT Hub
CosmosDB
1010111101010101001111011
1101010101010111110101
JSON
JSON
Raspberry Pi
UART Communication receiving data for laundry
DetergentLevel,Temp,RotationSpeed, etc.
1111010101011111010
1101010101010101011
UART Communication sending RFID data
UID,Cloth_code,Material,Type,Buttons,etc.
RFID READ/ WRITE DEVICE (FEIG)
SYSTEM DISPLAY
WASHING MACHINE MODEL DRUM DOOR
RFID SYSTEM EXTERNAL ANTENNAS
WASHING MACHINE MODEL CHASSIS
The prototype of inteligent washing machine

Fig. 4. Data transmission between a washing machine and a computing cloud

The data obtained from the cloud computing (Azure IoT service) is transmitted to Raspberry Pi, which modifies the incoming information accordingly. It then sends the information via the UART interface to the STM32 microcontroller, along with laundry-related details such as detergent dosing, drum rotation settings, and the amount of powder to be dispensed. Additionally, the LCD screen displays information related to the recommended detergent and laundry powder provided by the manufacturer.

III. Reading textile data

As part of the research work, accuracy analyses were conducted on the detected objects in the smart washing device. For this purpose, 30 RFID identifiers simulating clothes were used to determine the number of scans required to correctly read all the identifiers inside the washing machine. The results are presented in Figure 5, divided into the number of scans as 1-3 and 4-6, respectively.

As can be observed in Fig. 5 on the left chart, during a single full drum rotation with the identifier scanning, the accuracy of reading individual identifiers decreases as the number of clothes in the drum increases. In the case of a small quantity of 1-2 clothes, a 100% reading accuracy can be determined. In the range of 3 to 15 identifiers (clothes), the success rate of scanning ranges from 93% to 97%, which seems a pretty good result for a single rotation. Unfortunately, with a larger quantity of clothes, the level of accuracy decreases. This is due errors in reading data from the identifiers or a lack of power supplied to the identifier memory during a single rotation. It is worth noting that the device is equipped with 3 antennas for reading identifiers, which are activated during one rotation, resulting in good reading accuracy even with a large number of clothes.

In the case of two full drum rotations, a significant improvement in reading the fabric information can be observed. The reading accuracy is 100% for a range of 1 to 16 clothes inside the washing machine. The accuracy starts to decrease from 17 clothes, but within the range of 17 to 21 identifiers, the accuracy level still remains above 90%, which is a satisfactory result. The lowest value, 60%, appears only at 30 clothes, whereas in the case of a single rotation, the lowest value appeared at 29 clothes and was 44%.

Another measurement was conducted by performing 3 scans, which means 3 full drum rotations. As can be seen, the measurement accuracy increased for a larger number of clothes. The scanning achieved a 100% value even for 21 RFID identifiers. The accuracy level remained above 90% for 22 and 24 identifiers. In the case of measuring 23 identifiers, the value reached 88%. This may be related to a lack of power supplied to the memory of the RFID identifiers, which reduced the accuracy measurement results. It is worth noting that even for 30 clothes inside the washer, the reading accuracy level was above 70%.

The right chart of figure 5 presents measurements conducted for 4 to 6 drum rotations to verify the influence of rotations on reading accuracy. Although the previous results showed a significant increase in accuracy level, the readings were still below 80% in the most challenging case. As can be observed, the accuracy level for 4 drum rotations, up to 21 clothes, does not deviate from the results presented on the left chart. In the case of values 20-21, there is a slight decrease to approximately 99%. However, it is worth noting the change in reading results for the range of 27-30 clothes. The reading accuracy significantly improved for 4 rotations, resulting in over 80% accuracy for 30 RFID identifiers. For further analysis, two additional scans were conducted, namely 5 and 6 drum rotations. For the 5 scans (5 rotations), another accuracy leap can be observed for the final values. In the case of 30 clothes inside the intelligent washing device, the accuracy level reached over 90%, as well as for all the remaining cases. However, it can be noticed that an increase in the number of drum rotations did not significantly affect the values in the range of 23-26 RFID identifiers. To validate the results, a 6-fold drum rotation was performed, and the results are presented in Fig. 5. It can be observed that the accuracy for 30 clothes did not change significantly, with only a 1% increase. However, the reading levels for the range of 23-28 RFID identifiers slightly improved. The last two scans show that the accuracy level stabilized, and subsequent rotations do not significantly affect the data readings from the memory of the identifiers inside the drum of the device. Additional scans may cause changes at a maximum of 1-2%, which does not bring significant changes considering the lowest value of 92%.

For each identified textile, parameters stored in the Microsoft Azure cloud database are analyzed. Dedicated washing parameters for individual materials are retrieved. Then the data is processed to select the best possible washing agents and recommendations. In the case of a different fabrics inside the washing machine, averaged washing conditions are selected to avoid damaging any of the materials.

IV. Conclusion

The presented paper showcases a prototype of an intelligent laundry management system connected to the cloud. Parameters and washing configurations are retrieved from the database in the Microsoft Azure service. As part of the research, an analysis of textile recognition accuracy was conducted. This is crucial for washing automation, especially in public laundries, where each fabric should be correctly identified. The analyses demonstrated high accuracy in reading data from the identifiers after performing 5 and 6 scans, which correspond to full drum rotations, at a level exceeding 90%.

In further work, the authors aim to examine the selection of washing parameters for incompatible materials and introduce the ability to adjust material information in the washing machine before starting the washing process. Additionally, they plan to develop a mobile application that allows managing the washing machine, including reading the current materials inside the device. The development of the application will enable further advancement of the project towards an intelligent virtual wardrobe with enhanced material control and storage capabilities.

Acknowledgment

This project is financed by the Minister of Education and Science of the Republic of Poland within the “Regional Initiative of Excellence” program for years 2019–2023. Project number 027/RID/2018/19, amount granted 11 999 900 PLN.

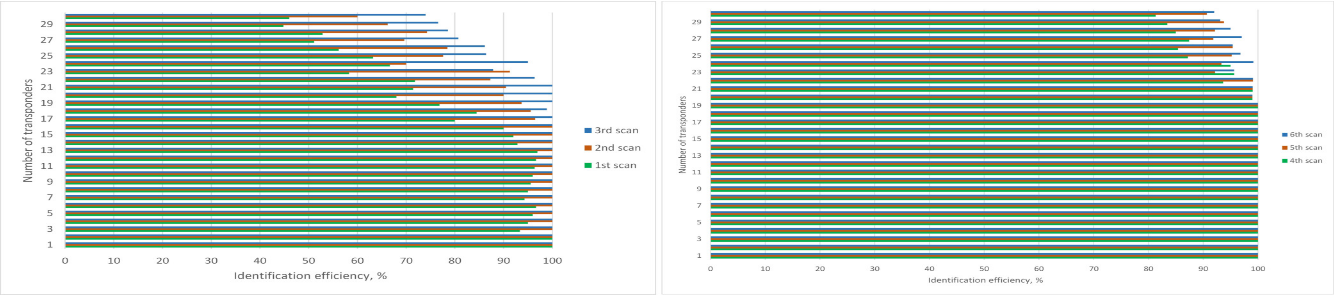


Fig. 5. Reading efficiency of identifiers depending on their number for 1 - 6 scans

REFERENCES

[1] K. Rathi, V. Sharma, S. Gupta, A. Bagwari, and G. S. Tomar, “Home Appliances using loT and Machine Learning: The Smart Home,” in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*. Al-Khobar, Saudi Arabia: IEEE, Dec. 2022, pp. 329–332. [Online]. Available: https://ieeexplore.ieee.org/document/10008294/

[2] O. Ameri Sianaki, A. Yousefi, A. Rajabian Tabesh, and M. Mahdavi, “Internet of everything and machine learning applications: Issues and challenges,” in *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2018, pp. 704–708.

[3] A. J. Chinchawade and O. S. Lamba, “Authentication schemes and security issues in internet of everything (ioe) systems,” in *2020 12th International Conference on Computational Intelligence and Communication Networks (CICN)*, 2020, pp. 342–345.

[4] J. Ryoo, S. Kim, J. Cho, H. Kim, S. Tjoa, and C. Derobertis, “Ioe security threats and you,” in *2017 International Conference on Software Security and Assurance (ICSSA)*, 2017, pp. 13–19.

[5] H. Khalid Alkahtani, K. Mahmood, M. Khalid, M. Othman, M. Al Duhayyim, A. E. Osman, A. A. Alneil, and A. S. Zamani, “Optimal Graph Convolutional Neural Network-Based Ransomware Detection for Cybersecurity in IoT Environment,” *Applied Sciences*, vol. 13, no. 8, p. 5167, Apr. 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/8/5167

[6] H. A. Abdulghani, A. Collen, and N. A. Nijdam, “Guidance Framework for Developing IoT-Enabled Systems’ Cybersecurity,” *Sensors*, vol. 23, no. 8, p. 4174, Apr. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/8/4174

[7] B. V. Albons, K. H. Yusof, N. H. Mahamarowi, A. S. Ahmad, and A. S. M. Azlan, “Designation of a Home Automation System using Nodemcu with Home Wireless Control Appliances in Traditional Malay House,” in *2022 Engineering and Technology for Sustainable Architectural and Interior Design Environments (ETSAIDE)*. Manama, Bahrain: IEEE, Jun. 2022, pp. 1–3. [Online]. Available: https://ieeexplore.ieee.org/document/9906385/

[8] M. Ibne Joha, M. Shafiul Islam, and S. Ahamed, “IoT-Based Smart Control and Protection System for Home Appliances,” in *2022 25th International Conference on Computer and Information Technology (ICCIT)*. Cox’s Bazar, Bangladesh: IEEE, Dec. 2022, pp. 294–299. [Online]. Available: https://ieeexplore.ieee.org/document/10054941/

[9] M. A. Khan, I. A. Sajjad, M. Tahir, and A. Haseeb, “IOT Application for Energy Management in Smart Homes,” in *IEEC 2022*. MDPI, Aug. 2022, p. 43. [Online]. Available: https://www.mdpi.com/2673-4591/20/1/43

[10] Nur-A-Alam, M. Ahsan, M. A. Based, J. Haider, and E. M. G. Rodrigues, “Smart Monitoring and Controlling of Appliances Using LoRa Based IoT System,” *Designs*, vol. 5, no. 1, p. 17, Mar. 2021. [Online]. Available: https://www.mdpi.com/2411-9660/5/1/17

[11] S. Venkatraman, A. Overmars, and M. Thong, “Smart Home Automation—Use Cases of a Secure and Integrated Voice-Control System,” *Systems*, vol. 9, no. 4, p. 77, Oct. 2021. [Online]. Available: https://www.mdpi.com/2079-8954/9/4/77

[12] S. Subramanian, M. Bindhu, S. Umathe, S. Rao, S. Deivasigamani, and M. Ramarao, “Wireless Sensor & RFID Based Smart Energy Management for Automated Home,” in *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. Trichy, India: IEEE, Nov. 2022, pp. 1125–1129. [Online]. Available: https://ieeexplore.ieee.org/document/10010710/

[13] S. P. Ramalingam and P. K. Shanmugam, “Hardware Implementation of a Home Energy Management System Using Remodeled Sperm Swarm Optimization (RMSSO) Algorithm,” *Energies*, vol. 15, no. 14, p. 5008, Jul. 2022. [Online]. Available: https://www.mdpi.com/1996-1073/15/14/5008

[14] M. Bolanowski, A. Gerka, A. Paszkiewicz, M. Ganzha, and M. Paprzycki, “Application of Genetic Algorithm to Load Balancing in Networks with a Homogeneous Traffic Flow,” in *Computational Science – ICCS 2023*, J. Mikyška, C. De Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. Sloot, Eds. Cham: Springer Nature Switzerland, 2023, vol. 14074, pp. 314–321, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-031-36021-3_32

[15] M. Khan, J. Seo, and D. Kim, “Towards Energy Efficient Home Automation: A Deep Learning Approach,” *Sensors*, vol. 20, no. 24, p. 7187, Dec. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/24/7187

[16] O. Sinkevych, L. Monastyrskyi, B. Sokolovskyi, Y. Boyko, and Z. Matchyshyn, “Estimation of Smart Home Thermophysical Parameters Using Dynamic Series of Temperature and Energy Data,” in *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*. Lviv, Ukraine: IEEE, Jul. 2019, pp. 934–937. [Online]. Available: https://ieeexplore.ieee.org/document/8879944/

[17] V. I. Akimov, E. N. Desyatirikova, A. V. Polukazakov, S. I. Polyakov, and V. E. Mager, “Development and Research of a "Smart Home" Heating Control System,” in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. St. Petersburg and Moscow, Russia: IEEE, Jan. 2020, pp. 574–580. [Online]. Available: https://ieeexplore.ieee.org/document/9039541/

[18] M. Aibin, “The Weather Impact on Efficient Home Heating with Smart Thermostats,” in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*. Edmonton, AB, Canada: IEEE, May 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8861778/

[19] N. Stroia, D. Moga, D. Petreus, A. Lodin, V. Muresan, and M. Danubianu, “Integrated Smart-Home Architecture for Supporting Monitoring and Scheduling Strategies in Residential Clusters,” *Buildings*, vol. 12, no. 7, p. 1034, Jul. 2022. [Online]. Available: https://www.mdpi.com/2075-5309/12/7/1034

[20] M. Bolanowski, A. Paszkiewicz, and A. Kraska, “Integration of the elements of a distributed IT system with a computer network core using island topology,” *Enterprise Information Systems*, vol. 15, no. 10, pp. 1354–1375, Nov. 2021. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/17517575.2020.1790042

[21] Y. He, J. Tian, and Y. Cao, “Intelligent home temperature and light control system based on the cloud platform,” in *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*. Xi’an, China: IEEE, Apr. 2022, pp. 1437–1441. [Online]. Available: https://ieeexplore.ieee.org/document/9778483/

[22] B. Pawłowicz, K. Kamuda, M. Skoczylas, P. Jankowski-Mihułowicz, M. Węglarski, and G. Laskowski, “Identification efficiency in dynamic uhf rfid anticollision systems with textile electronic tags,” *Energies*, vol. 16, no. 6, p. 2626, Mar 2023. [Online]. Available: http://dx.doi.org/10.3390/en16062626

DOI: 10.15439/2023F6275
ISSN 2300-5963 ACSIS, Vol. 35

Text embeddings and clustering for characterizing online communities on Reddit

Jan Sawicki
0000-0002-8930-7564
Warsaw University of Technology
Email: jan.sawicki2.dokt@pw.edu.pl

***Abstract*—This work analyses Reddit, the largest public, topic-centered social forum. In the experiments, contextualized text embeddings, obtained using DistilBERT, represented subreddit content. Next, clustering was performed, using an unsupervised K-means algorithm and evaluated with multiple clustering metrics. The obtained clusters were analyzed. Moreover, changes of cluster structure, between 2019 and 2022 have been examined.**

I. Introduction

REDDIT is the largest, public, topically-separated forum [1]. Its unique structure allows users with common interest to find their place for discussion, i.e. a subreddit; a subforum dedicated to a particular topic. The platform policy, and Reddit administration, put no restrictions on the topic of subreddits (except for the rules regarding illegal content). Moreover, any user with at least 30-day old account and a non-negative "karma score" (reputation metric) can create a subreddit. This allows "communities" to blossom, with little-to-no supervision. There are subreddits, which are very close thematically, e.g. r/worldnews or r/news (news and information), or r/leagueoflegends and r/Overwatch (video games). There are also subreddits with distant, or even opposite, topics, e.g. r/Conservative and r/Libertarian.

The freedom and scale of subreddits raise multiple research questions, e.g. what are the most popular topics? Are there topically similar subreddits? Can subreddits be reasonably grouped into clusters? Are there, and if so what are, migrations of subreddits between clusters? This contribution explores these questions, for a Reddit dataset spanning 2019-2022, using natural language processing and data clustering.

II. Related works

Reddit's communities have been analyzed with different methods and from different perspectives. The main inspiration for this work are the results of a 2015 study [2] clustering 15,000 subreddits, from the first half of 2013 using scale-free backbone graph networks. The subreddits were grouped into 57 clusters and further, manually, annotated into 10 metaclusters (categories) such as: Electronic Music, Fitness, Sports, Soccer, Video Games, my Little Pony, LGBT, Pornography, Programming, Guns. Captured relations were based on interactions of over 800,000 users. However, the actual content of the posts, or comments, were not analyzed.

Two years later, "community2vec" [3], was introduced. This study also focused on users, by encoding post authors and user co-occurrences and applying PCA . Additionally, post content was encoded with static GloVe embeddings [4]. The main result was showing that vector representations of communities can encode meaningful analogies and semantic relationships, similarly to what has been previously seen for words.

A 2020 study of Reddit and Twitter [5] focused exclusively on texts of 54.5 million Reddit comments and 23,684 tweets. Its goal was to compare text embedding methods: TF-IDF Word2Vec [6] and Doc2Vec [7] applied to topic modelling, with document clustering using k-means, k-medoids, hierarchical agglomerative clustering and non-negative matrix factorization (NMF). For these, different settings and hyperparameters have been tested. It was established that combining Doc2Vec and K-means achieved the best results.

Finally, in [8], instead of static clustering, community evolution over time was analyzed. Here, active users and textual content, processed with LIWC analysis , has been applied. The results represent patterns of user engagement at different stages of community lifespan. However, they do not show how the subreddit topic clusters evolve over time.

To summarize, over time, a shift from user-based to content-based embeddings can be observed. Additionally, since 2018, an influence of NLP advancements [9] is noticeable; from basic text processing (e.g. LIWC, TFIDF, PCA) and static embeddings (e.g. GloVe, Word2Vec and Doc2Vec), to contextualized text embeddings (e.g. BERT [10] and BERT-like models, e.g. DistilBERT [11]). Here, note that older techniques underperform, against BERT-like models (e.g. LIWC [12]).

Moreover, Reddit continues to grow, since its launch in 2005, with past studies completed in 2015, 2017 and 2020. Therefore, a Reddit structure study needs to be revisited, applying modern approaches, to understand what the subreddits communities look today like and how they change in time. Therefore, this contribution presents results of explorations based on a dataset spanning four years (2019-2022), while applying contextual text embeddings with a BERT-like model, with the goals of analysis of subreddit community structure and its evolution over time.

III. Methodology

Let us now briefly discuss (1) dataset, (2) text embedding method, (3) clustering approaches, and (4) cluster quality assessment methods used in this contribution.

A. *Dataset*

Reddit consists of over 3.5 million communities (and over 1.5 billion monthly visitors). The most popular Reddit data source is the Pushshift database [13], [14]. The subreddit data was extracted from Pushshift subreddit dumps. Furthermore, Pushshift REST API could not have been used due to an outage that happened in early 2023. Overall, content of 3090 "largest" subreddits, i.e. subreddits with at least 100,000 subscribers, has been extracted.

Overall, Reddit is a subject to the 1% rule that appears in the majority of social networks [15]. The majority of posts gain little-to-no attention ("upvotes"), while a small fraction "goes viral" and appears on the frontage of Reddit (the main Reddit forum r/all). Hence, to reduce the computational cost, while capturing subreddit structure, 1000 posts with the highest *scores*, have been extracted from each subreddit. The *score* is the Reddit's measure of "appraisal by a community of Reddit subscribers of an item" [16]. Finally, the dataset spans 4 years: 2019-2022, to allow the analysis of subreddits cluster evolution over time. The resulting dataset consisted of over 12 million unique user posts.

B. *Text embedding*

After gathering, text embeddings has been applied to the posts. Since the introduction of BERT (in 2018), multiple models, for different NLP goals, were introduced [17], [18]. This work needs a general feature extraction models that deliver multipurpose text embeddings. The model should be "general" and multipurpose, because input data originates from over 3000 communities, and covers topics from politics and news (r/politics, r/news), through video games (r/DOTA, r/gaming), memes (r/hmmm, r/me_irl), drug usage (r/LSD, r/shrooms), to plants (r/Bonsai, r/gardening), fishkeeping (r/Aquariums, r/PlantedTank) or military (r/military, r/guns).

Moreover, the NLP part takes the longest processing time (over 50% of total runtime). Therefore, a general multipurpose and fast, but efficient model is required. In 2019 a "smaller, faster, cheaper and lighter" version of the BERT model has been introduced, the DistilBERT. It retains 97% of the original BERT performance on downstream tasks, while being 40% smaller and 60% faster [11]. Therefore, to reduce computation time, DistilBERT has been selected.

Here, it should be noted that different Reddit communities have different posts "styles". For example, r/politics consists mostly of links to news websites, while r/AbruptChaos contains mostly GIFs or short videos. There is, however, one part of posts that is forced by Reddit – the post's title, which has to be present on every posts regardless of subreddit. While there are ways to overcome this (Reddit post, over 99% of posts in the dataset have a textual title. Overall, DistilBERT embedded posts titles to 768 dimensional vectors, which were clustered.

C. *Clustering*

Use of K-means for clustering followed results found in [19], [20]. However, the biggest downside of K-means is that it requires specification of the number of clusters. This problem can be overcome by using unsupervised clustering metrics [21], [22] to find "best" clustering. In this context, Silhouette Score (previously used on Reddit [23]), Davis-Bouldin score , Caliński-Harabasz score and k-means inertia (sum of squared distances of samples to their closest cluster center) have been tried. The most suitable cluster size has been sought by evaluating clustering results for cluster sizes: 10, 20, 30, ... 1530, 1540, where 1540 is half o the number of subreddits. Davis-Bouldin metrics is the only metric where lower values are better (for others, higher is better). For easier interpretability of the results, presented in Figure 1, Davis-Bouldin metrics is presented with a minus sign. Interestingly, the metrics were practically identical for considered time periods (annually for 2019-2022). Hence, it can be stated that the number of subreddit clusters, and hence the topical dispersion, does not change much over time (see, also, Section IV).

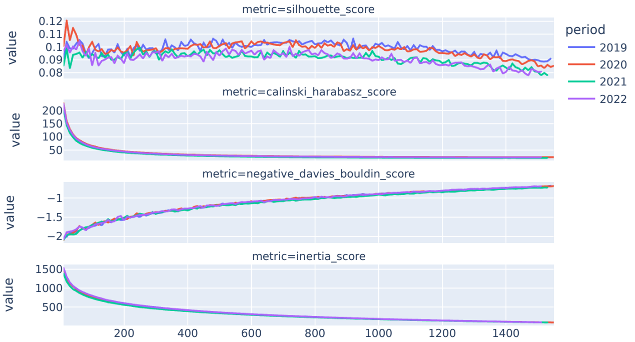


Fig. 1. Cluster evaluation with different metrics (Davis-Bouldin score is actually negative score, to keep up with "higher is better" interpretation.

The choice of the cluster number was a challenge since not all metrics have been consistent (see, Figure 1). The Silhouette Score differed a lot. This is related to the fact that the Silhouette Score ranges from -1 to 1, where the best value is 1 (all points assigned to the right cluster) and the worst value is -1 (all points assigned to the wrong cluster), while scores near 0 indicate cluster overlap. All Silhouette Scores were close to 0, indicating existence of overlaps. To find a compromise between Davis-Bouldin, Caliński-Harabasz and inertia metrics, the Elbow Method was applied, as previously used in similar settings [24], [25] (also on Reddit [26]). Overall, any number between 180 and 450 represented a "good fit". However, to achieve interpretability of results, 200 clusters were selected. Here, note that smaller numbers of clusters were checked (e.g. 100), but they produced clusters with "not-fitting" topics. Larger numbers (e.g. 300), on the other hand, resulted in fragmented topics.

After clustering, the subreddit groups were manually assigned to meta groups (categories). For example, subreddits related to games (r/LeagueOfLegends, r/DOTA, r/Overwatch, r/gtaonline) or subreddits related to politics (r/Conservative, r/politics, r/Libertarian, r/Political_Revolution, r/geopolitics). These groups are further described in Section IV-A.

D. Cluster similarity and time-evolution

To automatically detect cluster dynamics, they were compared annually using the Jaccard Index [27]. Each cluster from a period was compared to each cluster from the next chronological period. The pair of sets with the highest Jaccard Index is considered a transition, from the predecessor to the successor. Note that the predecessor and the successor may be the same, i.e. the cluster did not change from period to period. This way, ordered lists of cluster transitions were created. Then for each list a generalized multi-set Jaccard index was calculated for all sets (2019, 2020, 2021, 2022). The results are discussed in Section IV.

IV. Results and their analysis

Let us now discuss the key experimental findings.

A. Subreddit clusters characterization

Let us first look into clusters of subreddits established for year 2022. Table I presents number of subreddit by manually annotated categories. Most categories are obvious, but some require some explaining.

"**Pictures**" aggregates subreddits dedicated to posting pictures, GIFs and videos. The themes range from wallpapers (r/wallpapers), to content that is supposed to amaze on-lookers (r/nextfuckinglevel, r/woahdude) or disquiet/scare them (r/cursedcomments, r/cursedimages, r/cursedvideos).

Some subreddits do not have a "theme", and are extremely broad, such as r/gif, r/gifs, r/pics. Interesting is a group of "X_Porn" subreddits, where X is some subject. Here, the term "porn" is a synonym to "amazing", "beautiful", "wonderful" (not pornography). Subreddits in this group (r/CabinPorn, r/CityPorn, r/EarthPorn, r/InfrastructurePorn) showcase pictures of things, places or phenomena that are to be perceived as "porn", i.e. most spectacular of its kind. There are also "meta-themes", such as r/BetterEveryLoop, where the author of a post claims that the more times a GIF/video is watched, the better it gets. The actual content is discretionary. Finally, there are also subreddits with random pictures, e.g. r/nocontextpics.

In the "**states**" category, there are subreddits related to individual US states and cities, e.g. r/Atlanta, r/Austin, r/Calgary, r/California, r/Dallas, r/Denver, r/LosAngeles.

Interestingly, while the applied NLP model is meant for English, it clustered subreddits in other languages into the category "**language specific**". There are also separate clusters for: German subreddits (r/de, r/de_IAmA), the Polish subreddit r/Polska, the Netherlands, containing r/thenetherlands and a cluster related to Scandinavian subreddits, containing r/norge, r/svenskpolitik, r/swedishproblems.

The "ask" category contains subreddits with questions. Here, questions can be general (r/AskReddit), topic specific (r/morbidquestions, r/AskRedditAfterDark), or answerer specific (r/AskMen, r/AskMenOver30, r/AskWomen, r/AskEurope, r/AskUK).

There was a group that was separated from "pictures" were "**animals**". This group contains clusters of subreddits about (mostly) dogs and cats and other small animals. It appears that the similarity between some "animals" subreddits and some "pictures" is in the feelings that the pictures are supposed to invoke, i.e. happiness, or cuteness. For example, subreddits: r/aww (described as: "Things that make you go AWW! Like puppies, bunnies, babies, and so on... A place for really cute pictures and videos!" and r/MadeMeSmile ("A place to share things that made you smile or brightened up your day. A generally uplifting subreddit." . On the "other side", one can find "animals" in subreddits dedicated to brutality and violence in animal kingdom (r/natureismetal, r/Natureisbrutal).

Next, there is the "**irl**" group, standing for "in real life". It contains subreddits, such as: r/meirl, r/2meirl4meirl, r/bi_irl, r/discord_irl, r/egg_irl, r/anime_irl, r/gay_irl, r/me_irlgbt, r/woof_irl, r/ich_iel . All of them contain pictures with strict post title policy. Depending on the subreddit, the titles are always "meirl" (r/meirl), "2meirl42meirl4meirl" (r/2meirl42meirl4meirl), etc. It was clear how to characterize the content of these subreddits, other than it being memes. Interestingly, even though r/ich_iel is a German subreddit, it got clustered with other English "irl" subreddits.

Let us now consider "**social chatting**" group. Here, clusters include both general (r/CasualConversation, r/MakeNewFriendsHere) and specialized chatting topics (r/BreakUps, r/LongDistance, r/Marriage). There are also subreddits where users explicitly ask for an advice: r/Advice, r/askwomenadvice, r/dating_advice or seek approval/disapproval of their actions: r/AmItheAsshole (the latter with a dedicated study, from 2023 [28]).

Moving to smaller subreddits, there is the "**NSFW**" (Not Safe For Work) group. Here, confirmation that the user is an adult is required. However, these are different from "pornography", since they discuss adult topics, such as fetishes, fantasies and other sex-related issues. The examples are: r/BDSMAdvice, r/BDSMcommunity, r/Swingers, r/polyamory, r/DeadBedrooms, r/NoFap, r/bigdickproblems. There are also subreddits devoted to looking for other people with similar interests. These often use the acronym "r4r" (Redditors for Redditors): r/DirtySnapchat, r/Kikpals, r/dirtykikpals, r/dirtyr4r, r/exxxchange, r/r4r, r/snapchat, r/swingersr4r.

The "**Reddit meta**" category contains Reddit administration (r/announcement) and technical support subreddits (r/help). Next, "Deals" category contains subreddits about free goods, or goods on sale, e.g. r/GameDeals, r/NintendoSwitchDeals, r/PS4Deals, r/deals, r/eFreebies, r/freebies, r/googleplaydeals. Finally, "**Help me find**" is the group of subreddits where users ask others to help them find something, or find what something is, e.g. r/HelpMeFind, r/RBI (Reddit Bureau of Investigation), r/Whatisthis. Here, subreddits for identifying pornographic performers or scenes r/pornID, r/sources4porn, r/tipofmypenis are included.

What is clearly visible in Table I, is that the most of the clusters are related to pornography, pictures with vague themes, video games, memes and technology. These themes also aggregate the biggest number of subreddits.

TABLE I
CLUSTERING EVALUATION

category	subreddits count	cluster count
pornography	692	33
pictures	253	22
games	215	6
memes	201	8
mixed	195	13
tech	135	9
social chatting	119	6
tv series	99	3
animals	95	5
politics	92	5
music	64	6
sports	64	3
finance	58	4
NSFW	52	4
hate	48	1
cooking	39	3
drugs	38	2
popculture	36	1
science	31	3
states	29	1
education	27	3
fashion	27	2
game consoles	27	1
language specific	26	12
military	24	3
ask	23	2
fitness	21	2
science fiction	21	1
art	20	2
camping	20	1
irl	19	7
movies	19	2
cars	17	2
plants	17	1
Reddit meta	16	3
craftsmanship	16	1
food	14	1
horror	14	1
mental health	12	1
deals	11	3
writing	11	2
crime	11	1
religion	11	1
anime	10	1
trading	9	3
help me find	7	1
hiring	5	1
surveys	1	1

B. Subreddit clusters findings

Let us now report key findings regarding the clustering.

1) Subreddit naming: There are naming patterns and conventions of subreddits. Reddit's users employ multiple acronyms, e.g. "IRL" (In Real Life), "AMA" ("Ask Me Anything") or "NSFW" ("Not Safe For Work"). These 3 alone materialize in 42 subreddits. There are also subreddits acronym names, e.g. r/ATBGE ("Awful Taste But Great Execution"), including the longest name: r/UNBGBBIIVCHIDCTIICBG ("Upvoted Not Because Girl, But Because It Is Very Cool; However, I Do Concede That I Initially Clicked Because Girl."). As noted, common is using the word "porn" to name content that is supposed to be beautiful, aesthetically pleasing, interesting, well-made, etc. Moreover, only 10 out of 71 "X_porn" subreddits contain actual pornography.

Finally, while many subreddits are descriptive of the topic (e.g. movie or TV series title, music genre, area of science or name of a video game), multiple subreddits focus on describing a general phenomenon/feeling (r/aww, r/INEEEEDIT, r/iwanttobeher). This shows how important it is to analyze the content of the subreddits instead of just the names.

2) Country subreddits: There is a group of subreddits, in English, dedicated to countries, e.g. r/UnitedKingdom, r/russia, r/China, r/canada. Interestingly, they appear in *different* clusters, but all in the category "politics". Interestingly, r/russia and r/China appear in a single cluster, consisting of: {r/ANormalDayInRussia, r/China, r/MapPorn, r/PropagandaPosters, r/imaginarymaps, r/russia, r/vexillology, r/vexillologycirclejerk}. This suggests that text embeddings of map-related subreddits and a propaganda poster subreddit, are similar to the content of r/russia and r/China. Subreddits r/canada, r/europe, r/unitedkingdom are in the same cluster with political and general news subreddits: {r/CanadaPolitics, r/Conservative, r/DemocraticSocialism, r/canada, r/Economics, r/Libertarian, r/Political_Revolution, r/europe, r/Republican, r/news, r/The_Mueller, r/democrats, r/geopolitics, r/politics, r/ukpolitics, r/unitedkingdom, r/worldnews}.

Note that even though the US is the third-largest country by population, and has the highest number of users on Reddit, it does not have a dedicated subreddit. However, as noted (in Section IV-A), there exist subreddits dedicated to US states and cities, and they form a separate cluster.

3) Technology + Finance = Cryptocurrencies: There is an interesting overlap between subreddits in "finance" and "tech". There is a cluster containing both investment subreddits (r/algotrading, r/pennystock, r/RobinHoodPennyStocks), cryptocurrencies subreddits (r/BitcoinMarkets, r/btc, r/ethereum, r/ethtrader) and technology subreddits (r/tech, r/technews, r/technology). This captures that fact that cryptocurrencies became a "middle-ground" conversation joining finances and technology.

4) Real life and gaming: In the cluster dedicated to crafts, there is a subreddit with "digital craftsmanship". It is r/Minecraftbuilds, containing building concepts created in the game Minecraft. It appears in the cluster with subreddits such as: {r/HomeImprovement, r/Justrolledintotheshop, r/Tools, r/electricians, r/longboarding, r/redneckengineering, r/whatisthisthing, r/woodworking}. Similarly, among "fashion" subreddits there is one related to the Animal Crossing video game fashion designs (r/ACQR): {r/AsianBeauty, r/Embroidery, r/Makeup, r/MakeupAddiction, r/RedditLaqueristas, r/crafts, r/crochet, r/femalefashionadvice, r/knitting, r/malefashion, r/malefashionadvice}. This illustrates interfusion of real life craftsmanship and fashion with in-game craftsmanship and fashion.

5) Are you eating? Watch a documentary.: There exists a relatively small cluster: {r/Documentaries, r/mealtimevideos}, where the first subreddit is about a documentary movie and the second contains video suggestions for watching during lunch or dinner. It appears that both of them have similar content, meaning that documentary videos would be a good suggestion for watching during mealtime.

6) Pornography mix: As visible previously in Table I, most of the clusters are dedicated to pornography. There are subreddits dedicated to fetishes, body parts, looks activities performers, sexual preference (e.g. heterosexual, homosexual etc.), amateur vs professional or performers. However, all of their content seems alike, as there are no particular patterns in pornography clusters, except for one. The subreddits dedicated to particular performers have similar content (often about praising a particular performer), e.g. {r/AdrianaChechik, r/AlexisTexas, r/AngelaWhite, r/DaniDaniels, r/KimmyGranger, r/Miakhalifa, r/RileyReid, r/abelladanger, r/leahgotti}.

The lack of any other patterns when it comes to clustering pornography subreddits shows that their content is extremely overlapping and similar regardless of the subreddit.

C. r/worldpolitics in NSFW subreddits

There is an interesting anomaly in one of the clusters. Subreddit r/worldpolitics appears in a cluster nearly exclusive to NSFW content, e.g. {r/BDSMAdvice, r/Rapekink, r/SexWorkers, r/Swingers, r/mbti, r/bigdickproblems, r/bisexual, r/lgbt, r/polyamory, r/sexover30, r/BDSMcommunity}. At first, this looks like a clustering error, since r/worldpolitics should be in a political cluster with subreddits such as r/politics. Due to permissive rules of this subreddit it is full of all kinds of posts. Its description states "reddit's anything goes subreddit, no topic imposed or opposed by the mods" . Additionally, when posts are sorted by Reddit's "top of all time", the first 100 are marked NSFW, even though they do not include adult content. Overall, r/worldpolitics seen from the perspective of text-embedding, is very close to other adult-content subreddits, i.e. it is either very chaotic, or contains adult content.

1) Current clustering vs. previous studies: As mentioned, the study from 2015 [2] performed similar clustering and manual annotation into "meta clusters". Let us compare meta-clusters from 2015 with these from 2022.

First, the 2015 groups: "Fitness", "Sports", "Video Games", "Pornography" all map one-to-one to cluster groups established in 2022. Second, "Electronic Music", "Programming", "Soccer" and "Guns" map to wider/similar cluster categories, which are "music", "tech", "sports", and "military", respectively. Third, there are 2 groups of clusters with no one-to-one correspondence: "my Little Pony" and "LGBT". Subreddits marked as "LGBT" in 2015 appear in clusters of "NSFW" (e.g. {r/BDSMAdvice, r/Rapekink, r/SexWorkers, r/Swingers, r/mbti, r/bigdickproblems, r/bisexual, r/lgbt, r/polyamory, r/sexover30, r/BDSMcommunity}). This can be a question of the naming convention. Moreover, it seems that LGBT issues are close to NSFW, which is logical, since many of them involve sexuality and sex. "My Little Pony" subreddits were completely absent in this analysis. Even though they are large enough (over 100,000 subscribers), they did not appear in the Pushshift dumps, probably due to inconsistencies in the Pushshift database. Hence, this cluster has no mapping to 2022 clusters.

Similarly to the original work, a dimensionality reduction of the embeddings has been performed with the t-SNE method, to create a two-dimensional visualization. Figure 2 shows the clusters in the top 20 categories, by subreddit count in cluster. T-SNE reduced the dimensions of vectors 768 to 2, Even in two dimensions the clusters such as "pornography", "sports", "music" or "tech" appear close within the group and far between the groups. This is consistent with the 2015 study. Hence, it supports quality of embeddings and category annotations reported in current contribution.

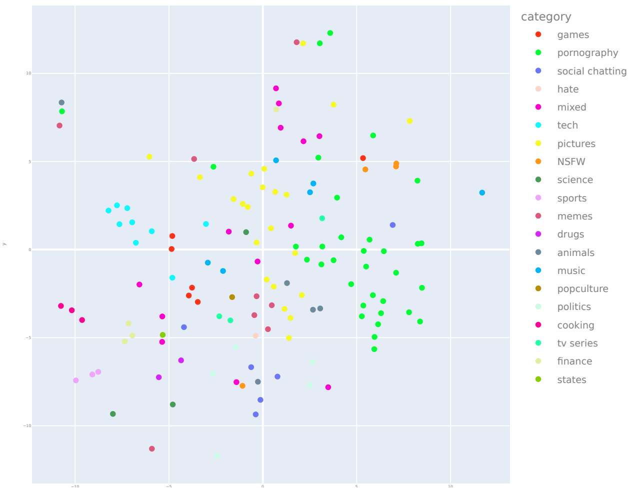


Fig. 2. The clusters in the top 20 categories by subreddit count in cluster. The X and Y axis are insignificant due to dimensionality reduction with t-SNE

D. Subreddit cluster transitions

The second group of results focuses on cluster evolution between 2019 and 2022. Due to space limitations, only selected key findings are presented.

1) Gardening, hair, writing and vehicles stay unchanged: The highest Jaccard index (0.78) between subreddit clusters is achieved for clusters about plants, gardening and fish tanks.

Here, almost no change has been observed for 4 years. The cluster lost one subreddit: r/shrooms, and gained 3 new ones: r/snakes, r/thingsforants, r/whatsthisbug. Interestingly, in 2021, the r/shrooms subreddits migrated to a drug-related cluster. Similarly, between 2019 and 2022, the hair-related cluster (Jaccard index of 0.7) went through a couple changes, but finally lost one subreddit (r/FancyFollicles) and gained one (r/beauty).

With Jaccard index of 0.61 there is also the writing cluster, which moved closer to it's writing theme by dropping r/MovieSuggestions, r/TrueFilm and gaining r/stephenking.

Another barely changed cluster concerns vehicles. In 2019, it was mostly related to cars, but in 2020 it gained and retained r/MTB (mountain bike), r/bicycling and r/cycling subreddits. Interestingly, the r/Cartalk and r/MechanicAdvice subreddits, in 2022 formed a completely new cluster. The main difference between these two and other clustered subreddits is that they focus on discussions rather than showcasing vehicle models.

2) COVID-19: As expected, COVID-19 pandemic is noticeable in subreddit evolution. A cluster that, in 2019, contained general science, health and chemistry subreddits ({EverythingScience, r/Health, r/physicsgifs, r/science}) was extended, in 2020, with r/COVID19, r/China_Flu, r/Coronavirus. It remained unchanged until 2022.

3) Pornographic subreddits migrations: Over time, significant migrations between pornographic clusters have been observed. "Pornography" category has the second-lowest mean, 4-year, Jaccard index of about 0.02. There even was one cluster in 2019 of 30 subreddits which finally got reduced to a single-subreddit cluster (containing only r/sarah_xxx). However, similarly as described in Section IV-B6 these cluster migrations are chaotic and random, and no pattern was detected.

V. Concluding remarks

This work is devoted to study of structure of and time evolution of the Reddit platform. Current text-embedding methods have been applied to the dataset covering 2019-2022 period. Overall, Reddit is a place containing content and discussion on various topics, with both very wide and very narrow scopes. Majority of the most popular subreddits are dedicated to pornography, pictures and videos about "anything and everything". Furthermore, popular are video games, memes and technology subreddits. While some of the topical clusters stay unchanged over the years, there are subreddit migrations between most of the clusters. Future studies will focus on more particular groups of subreddits and researching new methods for inter-subreddit topical modelling, such as crossposts.

References

[1] N. Proferes, N. Jones, S. Gilbert, C. Fiesler, and M. Zimmer, "Studying reddit: A systematic overview of disciplines, approaches, methods, and ethics," *Social Media+ Society*, vol. 7, no. 2.

[2] R. S. Olson and Z. P. Neal, "Navigating the massive world of reddit: Using backbone networks to map user interests in social media," *PeerJ Computer Science*, vol. 1, p. e4, 2015.

[3] T. Martin, "community2vec: Vector representations of online communities encode semantic relationships," in *Proceedings of the Second Workshop on NLP and Computational Social Science*.

[4] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing*.

[5] S. A. Curiskis, B. Drake, T. R. Osborn, and P. J. Kennedy, "An evaluation of document clustering and topic modelling in two online social networks: Twitter and reddit," *Information Processing & Management*, vol. 57, no. 2, p. 102034, 2020.

[6] K. W. Church, "Word2vec," *Natural Language Engineering*, vol. 23, no. 1, pp. 155–162, 2017.

[7] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*. PMLR, 2014, pp. 1188–1196.

[8] H. Mensah, L. Xiao, and S. Soundarajan, "Characterizing the evolution of communities on reddit," in *International Conference on Social Media and Society*, 2020, pp. 58–64.

[9] Q. Liu, M. J. Kusner, and P. Blunsom, "A survey on contextual embeddings," *arXiv preprint arXiv:2003.07278*, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[12] J. Biggiogera, G. Boateng, P. Hilpert, M. Vowels, G. Bodenmann, M. Neysari, F. Nussbeck, and T. Kowatsch, "Bert meets liwc: Exploring state-of-the-art language models for predicting communication behavior in couples' conflict interactions," in *Companion Publication of the 2021 International Conference on Multimodal Interaction*, 2021, pp. 385–389.

[13] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, "The pushshift reddit dataset," in *Proceedings of the international AAAI conference on web and social media*, vol. 14, 2020, pp. 830–839.

[14] J. Sawicki, M. Ganzha, M. Paprzycki, and A. Bădică, "Exploring usability of reddit in data science and knowledge processing," *Scalable Comput. Pract. Exp.*, vol. 23, pp. 9–22, 2021.

[15] E. Hargittai and G. Walejko, "The participation divide: Content creation and sharing in the digital age," *Information, Community and Society*, vol. 11, no. 2, pp. 239–256, 2008.

[16] P. Van Mieghem, "Human psychology of common appraisal: The reddit score," *IEEE Transactions on Multimedia*, vol. 13.

[17] P. Xia, S. Wu, and B. Van Durme, "Which* bert? a survey organizing contextualized encoders," *arXiv preprint arXiv:2010.00854*, 2020.

[18] M. Koroteev, "Bert: a review of applications in natural language processing and understanding," *arXiv preprint arXiv:2103.11943*, 2021.

[19] G. Ahalya and H. M. Pandey, "Data clustering approaches survey and analysis," in *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management*.

[20] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*.

[21] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE access*.

[22] T. Sai Krishna, A. Yesu Babu, and R. Kiran Kumar, "Determination of optimal clusters for a non-hierarchical clustering paradigm k-means algorithm," in *Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2017*.

[23] J. Sirait, "Investigating news source characterizations using reddit audiencebased metrics," 2022.

[24] M. Syakur, B. Khotimah, E. Rochman, and B. D. Satoto, "Integration k-means clustering method and elbow method for identification of the best customer profile cluster," in *IOP conference series: materials science and engineering*, vol. 336. IOP Publishing, 2018, p. 012017.

[25] M. Cui *et al.*, "Introduction to the k-means clustering algorithm based on the elbow method," *Accounting, Auditing and Finance*, vol. 1.

[26] V. Veselovsky, I. Waller, and A. Anderson, "Imagine all the people: Characterizing social music sharing on reddit," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15.

[27] L. d. F. Costa, "Further generalizations of the jaccard index," *arXiv preprint arXiv:2110.09619*, 2021.

[28] S. Giorgi, K. Zhao, A. H. Feng, and L. J. Martin, "Author as character and narrator: Deconstructing personal narratives from the r/amitheasshole reddit community."

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1137–1141

DOI: 10.15439/2023F7199
ISSN 2300-5963 ACSIS, Vol. 35

Runge-Kutta Method and WSM6 Microphysics for Weather Prediction on Hybrid Parallel Platform

Hércules Cardoso da Silva
Faculty of Computer Science
Federal University of MS
Campo Grande, Brazil
hercules.cardoso.silva1@gmail.com

Marco A. Stefanes
Faculty of Computer Science
Federal University of MS
Campo Grande, Brazil
marco@facom.ufms.br

Vinícius Capistrano
Physics Institute
Federal University of MS
Campo Grande, Brazil
vinicius.capistrano@ufms.br

***Abstract*—In Numerical Weather Prediction (NWP) models we need to model the dynamics of the atmosphere and the physical variables that take place in a moment. In grid point models the temporal evolution of the model variables are calculated in a 3D grid which covers the atmosphere from the surface up to the model top. NWP models include two import routine namely Runge-Kutta method and microphysics scheme WSM6. This paper describes advances in the performance of the Runge-Kutta method and WSM6 microphysics by exploiting multi-level parallelism using CUDA-based GPU on hybrid parallel platform. We applied pipeline parallelism technique, workload balancing and asynchronous data exchange strategy each demonstrating be useful to improve performance. Our experiments show that the solution is scalable. We also realized an analysis of the accuracy of our implementation with good results.**

***Index Terms*—Runge-Kutta Method, WSM6 microphysics, Hybrid Parallel Algorithms, Multi-GPU Algorithms, High performance Computing**

I. Introduction

WEATHER prediction systems assess atmospheric changes. They have diverse uses, from aiding agriculture, aviation, navigation, to assisting in natural disaster prevention. Weather prediction, a complex system, requires modeling atmospheric dynamics and physical variables such as pressure, temperature, wind, water vapor, clouds, and precipitation. Outputs typically include future temperature, humidity, and rainfall based on initial conditions. Some small-scale processes, like cloud formation, which can't be resolved numerically, are incorporated into models through parameterization schemes.

There are several numerical prediction systems based on sets of physics-based equations which weather can be predicted from atmospheric data as temperature, radiation, air pressure, wind speed, wind direction, humidity, and rainfall, and how they behave in the atmosphere [1], [2].

Among them, Model for Prediction Across Scales (MPAS) [3], Global/Regional Assimilation and PrEdiction System (GRAPES) [4] and Weather Research Forecasting (WRF) model [5] are popular tools used for both research and operational purposes. The Runge-Kutta third order method (RK3), used to time integration, and the Single-Moment 6-Class Micro-physics (WSM6), which calculates several hydrometeors variables, are tasks present in important models of weather prediction. Moreover, they are the most time-consuming tasks in a weather forecasting system.

The main goal of this paper is to present some enhancement for the RK3 and for the WSM6 using multi-level parallelism and pipelining on hybrid parallel platform. Our tests show that we achieved speedup ranging from 10 to 39 for the RK3 and from 5 to 26 for the WSM6 when compared to a 12-thread CPU. The pipelining and asynchronous data exchange strategies improved the runtime by up to 37%.

II. Background and related Works

There are several works that implement RK3 method and the WSM6 using parallel platforms. Korch and Rauber [6] investigated the RK3 method They compare the RK3 implementation using MPI, Pthreads and Java on Sun SMP and on Cray T3E. The speedup ranged from 3 to 10 with 8 processors. The authors of [7] show an enhancing of twelve-fold of a CUDA RK4 implementation on a Tesla compared to an OpenMP implementation. Murray [8] describes an OpenMP RK4 implementation for physics simulation that achieved a speedup of 2 on a 4-core CPU. Wo *et all* [9] achieve a speedup of 2 on a Geforce GT450M. The authors of [10] achieved a gain of 45 times on 2x Xeon Phi versus the same implementation using OpenMP on 2x Intel Xeon.

An implementation of WSM6 on CUDA [11] obtained a speedup of up to 246 using a K40 GPU and up to 295 using two GPUs when compared to a single-threaded CPU. Kim *et al* [12] implemented the WSM6 using OpenACC for Model for Prediction Across Scales (MPAS) [1]. They achieved a speedup of 5.7 running on a V100 GPU compared to a multi-thread version executed on 48 CPUs. More recently, Silva *et al* [13] improved those results and reached speedup of 371 using four V100 GPUs compared to single-thread CPU e 108 fold compared to 24-thread CPU.

III. Weather Prediction: RK3 and WSM6

The dynamic core of an NWP is responsible for discretization in space. Variables such as temperature, pressure, wind, humidity are time integrated by this dynamic core model. RK3 method is responsible for this time integration. However, many processes cannot be spatially discretized. Therefore, these processes are parameterized in terms of other variables available in the a model time-step state. Among these processes is the microphysics scheme.

The equations of the Runge-Kutta are formulated using the pressure vertical coordinate $\eta = (P_p - P_{ht})/\mu$, where $\mu = (P_{hs} - P_{ht})$, P_p is the hydrostatic component of the pressure, P_{hs} and P_{ht} are representing the pressure at surface and top boundaries. The transport equations in the flux form can be written as:

$$\frac{\partial \mu_d \phi}{\partial t} = -\nabla_\eta \cdot \mu_d \mathbf{v_h} - \frac{\partial \mu_d \omega \phi}{\partial \eta} \tag{1}$$

The quantity μ_d is the mass of the dry-air column (between the bottom and the top of the atmosphere for a vertical coordinate of mass type $s = [\pi_d - (\pi)_t]/\mu_d$, with $\mu_d = \partial \pi_d / \partial s$. Moreover, $\mathbf{v_h} = (u, v, \omega)$, which is the zonal, meridional and covariant vertical velocities, respectively. $\phi = (u, v, w, \theta, q_m)$, where w is the vertical movement, θ is the potential temperature, and q_m represents the scalar quantities, such as water vapor, hydrometeors and aerosol mixing ratio.

The WRF use third order Runge-Kutta for temporally discretized [14], which can be described by the equations:

$$\Phi^{*} = \Phi^{n} - \frac{\Delta t}{3} \nabla_\eta \cdot (\mu_d \mathbf{v_h} \phi)^{n} - \frac{\Delta t}{3} \frac{\partial (\mu_d \omega \phi)^{n}}{\partial \eta} \tag{2}$$

$$\Phi^{**} = \Phi^{n} - \frac{\Delta t}{2} \nabla_\eta \cdot (\mu_d \mathbf{v_h} \phi)^{*} - \frac{\Delta t}{2} \frac{\partial (\mu_d \omega \phi)^{*}}{\partial \eta} \tag{3}$$

$$\Phi^{n+1} = \Phi^{n} - \Delta t \nabla_\eta \cdot (\mu_d \mathbf{v_h} \phi)^{**} - \Delta t \frac{\partial (\mu_d \omega \phi)^{**}}{\partial \eta} \tag{4}$$

where $\Phi = \mu_d \phi$.

Now, the WSM6 scheme is based on six classes of cloud particles: water droplets, ice crystals, snow, hail, aggregates of ice crystals, and raindrops. In an NWP system, the WSM6 is a module that is coupled with other parameterization schemes which are described by differential equations that can be seen in more detail by Hong *et al* [15]. The WSM6 also considers the interactions between the different classes of hydrometeors.

IV. Hybrid Parallel platform

A Hybrid Parallel platform consists of a set of computer nodes which each node has a multi-core CPU and a many-core GPU. Each CPU as well GPU in the set can be heterogeneous. Under this model, we use MPI [16] to dispatch two threads by nodes using massage passing. One of them containing the code that will be executed on the multi-core CPU using OpenMP [17] and another thread containing CUDA [18] code that will be executed on GPU.

V. Multi-level Parallelism

Our weather model enables to use various levels of parallelism. The first level employs the MPI for coordinating various computer nodes. This level helps the task distribution across several nodes and their intercommunication, allowing concurrent execution of tasks. The second level is the division of tasks between different CPUs using shared memory. The last level is the use of CUDA-based GPUs to accelerate the processing. Each GPU executes the pipeline, which allows the model to run faster and more efficiently.

In our implementation of the method RK3 on CUDA, we performed a reorganization of the original code due to the different features between the Fortran compiler and CUDA. The first step in the implementation process involved consolidating all the subroutines into a single subroutine. The main goal of this initial stage is to identify sections that can be optimized and subsequently translated into CUDA. This strategy also helps avoid unnecessary copying of data from the CPU to the GPU. Besides, the consolidation of subroutines contributes to reducing the number of kernel calls.

Our CUDA WSM6 employs parallel optimization techniques similar to those used in the RK3. However, there is a distinction between the two implementations: WSM6 exhibits lower dependency on its operations. This feature enables the WSM6 to be executed in parallel over different portions of the data. We leverage this inherent characteristic of microphysics scheme to execute multiple WSM6 kernels simultaneously. This facilitates a better workload balance, improving distribution of tasks. Moreover, we distribute a workload to each node proportionally to its computational power.

A. Parallel Runge-Kutta

Take as input a grid of $nx \times ny \times nz$ of points which are the starting points of RK3, a function $f'(x, y)$ and a timestamp h. Moreover, it is given the computational power of each node and an integer N representing the total number of steps.

```
1.   - Let $T_f$ be the total of Gflops of the platform and let $T_i$
       be the computational power of the compute node $i$. Send
       to node $i$ $(nz * zy * nz) * (\frac{Ti}{Tf})$ points.
2.   - If node $i$ is a CPU then perform the following steps:
2.1.   Suppose we have $t$ threads so that each thread computes
       $((nz * zy * nz) * (Ti/T_f))/t$ elements:
       Compute the new x and y based on Eq. 2, 3, and 4.
3.   - If node $i$ is a GPU then perform the following steps:
3.1.   Let $v$ be an array of points attributed to $i$. Transfer
       array $v$ to global memory of the device.
3.2.   Invoke the kernel RK3 with $((k_1 + k_2 + 4k_3)/6)/32$
       blocks and 32 threads per block.
3.3.   Set $e = blockIdx.x * blockDim.x + threadIdx.x$;
3.4.   For $n_i$ from 0 to $N - 1$ do:
       Compute the new x and y based on Eq. 2, 3, and 4.
```

B. Parallel WSM6

We developed the CUDA WSM6 to support multi-level parallelism and pipelining. Each equation responsible for calculating the transition from one hydrometeor to another (as described in details by [15]) can to be computed in parallel.

Let p be a partition of the domain points with $x \times y \times z$ points according to Fig. 1.

```
1.   For $i = 0$ to $x$, do:
1.1.   Let the set of blocks $B = (b_0, b_1 ... b_y)$ be such that
       each $b_i \in B$ has $z$ threads.
       Each thread $t \in b_i$ computes the hydrometeors droplets
```

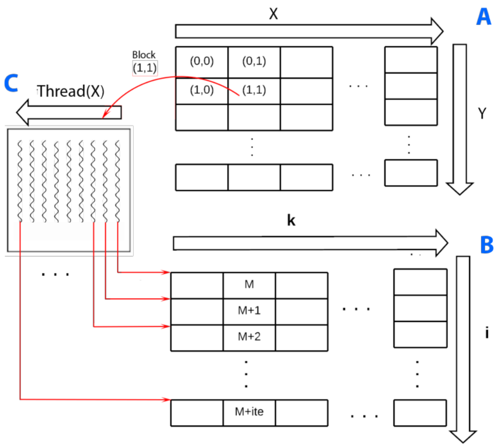


Fig. 1. GPU memory access of the M(k,i): (A) GPU blocks organized in two dimensions; (B) Matrix M(k,i), (C) A threads block of the (A) where each block thread accesses the matrix den with coalesced memory access.

(qc), ice particles (qi), rain droplet (qr), snow crystal (qs), graupel (qg) and water vapor (qv) as folows: Let k be the thread index and j the block index. For each (qc,qi,qr,qs,qg,qv) in $p(k,i,j)$ compute the transition considering the variables of that point.

VI. Pipeline Parallel

Pipelining is extensively utilized by hardware designers. Enhancement is accomplished by dividing each processed instruction into multiple stages. Below we describe how we apply the pipeline technique at the software layer, where the domain to be computed on the GPU is divided into partitions. Each partition represents a stage, allowing parallel processing.

We note that there are two main tasks of system working alternately, namely the dynamic RK3 and the microphysics WSM6. Each task is implemented in a different kernel. In this approach, as soon as a kernel finishes its job another kernel is launched. Hence, during some time the GPU is idle or wasting time with data exchange and, as result, reducing efficiency.

Firstly we identify data used by both RK3 and WSM6 and that may remain within the GPU memory during two consecutive call of different kernels avoiding data exchange.

Additionally, we noted that the idle GPU is the most important bottleneck to achieve better performance. To solve this problem the pipeline parallelism could be a interesting strategy. So, each part of the domain attributed to a GPU is divided into small partitions such that can be processed in parallel by the pipeline stages. Once a stage completes the WSM6 processing for its partition, that partition can start the task RK3 on it corresponding to the next stage in the pipeline. Similarly, as the next stage completes its RK3 processing for its partition it starts the WSM6 processing.

Each GPU executes its own pipeline independently. For clarity of presentation let us assume we have a two-stage pipeline. In this version we split the domain points given as input in two partitions p_1 and p_2: The RK3's and WSM6's kernels were implemented in such a way as to allow splitting the entry points into two partitions. Thus, we are able to execute RK3 with input p_2, while simultaneously, on the same GPU, we execute WSM6 with input p_2.

VII. Multi-Stages Pipeline RK3 WSM6

Using the same ideas of the previous section we may divide the domain in n blocks. Hence, we can generalize the pipelining for $2n$ stages considering n partitions of points.

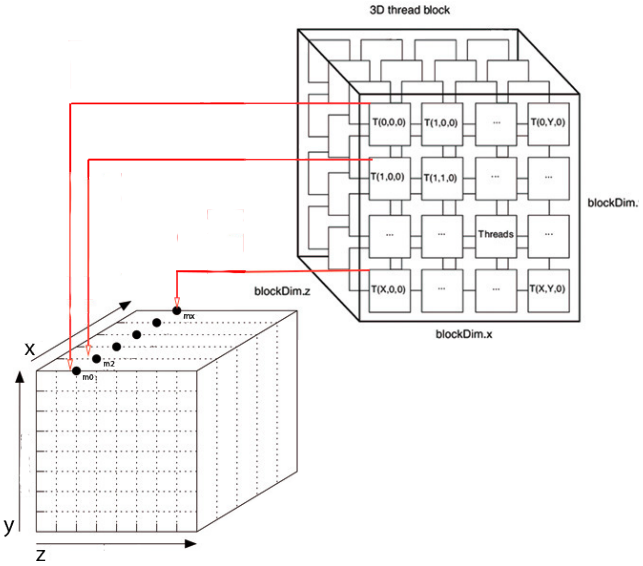


Fig. 2. GPU Memory access of the M(x,y,z): The thread (x,y,z) accesses the rank (x,y,z) of the M such that if the rank in the global memory of the element (x,y,z) is m, the rank of the element $(x+1,y,z)$ will be m+1.

The second step in implementing involves properly modeling the data structures to ensure coalesced access to the GPU's memory. This step is performed both in loops operating on three-dimensional (3D) portions of the domain and in two-dimensional (2D) portions. The way to map threads and blocks depends on the type of loop and its interaction with the data structure. In general, the structures are mapped as follows.

For 3D loops, threads in a block are mapped to a contiguous subset of elements in a specific dimension as can be seen in Figure 2, blocks are mapped to different portions of the 3D domain and for 2D loops, threads in a block are mapped to a contiguous rectangular region of the data in two dimensions, blocks are mapped to different sections of the 2D domain. The goal is to maximize coalesced access to the GPU's memory, allowing threads to simultaneously access data, taking advantage of the CUDA architecture.

We employed an optimization involving the use of constant memory and texture memory to variables and small arrays that remain constant during the execution. By storing these constant values in dedicated memory, we can benefit from their optimized access and reduce the memory traffic.

Shared memory is a valuable resource on GPUs. It is notably more limited than GPU global memory and is visible to threads within the same block. Due to this feature, we chose to rub this implementation with a configuration that prioritizes a wider cache over shared memory. Shared memory is firstly used for some loops that allow for blocking optimizations. Moreover, it is used to store some intermediate results and, when possible, loop invariants. This strategy maximizes the efficiency of shared memory utilization.

Our second implemented optimization is loop fusion. Since loops often iterate over the same data, they can be combined into a single loop without compromising the accuracy, depending on the operation. Along with loop fusion, we also strive to organize memory accesses to the same structure as closely as possible in the code whenever feasible.

Another optimization we used involved rewriting code snippets related to floating-point operations. It's important to exercise caution when applying this type of optimization to avoid introducing discrepancies between the CPU and GPU code. Taking care, we ensure that our CUDA implementation is both accurate and efficient.

Our last optimization creates a pipeline that splits RK3 processing into up to four stages. It works as follows: initially, a CPU thread is assigned for each stage and a master thread oversees operations. At the start, all threads are blocked. The master selects the thread responsible for processing the initial data and enters a busy-wait state. Once the kernel signals that more data can be processed, the master unblocks a thread. This process continues until no CPU threads are blocked. When a thread finish, it chooses a new data block and blocks itself, waiting for the master to unblock it.

A. *WSM6-RK3-Pipeline*

The joint execution of the RK3 pipeline and the multi-kernel implementation of WSM6 enables the creation of an integrated pipeline that performs the combined computation of both tasks. This implementation resembles the thread management described in the RK3 pipeline. Each of these stages is assigned to a thread on CPU. This mechanism has a similar principle to the RK3 pipeline. This case requires monitoring two distinct CUDA streams, one for RK3 and another for WSM6. Moreover, it is crucial to check for dependencies that arise between the stages of WSM6, thus ensuring proper synchronization between stages.

VIII. Experimental results

We performed a two-domains WRF simulation for March 4th of 2021, starting at 00UTC with duration of 12h. See Fig. 3 for details. The set of physical parameterization used in the simulation was: (a) WSM6 microphysics [15]; (b) longwave and (c) shortwave radiation schemes of Community Atmospheric Model 3 (CAM) [19]; (d) planetary boundary layer scheme of Yonsei University [20]; and, (e) Kain-Fritsch for cumulus parametrization [21].

A. *Experiments*

In our tests we first measured the time of the RK3 and WSM6 separately. After that, we measured the overall system performance considering both the stages.

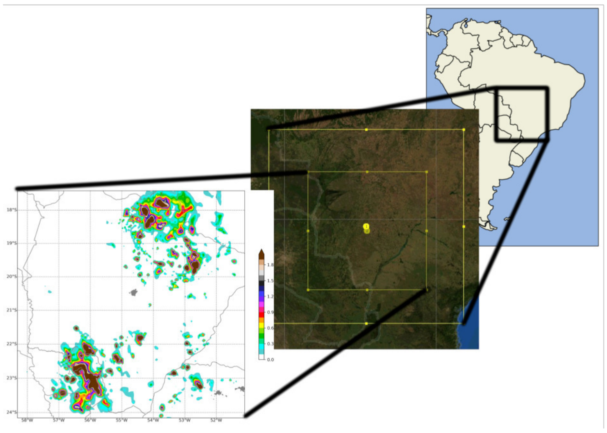


Fig. 3. Domains of the WRF simulation with 12 km of horizontal resolution: X-axis and Y-axis represent latitude and longitude, respectively. The 4 km horizontal resolution domain is represented by the highlighted square in the center of the 12-km resolution domain. The model is set to 34 vertical levels. Colors represent the vertical levels sum for q_c, q_i, q_r, q_s and q_g.

TABLE I
Time and acceleration of the RK3 and WSM6

	RK3		WSM6		WRF
Hardware	time(s)	speedup	time(s)	speedup	Total
12-thread-CPU	1335	-	2956	-	6156
1xP100	125	10.2	520	5.6	2857
1xP100+1xk80	96	13.9	436	6.7	2532
1xP100+2xk80	77	17.3	280	10.5	2410
2xP100	71	18.8	225	13.1	2337
1xP100+4xk80	53	25.2	169	17.4	2009
2xP100+4xk80	34	39.3	112	26.3	1776

With the CUDA-RK3, we achieved a speedup of 10.6 using one GPU P100 and 39.2 using 2x P100 + 4x k80 when compared to 12 thread version (see Table I). we also see the run-time and speedup of the CUDA-WSM6. We achieved speedups ranging from 5.6 on one GPU P100 to 26.3 on the 2x P100 + 4xk80 set. We note we obtained increasing speedup consistently with the growth of computational power.

TABLE II
Time (in seconds) of the RK3+WSM6 with pipeline stages

stages RK3+WSM6	1	2	4	8	16
12-Thread-CPU	6156	6156	6156	6156	6156
1x P100	3023	2802	2525	2420	2318
1x P100 + 1xk80	2736	2445	2212	2116	2048
1x P100 + 2xk80	2614	2343	2114	2022	1979
2x P100	2535	2214	1998	1911	1828
1x P100 + 4xk80	2213	1892	1707	1633	1502
2x P100 + 4xk80	1980	1698	1523	1463	1448

Table II show the overall performance when we increase the computational power(in each column). Besides, we notice the time gain when varying the number of pipeline stages.

Besides we apply the technique of overlapping transfer and processing through the use of asynchronous transfers. This enabled performance gains as can be seen in the Table III.

TABLE III
TIME (IN SECONDS) OF THE WSM6+RK3 WITH OVERLAPPING

stages RK3+WSM6	1	2	4	8	16
Ryzen 1600 12 Ths	6156	6156	6156	6156	6156
1x P100	2857	2613	2315	2216	2079
1x P100 + 1xk80	2532	2242	2009	1913	1844
1x P100 + 2xk80	2410	2140	1909	1818	1774
2x P100	2337	2014	1869	1728	1612
1x P100 + 4xk80	2009	1690	1502	1430	1300
2x P100 + 4xk80	1776	1493	1319	1260	1246

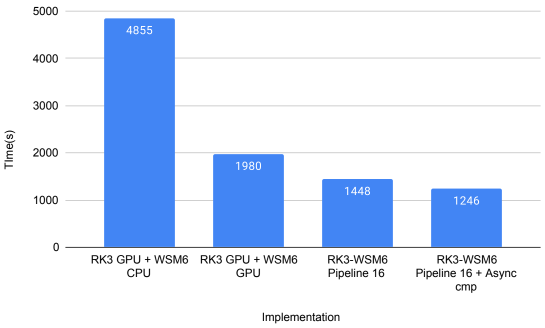


Fig. 4. Runtime of the implementation on platform 2xP100 + 4xK80.

Fig. 4 show the result using our load balance heuristic. We see that with each new device added, we have a performance gain nearly proportional to the increase in computational power, so that our implementation proves to be scalable.

B. Accuracy Analysis

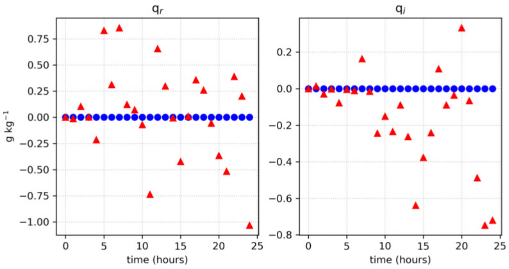


Fig. 5. The bias of the vertical profile in relation to the CPU simulation for mixing ratio of rain droplets (q_r) and ice particles (q_i) from 00UTC on March 4th to 00UTC on March 6th, 2021. The x-axis is in g kg^{-1} and the y-axis is in σ vertical coordinate. The blue and red markers are related GPU arithmetic optimizations off and GPU arithmetic optimization on, respectively.

To measure the accuracy of our GPU implementation, we used the results of the CPU implementation as a reference and analyzed the deviations of the GPU solution. We examined the results of the solution with the configuration 2xP100+4xK80.

The Fig. 5 show the time biases. We illustrate the output hydrometeors ice particles and rain droplets to compare the optimization levels. We can note a deviation between the GPU and the CPU implementation. We analyzed a version of the GPU implementation with enabled optimizations and another without optimizations. Note that when these optimizations are disabled, the deviation from the CPU implementation is negligible. When they are enabled, some points show a reasonable deviation compared to the CPU.

REFERENCES

[1] M. R. Petersen, X. S. Asay-Davis, D. W. Jacobsen, M. E. Maltrud, T. D. Ringler, L. Van Roekel, C. Veneziani, and P. J. Wolfram, Jr., "MPAS-Ocean model user's guide version 6.0," OSTI, Tech. Rep., 4 2018.

[2] S. R. Freitas, J. Panetta, K. M. Longo, L. F. Rodrigues, D. S. Moreira *et al.*, "The brazilian developments on the regional atmospheric modeling system (BRAMS 5.2): an integrated environmental model tuned for tropical areas," *Geosci. Model Dev.*, vol. 10, pp. 189–222, 2017.

[3] W. Skamarock, J. Klemp, M. Duda, L. Fowler, S. Park, and T. Ringler, "A multiscale nonhydrostatic atmospheric model using centroidal voronoi tesselations and c-grid staggering," *Monthly Weather Review*, vol. 140, no. 9, pp. 3090–3105, Sep. 2012.

[4] Z. Ma, Q. Liu, C. Zhao, X. Shen, Y. Wang, J. H. Jiang, Z. Li, and Y. Yung, "Application and evaluation of an explicit prognostic cloud-cover scheme in grapes global forecast system," *Journal of Advances in Modeling Earth Systems*, vol. 10, no. 3, pp. 652–667, 2018.

[5] W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, M. Barker, K. G. Duda, X. Y. Huang, W. Wang, and J. G. Powers, "A description of the Advanced Research WRF Version 3," National Center for Atmospheric Research, Tech. Rep., 2008.

[6] M. Korch and T. Rauber, "Comparison of parallel implementations of runge-kutta solvers: Message passing vs. threads," in *Parallel Computing - Advances in Parallel Computing*, G. Joubert, W. Nagel, F. Peters, and W. Walter, Eds. North-Holland, 2004, vol. 13, pp. 209–216.

[7] L. Murray, "GPU acceleration of runge-kutta integrators," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, pp. 94–101, 2012.

[8] C. Chantrapornchai, S. Dainpraprai, and O. Wongtaweesap, "On the Computer Simulation of Microparticles Capture in Magnetic Filters using OpenMP," *International Journal of Computer Applications*, vol. 51, no. 14, pp. 23–30, Aug. 2012.

[9] M. S. Wo, R. Gobithaasan, and K. Miura, "GPU acceleration of runge kutta-fehlberg and its comparison with dormand-prince method," *AIP Conference Proceedings*, vol. 1605, pp. 16–21, 2014.

[10] B. Bylina and J. Potiopa, "Explicit fourth-order runge—-kutta method on intel xeon phi coprocessor," *International Journal of Parallel Programming*, vol. 45, no. 5, p. 1073–1090.

[11] M. Huang, B. Huang, L. Gu, H. Huang, and M. Goldberg, "Parallel GPU architecture framework for the WRF single moment 6-class microphysics scheme," *Computers & Geosciences*, vol. 83, 06 2015.

[12] J. Y. Kim, J.-S. Kang, and M. Joh, "GPU acceleration of MPAS microphysics WSM6 using OpenACC directives: Performance and verification," *Computers and Geosciences*, vol. 146, p. 104627, Jan. 2021.

[13] H. C. da Silva, M. A. Stefanes, and V. Capistrano, "OpenACC Multi-GPU approach for WSM6 microphysics," in *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*, 2021, pp. 382–387.

[14] J. B. Klemp, W. C. Skamarock, and J. Dudhia, "Conservative Split-Explicit Time Integration Methods for the Compressible Nonhydrostatic Equations," *Monthly Weather Review*, vol. 135, pp. 2897–2913, 2007.

[15] S. Hong and J. Lim, "The WRF Single-Moment 6-Class Microphysics Scheme (WSM6)," *J. Korean Meteor. Soc.*, vol. 42, pp. 129–151, 2006.

[16] Message Passing Interface Forum, *MPI: A Message-Passing Interface Standard Version 4.0*, Jun. 2021.

[17] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[18] S. Cook, *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*. Elsevier Science, 2012.

[19] W. Collins, P. Rasch, B. Boville, J. McCaa, D. Williamson, J. Kiehl, B. Briegleb, C. Bitz, S. Lin, M. Zhang, and Y. Dai, "Description of the NCAR Community Atmosphere Model (CAM 3.0)," Tech. Rep., 2004.

[20] S.-Y. Hong, Y. Noh, and J. Dudhia, "A New Vertical Diffusion Package with an Explicit Treatment of Entrainment Processes," *Monthly Weather Review*, vol. 134, no. 9, pp. 2318–2341, Sep. 2006.

[21] J. S. Kain, "The Kain–Fritsch Convective Parameterization: An Update," *Journal of Applied Meteorology*, vol. 43, no. 1, pp. 170 – 181, 2004.

DOI: 10.15439/2023F6021
ISSN 2300-5963 ACSIS, Vol. 35

Incident Detection with Pruned Residual Multilayer Perceptron Networks

Mohamad Soubra
0000-0002-5195-9540
AGH University
Faculty of Computer Science
Electronics and Telecommunications
Krakow, Poland
Email: soubra@agh.edu.pl

Marek-Kisiel Dorohinicki
0000-0002-8459-1877
AGH University
Faculty of Computer Science
Electronics and Telecommunications
Krakow, Poland
Email: doroh@agh.edu.pl

Marcin Kurdziel
0000-0003-2022-7424
AGH University
Faculty of Computer Science
Electronics and Telecommunications
Krakow, Poland
Email: kurdziel@agh.edu.pl

Marek Zachara
0000-0002-8560-9696
AGH University
Faculty of Electrical Engineering,
Automatics, Computer Science
and Biomedical Engineering
Krakow, Poland
Email: mzachara@agh.edu.pl

***Abstract*—Internet of things (IoT) has opened new horizons in connecting all sorts of devices to the internet. However, continuous demand for connectivity increases the cybersecurity risks, rendering IoT devices more prone to cyberattacks. At the same time, rapid advances in Deep Learning (DL)-based algorithms provide state-of-the-art results in many classification tasks, including classification of network traffic or system logs. That said, deep learning algorithms are considered computationally expensive as they require substantial processing and storage capacity. Sadly, IoT devices have limited resources, making renowned DL models hard to implement in this environment. In this paper we present a Residual Neural Network inspired DL-based Intrusion Detection System (IDS) that incorporates weight pruning to make the model more compact in size and resource consumption. Additionally, the proposed system leverages feature selection algorithms to reduce the feature-space size. The model was trained on the NSL-KDD dataset benchmark. Experimental results show that the proposed system is effective, being able to classify network traffic with an F_1 score of up to 98.9% before the pruning and an F_1 score of up to 97.5% after pruning 90% of network weights.**

I. Introduction

INTERNET of Things (IoT) is booming in markets, driving efforts for increasing device inter-connectivity. However, this strive for increased connectivity poses requirements related to provision of security protocols and measures that would secure communication between devices and build trust in users that their data is communicated privately [1]. In order to meet these requirements current security solutions typically endorse defense in depth approach [2] in which the security layers span across network perimeter, intranet and endpoint systems. Such security mechanisms involve many attack detection and prevention technologies. One of the most important class of these technologies, namely Intrusion Detection Systems (IDS) [3], come in various flavors. Host-IDS examines the actions of the users and compares them to decide which actions can be considered as malevolent and which are likely benign. On the other hand, Network-based IDS, examines the traffic traversing through the network and compares it with already known signatures to distinguish between normal and malevolent flow. Though popular, these systems still face various challenges, such as detection accuracy, high false-alarm rates or the inability to detect zero-day attacks [4].

Machine Learning (ML) and Deep Learning (DL)-based technologies recently enjoy numerous practical deployments, e.g., in speech recognition, object detection, natural language processing, etc. It is also increasingly used in the cybersecurity domain [5][6]. Consequently, ML- and DL-based IDS gained popularity in the recent years. In particular, they have proven to be more robust than their predecessors, having lower false-positive rates and higher accuracy [7]. However, this line of research often adopted renowned image classification algorithms [8] to the traffic classification tasks [9][10][11]. Consequently, the proposed systems tend to be computationally cumbersome. Accordingly, for IoT devices, which have limited storage and processing resources, research increasingly focuses on replacing such burdensome algorithms with much lighter solutions.

In this paper, we introduce a new DL-based IDS designed around lightweight residual network [12] architectures. Our solution is coupled with the Extra Tree classification algorithm, which allows us to extract the most important features from the dataset. This makes the proposed system compact, while retaining high accuracy and detection rates. The small computational footprint of the proposed system is suitable for inference on a CPU, instead of resource-hungry GPU accelerators. Thus, our results show that attaining high accuracy while substantially reducing the size of the model is achievable in IDS tasks.

The following sections begin with review of the state-of-the-art results in ML-based intrusion detection systems. Next, we present the proposed attack detection architecture. Subsequently we describe the experimental setup and report obtained results. Finally, we give conclusions from experiments and outline future work.

II. Related Work

Deep Learning-based intrusion detection systems enjoyed rapid advances in recent years. Some researchers utilized DL capabilities for categorical data classification, where the task is to recognize specific attack instances. Haddad Pajouh et al. [13] proposed a Long-Short Term Memory (LSTM)-based IDS. First, they extracted OpCodes from the traffic and assigned them to input vectors. Next, they leveraged Principal Component Analysis (PCA) to extract the most significant features from the vectors. The model was trained using Adam optimizer [14]. Dropout layers were used to avoid overfitting. The performance of the model was evaluated with 10-fold Cross Validation (CV). Swarna Priya et al. [15] proposed a DNN-based IDS that, similarly to Haddad Pajouh et al. approach, also used PCA as a feature extractor. The system also utilized feature scaling to normalize the input data before feeding it to the classifier. Furthermore, they used Grey Wolf optimization algorithm (GMO) [16] to construct a feature hierarchy. This hierarchy provided features' fitness values. McDermott et al. [17] proposed a Bidirectional LSTM-based IDS. Word embeddings were used to embed the captured packets' content in a vector space suitable for the model. Subsequently, they used word embeddings to establish a dictionary of tokenized words. Sigmoid function, Mean Absolute Error (MSE) and Adam were selected as the activation function, loss function and optimizer, respectively. Zhang et. al [18] proposed a Deep-Belief Network (DBN)-based IDS that employed an improved genetic algorithm. The algorithm incorporated improved crossover and elite retention strategies to prevent the loss of the best individuals. The proposed system was trained and evaluated on the NSL-KDD dataset. Another DBN-based IDS was proposed by Tama et al. [19]. The system incorporated a grid search strategy to select the most significant input features. Evaluation was carried out on three datasets, namely, UNSW-NB15 [20], CIDDS-001 [21], and GPRS [22] using 10-folds cross validation, Repeated Cross-Validation (RepCV) [23] and data sub-sampling. Their model was able to maintain the same detection rate after sub-sampling. Overfitting was prevented with L1 and L2 regulations and an adaptive learning rate. Muna et al. [24] proposed a Deep Autoencoder to reduce the features dimensionality. Their system also encompassed a deep feed-forward Neural Network to detect and classify traffic. It was trained and evaluated on the NSL-KDD dataset. Latif et al. [25] emphasized the importance of providing lightweight DL-based IDS solutions. To this end, they proposed an intrusion detection algorithm employing random neural networks, in which the Poisson distribution was used to estimate the probability of the signals that made the neurons either active or inhibited. The proposed system was evaluated on the DS2OS dataset [26]. Shone et al. [27] proposed a Non-Symmetric Deep Auto-Encoder for unsupervised feature learning. The system employed Random Forest [28] to classify the traffic between benign and malevolent. Both NSL-KDD and KDD Cup '99 datasets were used in training and evaluation. Min et al. [29] proposed a system which uses an ensemble of byte-level word embeddings and text convolutional neural networks. Skip-Gram algorithms was used to create the byte-level word embeddings. Text convolutional neural networks were constructed from one-dimensional convolutions that extracted word-based features. Similarly to Shone et al., Random Forest was chosen as a classifier. The system was trained and evaluated on the ISCX2012 dataset [30]. Zhou et al. [31] proposed Deep Feature Embedding Learning method that reduces input features' dimensionality, thereby decreasing the time needed to train the model. They trained and evaluated their model on the NSL-KDD and UNSW-NB15 datasets. Leaky ReLU was chosen as the activation function for the hidden layers, while Sigmoid function was used as an activation function in the classification layer. Additionally, Dropout was used to avoid overfitting.

Other researchers choose to use deep learning for binary classification, where the goal is to distinguish attack signatures from normal traffic, irrespective of specific attack classes. Diro et al. [32] proposed a DL-based IDS trained in a distributed optimization scheme which involved fog nodes, i.e. mini-clouds implemented as edge devices in the cloud [33]. To avoid overfitting, the parameters were collected in the fog coordinator, which was responsible for their updating and distribution for subsequent epochs. Diro et al. [32] evaluated their system on the NSL-KDD dataset [34]. Similarly, Abeshu et al. [35] proposed a novel DL-based IDS that takes its parameters from the master fog node, while performing system fine-tuning on the worker nodes. Again, NSL-KDD was chosen as training and evaluation dataset. Almiani et al. [36] proposed an RNN-based IDS. Their system employed data oversampling to balance the minority classes, a modified back-propagation algorithm, and the min-max normalization. Kasongo et al. [37] proposed a feed-forward Neural Network-based IDS that was coupled with a wrapper-based feature extraction unit. The wrapper used the Extra Tree algorithm to classify and specify which features are most significant. The proposed system was trained and evaluated on the NSL-KDD dataset. Devan et al. [38] proposed an XGboost DL-based IDS composed of three main steps, namely, input feature normalization, feature selection using a classifier based on a collection of decision trees that derive the significant features, and final classification. Their system also leveraged neural networks with ReLU and Softmax activation functions for the hidden and classification layers, respectively. Nagisetty et al. [39] proposed a DL-based IDS that incorporated three

DL architectures: Multi Layer Perceptrons (MLP), CNNs, and an Autoencoders. The proposed system was trained and evaluated on two datasets, namely, UNSW-NB15 and NSL-KDD99. The system employed Root Mean Square Root Error (RMSE) as the cost function. DNN was used mainly to sort the features and create a feature hierarchy. Zhihan et al. [40] proposed a hierarchical Supporting Vector Machine-based IDS. In addition, a stacked autoencoder was used to denoise the data. The system was evaluated on the NSL-KDD dataset.

In this paper we will benchmark our results with the papers that focus on the binary classification task. To this end, we will evaluate our algorithm with respect to the metrics that they have discussed in their papers as our goal is to see if our pruned networks could compete with the state-of-the-art.

III. Proposed Architecture

In order to protect devices from attacks, while preserving processing and storage resources, we propose an Intrusion Detection System based on pruned residual neural networks [12]. The proposed system is trained in several steps. First, input data is pre-processed, including encoding of symbolic features. The data is then fed to an Extra Tree Classifier [41], which selects the most important features from the feature set. In the next step, the data is normalized and used to train the proposed classification model. Finally, the model is pruned in fine-tuning steps, which minimizes its size and the inference cost.

We evaluate the final model with respect to precision, recall and F_1 score both before and after network pruning. Evaluation is carried on the NSL-KDD dataset.

A. NSL-KDD Dataset

The NSL-KDD dataset is the successor of the KDD'99 [42] dataset, which was introduced by DARPA in 1998. The dataset was firstly proposed by Tavallaee et al. [43] and is composed of 4 different attack classes, namely, Denial of Service (DoS), Probe, User-to-root (U2R), and Remote-to-Local (R2L). In DoS attacks the computing or network resources are exhausted, making the attacked system unable to serve the user's requests. Signatures of a DoS attack in the NSL-KDD dataset would be, e.g., the `Src_byte` and the `Wrong_fragment` features. Probe attacks are mostly used for surveillance, in order to to gain information on the potential victim system. The relevant signatures for probe attacks in the NSL-KDD dataset are the `Src_bytes` and the `Duration` features. User-to-root attacks attempt to grant superuser privileges to the attacker. One way of doing this is accessing the user's system via a normal account and then attempting to escalade privileges by exploiting a vulnerability. Relevant signatures for U2R attacks with respect to the NSL-KDD dataset are, e.g., `Num_file_creations` and `Num_shells` features. In Remote-to-Local attacks the attacker attempts to gain access of the user's system via a remote machine. Relevant signatures for R2L attacks in the NSL-KDD dataset are, e.g., `Duration`, `Service` and `Num_failed_logins` features.

TABLE I: NSL-KDD traffic statistics.

	NSL-KDD		
	Attack Type	***KDDtrain+***	***KDDTest***
1	DOS	45926	7458
2	Probe	11655	2421
3	R2L	995	2754
4	U2R	52	200
5	Normal	67345	9711
Total		125973	22544

The NSL-KDD dataset encompasses two subsets, namely, the KDDtrain+ and KDDtest. In standard classification setup, the proposed system should assign the signatures into four major categories, namely, `DOS`, `Probe`, `R2L`, `U2R`, and `Normal` traffic. Table I reports datasets statistics for these categories. Note that `DOS`, `Probe`, `R2L`, `U2R` and normal traffic makes 36.45%, 9.25%, 0.78%, 0.04% and 45.52% of the dataset instances, respectively. In binary classification setup the proposed system should be able to classify the traffic into two classes, namely, attack and non-attack. Note that classes in this case are balanced, with attack and non-attack traffic making 46.5% and 53.5% of the dataset, respectively.

The NSL-KDD consists of a total of 41 features that comes in four main categories: (a) intrinsic features that can be extracted from the packet's headers, (b) content features which reflect the data content of the packets, (c) time-based features which reflect the connection rates with the hosts, and finally (d) the host-based features. It is also worth mentioning that the KDDtrain+ subset has 3 categorical features, namely:

- `Protocol Type` which consists of 3 categories,
- `Services` which consists of 70 categories,
- `Flag` which consists of 11 main categories.

These features require preprocessing into one-hot encoding before they can be used in the subsequent steps.

B. Data Preprocessing

In this work we focus on a binary classification task, i.e., distinguishing normal network traffic from attacks. We therefore convert the provided labels into attack and non-attack classes before selecting the important features. Next, we remove data duplicates and rows that contain null values. The KDDtrain+ subset consists of both numerical and categorical data. We normalize the numerical features via z-scores:

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where x represent the current instance of the feature while μ and σ represent the mean and the standard deviation of the feature respectively. The categorical features, on the other hand, are encoded in one-hot vectors.

In the next step we use Extra Tree classifier with 100 estimators (trees) to identify the most significant features. Specifically, we use the Gini coefficients [44] returned by the Extra Tree classifiers to select the most prominent features. Importantly, for one-hot-encoded features, the feature is retained if the Extra Tree classifier selects any of its dimensions according to the Gini coefficient. After a series of features selection iterations, the features listed in Table II were used to train the neural network for the binary classification task.

TABLE II: Features selected by the Extra Tree classifier.

Index	NSL-KDD Features	Index	NSL-KDD Features
1	count	12	protocol_type
2	same_srv_rate	13	logged_in
3	dst_host_count	14	rerror_rate
4	dst_host_same_srv_rate	15	same_srv_rate
5	dst_host_serror_rate	16	serror_rate
6	dst_host_same_src_port_rate	17	service
7	dst_host_same_srv_rate	18	flag
8	dst_host_rerror_rate	19	src_bytes
9	dst_host_srv_count	20	srv_rerror_rate
10	dst_host_srv_diff_host_rate	21	srv_serror_rate
11	dst_host_srv_serror_rate	22	dst_host_srv_rerror_rate

C. Pruning

We use pruning to reduce the overall size of the trained neural model. There are several pruning strategies that can be used to this effect:

- The classical approach in which the model is firstly trained with all parameters and then subset of the trained parameters is removed during additional fine-tuning epochs.
- Pruning at initialization, where parameters are pruned before the model is trained [45].
- Pruning during the main training run.

Furthermore, pruning can carried out globally, i.e., across the whole model, or locally, i.e., in each network layer [46].

In this work we use global, magnitude-based pruning which employs fine-tuning epochs after the main training run, during which weights with low magnitudes are gradually set to zero.

IV. Experimental Setup

Table III summarizes the hyper-parameters used in the experiments. These training hyperparameters were selected with few trial training runs. We evaluate variants of this architecture with varying widths. In particular, we vary the number of neurons inside the residual blocks while keeping a fixed network width of the skip-connection nodes. Furthermore, batch normalization layers [47] are used to improve the training. This architecture proved to work well, while saving on the number of model parameters. Each constructed model was run five times with different random seeds.

We also carried out evaluation of pruned variants of our models. To this end, a 20 epoch fine-tuning run with magnitude-based pruning was done. For each network instance the sparsity schedule started with 85% initial sparsity and increased with each iteration, until it reached a final sparsity of 90% by the end of the last fine-tuning epoch. For the performance numbers we report mean and variance of training time, test accuracy, precision, recall and F_1 score:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

TABLE III: List of training hyper-parameters.

Models width	1024, 256, 32, 8
Activation function	ReLU [48]
Optimizer	Stochastic Gradient Descend (SGD) with Nesterov accelerated gradient = 0.9 [49]
Loss function	Binary Cross Entropy [50]
Learning rate	0.1
Decay for unpruned models	1e-6
Decay for pruned models	Polynomial Decay
Batch size	120
Number of epochs (for both pruned and unpruned networks)	20

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

where TP is the true-positive count, FP is the false-positive count, TN is the true-negative count and FN is the false-negative count. Due to the substantial pruning rate, the sparse models of the same width tended to have the same performance across random seeds. Consequently, the variance estimates are not meaningful in this case and we don't report them.

V. Results

The proposed models were trained using the Google Collab environment. Table IV reports the results for unpruned networks. The model with the highest width achieved 98.96% accuracy, 99.39% precision, 98.38% recall and 98.91% F_1 score. Note that this is the most computationally expensive of our models. That said, the model with quarter the width preformed equally well up to the variance across random seeds. The remaining two models performed slightly worse, with F_1 score around 0.6% below that of the larger models. These models were, however, much more computationally efficient, with the training time stabilizing below width equal to 32 units. Our results also shows that the variance across the training runs is low for all models, which shows that the performance is not highly affected by the initial seeds.

Results for the pruned networks are reported in Table V. The F_1 score of the model with 1024-unit width dropped by about 1.5% after pruning, with performance decrease manifesting mostly in model's recall. For the pruned model with quarter the width, the performance metrics were about 0.5% below those of the larger pruned model and up to 2% below the unpruned network. The two smallest models scored the lowest after pruning, with an F_1 score approximately 2% below larger pruned networks. Overall, our results shows that even with aggressive pruning and small initiated models residual fully-connected networks perform well in this task, with precision recall and F_1 score above 95%.

TABLE IV: Performance metrics for unpruned models.

Model	Training Time(sec)	Accuracy	Precision	Recall	F1 Score
1024	360.6±35.7	98.96±0.03%	99.39±0.10%	98.38±0.11%	98.91±0.07%
256	155.4±6.5	98.91±0.05%	99.44±0.02%	98.22±0.08%	98.82±0.05%
32	72.9±7.1	98.38±0.07%	81.07±0.16%	97.45±0.23%	98.25±0.08%
8	73.7±8.5	98.38±0.07%	99.07±0.16%	97.45±0.23%	98.25±0.08%

TABLE V: Performance metrics for pruned models.

Model	Training Time(sec)	Accuracy	Precision	Recall	F1 Score
1024	471.3	97.67%	99.42%	95.59%	97.46%
256	217.9	97.31%	99.22%	95.01%	97.07%
32	118.6	95.82%	95.91%	95.13%	95.52%
8	119.3	95.82%	95.91%	95.13%	95.52%

TABLE VI: Performance metrics reported in related work.

Reference	Accuracy	Precision	Recall	F1 Score
[32]	99.20%	99.02%	99.27%	99.14%
[35]	99.20%	-	99.27%	-
[36]	92.42%	90.20%	-	92.29%
[37]	99.37%	-	92%	-
[38]	97.60%	97%	97%	97%
[39]	98.96%	-	-	92.28%
[40]	97.83%	-	-	-
Our unpruned 1024 model	98.96±0.03%	99.39±0.10%	98.38±0.11%	98.91±0.07%
Our pruned 32 model	95.82%	95.91%	95.13%	95.52%

To benchmark our results against the state of the art, we selected peer-reviewed papers which addressed the binary classification task with respect to the same NSL-KDD dataset. Some of these papers reported all the metrics mentioned earlier, while others took into consideration only a subset of them. Comparison between the benchmarks and our results is summarized in in table VI.

Comparing with the state-of-the-art for this benchmark dataset in binary classification setup, we observe that all of the proposed unpruned networks give competitive or better precision in detecting attacks (Table VI). More precisely, the models with 1024 and 256 widths achieved better accuracy compared to [36] [38] [39] [40], recall compared to [37] [38] and F_1 score compared to [36] [38] [39]. The pruned models achieved slightly lower results, but still maintained strong performance while requiring only 10% of the initial parameters.

VI. Conclusions and Future Work

Proliferation of IoT devices is making a huge impact on the communication sector. The increased interconnectivity comes not only with new business opportunities, but also increases security risks related to prevalence of network vulnerabilities and persistent cyberattack threats. Conventional IDS and firewalls deployed to counter cyber-threats are often inadequate for IoT environments, e.g., due to high false-positive rates or large resource requirements. In this paper we proposed an ML-based IDS that employs residual MLP networks and demonstrated that it provides strong results with respect to the precision and recall of attack detection, even when implemented with relatively small networks. We also demonstrated that it retains most of its accuracy after pruning of as much as 90% of its parameters.

In our future work we intend to extend this line of research with novel and promising neural architectures, e.g., transformer models. These models excel at text embedding and classification. We therefore intend to explore their ability to classify network and system logs. We also intend to explore more pruning strategies, e.g., unit-based pruning which removes entire neurons, rather than individual weights. Such pruning strategies may result in lower computational footprint, while still maintaining strong attack detection performance.

References

[1] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on smart homes system using internet-of-things," in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*. IEEE, 2015, pp. 0330–0335.

[2] D. Kuipers and M. Fabro, "Control systems cyber security: Defense in depth strategies," Idaho National Lab.(INL), Idaho Falls, ID (United States), Tech. Rep., 2006.

[3] Y. Lin, C. Wang, C. Ma, Z. Dou, and X. Ma, "A new combination method for multisensor conflict information," *J. Supercomputing*, vol. 72, no. 7, pp. 2874–2890, 2016.

[4] H. Liao, C. R. Lin, Y. Lin, and K. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network Computing Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[5] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[6] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[7] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, "On the effectiveness of machine and deep learning for cyber security," in *10th International Conference on Cyber Conflict, CyCon 2018, Tallinn, Estonia, May 29 - June 1*, T. Minárik, R. Jakschis, and L. Lindström, Eds. IEEE, 2018, pp. 371–390.

[8] S. H. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, ser. Synthesis Lectures on Computer Vision. Morgan & Claypool Publishers, 2018.

[9] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, "An intrusion detection model based on feature reduction and convolutional neural networks," *IEEE Access*, vol. 7, pp. 42 210–42 219, 2019.

[10] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking, ICOIN 2017, Da Nang, Vietnam, January 11-13*. IEEE, 2017, pp. 712–717.

[11] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50 850–50 859, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30*. IEEE Computer Society, 2016, pp. 770–778.

[13] H. H. Pajouh, A. Dehghantanha, R. Khayami, and K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computing Systems*, vol. 85, pp. 88–96, 2018.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[15] R. M. S. Priya, P. K. R. Maddikunta, P. M., S. Koppu, T. R. Gadekallu, C. L. Chowdhary, and M. Alazab, "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in iomt architecture," *Computing and Communication*, vol. 160, pp. 139–149, 2020.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[17] C. D. McDermott, F. Majdani, and A. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13*. IEEE, 2018, pp. 1–8.

[18] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for iot based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31 711–31 722, 2019.

[19] B. A. Tama and K.-H. Rhee, "Attack classification analysis of IoT network via deep learning approach," *Res. Briefs Inf. Commun. Technol. Evol.(ReBICTE)*, vol. 3, pp. 1–9, 2017.

[20] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015, Canberra, Australia, November 10-12*. IEEE, 2015, pp. 1–6.

[21] M. Ring, S. Wunderlich, D. Grüdl, D. Landes, and A. Hotho, "Flow-based benchmark data sets for intrusion detection," in *Proceedings of the 16th European conference on cyber warfare and security. ACPI*, 2017, pp. 361–369.

[22] D. W. Vilela, T. F. Ed'Wilson, A. A. Shinoda, N. V. de Souza Araújo, R. De Oliveira, and V. E. Nascimento, "A dataset for evaluating intrusion detection systems in [ieee] 802.11 wireless networks," in *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*. IEEE, 2014, pp. 1–5.

[23] J. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Computational Statistics and Data Analysis*, vol. 53, no. 11, pp. 3735–3745, 2009.

[24] M. Al-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *Journal of Information Security and Applications*, vol. 41, pp. 1–11, 2018.

[25] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," *IEEE Access*, vol. 8, pp. 89 337–89 350, 2020.

[26] M. Pahl and F. Aubet, "Ds2os traffic traces IoT traffic traces gathered in a the ds2os iot environment," 2018.

[27] N. Shone, N. N. Tran, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[28] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, 2016.

[29] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest," *Secur. Commun. Networks*, pp. 4 943 509:1–4 943 509:9, 2018.

[30] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.

[31] C. Lin, Z. Wang, J. Deng, L. Wang, J. Ren, and G. Wu, "mts: Temporal-and spatial-collaborative charging for wireless rechargeable sensor networks with multiple vehicles," in *2018 IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April 16-19*. IEEE, 2018, pp. 99–107.

[32] A. A. Diro and N. K. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," *Future Generation Computing Systems*, vol. 82, pp. 761–768, 2018.

[33] A. A. Bukhari, F. K. Hussain, and O. K. Hussain, "Fog node discovery and selection: A systematic literature review," *Future Generation Computing Systems*, vol. 135, pp. 114–128, 2022.

[34] L. Dhanabal and S. Shantharajah, "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms," *International journal of advanced research in computer and communication engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[35] A. A. Diro and N. K. Chilamkurti, "Deep learning: The frontier for distributed attack detection in fog-to-things computing," *IEEE Communincation Magazine*, vol. 56, no. 2, pp. 169–175, 2018.

[36] M. Almiani, A. A. Ghazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, "Deep recurrent neural network for iot intrusion detection system," *Simulation Modelling and Practice Theory*, vol. 101, p. 102031, 2020.

[37] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computer Security*, vol. 92, p. 101752, 2020.

[38] P. Devan and N. Khare, "An efficient xgboost-dnn-based classification model for network intrusion detection system," *Neural Computations and Applications*, vol. 32, no. 16, pp. 12 499–12 514, 2020.

[39] A. Nagisetty and G. P. Gupta, "Framework for detection of malicious activities in iot networks using keras deep learning library," in *2019 3rd international conference on computing methodologies and communication (ICCMC)*. IEEE, 2019, pp. 633–637.

[40] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep-learning-enabled security issues in the internet of things," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9531–9538, 2021.

[41] M. Pal, "Random forest classifier for remote sensing classification," *International journal of remote sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[42] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on KDD 99 intrusion detection datasets," in *Proceedings of the third annual conference on privacy, security and trust*, vol. 94. Citeseer, 2005, pp. 1723–1722.

[43] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, Ottawa, Canada, July 8-10*. IEEE, 2009, pp. 1–6.

[44] G. Pyatt, "On the interpretation and disaggregation of gini coefficients," *The Economic Journal*, vol. 86, no. 342, pp. 243–255, 1976.

[45] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7*. OpenReview.net, 2021.

[46] D. W. Blalock, J. J. G. Ortiz, J. Frankle, and J. V. Guttag, "What is the state of neural network pruning?" in *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4*, I. S. Dhillon, D. S. Papailiopoulos, and V. Sze, Eds. mlsys.org, 2020.

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July*, ser. JMLR Workshop and Conference Proceedings, F. R. Bach and D. M. Blei, Eds., vol. 37. JMLR.org, 2015, pp. 448–456.

[48] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3-4, pp. 121–136, 1975.

[49] I. Sutskever, J. Martens, G. E. Dahl, and G. E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June*, ser. JMLR Workshop and Conference Proceedings, vol. 28. JMLR.org, 2013, pp. 1139–1147.

[50] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

DOI: 10.15439/2023F8601
ISSN 2300-5963 ACSIS, Vol. 35

Optimizing Machine Translation for Virtual Assistants: Multi-Variant Generation with VerbNet and Conditional Beam Search

Marcin Sowański
ORCID: 0000-0002-9360-1395
Samsung R&D Institute Poland, Warsaw, Poland
Warsaw University of Technology, Warsaw, Poland
Email: m.sowanski@samsung.com

Artur Janicki
ORCID: 0000-0002-9937-4402
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warsaw, Poland
Email: artur.janicki@pw.edu.pl

***Abstract*—In this paper, we introduce a domain-adapted machine translation (MT) model for intelligent virtual assistants (IVA) designed to translate natural language understanding (NLU) training data sets. This work uses a constrained beam search to generate multiple valid translations for each input sentence. The search for the best translations in the presented translation algorithm is guided by a verb-frame ontology we derived from VerbNet. To assess the quality of the presented MT models, we train NLU models on these multiverb-translated resources and compare their performance to models trained on resources translated with a traditional single-best approach. Our experiments show that multi-verb translation improves intent classification accuracy by 3.8% relative compared to single-best translation. We release five MT models that translate from English to Spanish, Polish, Swedish, Portuguese, and French, as well as an IVA verb ontology that can be used to evaluate the quality of IVA-adapted MT.**

I. Introduction

MULTILINGUAL natural language understanding (NLU) models are a major focus in natural language processing (NLP) as they enable virtual assistants to manage multiple languages. However, the scarcity of multilingual training data often leads to under-representation of some languages. While the manual translation of training sentences can address this problem, it is a time-consuming and costly process prone to errors and ambiguities that can compromise model quality. Moreover, manual translation struggles to adapt to language changes or the introduction of new languages to the virtual assistant.

In this context, using machine translation (MT) systems as a source of translations seems to be an attractive alternative for acquiring multilingual learning data. Creating multilingual NLU models by translating a learning sentence into multiple languages using MT models seems possible and promising.

MT systems, used to generate sentences for training NLU models, should produce multiple correct translation variants. This is crucial as languages often have numerous grammatical forms and ways of conveying information. For instance, English has various verb forms, such as regular, irregular, and modal verbs, with potentially different translations in other languages. If an MT system generates only one translation variant, the NLU model might not learn to recognize others, compromising the model's quality. Hence, MT systems should create multiple accurate translation variants to cover all possible patterns, enhancing the performance of NLU models.

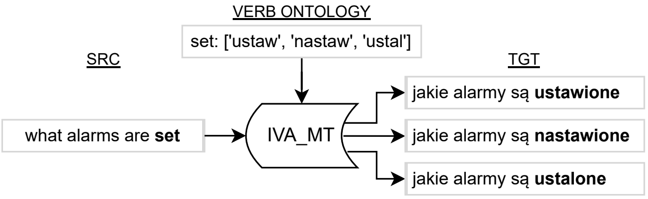

VERB ONTOLOGY
set: ['ustaw', 'nastaw', 'ustal']
SRC
TGT
what alarms are **set**
IVA_MT
jakie alarmy są **ustawione**
jakie alarmy są **nastawione**
jakie alarmy są **ustalone**

Fig. 1. Example of multiple variants translations based on verb ontology and constrained beam search.

Fig. 1 presents the system schema proposed in this article. Source utterance is translated to the target language with the help of verb ontology. Translations generated by the system are rich in terms of verb coverage and improve NLU model generalization capabilities.

In this work, to the best of our knowledge, we present the first analysis of language (verb analysis) used in available IVA corpora. The results of this analysis are used to construct verb ontology, based on VerbNet and WordNet, that is later used to generate multiple correct hypotheses in the MT system designed to translate training resources of multilingual NLU.

II. Related Work

At first glance, our work conceptually resembles early machine learning efforts to introduce linguistic knowledge into neural network models. Our goal is different, however, as we aim to use methods that utilize semantic information and linguistic knowledge in the context of machine translation to explain better and analyze its results. Our research focuses on explaining how the model works and how to improve its output.

This work relates to the methods of generating multiple correct translations. Fomicheva et al. [1] used MT model uncertainty to generate multiple diverse translations. In our

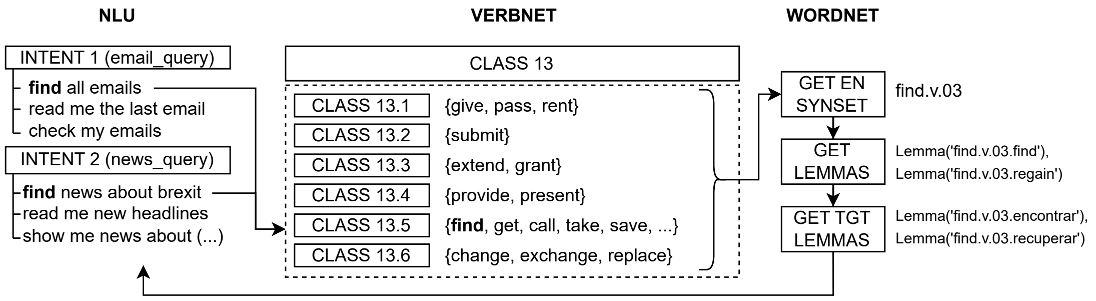


Fig. 2. Overview of the presented method. NLU verbs are matched to VerbNet, which consists of a WordNet synset from which a lemma in the target language can be extracted.

work, we used constrained beam search proposed by Anderson et al. [2] to generate multiple correct variants of translations.

Another area related to this work is using machine translation to translate training resources of NLU. Gaspers et al. [3] use MT to translate the training set of IVA and reported improvement in performance compared to grammar-based resources and in-house data collection methods. Abujabal et al. [4] used the MT model in conjunction with an NLU model trained for the source language to annotate unlabeled utterances reporting that 56% of the resulting automatically labeled utterances had a perfect match with ground-truth labels, and 90% reduction in manually labeled data.

We used VerbNet [5] and WordNet [6] to construct a dictionary to guide constrained beam search. WordNet is a linguistic resource that can be used to identify shallow semantic features that can be attached to lexical units. WordNet covers the vast majority of nouns, verbs, adjectives, and adverbs. It was initially developed for English, but more languages were recently added to the project Open Multilingual Wordnet. The words in WordNet are organized in synonym sets called synsets that share the same meaning. WerbNet is a verb lexicon with syntactic and semantic information based on Levin's verb classes. VerbNet is compatible with WordNet as verbs have links defined to WordNet synsets. VerbNet has been widely used in the context of NLU [7], [8].

Finally, this work relates to work that uses linguistic resources to improve the quality of NLU systems. Moneglia [9] created the ontology of action verbs to improve the performance of NLU and MT systems.

III. Method

In this work, we aim to build a multi-variant MT model that is guided by verb ontology, adapted to the IVA domain. Our secondary goal is that our ontology would be easy to edit, inspect and analyze by NLU developers. To do that, we extracted verbs from several VA corpora, matched them to their semantically equivalent class in VerbNet, and finally, using the link to WordNet, we extracted all their translations in the target language. In Fig. 2, we present steps of processing used to find verb equivalent in the target language to increase the variance of training resources. The proposed method consists of the following stages:

1) Creation of multilingual dictionary with verb translation for the IVA domain,
2) Creation of MT model (based on M2M100 architecture) from parallel corpora and creation of tools that guide decoding (constrained beam search) to generate multiple hypotheses,
3) Translation of NLU training resources, training of NLU model, and evaluation and analysis of the impact of MT on NLU quality.

A. Verb analysis of the NLU corpora

We start our investigation by analyzing verbs in NLU corpora. Verbs are carriers of key information about the event or action being described [10]. IVA commands semantics is composed of a verb and its parameters. In this work, we analyzed eight popular NLU corpora (listed in Table I) and extracted 374 English verbs. We then created a ranking list where the frequency of occurrences of verbs in all corpora is counted. The first verb on the list represents the most frequently used verb in all analyzed corpora.

In Table I, we present the top five positions on verb occurrence ranking. The highest-ranked verbs are: *set*, *show*, *remind*, *play* and *give*. Most analyzed NLU corpora consisted of calendar, alarm, and music domains which explain why given verbs are most popular.

While creating the ranking list, we noticed that each NLU corpus presents the same trend where the most frequent verbs can be found in around 20% of utterances. Fig. 3 illustrates that trend in IVA corpora follows the Zipf distribution. A similar trend can be found in other linguistic resources, for example, VerbNet [11].

B. Mapping IVA verbs to Levin classes and VerbNet

Most of the verbs we extracted from NLU corpora and analyzed are used in more than one domain. For example, a verb *set* can be used to set the alarm and the screen's brightness. For that reason, we decided to classify verbs of similar meaning. We used Levin verb classification [12] to investigate if IVA verbs are to be found there. In her work,

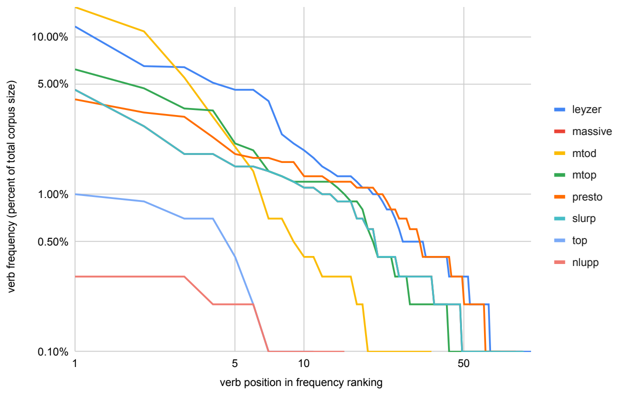


Fig. 3. Verb frequency and verb position on the ranking list for selected VA data sets presented in logarithmic scale.

Levin assigned 3,024 verbs to 48 broad and 192 fine-grained classes used in this article to find IVA verb frames.

Although verb classification can be automated [13], we found that research on language used in IVA is almost non-existent. Therefore, the automatic or semi-automatic methods will not perform well as they cannot be verified with certainty. For that reason, we decided to assign verbs to Levin classes manually. We first read each class description, including example verb frames, to decide if the same frame is used in the IVA context.

Out of 270 verbs, 14.88% could not be found in VerbNet or did not consist of WordNet class, making it impossible to use in our algorithm. 7.04% verbs matched more than one VerbNet class. 7.27% verbs belong to a VerbNet class where no other verb from NLU corpora belongs.

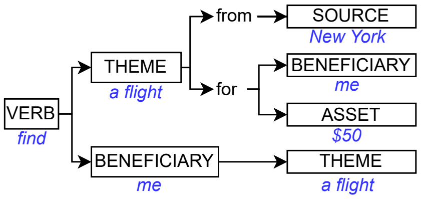


Fig. 4. Example of frames available in VerbNet for class 13 (Verbs of Change of Possession).

VerbNet defines semantic frames in which a given verb can be found. In the example presented in Fig. 4, we show four semantic frames belonging to class 13 where verb *find* appears. Verbs that belong to that class reflect the change of possession. From the frames presented in the example, we can construct several utterances belonging to the different IVA domains.

Below we present verbs found in NLU corpora that have been successfully matched to VerbNet classes. We can find other instances (verbs) of the same frame using those classes. We present the ten most frequent classes found in NLU corpora:

1) Class 13 (Verbs of Change of Possesion) where 10.73% of verbs belong with following sub-classes:
   a) 13.1 with *give*, *pass*, *rent*,
   b) 13.2 with *submit*,
   c) 13.3 with verbs such as *extend* and *grant* that relate to the change of possession that will take place in the future,
   d) 13.4 with *provide*, *present* that can be described as “X gives something to Y that Y needs or deserves”,
   e) 13.5 (Get and Obtain Verbs) with *find*, *get*, *call*, *take*, *save*, *order*, *keep*, *book*, *buy*, *select* and other,
   f) 13.6 with *change*, *exchange*, *replace* that relate to exchanging one thing for another,
2) Class 37 (Verbs of Communication) where 9.34% of verbs belong with the following sub-classes:
   a) 37.1 (Verbs of Transfer of Message) with *tell*, *read*, *write*, *ask*, *explain*, *dictate*, *summarize* that are verbs of type of communicated message,
   b) 37.2 with *remind*, *update*, *notify*, *inform*
   c) 37.3 with *call*, which is the verb of a manner of speaking and are distinguished from each other by how the sound is expressed. This is not a perfect match for IVA, but members are also not very far from IVA context,
   d) 37.4 with *email*, *phone*, *broadcast*, *ring* that relate to communication via these instruments of communication and are zero-related to the same noun,
   e) 37.5 with *speak*, *talk* that do not take sentential complement,
   f) 37.6 with *chat*
   g) 37.7 with *repeat*, *say*, *report*, *note*, *suggest*
   h) 37.8 with *complain* that specify the speaker’s attitude or feeling towards what is said,
   i) 37.9 with *alert*, *brief*
3) Class 26 (Verbs of Creation and Transformation) where 6.92% of the verbs belong. Members of that class are transitive verbs where one argument (agent) creates or transforms an entity,
4) Class 55 (Aspectual Verbs) where 5.19% of verbs belong. These verbs describe the initiation, termination, or continuation of an activity,
5) Class 45 (Verbs of Change of State) where 4.50% of the verbs belong. All of the verbs in this class relate to the change of state, with several sub-classes that define this state in more detail,
6) Class 9 (Verbs of Putting) where 4.15% of the verbs belong. These verbs refer to putting an entity at some location,
7) Class 29 (Verbs of Predicative Complements) where 4.15% of verbs belong,
8) Class 11 (Verbs of Sending and Carrying) where 3.81% of verbs belong,
9) Class 10 (Verbs of Removing) where 3.11% of verbs belong,
10) Class 51 (Verbs of Assuming Position) where 2.77% of verbs belong,
11) Remaining 30.45% consists of 38 verb classes.

TABLE I
TOP 5 ENGLISH VERBS FROM OCCURRENCE RANKING AND OCCURRENCE FREQUENCY IN EACH OF SELECTED NLU CORPORA.

Data set	***Set***	***Show***	***Remind***	***Play***	***Give***
Leyzer [14]	0.7%	11.6%	0.3%	1.1%	6.5%
MASSIVE [15]	1.8%	1.5%	1.3%	4.6%	1.1%
MTOD [16]	15.4%	3.1%	10.8%	0.0%	0.4%
MTOP [17]	6.2%	2.1%	4.7%	3.5%	1.2%
PRESTO [18]	0.4%	3.1%	0.2%	0.7%	0.3%
SLURP [19]	1.8%	1.5%	1.3%	4.6%	1.1%
TOP [20]	0.0%	0.7%	0.0%	0.0%	0.7%
NLU++ [21]	0.1%	0.2%	0.0%	0.0%	0.1%

TABLE II
AVERAGE NUMBER OF TARGET VERBS GENERATED IN VERB ONTOLOGY.

Language	**English Verbs**	**Avg. Num. of Target Verbs**
es-ES	185	3.51
fr-FR	200	5.09
it-IT	187	4.24
pl-PL	89	2.63
pt-PT	188	3.76
sv-SE	116	2.46

C. Mapping VerbNet to WordNet

VerbNet maps each verb to the corresponding synset in WordNet. We used NLTK implementation of VerbNet and WordNet to find target language synsets.

As a result of mapping VerbNet to WordNet, we created verb ontology[1] that is represented by a dictionary where the key is an English verb, and values are verb translations in the target language as presented in the below examples. In Table II, we present how many English verbs and, on average, how many target verbs were extracted for them. In the case of Polish ontology, only 89 English verbs were matched as Polish WordNet has a small subset of the entire WordNet mapped, and we had to perform mapping manually.

1) en-es: {*find*: [*encontrar*, *recuperar*, *conseguir*]}
2) en-es: {*find*: [*encontrar*, *recuperar*, *conseguir*]}
3) en-fr: {*find*: [*retrouver*, *trouver*, *analyser*]}
4) en-pl: {*find*: [*znajdź*, *poszukaj*, *odnajdź*]}
5) en-pt: {*find*: [*achar*, *encontrar*, *atingir*]}
6) en-sv: {*find*: [*upptäcka*, *hitta*, *finna*]

D. Constrained variant generation using verb ontology

Verb ontology guides MT to generate translation variants that consist of the target verb. We use constrained decoding implemented in the Transformers library to generate a translation consisting of a target verb (force word). We choose a beam size equal to 5, translations cannot consist of n-grams bigger than two more than once, and a single translation is generated for each constrained verb. All translations with more than two tokens bigger or smaller than the first-best are removed. If the input sentences consist of slot annotation, then we expect constrained examples also to have slot annotations.

Our translator (multiverb_iva_mt[2]) generate translations using following algorithm:

1) First translation is always a result of unconstrained translation (single-best),
2) For each target verb from verb ontology, we replace the verb from the single-best translation with the target verb,
3) Finally, we add variants generated by constrained beam search.

The final result is a list of translations that consist of at least one translation, but in the case when the input verb is found in verb ontology, typically, three variants are generated.

IV. EXPERIMENTS

To demonstrate the impact of the proposed method on translation quality, we designed experiments in which we compared the baseline model with two different translation methods: single-best and multi-verb. We use a model trained and evaluated on an untranslated subset of the Polish data set as a baseline. In the second step, we translated the English subset of the same data set to Polish. In a typical scenario, one Polish translation is generated for one input utterance (English). We call this single-best translation as the typical MT model returns the best translation candidate using the beam-search algorithm. In contrast, multi-verb translation generates multiple translation variants using constrained beam search guided by the proposed verb ontology.

A. Data

We used the second version (0.2.0) of the Leyzer[3] data set to conduct the experiments. Leyzer is a multilingual data set created to evaluate virtual assistants. It comprises 192 intents and 86 slots across three languages (English, Polish, and Spanish) and 21 IVA domains. The corpus primarily consists of imperative commands uttered to a device, with most languages and utterances using subject-initial word order (Subject-Verb-Object and Subject-Object-Verb). We selected Leyzer to conduct our experiments because each intent comprises several verb patterns and levels of naturalness. For example, *ChangeTemperature* intent, which represents the goal of changing the temperature of a home thermostat system, distinguishes three levels of naturalness, where the most natural way (level 0) of uttering this goal by the user would be to say *change temperature on my thermostat*, less natural (level 1) would be *set the temperature on my thermostat*, and finally least natural (level 2) yet still correct would be *modify the temperature on my thermostat*. These two pieces of information that are also available in the test set of the Leyzer corpus allow us to measure the impact of the multi-verb translation better.

The training subset of Polish corpora that we used to measure baseline results includes 15748 train utterances, 4695 development utterances, and 5839 test utterances. The English subset of corpora that we used to translate and report results

[1] https://github.com/cartesinus/multiverb_iva_mt/tree/main/data/verb_translations

[2] Code available at: https://github.com/cartesinus/multiverb_iva_mt

[3] Data set available at https://github.com/cartesinus/leyzer

of single-best and multi-verb includes 17289 training and validation utterances. All training utterances were translated with the third version of verb ontology (v.0.3.0) available in the proposed system. We extracted 3997 utterances from translated training set for validation, ensuring at least one sentence is available for every intent, level, and verb pattern.

B. Machine translation

We used the M2M100 model [22] as a base for our MT model. It provides an excellent base for future expansion, especially when considering low-resource languages, as it was trained to translate 100 languages. Moreover, this architecture is considered state-of-the-art, and most systems participating in WMT-22 implemented similar, Transformer architecture.

The foundation model was already pre-trained on the MT task; therefore, we performed light adaptation for ten epochs on the MASSIVE data set [15]. Adam [23] was used for optimization with an initial learning rate of $2e-5$. We used all data available in the training part of the corpus. Each epoch was evaluated on the validation subset. The batch size was 4, which is a relatively small value, but in our experiments on A100 GPU (40GB VRAM), it was impossible to set a larger batch size due to insufficient memory.

C. Natural language understanding

We used multilingual XLM-RoBERTa [24] models for intent classification (IC) and slot-filling (SF) and fine-tuned the models on the Leyzer data set. We chose this architecture for NLU as it can be easily compared to models presented in MASSIVE and achieves better results in a multilingual setting when compared to multilingual BERT (mBERT).

The foundation model was trained on 2.5TB of filtered CommonCrawl data containing 100 languages. During fine-tuning on the Leyzer data set, we used Adam [23] for optimization with an initial learning rate of $2e - 5$.

The quality of the IC model was evaluated using the accuracy metric that represents the number of utterances correctly classified to given intent. SF model was evaluated using a micro-averaged F1-score.

D. Impact of multi-verb translation on NLU

In Table III, we present the impact of multiple variant generation on IC and SF model results. Baseline models achieve results above 95% for both IC and SF, which means that test set annotations are consistent with a train set, and if good translated training data are present, also good results can be obtained.

The proposed improvement to the translation generation positively impacts IC model results. The accuracy of multi-verb translation is 3.8% relatively better than single-best translation. However, it is 7.95% relatively lower than the baseline model. As presented in Table IV, each English sentence generates an average of 1.74 Polish translations. In our opinion, this is the main reason why multi-verb translation generates a better training data set for the IC model. Leyzer test set evaluates multiple variants in which given intent can be uttered, including different levels of naturalness and verb patterns; therefore, more variant training set improves results. Further, IC results could be improved if more variants were created in verb ontology. Polish ontology (Table II) consists of 89 verbs, which is the smallest of all presented languages.

TABLE III

COMPARISON OF NLU INTENT ACCURACY AND SLOT F1-SCORE BETWEEN BASELINE, SINGLE-BEST TRANSLATION, AND MULTI-VERB TRANSLATION ON LEYZER DATA SET.

Model	**Intent Accuracy [%]**	**Slot F1 [%]**
Baseline	95.48	98.07
Single-best	83.73	88.21
Multi-verb	87.53	88.15

TABLE IV

AVERAGE NUMBER OF TRANSLATIONS GENERATED FOR A SINGLE ENGLISH INPUT PER LANGUAGE.

Target Language	**Avg. Num. Translations**
es-ES	1.73
fr-FR	2.63
pl-PL	1.74
pt-PT	1.91
sv-SE	1.46

Multi-verb translation does not improve the results of the SF model. Our method does not generate different variants of slot values; therefore, during training, the SF model cannot generalize to new test cases. The difference in F1-score between single-best and multi-variant is not statistically significant.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a method to create verb ontology for IVAs that can be used to generate multiple variants of translations. We tested our method on the NLU training set translation task, where we translated English corpora to Polish and trained NLU models from them. The results of our experiments show that verb ontology can significantly improve IC while maintaining SF results intact compared to single-best translation.

To the best of our knowledge, our MT models extended with verb ontology presented in this work are the first open-source models adapted to the domain of IVA that can return multi-variant translation. We released verb ontology, verb ranking list, and source code of IC and SF training codes to the research community. Data for the following five language pairs were published: English-Spanish[4], English-French[5], English-Polish[6], English-Portuguese[7], and English-Swedish[8]. In the future, we plan to extend our experiments to other languages.

REFERENCES

[1] M. Fomicheva, L. Specia, and F. Guzmán, "Multi-hypothesis machine translation evaluation," in *Proceedings of the 58th Annual Meeting*

[4] https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-es

[5] https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-fr

[6] https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-pl

[7] https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-pt

[8] https://huggingface.co/cartesinus/iva_mt_wslot-m2m100_418M-en-sv

of the Association for Computational Linguistics. Association for Computational Linguistics, 2020, pp. 1218–1232.

[2] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Guided open vocabulary image captioning with constrained beam search," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 936–945.

[3] J. Gaspers, P. Karanasou, and R. Chatterjee, "Selecting machine-translated data for quick bootstrapping of a natural language understanding system," in *North American Chapter of the Association for Computational Linguistics*, 2018.

[4] A. Abujabal, C. D. Bovi, S.-R. Ryu, T. Gojayev, F. Triefenbach, and Y. Versley, "Continuous model improvement for language understanding with machine translation," in *North American Chapter of the Association for Computational Linguistics*, 2021.

[5] K. K. Schuler, *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.

[6] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[7] K. Basu and G. Gupta, "Natural language question answering with goal-directed answer set programming." in *ICLP Workshops*, 2021.

[8] L. K. Schubert, "What kinds of knowledge are needed for genuine understanding," in *IJCAI 2015 Workshop on Cognitive Knowledge Acquisition and Applications (Cognitum 2015)*, 2015.

[9] M. Moneglia, "Natural language ontology of action: A gap with huge consequences for natural language understanding and machine translation," in *Language and Technology Conference*, 2011.

[10] O. Majewska and A. Korhonen, "Verb classification across languages," *Annual Review of Linguistics*, vol. 9, 2023.

[11] A. Huminski, F. Liausvia, and A. Goel, "Semantic roles in verbnet and framenet: Statistical analysis and evaluation," in *Computational Linguistics and Intelligent Text Processing: 20th International Conference, CICLing 2019, La Rochelle, France, April 7–13, 2019, Revised Selected Papers, Part II*. Springer, 2023, pp. 135–147.

[12] B. Levin, *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

[13] L. Sun, A. Korhonen, and Y. Krymolowski, "Verb class discovery from rich syntactic data," *Lecture Notes in Computer Science*, vol. 4919, p. 16, 2008.

[14] M. Sowański and A. Janicki, "Leyzer: A dataset for multilingual virtual assistants," in *Proc. Conference on Text, Speech, and Dialogue (TSD2020)*, P. Sojka, I. Kopeček, K. Pala, and A. Horák, Eds. Brno, Czechia: Springer International Publishing, 2020, pp. 477–486.

[15] J. FitzGerald, C. Hench, C. Peris, S. Mackie, K. Rottmann, A. Sanchez, A. Nash, L. Urbach, V. Kakarala, R. Singh *et al.*, "MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages," *arXiv preprint arXiv:2204.08582*, 2022.

[16] S. Schuster, S. Gupta, R. Shah, and M. Lewis, "Cross-lingual transfer learning for multilingual task oriented dialog," in *Proceedings of NAACL-HLT*, 2019, pp. 3795–3805.

[17] H. Li, A. Arora, S. Chen, A. Gupta, S. Gupta, and Y. Mehdad, "Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2950–2962.

[18] R. Goel, W. Ammar, A. Gupta, S. Vashishtha, M. Sano, F. Surani, M. Chang, H. Choe, D. Greene, K. He *et al.*, "Presto: A multilingual dataset for parsing realistic task-oriented dialogs," *arXiv preprint arXiv:2303.08954*, 2023.

[19] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A Spoken Language Understanding Resource Package," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[20] S. Gupta, R. Shah, M. Mohit, A. Kumar, and M. Lewis, "Semantic parsing for task oriented dialog using hierarchical representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2787–2792.

[21] I. Casanueva, I. Vulić, G. Spithourakis, and P. Budzianowski, "Nlu++: A multi-label, slot-rich, generalisable dataset for natural language understanding in task-oriented dialogue," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 1998–2013.

[22] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, E. Grave, M. Auli, and A. Joulin, "Beyond english-centric multilingual machine translation," 2020.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 6th International Conference on Learning Representations (ICRL 2015), San Diego, CA*, 2015.

[24] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.

DOI: 10.15439/2023F9640
ISSN 2300-5963 ACSIS, Vol. 35

Performance of Portable Sparse Matrix-Vector Product Implemented Using OpenACC

Kinga Stec, Przemysław Stpiczyński
0009-0008-6562-8954, 0000-0001-8661-414X
Maria Curie-Skłodowska University, Institute of Computer Science
Akademicka 9, 20-033 Lublin, Poland
Email: kingastec439@gmail.com, przemyslaw.stpiczynski@umcs.pl

***Abstract*—The aim of this paper is to study the performance of OpenACC implementations of sparse matrix-vector product for several storage formats: CSR, ELL, JAD, pJAD, and BSR, achieved on Intel CPU and NVIDIA GPU platforms to compare them with the performance of SpMV implementations using the BSR storage format provided by *Intel MKL* and *NVIDIA cuSPARSE* libraries. Numerical experiments show that vendor-provided BSR is the best format for CPUs but in the case of GPUs, the pJAD storage format allows to achieve better performance.**

I. Introduction

Sparse matrix-vector product (SpMV) is a central part of many numerical algorithms and its performance can have a very big impact on the performance of scientific and engineering applications [1], [2]. There are a lot of various sparse matrix storage formats and sophisticated techniques for developing efficient implementations of SpMV that utilize the underlying hardware of modern multicore CPUs and GPUs [3], [4], [5], [6], [7], [8], [9]. Unfortunately, these methods are rather complicated and usually depend on particular computer architecture, thus developing efficient and portable sparse matrix source code is still a challenge. However, the results presented in [10] and [11] show that simple SPARSKIT SpMV routines using various storage formats (CSR, ELL, JAD) [1] can be easily and efficiently adapted to modern CPU-based or GPU-accelerated architectures. Loops in source codes can be easily parallelized using OpenMP [12] or OpenACC [13], [14] directives, while the rest of the work can be done by a compiler. Such parallelized SpMV routines achieve performance comparable with the performance of the SpMV routines available in libraries optimized by hardware vendors (i.e. Intel MKL, NVIDIA cuSPARSE). OpenACC, a standard for accelerated computing, provides compiler directives for offloading C/C++ programs from host to attached accelerator devices. Such simple directives allow marking regions of source code for automatic acceleration in a portable vendor-independent manner. Moreover, OpenACC programs can be compiled using the `multicore` option, and then such programs can also be run on CPU-based architectures [15], [16], [17] without any changes in source codes.

Recently, the Block Compressed Row (BSR) format [18], [19], which is a generalization of the Compressed Sparse Row (CSR) format, has become very popular. Intel MKL and NVIDIA cuSPARSE provide optimized SpMV implementations for this format. Moreover, the other formats have been deprecated. Especially, BSR has replaced the HYB format in cuSPARSE. In this paper we compare the performance of portable OpenACC implementations of sparse matrix-vector product for CSR, ELL, JAD, pJAD, and BSR with the performance of SpMV implementations using the BSR storage format provided in *Intel MKL* and *NVIDIA cuSPARSE* libraries.

II. Sparse Matrix Representations

Let us assume that A is a sparse matrix with a significant number of zero entries, and $\mathbf{x}$, $\mathbf{y}$ are dense vectors. The SpMV operation is defined as follows:

$$\mathbf{y} \leftarrow A\mathbf{x}. \tag{1}$$

It is clear that if we do not multiply entries of $\mathbf{x}$ by zero entries of A, then (1) requires $2 \cdot n_{nz}$ floating point operations (one multiplication and one addition per nonzero entry of A). The structure of a sparse matrix can be characterized by n, n_{nz}, n_{nz}/n, and $\max_{nz}$, where n is the number of rows, n_{nz} is the total number of nonzero elements, n_{nz}/n the average number of nonzero elements per row, $\max_{nz}$ is the biggest number of nonzero elements per row. Table I shows values of these parameters for a set of test matrices, selected from *Matrix Market* [20] and *University of Florida Sparse Matrix Collection* [21]. It is clear that the performance of SpMV depends on the matrix storage format that utilizes the underlying hardware.

For description purposes of several possible sparse matrix storage formats, let us consider the following matrix as an example:

$$A = \begin{bmatrix} 7 & 0 & 1 & 0 \\ 0 & 4 & 2 & 3 \\ 1 & 8 & 0 & 0 \\ 0 & 9 & 0 & 0 \end{bmatrix}, \tag{2}$$

where $n = 4$, $n_{nz} = 8$, $n_{nz}/n = 2$, and $\max_{nz} = 3$. Now let us consider a few basic (ELL, JAD, CSR [1], [22]), as well as, more sophisticated (pJAD [11], BSR [18], [19]) storage formats for sparse matrices.

TABLE I: Set of test matrices [11]

Matrix	n	n_{nz}	n_{nz}/n	$\max_{nz}$
cry10000	10000	49699	5.0	5
possion3Da	13514	352762	26.1	110
af23560	23560	484256	20.6	21
g7jac140	41490	565956	13.6	153
fidapm37	9152	765944	83.7	255
bcsstk36	23052	1143140	49.6	178
majorbasis	160000	1750416	10.9	11
bbmat	38744	1771722	45.7	126
cfd1	70656	1828364	25.9	33
ASIC_680ks	682712	2329176	3.4	210
FEM_3D_thermal2	147900	3489300	23.6	27
parabolic_fem	525825	3674625	7.0	7
ecology2	999999	4995991	5.0	5
pre2	659033	5959282	9.0	628
boneS01	127224	6715152	52.8	81
torso1	116158	8516500	73.3	3263
thermal2	1228045	8580313	7.0	11
atmosmodl	1489752	10319760	6.9	7
bmw3_2	227362	11288630	49.7	336
af_shell8	504855	17588875	34.8	40
cage14	1505785	27130349	18.0	41
nd24k	72000	28715634	398.8	520
inline_1	503712	36816342	73.1	843
ldoor	952203	46522475	48.9	77
cage15	5154859	99199551	19.2	47

A. ELL

The ELL storage format was introduced in *Ellpack-Itpack* package. It assumes that a sparse matrix is represented by two arrays (Figure 1). Nonzero elements are stored in the first one called `a`. The second one called `ja` contains the corresponding column indices [23]. Both arrays are $n \times ncol$, where $ncol = \max_{nz}$. While ELL is simple and provides easy access to matrix entires, when $n_{nz}/n \ll \max_{nz}$, the number of stored zero entries of the matrix increases significantly.

`a:`

7	1	*
4	2	3
1	8	*
9	*	*

`ja:`

0	2	*
1	2	3
0	1	*
1	*	*

Fig. 1: ELL format for (2)

B. JAD

The JAD (i.e. *Jagged Diagonal*) format storage is represented by three arrays (Figure 2). It is similar to ELL, but removes the assumption on the fixed-length rows [22]. Firstly, a sparse matrix needs to be sorted in non-increasing order of the number of nonzeros per row

$$PA = \begin{bmatrix} 0 & 4 & 2 & 3 \\ 7 & 0 & 1 & 0 \\ 1 & 8 & 0 & 0 \\ 0 & 9 & 0 & 0 \end{bmatrix}.$$

The arrays `a` and `ja` of dimension nz contain nonzero elements (i.e. jagged diagonals) and the corresponding column indices. The array `ia` contains the beginning position of each jagged diagonal. Additionally, we can add array `rlen` which contains the number of nonzero elements in each row. Entries of this array can be calculated (in parallel) using the following formula. Let $jdiag$ be the number of jagged diagonals. Then for each row, $i = 0, \ldots, n-1$, we have

$$\mathbf{rlen}[i] = |\{j : 0 \leq j \leq jdiag - 1 \wedge \mathbf{ia}[j+1] - \mathbf{ia}[j] > i\}|.$$

Note that this format is devoid of the inconvenience associated with the need to store zero elements in rows completed to the width of $\max_{nz}$.

`a:`

4	2	3
7	1	
1	8	
9		

`ia:`

0
4
7
8

`rlen:`

3
2
2
1

`ja:`

1	2	3	0	2	0	1	2

Fig. 2: JAD format for (2)

C. pJAD

The pJAD storage format is an optimized version of JAD (Figure 3). This format assumes aligning (padding) columns of the arrays `a` and `ja` [11]. We add zero elements, thus the number of elements of each column should be a multiple of a given $bsize$ and rows of each block should have the same length. Entries of the array `brlen` contain widths of blocks of $bsize$ rows. Note that pJAD assumes to store at most $jdiag \cdot (bsize - 1)$ additional zero entries, where $jdiag$ is the number of jagged diagonals stored in `a`. Padding of jagged diagonals is important especially for GPUs. It allows *coalesced memory access* and reduces *thread divergence* within a block of threads [24].

`a:`

4	2	3
7	1	0
1	8	
9	0	

`ia:`

0
4
8
10

`brlen:`

3
2

`ja:`

1	2	3	0	2	0	1	*	2	*

Fig. 3: pJAD format for (2)

D. CSR

A sparse matrix in CSR (i.e. *Compressed Sparse Rows*) is stored in three arrays (Figure 4). The first array called `data` contains nonzero elements, and the second one called `cols` contains corresponding column indices of nonzero values. Indices of the beginning of rows in `data` array are stored at the `ptr` array.

`data:`	7	1	4	2	3	1	8	9
`cols:`	0	2	1	2	3	0	1	2
`ptr:`	0	2	5	7	8			

Fig. 4: CSR format for (2)

E. BSR

The BSR storage format can be treated as a generalization of CSR. A sparse matrix is represented by four arrays (Figure 5). Array `vals` contains column ordered values from blocks with nonzero values. Array `cols` stored columns indices of the first element per block. The `ptrB` and `ptrE` arrays contain the indices of the beginning and ending positions of the elements in the block row respectively.

`vals:`	7	0	0	4	1	2	0	3	1	0	8	9
`cols:`	0	1	0									
`ptrB:`	0	2										
`ptrE:`	2	3										

Fig. 5: BSR format for (2)

III. Algorithms

The SpMV operation for all storage formats presented in Section II can be implemented using OpenACC to be executed on both GPU-accelerated and CPU-based systems. OpenACC offers compiler directives for offloading selected computations from host to attached accelerator devices. It allows to indicate regions of source code for automatic parallelization in a portable manner. Algorithms 1, 2, 3, 4, and 5 show how to implement SpMV in C/C++ using OpenACC for all considered formats: ELL, JAD, pJAD, CSR, and BSR formats, respectively. OpenACC-specific parts of the implementation start with `#pragma acc` directives. The `parallel loop` directive defines a loop to be accelerated on GPU. Additional clauses, namely `gang` and `vector_length` tell that `gangs` (i.e. blocks of threads) should perform an iteration of loops. Threads within gangs work in vector or SIMD mode [13]. The `loop seq` construct placed before a loop within `parallel loop` says that such a loop should be executed sequentially by a single thread. The `present` clause says that indicated variables are previously allocated on GPU. It allows to avoid unnecessary data movements between host and device memory systems. OpenACC provides the `data` construct that can be used to specify such scope of data in accelerated regions. Data transfers can also be initialized using the `enter data` and `exit data` constructs [13]. Figure 6 shows output messages generated by the compiler using the `-acc=gpu` option.

Algorithm 1 SpMV using ELL in OpenACC

```
// auxiliary routine
double count_per_row(double *a, double *x, int *ja,
     int n,  int ncol, int i){
  double t = 0;
  #pragma acc loop seq
  for( int j = 0; j < ncol; ++j ) {
      t += a[j*n+i] * x[ja[j*n+i]];
  }
  return t;
}

// driver routine
void ELL_SpMV(int n, double *x, double *y, int ncol,
     double *a, int *ja){
  #pragma acc parallel loop gang vector_length(128)\
       present(y,x,a,ja)
  for(int i=0; i<n; i++) {
    y[i] = count_per_row(a, x, ja, n, ncol, i);
  }
}
```

Algorithm 2 SpMV using JAD in OpenACC

```
// auxiliary routine
double count_per_row(int rlen, int *ia, int i,
     double *a, double *x, int *ja){
  double t = 0;
  #pragma acc loop seq
  for(int j = 0; j<rlen; j++){
    int k = ia[j]+i;
    t+=a[k]*x[ja[k]];
  }
  return t;
}

// driver routine
void JAD_SpMV(int n, int *perm, double *a, int *rlen
     , int *ia, int *ja, double *x, double *y){
  #pragma acc parallel loop gang vector_length(128)\
       present(perm,a,rlen, ia, ja, x, y)
  for(int i = 0; i<n; i++){
    y[perm[i]] = count_per_row(rlen[i], ia, i, a, x,
         ja);
  }
}
```

When OpenACC programs are compiled using the `-acc=multicore` option, the compiler generates appropriate parallel regions to be executed in parallel on CPU cores (Figure 7). It should be noticed that if we omit OpenACC directives, we will get sequential implementations of SpMV.

IV. Performance of SpMV

All OpenACC implementations of SpMV have been tested on the computer equipped with two Xeon Gold 6342 @ 2.80GHz (48 cores) and NVIDIA A40 GPU (10752 cores, FP64 Peak perf. 584.6 GFLOPS), running under Linux Oper-

Algorithm 3 SpMV using pJAD in OpenACC

```
// auxiliary routine
double count_per_row(double *a, double *x, int *ia,
    int *ja, int brlen, int bsize, int i){
  double t = 0;
  #pragma acc loop seq
  for( int j = 0; j < brlen; ++j ) {
    int k = ia[j]+i;
    t += a[k]* x[ja[k]];
  }
  return t;
}

// driver routine
void pJAD_SpMV(int n_block, double *x, double *y,
    double *a, int *ja, int *ia, int *brlen, int *
    iperm, int bsize){
  #pragma acc parallel loop gang vector_length(128)\
       present(y,x,a,ja,ia,brlen,iperm)
  for(int i=0; i<n_block; i++) {
    #pragma acc loop
    for (int j=0; j<bsize; j++){
      y[iperm[i*bsize+j]] = count_per_row(a, x, ia,
          ja, brlen[i], bsize, i*bsize+j);
    }
  }
}
```

Algorithm 4 SpMV using CSR in OpenACC

```
// auxiliary routine
double count_per_row(int nz_in_row, int idx_start,
    int *cols, double *data, double *x){
  double t = 0;
  #pragma acc loop seq
  for(int j = 0; j<nz_in_row; j++){
     t+=x[cols[idx_start+j]]*data[idx_start+j];
  }
  return t;
}

// driver routine
void CSR_SpMV(int n, double *data, int *cols, int *
    ptr, double *x, double *y){
  #pragma acc parallel loop gang vector_length(128)\
       present(ptr,cols,x,y,data)
  for(int i = 0; i<n; i++){
    y[i] = count_per_row(ptr[i+1]-ptr[i],ptr[i],cols
        ,data,x);
  }
}
```

```
count_per_row:
  4, Generating implicit acc routine seq
     Generating acc routine seq
     Generating NVIDIA GPU code
ELL_SpMV:
 13, Generating present(y[:],x[:],ja[:],a[:])
     Generating NVIDIA GPU code
     15, #pragma acc loop gang, vector(128)
                 /* blockIdx.x threadIdx.x */
```

Fig. 6: Compiler output messages for Algorithm 1 compiled using -acc=gpu

Algorithm 5 SpMV using BSR in OpenACC

```
// auxiliary routine
void count_per_block(int block_size, int rows_begin,
    int rows_end,int *cols, double *vals ,double *x,
    double *y, int i){
  #pragma acc loop seq
  for(int j=rows_begin;j<rows_end;j++){
    int base=j*block_size*block_size;
    for(int jdx=cols[j]*block_size;jdx<(cols[j]+1)*
        block_size;jdx++){
          for(int idx=i*block_size; idx<(i+1)*
              block_size; idx++){
                y[idx]+=vals[base]*x[jdx];
                base++;
          }
    }
  }
}

// driver routine
void  BSR_SpMV(int rows,int block_size, int *ptrB,
    int *ptrE, int *cols, double *vals, double *x,
    double *y){
  #pragma acc parallel loop gang vector_length(128)\
       present(x,y,vals,ptrB,ptrE,cols)
  for(int i=0;i<rows;i++){
    count_per_block(block_size, ptrB[i], ptrE[i],
        cols, vals, x, y, i);
  }
}
```

```
ELL_SpMV:
     13, Generating Multicore code
         15, #pragma acc loop gang
```

Fig. 7: Compiler output messages of Algorithm 1 compiled using -acc=multicore

ating System with Intel OneAPI and NVIDIA HPC compiler suits. The results have been compared with SpMV implementations using the BSR storage format that are provided in *Intel MKL* and *NVIDIA cuSPARSE* libraries. Table II shows the performance (GFLOPS) obtained for all considered implementations for both CPUs and GPU and the set of sparse matrices from Table I calculated as follows:

$$perf = \frac{2 \cdot n_{nz}}{t \cdot 10^9} \text{ GFLOPS}, \tag{3}$$

where t is the execution time of SpMV (in seconds). It should be noticed that in the case of BSR, the table shows the best performance achieved for the optimal block size determined empirically. All experiments have been performed for FP64.

V. Results of Experiments

On CPU, the best performance for the majority of matrices is obtained for *Intel MKL* BSR implementation. For the smaller matrices, the best results are achieved by OpenACC implementation of SpMV using the CSR format. Other OpenACC implementations achieve worse performance than *Intel MKL* BSR. Especially, OpenACC BSR is much slower than its well-optimized counterpart. In most cases pJAD achieves better performance than JAD, however its performance is

TABLE II: SpMV performance results (GFLOPS)

Matrix	**OpenACC CPU**					**MKL**	**OpenACC GPU**					**cuSPARSE**
	CSR	**ELL**	**JAD**	**pJAD**	**BSR**	**BSR**	**CSR**	**ELL**	**JAD**	**pJAD**	**BSR**	**BSR**
cry10000	**2.56**	1.07	1.16	1.02	1.56	0.71	5.68	**6.14**	5.21	5.28	4.29	1.06
poisson3Da	**7.57**	1.43	6.85	6.37	3.62	3.21	**18.21**	12.47	14.95	16.60	3.28	3.82
af23560	**7.78**	7.59	5.13	4.76	1.12	4.68	26.74	**32.04**	29.02	29.76	22.45	9.45
g7jac140	**8.47**	0.97	2.65	3.11	0.75	3.99	**24.47**	6.62	18.32	20.85	6.02	6.22
fidapm37	**14.36**	3.31	7.27	6.91	8.63	7.04	**26.75**	20.76	18.46	21.85	5.82	10.44
bcsstk36	**12.18**	2.40	6.46	7.47	2.41	10.23	**33.30**	19.44	29.40	32.67	18.21	17.65
majorbasis	**12.51**	11.05	8.38	9.35	0.85	11.09	41.19	51.85	50.34	**51.89**	10.60	14.82
bbmat	**13.41**	4.14	5.59	6.09	2.49	12.58	40.97	25.93	47.80	**52.15**	22.71	22.49
cfd1	**14.22**	8.38	8.50	8.89	1.55	11.71	32.97	46.88	53.83	**55.38**	16.61	15.76
ASIC_680ks	4.25	0.21	2.07	2.20	0.52	**4.88**	**37.88**	1.42	23.69	25.91	9.40	12.53
FEM_3D_thermal2	14.27	9.76	6.81	9.17	1.49	**16.54**	37.78	56.75	60.18	**64.30**	14.63	23.79
parabolic_fem	**6.15**	5.73	5.45	5.68	0.82	5.76	44.07	50.83	48.88	**51.46**	4.13	15.39
ecology2	5.86	6.48	5.43	5.67	0.92	**7.61**	59.81	**63.25**	56.38	60.11	8.59	29.02
pre2	6.99	0.13	2.34	3.14	1.24	**7.35**	**45.28**	1.18	26.94	28.47	6.16	16.49
boneS01	14.03	7.98	5.84	8.91	3.97	**24.22**	38.42	50.39	72.20	**74.71**	22.06	41.06
torso1	12.07	0.27	0.71	0.52	4.23	**20.15**	**31.94**	1.82	21.03	23.66	6.31	30.01
thermal2	6.19	5.36	4.54	5.43	1.43	**7.17**	44.80	**46.54**	26.04	27.73	4.17	22.74
atmosmodl	7.73	8.11	6.88	6.93	1.69	**10.38**	61.82	**72.87**	58.44	62.32	7.00	32.63
bmw3_2	10.83	1.53	5.14	9.73	3.54	**27.16**	36.42	12.38	63.03	**68.40**	15.74	47.84
af_shell8	4.24	4.98	4.86	6.21	3.30	**21.26**	38.86	71.83	74.69	**79.50**	20.64	64.34
cage14	6.29	2.83	4.12	3.68	3.53	**11.17**	43.37	34.83	45.00	**50.33**	5.18	29.19
nd24k	10.64	7.00	6.31	5.38	11.38	**28.06**	23.83	68.88	88.65	**90.49**	30.91	78.49
inline_1	9.95	0.72	6.08	5.72	5.35	**20.26**	33.21	8.17	57.97	**64.76**	9.82	41.29
ldoor	9.42	4.23	5.25	6.22	7.14	**25.91**	39.40	51.67	51.54	58.89	16.55	**67.83**
cage15	10.50	3.68	5.60	6.40	4.57	**12.72**	**41.34**	31.92	34.23	37.62	4.70	27.75

still worse than the optimized BSR provided by Intel. The ELL format gives the worse performance for matrices with $n_{nz}/n \ll \max_{nz}$, when the number of stored zero entries increases significantly.

On GPU, we can observe that pJAD implementation achieves the best results for the largest number of matrices (eleven matrices). It outperforms JAD format for all matrices and *Nvidia cuSPARSE* BSR for almost all matrices. Moreover, for several matrices pJAD outperforms *Intel MKL* BSR significantly. The second best format is CSR. It gains the best results for seven matrices. For the others, the pJAD format is always better and the JAD format is almost always better. The ELL format achieves best results for five matrices (*cry10000, af23560, ecology2, thermal2, atmosmodl*), most of which have almost the same row length ($n_{nz}/n \approx \max_{nz}$). *Nvidia cuSPARSE* BSR gains the highest performance for only one matrix (i.e. *ldoor*), however for larger matrices it is much faster than OpenACC BSR. As with CPU, the difference between OpenACC and *Nvidia cuSPARSE* implementations of BSR is significant.

VI. Conclusions and Future Work

We have shown that sparse matrix-vector product using several formats can easily be implemented using OpenACC in order to utilize underlying hardware of modern CPUs and GPUs. Our implementations achieve reasonable performance on GPU and CPU, in some cases comparable with the performance of vendor optimized implementations using the BSR format, and sometimes even better.

It seems that the use of pJAD is very promising for GPUs. Its OpenACC portable implementation achieves much better performance than BSR optimized by the vendor. In the future we plan to provide non-portable optimized version of SpMV using pJAD.

References

[1] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[2] R. Helfenstein and J. Koko, "Parallel preconditioned conjugate gradient algorithm on GPU," *J. Computational Applied Mathematics*, vol. 236, no. 15, pp. 3584–3590, 2012. doi: 10.1016/j.cam.2011.04.025

[3] X. Feng, H. Jin, R. Zheng, Z. Shao, and L. Zhu, "A segment-based sparse matrix-vector multiplication on CUDA," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 1, pp. 271–286, 2014. doi: 10.1002/cpe.2978

[4] J. C. Pichel, J. A. Lorenzo, F. F. Rivera, D. B. Heras, and T. F. Pena, "Using sampled information: is it enough for the sparse matrix-vector product locality optimization?" *Concurrency and Computation: Practice and Experience*, vol. 26, no. 1, pp. 98–117, 2014. doi: 10.1002/cpe.2949

[5] F. Vázquez, G. O. López, J. Fernández, and E. M. Garzón, "Improving the performance of the sparse matrix vector product with GPUs," in *10th IEEE International Conference on Computer and Information Technology, CIT 2010, Bradford, West Yorkshire, UK, June 29-July 1, 2010*, 2010. doi: 10.1109/CIT.2010.208 pp. 1146–1151.

[6] S. Williams, L. Oliker, R. W. Vuduc, J. Shalf, K. A. Yelick, and J. Demmel, "Optimization of sparse matrix-vector multiplication on emerging multicore platforms," *Parallel Computing*, vol. 35, no. 3, pp. 178–194, 2009. doi: 10.1016/j.parco.2008.12.006

[7] K. K. Matam and K. Kothapalli, "Accelerating sparse matrix vector multiplication in iterative methods using GPU," in *International Conference on Parallel Processing, ICPP 2011, Taipei, Taiwan, September 13-16, 2011*, 2011. doi: 10.1109/ICPP.2011.82 pp. 612–621.

[8] C. Stylianou and M. Weiland, "Exploiting dynamic sparse matrices for performance portable linear algebra operations," in *IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC, P3HPC@SC 2022, Dallas, TX, USA, November 13-18, 2022*. IEEE, 2022. doi: 10.1109/P3HPC56579.2022.00010 pp. 47–57.

[9] B. Yilmaz, "A novel graph transformation strategy for optimizing SpTRSV on CPUs," *Concurrency and Computation Practice and Experience*, 2023. doi: 10.1002/cpe.7761

[10] B. Bylina, J. Bylina, P. Stpiczyński, and D. Szałkowski, "Performance analysis of multicore and multinodal implementation of SpMV operation," in *Proceedings of the Federated Conference on Computer Science and Information Systems, September 7-10, 2014, Warsaw, Poland*. IEEE, 2014. doi: 10.15439/2014F313 pp. 575–582.

[11] P. Stpiczyński, "Semiautomatic acceleration of sparse matrix-vector product using OpenACC," in *Parallel Processing and Applied Mathematics, 11th International Conference, PPAM 2015, Kraków, Poland, September 6-9, 2015, Revised Selected Papers, Part II*, ser. Lecture Notes in Computer Science, vol. 9574. Springer, 2016. doi: 10.1007/978-3-319-32152-3_14 pp. 143–152.

[12] R. van der Pas, E. Stotzer, and C. Terboven, *Using OpenMP – The Next Step. Affinity, Accelerators, Tasking, and SIMD*. Cambridge MA: MIT Press, 2017.

[13] S. Chandrasekaran and G. Juckeland, Eds., *OpenACC for Programmers: Concepts and Strategies*. Addison-Wesley, 2018.

[14] R. Farber, Ed., *Parallel Programming with OpenACC*. Morgan Kaufmann, 2017.

[15] H. J. Eberl and R. Sudarsan, "OpenACC parallelisation for diffusion problems, applied to temperature distribution on a honeycomb around the bee brood: A worked example using BiCGSTAB," in *Parallel Processing and Applied Mathematics - 10th International Conference, PPAM 2013, Warsaw, Poland, September 8-11, 2013, Revised Selected Papers, Part II*, 2013. doi: 10.1007/978-3-642-55195-6_29 pp. 311–321.

[16] P. Stpiczyński, "Algorithmic and language-based optimization of Marsa-LFIB4 pseudorandom number generator using OpenMP, OpenACC and CUDA," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 238–245, 2020. doi: 10.1016/j.jpdc.2019.12.004

[17] B. Dmitruk and P. Stpiczyński, "Solving tridiagonal Toeplitz systems of linear equations on GPU-accelerated computers," *Concurrency and Computation Practice and Experience*, vol. 34, p. 6449, 2022. doi: 10.1002/cpe.6449

[18] R. W. Vuduc and H. J. Moon, "Fast sparse matrix-vector multiplication by exploiting variable block structure," in *High Performance Computing and Communications, First International Conference, HPCC 2005, Sorrento, Italy, September 21-23, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3726. Springer, 2005. doi: 10.1007/11557654_91 pp. 807–816.

[19] R. Shahnaz and A. Usman, "Blocked-based sparse matrix-vector multiplication on distributed memory parallel computers," *Int. Arab J. Inf. Technol.*, vol. 8, no. 2, pp. 130–136, 2011.

[20] R. F. Boisvert, R. Pozo, K. A. Remington, R. F. Barrett, and J. Dongarra, "Matrix Market: a web resource for test matrix collections," in *Quality of Numerical Software - Assessment and Enhancement, Proceedings of the IFIP TC2/WG2.5 Working Conference on the Quality of Numerical Software, Assessment and Enhancement, Oxford, UK, 8-12 July 1996*, ser. IFIP Conference Proceedings, R. F. Boisvert, Ed., vol. 76. Chapman & Hall, 1997, pp. 125–137.

[21] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, pp. 1–25, 2011. doi: 10.1145/2049662.2049663

[22] R. Li and Y. Saad, "GPU-accelerated preconditioned iterative linear solvers," *The Journal of Supercomputing*, vol. 63, no. 2, pp. 443–466, 2013. doi: 10.1007/s11227-012-0825-3

[23] R. Grimes, D. Kincaid, and D. Young, "ITPACK 2.0 user's guide," Center for Numerical Analysis, University of Texas, Tech. Rep. CNA-150, 1979.

[24] J. Cheng, M. Grossman, and T. McKercher, Eds., *Professional CUDA C Programming*. Wiley and Sons, 2014.

DOI: 10.15439/2023F2738
ISSN 2300-5963 ACSIS, Vol. 35

Using graph solutions to identify "troll farms" and fake news propagation channels

Patryk Sulej
Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
Email: patryksulej2@gmail.com

Krzysztof Hryniów
0000-0001-8044-3925
Institute of Control and Industrial Electronics,
Warsaw University of Technology,
ul. Koszykowa 75, 00-662 Warsaw, Poland
Email: krzysztof.hryniow@pw.edu.pl

***Abstract*—This paper addresses the issue of fake news detection, with a particular focus on solutions derived from graph theory. It covers identifying channels, which are sources of fake news, and identifying users spreading false information, considering users deliberately misleading their audience, forming clusters called ’troll farms’. It proposes a solution using graph theory, which includes classifying users based on the social context extracted in graph centrality measures built from user interactions or networks built from followers on the social network Twitter. The solution includes not only the identification of trolls but also potential unintentional users spreading false information, users exposed to false information, or automated scripts spreading information (bots). Thorough research on the efficiency of different features and classifiers is conducted on MIB and FakeNewsNet datasets. Conducted research confirms general conclusions from previous studies and offers some improvements.**

I. Introduction

WITH the rapid increase in accessibility to information caused by the development of the Internet, it has become much easier to manipulate and spread false information to any audience. Social media platforms have changed the way journalism can be conducted in the 21st century, causing anyone to be able to report on events to the masses. What is most commonly considered fake news is, in a broad sense, information that is not true, or in a more specific purpose, information that has been made available to mislead the recipient [1].

Information portals or social media enable targeting any audience, which, if used appropriately, can influence public sentiment and impact countries’ internal politics. It severely threatens a nation’s and its citizens’ stability and internal security. Examples of such events include the 2016 US presidential election campaigns, during which 20 of the most popular manipulated posts generated more shares and comments than 19 of the most prominent news sites [2]. ’Trolling’ can be defined as deviant, malicious, anti-social behavior aimed at destroying a conversation or creating conflict. The key features of this activity are deception, aggression, and negative disruptive actions, and the measure of success is to gain as much audience attention as possible [3].

An example of how vital the information domain is can be seen in the actions taken by Russia and Ukraine during the Russian-Ukrainian war that began on 24 February 2022. Building public support for an invasion of a neighboring country using manipulative techniques and a wide range of information channels preceded the Russian Federation’s attack on Ukraine [4]. This action also targeted the rest of the world – using messengers such as Telegram to release posts or videos distorting the picture of reality to present the Russian view of the conflict and gain support for its actions. While most Western countries did not succumb to disinformation, the manipulation work carried out domestically in the Russian Federation served its purpose and convinced most Russians that the war was necessary and consolidated citizens around the authorities.

II. Methods of fake news detection

Methods for detecting false information are classified as content-oriented, social context-oriented, and graph-based [5].

A. Content-oriented methods

Methods that use fact-checking, i.e., comparing the thesis presented in the news with external sources, are called knowledge-oriented methods. Manual fact-checking is poorly scalable and manpower-intensive. However, it allows for creating valuable datasets for developing automated solutions such as FakeNewsNet [6]. Fact-checking using ’crowd-sourcing’ has a high risk of obtaining biased results, but it is better scalable than the expert method [1].

Style-oriented methods are similar to knowledge-oriented methods, but in this case, the aim is not to assess the content’s veracity but to extract the author’s intentions and determine whether it was to mislead the audience [1].

Content-oriented methods also include linguistic analysis of the text [7]. It is based on analyzing the syntax and semantics of a sentence by extracting features that distinguish false information from accurate information, such as length of statements, word embedding, lexical context, discourse level, etc. [5]. This solution works in the case of longer forms of expression, but in social media, extracting these features proves difficult, or there are too few to determine the veracity of such information.

B. Social context-oriented methods

One method used is to analyze the life of information on the web. It allows one to observe how it evolves with each sharing and how information changes to form a 'rumor.' Analyzing the life cycle of such information over a period of time allows us to understand the diffusion patterns of rumors over time. Another way is to assess the information's veracity by analyzing the source's credibility [1][5]. The third popular solution for identifying fake news is to analyze the networks they form with other information, like social networks, friends, post sharing, interactions with other profiles, and profile data [1]. It allows for identifying the relationships between people spreading such information and extracting the characteristics of such interactions or profiles. An important aspect is the propagation pattern of such information, which differs between false and authentic information.

C. Graph-based solutions

Network and graph analysis is mainly based on studying features challenging to describe by standard data-averaging methods. In the case of graphs, there are often power relationships due to the uneven distribution of nodes or the high degree of links between data. Graph solutions allow the study of features such as the propagation speed of objects in the network, the relevance of individual nodes, or the way objects interact within the network and whether this can change.

Graph-based methods are used extensively in deep learning to detect internet trolls, fake news propagation channels, or fake news in general. Graph neural networks (GNNs) are characterized by the fact that they can encode the graph structure as well as the node features at the same time, which in the case of social networks or news propagation networks, dramatically increases the efficiency of classification [8][9].

To verify fake news, automatic fact-checking methods are often used, which consist of extracting facts from the content of the news and then comparing this fact with a knowledge base, the form of which can be a knowledge graph [1].

III. Dataset preparation and preprocessing

This paper decided to use graph centrality measures, which have served as features for machine learning algorithms. These measures were chosen because this area has yet to be fully explored despite some work on the subject.

Identifying 'fake' users based on a follower network is a method of detecting fake news based on social context. The source of false information can be identified in this way. When a new user arrives and 'adds' other users to his/her social network, there is a chance to identify whether he or she is an account that will spread false information. It creates a significant advantage because we can already take action, then – observe the user and start analyzing their content with other solutions to detect false information. References [10] and [11] examined follower networks and followed accounts using graph centrality measures. In addition, it was possible to classify online trolls from the 2016 US presidential campaign by creating a network of users who retweeted their posts [12]. Using graph algorithms, identifying Russian troll accounts extracted from a list provided by the US House Intelligence Committee from the 2016 US election was also feasible [13].

As part of this work, it was decided to use the centrality measures used in previous works, such as: centrality of agency, centrality of proximity (unnamed type), centrality of node degree (degree, in-degree, out-degree), PageRank centrality, centrality of eigenvector. In addition to this, the measures examined were: centrality of proximity (Wasserman-Faust), the centrality of harmonic closeness (harmonic closeness), ArticleRank, HITS (Hyperlink-Induced Topic Search).

A. Tools

Of the available tools for operating on graphs, it was decided to use Neo4j in the study because of the numerous previous uses of this tool for analyzing fake news and troll accounts [2][11][13]. This database is also fully adapted to operate on graphs. The research was performed on a computer with an Intel Core i7 7700HQ processor with 16GB RAM DDR5 and Google Colab. All collections were placed in the Neo4j database version 5.1.0. Additional libraries were used: APOC version 5.1.0 and Graph Data Science Library 2.2.5.

B. Datasets

The datasets used were those collected for the study of fake Twitter accounts [14]. This MIB dataset consists of five subsets: two sets of accounts run by humans (TFP and E13) and three sets of accounts with fake followers (INT, FSF, TWT). The data was collected before 2015. For machine learning, the collection was filtered, removing profiles with less than two edges due to their large number – they were considered noise. However, the complete set was used for feature extraction to capture the centrality features of all nodes as accurately as possible. In addition to the MIB collection, users extracted from the FakeNewsNet [6] collection were also used. This collection was created in 2018 and consisted of tweets spreading fake and real news, their retweets, the profiles of the users who sent them, and tweets from the users' timelines. The collection is based on manual fact-checking performed by the portals Gossipcop and Politifact.

Due to the known problems with the collection download and the Twitter limits [6][12][14][15], it was eventually possible to obtain 6 240 964 unique identifiers of users. Based on whether a profile was among the followers, or followers of an account that spread real or fake news, a label of true or false was assigned to that profile. Thus, profiles potentially at risk of seeing fake news were labeled as if they were spreading fake news. After filtering out the noise in the form of profiles that contained one or fewer relations and were irrelevant to the graph, 2 713 356 profiles were obtained. To speed up the Neo4j database feature extraction algorithms, once the collection was imported, the Random Walk with Restart algorithm was used to sample the collection at a ratio of 0.3. This algorithm preserves the structural features of the graph, which, in the case of centrality testing, is crucial for obtaining results close to the truth. Unfortunately, this procedure nevertheless introduced additional uncertainty into the study. The final result was 541 255 nodes labeled as potentially false and 272 743 as potentially genuine.

TABLE I: Size of follower datasets.

Dataset name	Number of profiles	Final number of profiles	Genuine/Fake accounts
TFP	240 961	198 621	Genuine
E13	996 438	147 955	Genuine
TWT	77 685	24 436	Fake
INT	57 266	17 578	Fake
FSF	20 173	5 914	Fake
FakeNewsNet #1	6 240 964	541 255	Fake
FakeNewsNet #2	-	272 743	Genuine
Sum	7 633 487	1 208 502	-

TABLE II: Size of user interaction sets from the FakeNewsNet skeleton and US Elections Trolls.

Dataset name	graphs (with fake news propagation)	profiles retweeting genuine news	profiles retweeting fake news
Politifact	314 (157)	18 042	23 012
Gossipcop	5 464 (2 732)	208 079	106 183
USElectionsTrolls	269 (269)	0	413
Sum	6 047 (3 158)	226 121	129 195

As the access to information about real troll accounts via the Twitter API was prevented, and the data contained in the Neo4j Sandbox about these accounts were small, another collection [1] from the GNN Fake News survey was used [8]. That survey used the FakeNewsNet dataset and provided the collection as a finished graph – the relationship between individual users who retweeted another user's post. The collection in this form contains much less memory because the original tweet identifiers have been mapped to unique numerical values starting from 0, and it does not contain additional information related to the user profile – it is a kind of skeleton. This processed collection yielded user profiles, with interaction in the form of retweeting a post. The original collection contained 425 842 profiles, but due to accounts being blocked, deleted, or unavailable, the authors obtained only 355 316. Tweet collection is a separate part of the MIB collection. It was created for the paper [15] on the study of spambots.

C. *Selection of characteristics of user interaction and followers sets*

To select features for machine learning algorithms, the following dependency measures were used: Pearson correlation, F-test, analysis of variance (F classifier), Mutual information, chi2 (chi-square test), tree classifier [16][17].

The feature selection analysis was started by determining the Pearson correlation matrix, identifying linearly dependent features, and then sifting them out. The selection was carried out on the complete set, with the awareness that some algorithms will show a linear relationship because they are similar in implementation – for example, closeness and harmonic closeness, or PageRank and ArticleRank.

The significance level of $\alpha = 0.1$ was assumed to reject the null hypothesis. Thus, for p > 0.1, we cannot reject the null hypothesis that the variables are independent. The choice to leave one of the two features was made when the linear Pearson correlation coefficient between the features exceeded the value of 0.3.

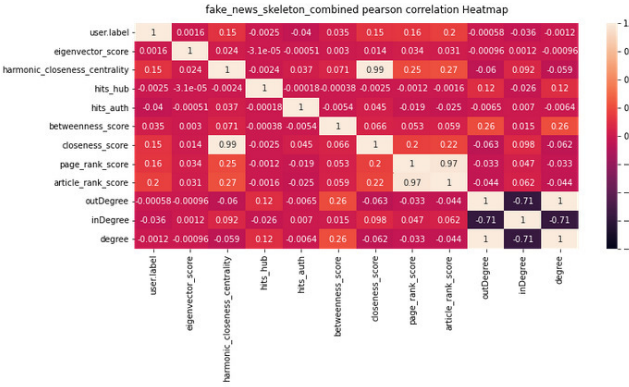


Fig. 1: Pearson correlation matrix of features extracted using graph algorithms for the FakeNewsNet skeleton - combined Politifact and Gossipcop sets.

Pearson correlation matrices were prepared for different cases: separate Politifact user interaction set, separate Gossipcop user interaction set, combined Politifact and Gossipcop user interaction set, MIB set of followers, and combined Politifact and Gossipcop set of followers. Results for different cases were similar. One of Pearson correlation matrices is shown in Figure 1. Based on them, selected features for user interaction sets were: eigenvector score, closeness score, hits auth, page rank, and inDegree. For the MIB set of followers and combined Politifact and Gossipcop set of followers, the following features were selected: eigenvector score, harmonic closeness score, hits auth, page rank.

IV. Classification

To roughly identify the classifiers that will bring the best effect, the extracted features were trained using the following algorithms:

- K-Neighbors Classifier, where k is set to n=3 by default;
- classifier with decision tree algorithm (Decision Tree Classifier);
- classifier with Random Forest Classifier, where the number of heuristic estimators was initially set at 300;
- adaptive boost classifier (AdaBoost Classifier);
- Gradient Boosting Classifier;
- Gaussian classifier with naive Bayes algorithm (GaussianNB);
- Linear Discriminant Analysis classifier;
- Quadratic Discriminant Analysis classifier;
- Support Vector Machines Classifier (SVC), with regularization parameter C=0.025, radial basis function kernel, and 5-fold cross-validation;
- Support vector classifier with support vector quantity control proposed by Bernhard Schölkopf (NuSVC - Nu Support Vector Machines Classifier) [18].

Algorithms with unspecified configurations used the default settings of the Sci-Kit Learn library. An initial test was carried out using the Accuracy index and the Log Loss function to determine the confidence with which the algorithm made the classification [19]. The sets were divided using the function

[1] https://github.com/safe-graph/GNN-FakeNews

TABLE III: The results of individual algorithms measuring data dependence for the combined set of Politifact and Gossipcop (skeleton of the FakeNewsNet set)

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.020	0.929	0.335	0.237	0.627	0.002	0.002
harmonic closeness centrality	0.020	7925.474	0.000	0.101	0.750	0.148	0.046
hits hub	0.000	2.148	0.143	1.530	0.216	-0.002	0.000
hits auth	0.011	583.671	0.000	25.271	0.000	-0.040	0.016
betweenness score	0.009	438.670	0.000	13913.119	0.000	0.035	0.018
closeness score	0.021	8073.976	0.000	0.086	0.770	0.149	0.047
page rank score	0.225	9778.541	0.000	74.900	0.000	0.164	0.386
article rank score	0.259	14958.180	0.000	25.026	0.000	0.201	0.445
outDegree	0.011	0.121	0.728	7.679	0.006	-0.001	0.018
inDegree	0.048	472.832	0.000	7.679	0.006	-0.036	0.003
degree	0.031	0.495	0.482	15.358	0.000	-0.001	0.019

TABLE IV: The results of individual algorithms measuring data dependence for the MIB set of followers

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.245	1950.344	0.000	9.942	0.002	-0.037	0.001
harmonic closeness centrality	0.271	196599.696	0.000	8538.933	0.000	-0.352	0.353
hits hub	0.190	26842.827	0.000	981.298	0.000	-0.138	0.286
hits auth	0.248	693.709	0.000	3.032	0.082	0.022	0.036
betweenness score	0.002	2.082	0.149	4.20e9	0.000	0.001	0.000
closeness score	0.271	190243.421	0.000	7738.225	0.000	-0.347	0.289
page rank score	0.264	3580.209	0.000	22986.895	0.000	-0.051	0.000
article rank score	0.260	1308.763	0.000	708.467	0.000	-0.031	0.000
outDegree	0.114	51.639	0.000	3397591.733	0.000	-0.006	0.018
inDegree	0.178	11221.404	0.000	3.4e7	0.000	-0.089	0.007
degree	0.053	204.437	0.000	6795183.465	0.000	-0.012	0.010

TABLE V: The results of individual algorithms measuring data dependence for the FakeNewsNet set of followers

Feature	Mutual info scores	F-test scores	F-test pvalues	Chi2 scores	Chi2 pvalues	Pearson scores	Tree classifier
eigenvector score	0.244	3560.221	0.000	19.645	0.000	-0.066	0.163
harmonic closeness centrality	0.263	2828.637	0.000	3.670	0.055	-0.059	0.168
hits hub	0.103	3.883	0.049	0.077	0.782	-0.002	0.069
hits auth	0.261	1850.662	0.000	9.523	0.002	-0.048	0.159
closeness score	0.260	1961.225	0.000	2.354	0.125	-0.049	0.174
page rank score	0.266	4.536	0.033	306.354	0.000	-0.002	0.136
article rank score	0.271	61.463	0.000	164.771	0.000	-0.009	0.131

TABLE VI: Stratified 10-fold cross-validation results for selected classifiers for the combined set of Politifact and Gossipcop.

Classifier	Mean Validation Accuracy	Mean Validation Precision	Mean Validation Recall	Mean Validation F1 Score
K-Neighbors	75.082	0.917	0.346	0.502
Random forest	80.714	0.812	0.611	0.697
AdaBoost	72.101	0.652	0.499	0.565
Gradient boosting	73.710	0.688	0.506	0.583

sklearn.model_selection.train_test_split from the sci-kit learn library, which split the set on a scale of 0.7 into training and test sets [17].

Finally, the following classifiers were subjected to further analysis: random forest, gradient boost, k-nearest neighbors, adaptive boost. These classifiers were subjected to a stratified 10-fold cross-validation study following a review of popular methods for testing the efficiency of classifiers [20]. Stratification is a good solution for unbalanced sets, and the K-fold method itself has already been used in previous works on this topic [10][11]. It is also widely used, and effective [21]. The study results are presented in Table VI.

Table VI shows that all models obtained a relatively low recall value, indicating many classifications of "fake" users as "real". A better result was obtained in the case of precision, which gives us information about how many "real" accounts were rated as "fake". Fewer false profiles in the set (Table II) could have contributed to obtaining a high value of the accuracy coefficient.

When detecting fake user accounts, it is essential to consider how much it will cost to recognize a user spreading accurate information when they are a "troll". This cost can be very high, making the built algorithm useless. Sometimes, however, the "forbearance" of the algorithm can be desirable.

The final proposed solution is a classifier based on the random forest algorithm, where the number of estimators n has been heuristically set to n=300. This classifier was tested on the set of Russian troll accounts described in Table II. This set consisted of 413 fake accounts and was used only as another measure of verifying the task's success. The final version of the random forest classifier learned from the Politifact and Gossipcop collections achieved an accuracy of 84.50%. This observation is consistent with previous conclusions for the set of low validity, but the obtained result is better than the tests would indicate.

A. Choosing a solution to detect fake propagation channels and bots by analyzing the network of followed users

The classifiers were studied for these sets by testing the best-performing algorithms using selected features. Studies for the

TABLE VII: Test results of different classifiers on a set of MIB followers using 10-fold cross-validation with stratification.

Classifier	Mean Val. Accuracy	Mean Val. Precision	Mean Val. Recall	Mean Val. F1 Score
KNN	99.822	1.000	0.998	0.999
Decision Tree	99.388	0.996	0.997	0.997
Random Forest	99.397	0.996	0.997	0.997
AdaBoost	98.626	0.996	0.988	0.992
Gradient Boosting.	99.388	0.996	0.997	0.997
Gaussian NB	90.731	0.927	0.973	0.949
Linear Disc. Anal.	92.734	0.925	1.000	0.961
Quadratic Disc. Anal.	90.615	0.926	0.972	0.948

TABLE VIII: Test results of various classifiers on a set of FakeNewsNet followers without using cross-validation. (70% training data, 30% test data)

Classifier	Accuracy	Precision	Recall	F1 Score	Log Loss
Decision Tree	77.454	0.832	0.828	0.830	7.784
Random Forest	82.035	0.816	0.941	0.874	0.346
Gradient Boosting	70.184	0.693	0.990	0.815	0.584
GaussianNB	66.330	0.669	0.975	0.794	0.815
KNN	77.484	0.814	0.857	0.835	2.710
AdaBoost	66.888	0.670	0.987	0.798	0.690
Linear Disc. Anal.	66.660	0.667	0.997	0.799	0.635
Quadratic Disc. Anal.	66.411	0.669	0.978	0.795	0.790

TABLE IX: Test results of various classifiers learned on a set of FakeNewsNet followers and tested on a set of MIB followers without cross-validation. (70% training data, 30% test data)

Classifier	Accuracy	Precision	Recall	F1 Score	Log Loss
Decision Tree	34.155	0.772	0.368	0.498	22.753
Random Forest	87.018	0.894	0.969	0.930	0.668
Gradient Boosting	89.342	0.894	0.998	0.943	0.376
GaussianNB	8.762	0.000	0.000	0.000	5.609
KNeighbors	89.956	0.901	0.996	0.946	3.487
AdaBoost	90.237	0.903	0.998	0.948	0.683

TABLE X: Stratified 10-fold cross-validation results for selected classifiers for the combined set of FakeNewsNet and MIB followers.

Classifier	Mean Val. Accuracy	Mean Val. Precision	Mean Val. Recall	Mean Val. F1 Score
Random Forest	93.109	0.937	0.981	0.958
Gradient Boosting	87.988	0.875	0.994	0.930
KNN	91.647	0.942	0.955	0.949
AdaBoost	84.504	0.870	0.950	0.908

sets were performed using 10-fold cross-validation to better compare the results with those in other studies. In addition, the results of training performed on one set and then testing the model on a second set were also examined.

Table VII shows that the classifiers obtained high confidence and accuracy on the MIB set. This may be because it consisted of accounts generated by bots, which may have resulted in more significant differences between the characteristics. An important fact is that this set is already about ten years old, so the algorithms creating the bots could have been less advanced then. Similar high accuracy was achieved for all algorithms except Gaussian naive Bayes and linear and quadratic discriminant analysis.

Other studies based on measures of centrality have yielded for this set:

- in 2016 - precision 89.0%, accuracy 100% and validity 95% [10];
- in 2021 - precision, accuracy, and validity of 99.5%[11].

The extracted features for this set allowed us to obtain results similar to the work [11] where closeness centrality was introduced. At the same time, it can be seen that betweenness centrality, in this case, does not play a significant role in classifying "fake" users. It was also possible to obtain better results with the KNN classifier than previous works did with the random forest.

Worse algorithm efficiency results were obtained for the set of FakeNewsNet followers, presented in Table VIII. In this case, the random forest classifier was the best, achieving the highest accuracy and the lowest Log Loss. The decision tree algorithm, KNN, and gradient boost also achieved high scores. Worse results could be obtained because accounts were classified as fake or genuine only because they had a person who tweeted false information to their followers or followed. However, achieving such accuracy means that we can identify people who may be potentially unwitting spreaders of fake news, and according to research, they constitute a large part of fake news propagation channels [1].

However, surprising results were obtained for the model trained on the FakeNewsNet set and tested on the MIB set containing bots. The results presented in Table IX show that both the decision tree algorithm and the naive Bayes classifier performed much worse in this case than before. An interesting result was obtained in the case of the adaptive gain algorithm, which turned out to be the best in terms of precision and in terms of accuracy. The gradient boost and random forest algorithms also performed well. The KNN method obtained a relatively high value of the Loss Log coefficient. This result was probably obtained because the MIB set profiles were relatively easy to detect. The model built on FakeNewsNet seems to be quite effective in this case. In the reverse situation, when the MIB model was used on the FakeNewsNet set, worse results were obtained – it can be assumed that the model built on this set will have a lower generalization ability.

The random forest, gradient boost, KNN, and adaptive boost classifiers were tested on a combined set of FakeNewsNet and MIB followers to maximize the efficiency. Table X shows the result of testing the effectiveness of classifiers using 10-fold cross-validation with stratification.

The obtained values are slightly worse than those obtained in the research from 2021 [11] on the exclusive MIB set. However, the MIB set allowed us to build a classifier and significantly lower ability to generalize in detecting fake users, in contrast to the set of FakeNewsFollowers obtained in this work. Building a classifier based on both sets significantly increases the generalization capabilities of the classifier.

Ultimately, the best overall results were obtained for the random forest, which confirms previous studies. At the same time, other algorithms have also been shown to be highly effective. Very high accuracy was obtained for the gradient boost, which may be beneficial in the case when a maximum "raw" classifier is needed in detecting fake users, even at the cost of considering some genuine users as fake.

V. Conclusions

The article presents graph techniques for detecting false information. An essential aspect of detecting fake news is combining knowledge from many disciplines and data from different contexts to get better results. Combining several methods gives better results, but creating a complete system that classifies information as genuine and false using content and social context analysis is time-consuming and complicated. Studying individual techniques of a complex solution, such as the one presented in the paper, requires much time and collecting appropriate training data for machine learning algorithms.

The problem of classification presented in both cases, for the analysis of connections between users based on retweeting posts and based on followers, turned out to be a complicated issue. In the case of the user interaction network, it was impossible to build an effective classifier to solve the problem. However, we created a classifier that dealt with accounts of Russian trolls from the 2016 US elections quite effectively, proving that research in this direction should be continued.

It is more difficult to determine whether a user is part of a fake news channel based on what users they retweet. In further research, the set should be enlarged with additional samples, more work should be done to remove potential outliers, and the set should be better balanced to avoid overfitting. An important area for improvement is set normalization, model regularization, and parameter tuning.

Classifier tests in the case of the follower network essentially confirmed the conclusions regarding the effective operation of the random forest from previous studies [10][11]. It turned out, however, that the KNN classifier on the same set of MIB followers achieved better results than the random forest used in previous studies. It is an important finding, considering that learning this algorithm took less time than in the case of a random forest for n=300 estimators. Also, learning on the set of FakeNewsNet followers and validation on the MIB set was reasonably practical – although it could have been more reliable among the algorithms, obtaining a considerable value of the Log Loss coefficient.

References

[1] X. Zhou and R. Zafarani, "A survey of fake news: Fundamental theories, detection methods, and opportunities," *ACM Comput. Surv.*, 9 2020. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3395046

[2] C. Silverman, "This analysis shows how viral fake election news stories outperformed real news on facebook. buz- zfeed news." [Online]. Available: https://www.buzzfeednews.com/article/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[3] H. F. Gylfason, A. H. Sveinsdottir, V. Vésteinsdóttir, and R. Sigurvinsdottir, "Haters gonna hate, trolls gonna troll: The personality profile of a facebook troll," *Int J Environ Res Public Health*, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8199376/#B1-ijerph-18-05722

[4] M. Marek, "Russian information war: the activities of the russian propaganda apparatus in the context of the war in ukraine (as of the first half of march 2022)," *Bezpieczeństwo teoria i praktyka*, 2022. [Online]. Available: https://btip.ka.edu.pl/btip-2022-nr3/

[5] S. Hangloo and B. Arora, "Fake news detection tools and methods – a review," *International Journal of Advance and Innovative Research*, 6 2021. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2112/2112.11185.pdf

[6] K. Shu, D. Mahudeswaran, D. L. Suhang Wang, and H. Liu, "Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media," 2018. [Online]. Available: https://arxiv.org/abs/1809.01286

[7] B. D. Horne and S. Adali, "This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news." The 2nd International Workshop on News and Public Opinion at ICWSM, 2017. [Online]. Available: https://arxiv.org/abs/1703.09398

[8] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, "User preference-aware fake news detection," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021. [Online]. Available: https://arxiv.org/abs/2104.12259

[9] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," vol. abs/1902.06673, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918318210

[10] A. Mehrotra, M. Sarreddy, and S. Singh, "Detection of fake twitter followers using graph centrality measures," in *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016. doi: 10.1109/IC3I.2016.7918016 pp. 499–504.

[11] Y. Zhao and J. Weber, "Detecting fake users on social media with a graph database," vol. 12, 10 2021. [Online]. Available: https://doi.org/10.18357/tar121202120027

[12] A. Badawy, E. Ferrara, and K. Lerman, "Analyzing the digital traces of political manipulation: The 2016 russian interference twitter campaign," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 8 2018. [Online]. Available: https://doi.org/10.1109\%2Fasonam.2018.8508646

[13] W. Lyon, "The story behind russian twitter trolls: How they got away with looking human – and how to catch them in the future," 3 2018. [Online]. Available: https://neo4j.com/blog/story-behind-russian-twitter-trolls

[14] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "Fame for sale: Efficient detection of fake twitter followers," *Decision Support Systems*, vol. 80, pp. 56–71, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167923615001803

[15] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, and M. Tesconi, "The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race." Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 963–972. [Online]. Available: https://doi.org/10.1145/3041021.3055135

[16] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLOS ONE*, vol. 9, no. 2, pp. 1–5, 02 2014. doi: 10.1371/journal.pone.0087357. [Online]. Available: https://doi.org/10.1371/journal.pone.0087357

[17] scikit learn.org, "Api reference." [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_selection

[18] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, p. 1207–1245, 5 2000. [Online]. Available: https://doi.org/10.1162/089976600300015565

[19] I. J. Good, "Rational decisions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, no. 1, pp. 107–114, 1952. [Online]. Available: http://www.jstor.org/stable/2984087

[20] M. Ojala and G. C. Garriga, "Permutation tests for studying classifier performance." *Journal of machine learning research*, vol. 11, no. 6, 2010. [Online]. Available: https://www.jmlr.org/papers/volume11/ojala10a/ojala10a.pdf

[21] P. Refaeilzadeh, L. Tang, and H. Liu, "On comparison of feature selection algorithms," in *Proceedings of AAAI workshop on evaluation methods for machine learning II*, vol. 3, no. 4. AAAI Press Vancouver, 2007, p. 5. [Online]. Available: https://www.aaai.org/Library/Workshops/2007/ws07-05-007.php

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1167–1172
DOI: 10.15439/2023F8893
ISSN 2300-5963 ACSIS, Vol. 35

A Stochastic Optimization Technique for UNI-DEM framework

Venelin Todorov*‡[0000-0001-7134-5901], Slavi Georgiev*†[0000-0001-9826-9603], Ivan Dimov‡, Tzvetan Ostromsky‡
*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,
8 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
†Department of Applied Mathematics and Statistics, Angel Kanchev University of Ruse, 8 Studentska Str., 7004 Ruse, Bulgaria
‡Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg, venelin@parallel.bas.bg, sggeorgiev@math.bas.bg, sggeorgiev@uni-ruse.bg,
ivdimov@bas.bg, ceco@parallel.bas.bg

***Abstract*—This paper introduces a sophisticated multi-dimensional sensitivity analysis, incorporating cutting-edge stochastic methods for air pollution modeling. The study focuses on a large-scale long-distance transportation model of air pollutants, specifically the Unified Danish Eulerian Model (UNI-DEM). This mathematical model plays a pivotal role in understanding the detrimental impacts of heightened levels of air pollution. With this research, our intent is to employ it to tackle crucial questions related to environmental protection.**

We suggest advanced Monte Carlo and quasi-Monte Carlo methods, leveraging specific lattice and digital sequences to enhance the computational effectiveness of multi-dimensional numerical integration. Moreover, we further refine the existing stochastic methodologies for digital ecosystem modeling. The main aspect of our investigation is to analyze the sensitivity of the UNI-DEM model output to changes in the input emissions of human-induced pollutants and the rates of a number of chemical reactions.

The developed algorithms are utilized to calculate global Sobol sensitivity measures for various input parameters. We also assess their influence on key air pollutant concentrations in different European cities, considering the diverse geographical locations. The overarching goal of this research is to broaden our understanding of the elements influencing air pollution and inform potent strategies to alleviate its negative impacts on the environment.

I. Introduction

THIS paper focuses on conducting sensitivity analysis (SA) studies in the field of air pollution modeling [23], [26], [27], [28], [29], specifically using the Unified Danish Eulerian Model (UNI-DEM) as a case study. UNI-DEM is chosen for its accurate representation of relevant chemical processes in the atmosphere. The extensive output data generated by UNI-DEM has been utilized in various real-world applications, necessitating the accurate assessment of data reliability for specific uses. The research objective is to evaluate the dependability of the substantial volume of output data produced by the model. The study primarily examines the variations in hazardous air pollutant concentrations in relation to human-made emission levels and chemical reaction rates.

When it comes to making decisions, doubts arise regarding the reliability of large-scale mathematical models. To enhance their reliability, the sensitivity of model outputs to variations in model inputs caused by natural variability is studied and analyzed. Sensitivity analysis, as defined in this paper, is a procedure used to measure how sensitive mathematical model outputs are to variations in input data. The input data for sensitivity analysis in this study is obtained through simulations of a large-scale mathematical model known as the Unified Danish Eulerian Model (UNI-DEM). The model, developed at the Danish National Environmental Research Institute, covers a vast geographical area of 4800 × 4800 km, encompassing Europe and the Mediterranean fully and parts of Asia and Africa. It accurately represents the primary chemical, photochemical, and physical processes between the species considered and the emissions under rapidly changing meteorological conditions. The choice of this model for the case study is motivated by its precise treatment of chemical processes compared to other atmospheric chemistry models.

UNI-DEM is mathematically represented by the following system of partial differential equations (PDE) [22]:

$$\begin{aligned}\frac{\partial c_s}{\partial t} &= -\frac{\partial (uc_s)}{\partial x} - \frac{\partial (vc_s)}{\partial y} - \frac{\partial (wc_s)}{\partial z} + \\ &+ \frac{\partial}{\partial x}\left(K_x \frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial c_s}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial c_s}{\partial z}\right) + \\ &+ E_s + Q_s(c_1, c_2, \ldots, c_q) - (k_{1s} + k_{2s})c_s, \quad s = 1, 2, \ldots, q,\end{aligned} \tag{1}$$

where c_s are the chemical species' concentrations; u, v, w are the wind components; K_x, K_y, K_z – the diffusion coeff.; E_s – the emissions; k_{1s}, k_{2s} – dry / wet deposition

The work is supported by the Project BG05M2OP001-1.001-0004 UNITe, funded by the Operational Programme "Science and Education for Smart Growth", co-funded by the European Union trough the European Structural and Investment Funds and by the Bulgarian National Science Fund under Project KP-06-M62/1 "Numerical deterministic, stochastic, machine and deep learning methods with applications in computational, quantitative, algorithmic finance, biomathematics, ecology and algebra" from 2022. Venelin Todorov is supported by the Bulgarian National Science Fund under Project KP-06-N52/5 "Efficient methods for modeling, optimization and decision making" and Project KP-06-Russia/17 "New Highly Efficient Stochastic Simulation Methods and Applications".

coeff.; $Q_s(c_1, c_2, \dots c_q)$ – non-linear functions used to depict the chemical reactions that occur between the species being studied.

The Carbon Bond Mechanism (CBM-IV) chemical scheme is utilized to account for both non-linearity and stiffness. [22], [25].

II. Global Sensitivity Analysis – Sobol Approach

Variance-based methods are frequently employed in quantitative global sensitivity analysis, with the aim of assessing the contribution of input variance (either individual or grouped) to the overall variance of model output. Among these methods, the Sobol approach is widely utilized [17], [5], [19]. This approach is based on the assumption that the mathematical model can be represented by a specific model function:

$$\mathrm{u} = f(\mathrm{x}), \tag{2}$$

where $\mathrm{x} = (x_1, x_2, \dots, x_d) \in U^d \equiv [0; 1]^d$ is the vector of input parameters with a joint **p**robability **d**ensity **f**unction (p.d.f.) $p(\mathrm{x}) = p(x_1, \dots, x_d)$.

The concept behind the Sobol approach involves decomposing the integrable model function f into terms of increasing dimensionality [18], [20]:

$$f(\mathrm{x}) = f_0 + \sum_{\nu=1}^{d} \sum_{l_1<\dots<l_\nu} f_{l_1\dots l_\nu}(x_{l_1}, x_{l_2}, \dots, x_{l_\nu}), \tag{3}$$

where f_0 is some constant.

According to Sobol [19], the ANOVA (Analysis of Variance) decomposition decompose the output variance of a mathematical model into components attributed to each input. The goal is to identify which inputs contribute most significantly to the output variance. Each input variable is given a sensitivity index, or Sobol index, indicating its relative contribution to the output variance.

In simple terms, the process involves running the model multiple times with different combinations of inputs and observing the changes in the output. The larger the change in output for a given change in input, the more 'sensitive' the model is to that input.

The expression (3), where each term is selected to fulfill the specified condition, is referred to as the ANOVA representation of the model function $f(\mathbf{x})$:

$$\int_0^1 f_{l_1\dots l_\nu}(x_{l_1}, x_{l_2}, \dots, x_{l_\nu}) \mathrm{d}x_{l_k} = 0, 1 \le k \le \nu, \nu = 1, \dots, d.$$

This condition ensures that the functions on the right-hand side of (3) have a unique definition and $f_0 = \displaystyle\int_{U^d} f(\mathrm{x})\mathrm{dx}$. The quantities

$$\mathbf{D} = \int_{U^d} f^2(\mathrm{x})\mathrm{dx} - f_0^2, \quad \mathbf{D}_{l_1 \ \dots \ l_\nu} = \int f^2_{l_1 \ \dots \ l_\nu} \mathrm{d}x_{l_1} \dots \mathrm{d}x_{l_\nu} \tag{4}$$

are referred to as total and partial variances, respectively. Similar is true for the total variance which is represented by the corresponding partial variances: $\mathbf{D} = \sum_{\nu=1}^{d} \sum_{l_1<\dots<l_\nu} \mathbf{D}_{l_1\dots l_\nu}$.

The definition of Sobol global sensitivity indices is the following[19], [17]:

$$S_{l_1 \ \dots \ l_\nu} = \frac{\mathbf{D}_{l_1 \ \dots \ l_\nu}}{\mathbf{D}}, \quad \nu \in \{1, \dots, d\}, \tag{5}$$

and the **t**otal **s**ensitivity **i**ndex (TSI) of an input parameter $x_i, i \in \{1, \dots, d\}$ defined by [19], [17]:

$$S_i^{tot} = S_i + \sum_{l_1 \neq i} S_{il_1} + \sum_{l_1, l_2 \neq i, l_1 < l_2} S_{il_1 l_2} + \dots + S_{il_1 \dots l_{d-1}}, \tag{6}$$

where S_i is named *the main effect (first-order sensitivity index)* of x_i and $S_{il_1\dots l_{j-1}}$ is the j^{th} order sensitivity index. The higher-order terms characterize the interaction effects between the unknown input parameters $x_{i_1}, \dots, x_{i_\nu}, \nu \in \{2, \dots, d\}$ on the output variance. Therefore comprehensive mathematical analysis of the global sensitivity analysis problem involves the calculation of total sensitivity indices (6) of the corresponding order. This calculation relies on the formulas (4)-(5), which require the computation of multidimensional integrals.

The authors of [9] discuss which formulation of

$$f_0^2 = \left(\int_{U^d} f(\mathrm{x})\mathrm{dx} \right)^2 \tag{7}$$

is better when calculating the total variance and the Sobol global sensitivity measures. The first approximation formula is

$$\hat{f_0^2} = \frac{1}{n} \sum_{i=1}^{n} f(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,d}) \ f(\mathrm{x}'_{i,1}, \dots, \mathrm{x}'_{i,d}) \tag{8}$$

and the second one is

$$\hat{f_0^2} = \left\{ \frac{1}{n} \sum_{i=1}^{n} f(\mathrm{x}_{i,1}, \dots, \mathrm{x}_{i,d}) \right\}^2, \tag{9}$$

where x and x′ are two independent sample vectors. If one estimates sensitivity indices of a fixed order, the expression (8) is better (as it is recommended in [9]), and this is why we apply it here as well.

III. A new optimization method for SA

Let us take into account a multidimensional integration task in dimension s:

$$I(f) := I = \int_{U^s} f(x)\mathrm{d}x. \tag{10}$$

We introduce the quadrature formula

$$I_N(f) = \frac{1}{N} \sum_{i=1}^{N} f(x_i), \tag{11}$$

where $P_N = x_1, x_2, \dots, x_N, x_i \in [0, 1)^s$ are the nodes for the integration of the formula. The selection of these nodes is critical because it establishes the discrepancy of the sequence and the precision of the quadrature. For equation (11), the integration nodes that we will employ are [13], [14]:

$$x_k = \left(\left\{ \frac{kz_1}{N} \right\}, \left\{ \frac{kz_2}{N} \right\}, \dots, \left\{ \frac{kz_s}{N} \right\} \right), \ k = 1, 2, \dots, N, \tag{12}$$

where N represents the quantity of nodes, z is an s-dimensional generating vector of the lattice set and $a = a-[a]$ is the fractional part of a. Now, the equation (11) with nodes (12) and generators z is referred to as rank-1 lattice rules [2]. We will adopt a particular category of rank-1 lattice: the symmetrized lattice (SL).

We put forth a unique SL, defined in the following manner. In the unidimensional scenario, we set up a function, appropriate for periodic integrand functions, to be used with a non-periodic function F by applying the SL to the function

$$L(x) = \big(F(x) + F(1-x)\big)/2,$$

in a single dimension. For the two-dimensional situation, the function L is established as

$$L(x_1, x_2) = (F(x_1, x_2) + F(x_1, 1-x_2) + F(1-x_1, x_2) + F(1-x_1, 1-x_2))/4.$$

The definition of the function $L(x_1, \dots, x_s)$ is extrapolated for s dimensions:

$$L(x_1, \dots, x_s) = 2^{-s} \sum_{\varepsilon\in 0,1^s} F\big(\varepsilon_1 x_1+(1-\varepsilon_1)(1-x_1), \dots, \varepsilon_s x_s+(1-\varepsilon_s)(1-x_s)\big). \tag{13}$$

The terms over which the summation takes place can be envisioned as vertices of a parallelotope, with diagonals converging at the point $(1/2, 1/2, \dots, 1/2) \in [0, 1]^s$. Formula (13) is identical to

$$L(x_1, \dots, x_s) = \sum_{\varepsilon\in 0,1^s} F\left(x_1^{\varepsilon_1}(1-x_1)^{1-\varepsilon_1}, x_2^{\varepsilon_2}(1-x_2)^{1-\varepsilon_2}, \dots, x_s^{\varepsilon_s}(1-x_s)^{1-\varepsilon_s}\right).$$

The lattice we will use in our study are defined as follows. The first one is a rank one lattice rule with prime number of points and with product weights, which symmetrized version would be denoted by `SL-1pt`. The next lattice is a rank one lattice rule with prime number of points and with order dependent weights, which symmetrized version would be designated with `SL-1od`. These two lattice rules have variant with number of points, which is a prime power instead of prime itself, and we would denote them with `SL-1expt` and `SL-1exod`, respectively. The last lattice that would be used is a polynomial rank one lattice sequence in base two and with product weights, designated by `SL-2poly`.

IV. Sensitivity Studies with Respect to Emission Levels

In this section, we report the findings of the Sensitivity Analysis performed on the output of UNI-DEM, with particular attention paid to the monthly average ammonia concentrations in Milan, Italy. This analysis scrutinizes how alterations in anthropogenic emission data, employed as input, impact these concentrations.

The input is composed of 4 distinct constituents $\mathbf{E} = (\mathbf{E^A}, \mathbf{E^N}, \mathbf{E^S}, \mathbf{E^C})$:

$\mathbf{E^A}$ – ammonia (NH_3);
$\mathbf{E^S}$ – sulphur dioxide (SO_2);
$\mathbf{E^N}$ – nitrogen oxides ($NO + NO_2$);
$\mathbf{E^C}$ – anthropogenic hydrocarbons.

The domain under examination is the 4-dimensional hypercube $[0.5, 1]^4$.

The primary determinant of ammonia output concentrations is the emission of ammonia itself, accounting for approximately 89% in Milan. The next most influential factor is the emission of sulphur dioxide, contributing around 11% to ammonia output. This depiction of first- and second-order sensitivity indices for ammonia in Milan was established through the use of correlated sampling as part of Sobol's variance-based approach for multidimensional sensitivity analysis. This was done to compute all potential sensitivity measures and investigate the impact of the selected four groups of air pollutant emissions on the concentration of three key air pollutants.

This signifies the degree to which ammonia emissions directly affect ammonia concentrations, emphasizing the need for effective monitoring and management of these emissions. The role of sulphur dioxide emissions, albeit smaller, also needs to be taken into account due to their noticeable influence. Using multidimensional sensitivity analysis aids in comprehensively understanding the role of various emissions in air pollution, thereby enabling more targeted strategies to mitigate these issues. The results provide a foundation for future work aimed at improving air quality, informing policy decisions, and guiding future research into pollution control methods.

The relative error estimation for quantities f_0, the overall variance $\mathbf{D}$, the first-order (S_i) and the total (S_i^{tot}) sensitivity indices is exhibited in Tables I, II, III, correspondingly. f_0 is represented by a 4-dimensional integral, whereas the remaining quantities are denoted by 8-dimensional integrals, drawing upon the concepts of the *correlated sampling* technique to compute sensitivity measures in a robust manner (refer to [9], [20]). Four distinct stochastic methods utilized for numerical integration are displayed in separate columns in the tables.

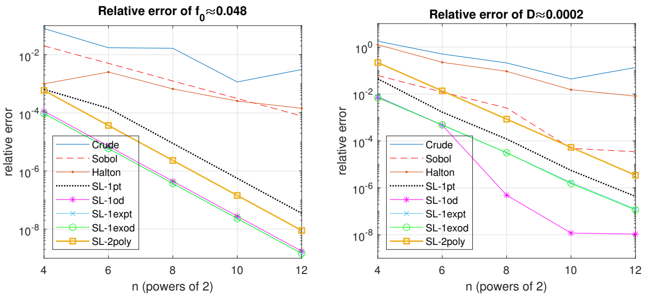


Fig. 1. Relative errors for the calculation of $f_0 \approx 0.048$ (left) and $\mathbf{D} \approx 0.0002$ (right)

When examining the model function f_0 with a sample size of $n = 2^{12}$, the most effective algorithm appears to be SL-1EXPT. This can be observed from the results given in Table I, which highlight outcomes for the maximum sample count.

When considering the total variance D for the same number of samples, SL-1OD comes out on top, as can be seen in Table II, which presents findings for the highest sample amount.

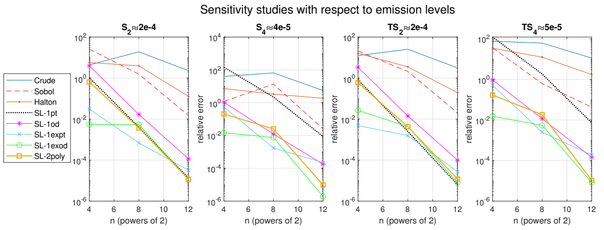


Fig. 2. Relative errors for the calculation of the small in value SIs

TABLE I
RELATIVE ERROR FOR THE EVALUATION OF $f_0 \approx 0.048$.

n	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**SL-1EXOD**	**SL-1POLY**
2^4	7.9e-02	2.0e-02	1.0e-03	6.4e-04	1.1e-04	9.5e-05	**9.5e-05**	5.9e-04
2^6	1.7e-02	5.0e-03	2.5e-03	1.4e-04	7.1e-06	**5.9e-06**	5.9e-06	3.7e-05
2^8	1.7e-02	1.2e-03	6.7e-04	8.9e-06	4.5e-07	**3.7e-07**	3.7e-07	2.3e-06
2^{10}	1.2e-03	3.1e-04	2.5e-04	5.6e-07	2.8e-08	**2.3e-08**	2.3e-08	1.4e-07
2^{12}	3.1e-03	7.8e-05	1.4e-04	3.5e-08	1.7e-09	**1.5e-09**	1.5e-09	9.0e-09

Regarding Sensitivity Indices (SIs), the optimal method is SL-1EXOD, as evident in Table III.

The efficiency and results of these algorithms can be further examined in Figures 1 and 2. The latter focuses particularly on SIs with smaller values, providing a more detailed look into their performance.

From the data displayed in Table III, it is evident that the SL-1EXOD algorithm enhances results in a majority of scenarios, particularly in determining the low-value sensitivity indices S_2, S_4, S_2^{tot}, and S_4^{tot}. These specific instances hold substantial significance as they play a crucial role in ascertaining the dependability of the model outcomes.

V. SENSITIVITY STUDIES WITH RESPECT TO CHEMICAL REACTIONS RATES

This section analyzes the sensitivity of the concentration levels of ozone in the atmosphere above Genova, Italy, with respect to modifications in the reaction rates of specific chemical reactions entailed in the condensed CBM-IV model ([22]). Notably, reactions # $1, 3, 7, 22$ (time-dependent) and # $27, 28$ (time independent) are the primary focus. The simplified formulas for the chemical reactions are as follows:

$$
\begin{aligned}
&[\#\mathbf{1}] \qquad NO_2 + h\nu \Longrightarrow NO + O;\\
&[\#\mathbf{3}] \qquad O_3 + NO \Longrightarrow NO_2;\\
&[\#\mathbf{7}] \qquad NO_2 + O_3 \Longrightarrow NO_3;\\
&[\#\mathbf{22}] \qquad HO_2 + NO \Longrightarrow OH + NO_2;\\
&[\#\mathbf{27}] \qquad HO_2 + HO_2 \Longrightarrow H_2O_2;\\
&[\#\mathbf{28}] \qquad OH + CO \Longrightarrow HO_2.
\end{aligned}
$$

The domain under examination is the 6-dimensional hypercube $[0.6, 1.4]^6$).

TABLE II
RELATIVE ERROR FOR THE EVALUATION OF THE TOTAL VARIANCE $\mathbf{D} \approx 0.0002$.

n	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**SL-1EXOD**	**SL-1POLY**
2^4	1.8e+00	6.1e-02	1.3e+00	4.4e-02	7.2e-03	7.9e-03	**6.9e-03**	2.2e-01
2^6	5.1e-01	1.2e-02	2.2e-01	1.7e-03	**4.7e-04**	4.7e-04	4.9e-04	1.4e-02
2^8	2.1e-01	2.5e-03	9.3e-02	1.2e-04	**4.8e-07**	3.1e-05	3.1e-05	8.5e-04
2^{10}	4.4e-02	4.8e-05	1.5e-02	5.5e-06	**1.2e-08**	1.7e-06	1.5e-06	5.3e-05
2^{12}	1.3e-01	3.4e-05	8.1e-03	4.3e-07	**1.1e-08**	1.2e-07	1.1e-07	3.4e-06

TABLE III
RELATIVE ERROR FOR ESTIMATION OF SENSITIVITY INDICES OF INPUT PARAMETERS USING DIFFERENT QUASI-MONTE CARLO APPROACHES ($n = 2^{12}$).

SI	EQ	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**SL-1EXOD**	**SL-1POLY**
S_1	9e-01	1.1e-03	5.7e-07	8.8e-04	4.8e-07	5.1e-08	1.7e-08	**8.3e-09**	3.3e-06
S_2	2e-04	2.3e+00	1.5e-02	1.3e-01	1.4e-05	1.2e-04	3.2e-05	**1.1e-05**	1.2e-05
S_3	1e-01	1.2e-02	1.2e-04	8.7e-03	6.1e-07	6.2e-07	1.4e-08	**1.1e-08**	2.7e-05
S_4	4e-05	5.5e+00	2.5e-02	1.9e+00	8.7e-03	1.8e-04	2.3e-04	**1.9e-06**	9.7e-06
S_1^{tot}	9e-01	1.9e-03	1.6e-05	1.2e-03	4.7e-07	4.2e-08	1.9e-08	**9.1e-10**	3.3e-06
S_2^{tot}	2e-04	3.1e+00	2.0e-02	1.9e-01	**6.3e-06**	9.9e-05	2.7e-05	7.8e-06	1.2e-05
S_3^{tot}	1e-01	1.6e-02	6.0e-05	5.8e-03	7.4e-07	7.0e-07	2.8e-07	**8.4e-08**	2.7e-05
S_4^{tot}	5e-05	1.1e+01	4.3e-02	1.7e+00	7.2e-03	1.4e-04	1.9e-04	**8.3e-06**	1.1e-05

The findings from our analysis, with a focus on the reactions as described by the CBM-IV scheme, led to several important insights. Reaction rates #1, 3, and 22 have a profound impact on O_3 concentrations, making them extremely influential in this context. On the other hand, reaction rates #7 and 27, while not as dominant, still hold a noticeable significance. Contrarily, the influence of reaction rate #28 can be deemed negligible.

In other words, the study has found that there are clear relationships between specific reaction rates and O_3 concentrations. While the reactions #1, 3, and 22 play a leading role, reactions #7 and 27 still contribute to a certain extent. This information suggests that these specific reactions could be potential targets for strategies to reduce O_3 concentrations. However, the role of reaction #28 appears to be minimal, suggesting that efforts aimed at this reaction are likely to be less effective. These observations provide a better understanding of the dynamics involved in O_3 concentrations, paving the way for more effective and targeted air pollution control strategies.

The estimated relative error for the values f_0, total variance $\mathbf{D}$, and a subset of the sensitivity indices are detailed in Tables IV, V, and VI, correspondingly.

The parameter f_0 is depicted by a 6-dimensional integral, while the remaining quantities being examined are shown by 12-dimensional integrals, in line with the *correlated sampling* principle.

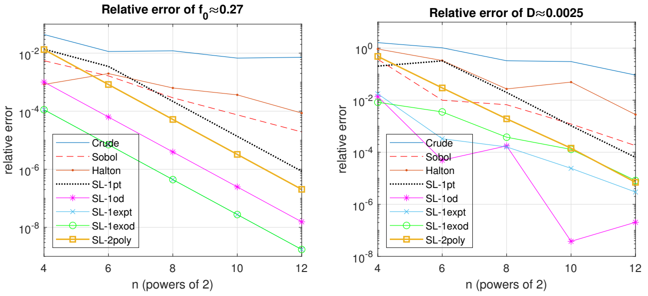


Fig. 3. Relative errors for the calculation of $f_0 \approx 0.27$ (left) and $\mathbf{D} \approx 0.0025$ (right)

In the case of the model function f_0, the optimal algorithm turns out to be the SL-1EXPT, with SL-1EXOD coming in as the second-best choice, as evidenced by the results displayed in Table IV. When dealing with a sample size of $n = 2^{12}$ for the total variance D, the top-performing algorithm is SL-1OD, as demonstrated by the results shown in Table V for the largest

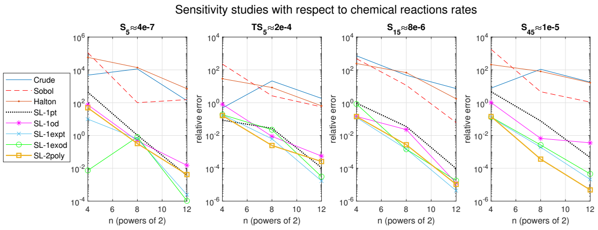


Fig. 4. Relative errors for the calculation of the small in value SIs

TABLE IV

RELATIVE ERROR FOR THE EVALUATION OF $f_0 \approx 0.27$.

n	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**SL-1EXOD**	**SL-1POLY**
2^4	4.4e-02	5.5e-03	8.3e-04	1.4e-02	1.0e-03	**1.1e-04**	1.1e-04	1.3e-02
2^6	1.1e-02	1.6e-03	2.0e-03	3.5e-03	6.3e-05	7.0e-06	**7.0e-06**	8.3e-04
2^8	1.2e-02	2.9e-04	6.3e-04	2.2e-04	3.9e-06	**4.4e-07**	4.4e-07	5.2e-05
2^{10}	6.8e-03	7.5e-05	3.7e-04	1.4e-05	2.5e-07	**2.7e-08**	2.7e-08	3.2e-06
2^{12}	7.2e-03	1.9e-05	8.6e-05	8.5e-07	1.5e-08	1.7e-09	**1.7e-09**	2.0e-07

number of samples. In terms of the Sensitivity Indices (SIs), SL-1EXPT proves to be the most efficient method, closely followed by SL-1POLY and SL-1OD. This is clearly shown in Table VI. The algorithms' performance can be visually inspected through Fig. 3 and 4, with the latter emphasizing the SIs of smaller values.

As evidenced by Table VI, the SL-1EXPT algorithm improves outcomes in most instances, most notably for the lower-valued sensitivity indices S_5, S_5^{tot}, S_{15}, and S_{45}. These indices are particularly significant as they greatly impact the trustworthiness of the model's outcomes.

VI. CONCLUSION

We have examined the computational effectiveness of various stochastic methodologies for multi-dimensional numerical integration in relation to relative error and computational resources. The subject of this study is the sensitivity analysis of the output from the UNI-DEM model to changes in input emissions of anthropogenic contaminants and alterations in a selection of chemical reaction rates.

We scrutinize the impact of emission levels on key air pollutants, specifically ammonia, ozone, ammonium sulphate, and ammonium nitrate.

The computational experiments reveal that the optimization methods developed are amongst the most effective stochastic strategies currently available for determining sensitivity indices, particularly for the most challenging task – assessing the least value sensitivity indices, which are crucial for the dependability of the model's outcomes.

TABLE V

RELATIVE ERROR FOR THE EVALUATION OF THE TOTAL VARIANCE $\mathbf{D} \approx 0.0025$.

n	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**SL-1EXOD**	**SL-1POLY**
2^4	1.6e+00	3.7e-01	9.3e-01	2.0e-01	1.4e-02	1.8e-02	**8.3e-03**	4.8e-01
2^6	1.0e+00	1.0e-02	3.3e-01	3.2e-01	**4.9e-05**	3.3e-04	3.5e-03	3.0e-02
2^8	3.3e-01	6.7e-03	2.7e-02	2.0e-02	1.8e-04	**1.6e-04**	3.8e-04	1.9e-03
2^{10}	3.0e-01	1.2e-03	5.0e-02	1.0e-03	**3.8e-08**	2.4e-05	1.3e-04	1.4e-04
2^{12}	9.2e-02	1.8e-04	2.8e-03	6.4e-05	**2.0e-07**	3.0e-06	8.2e-06	7.0e-06

TABLE VI

RELATIVE ERROR FOR ESTIMATION OF SENSITIVITY INDICES OF INPUT PARAMETERS USING DIFFERENT QUASI-MONTE CARLO APPROACHES ($n = 2^{12}$).

SI	EQ	**Crude**	**Sobol**	**Halton**	**SL-1PT**	**SL-1OD**	**SL-1EXPT**	**R1L-1EXOD**	**SL-1POLY**
S_1	4e-01	4.9e-04	1.8e-04	2.4e-02	6.0e-05	**7.6e-07**	2.2e-06	9.3e-06	9.9e-06
S_2	3e-01	2.9e-01	4.5e-04	3.0e-02	6.2e-05	4.9e-06	**2.6e-06**	8.1e-06	5.4e-06
S_3	5e-02	2.3e-01	4.3e-03	8.2e-02	1.5e-03	5.7e-05	**2.1e-06**	5.9e-06	1.4e-04
S_4	3e-01	3.1e-01	1.1e-03	3.3e-02	9.7e-05	**3.6e-07**	3.4e-06	8.7e-06	6.7e-06
S_5	4e-07	1.3e+02	1.5e+02	7.1e+02	3.6e-03	1.5e-02	2.4e-04	**1.0e-04**	4.2e-03
S_6	2e-02	8.2e-01	1.0e-02	4.1e-02	1.0e-05	2.4e-05	4.3e-05	4.6e-04	**1.7e-06**
S_1^{tot}	4e-01	2.7e-02	1.3e-04	1.5e-02	6.3e-05	**1.1e-06**	1.8e-06	9.5e-06	9.0e-06
S_2^{tot}	3e-01	3.4e-01	3.1e-04	3.9e-02	6.4e-05	5.3e-06	**2.8e-06**	8.9e-06	5.0e-06
S_3^{tot}	5e-02	1.6e-01	9.7e-05	7.6e-02	1.4e-03	5.4e-05	**1.4e-06**	6.3e-06	1.4e-04
S_4^{tot}	3e-01	3.7e-01	6.1e-04	2.7e-02	1.0e-04	**1.3e-06**	2.9e-06	9.6e-06	5.7e-06
S_5^{tot}	2e-04	1.8e+00	5.6e-01	6.4e-01	1.1e-04	5.6e-04	**1.7e-05**	3.1e-05	2.6e-04
S_6^{tot}	2e-02	9.6e-01	6.6e-04	5.7e-02	1.7e-05	2.7e-05	4.5e-05	4.4e-04	**3.9e-07**
S_{12}	6e-03	3.9e+00	2.1e-04	3.6e-01	8.4e-05	5.6e-06	4.7e-06	1.6e-05	**4.2e-06**
S_{14}	5e-03	2.0e+00	1.1e-02	1.9e-01	2.2e-04	**1.2e-05**	1.4e-05	1.6e-05	3.3e-05
S_{15}	8e-06	7.3e+00	6.2e-02	1.7e+00	9.7e-05	1.3e-05	**4.3e-06**	1.8e-05	1.1e-05
S_{24}	3e-03	1.6e+00	5.7e-03	1.6e-01	2.0e-04	3.1e-05	4.2e-06	6.1e-05	**2.6e-06**
S_{45}	1e-05	1.8e+01	1.1e+00	1.6e+01	4.5e-04	3.6e-03	2.1e-05	4.6e-05	**4.8e-06**

These findings are of considerable significance for environmental conservation and the credibility of future predictions.

REFERENCES

[1] I. Antonov, V. Saleev, An economic method of computing LP_τ-sequences, USSR Comput. Math. Phys. 19, (1979), 252–256.

[2] Bahvalov, N. On the Approximate Computation of Multiple Integrals. *Vestn. Mosc. State Univ.* **1959**, *4*, 3–18.

[3] I. Dimov, Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, (2008).

[4] I.T. Dimov, R. Georgieva, Tz. Ostromsky, Z. Zlatev, Advanced algorithms for multidimensional sensitivity studies of large-scale air pollution models based on Sobol sequences, Computers and Mathematics with Applications 65(3), "Efficient Numerical Methods for Scientific Applications", Elsevier, (2013), 338–351.

[5] F. Ferretti, A. Saltelli, S. Tarantola, Trends in sensitivity analysis practice in the last decade, Journal of Science of the Total Environment 568, (2016), 666–670.

[6] J. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numerische Mathematik 2, (1960), 84–90.

[7] J. Halton, G.B. Smith, Algorithm 247: radical-inverse quasi-random point sequence, Communications of the ACM 7, (1964), 701–702.

[8] S. Joe, F. Kuo, Remark on algorithm 659: implementing Sobol's quasirandom sequence generator, ACM Transactions on Mathematical Software 29(1), (2003), 49–57.

[9] T. Homma, A. Saltelli, Importance measures in global sensitivity analysis of nonlinear models, Reliability Engineering and System Safety 52, (1996), 1–17.

[10] A. Karaivanova, E. Atanassov, T. Gurov, R. Stevanovic, K. Skala, Variance reduction MCMs with application in eEnvironmental studies: sensitivity analysis. AIP Conference Proceedings 1067(1), (2008), 549–558.

[11] A Karaivanova, I. Dimov, S. Ivanovska, A quasi-Monte Carlo method for integration with improved convergence, In International Conference on Large-Scale Scientific Computing, Springer, Berlin, Heidelberg, (2001), 158–165.

[12] L. Kocis, W. J. Whiten, Computational investigations of low-discrepancy sequences, ACM Transactions on Mathematical Software 23(2), (1997), 266–294.

[13] N.M. Korobov (1960) Properties and calculation of optimal coefficients, *Soviet Mathematics Doklady* **1**, 696–700.

[14] N.M. Korobov, *Number-theoretical methods in approximate analysis* (Fizmatgiz, Moscow, 1963).

[15] J. Matousek, On the L2-discrepancy for anchored boxes, Journal of Complexity 14(4), (1998), 527–556.

[16] G. Ökten, A. Göncüb, Generating low-discrepancy sequences from the normal distribution: Box–Muller or inverse transform?, Mathematical and Computer Modelling 53, (2011), 1268–1281.

[17] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, Halsted Press, New York, (2004).

[18] I. Sobol, Numerical methods Monte Carlo, Nauka, Moscow, (1973).

[19] I.M. Sobol, Sensitivity estimates for nonlinear mathematical models, Mathematical Modeling and Computational Experiment 1(4), (1993), 407–414.

[20] I.M. Sobol, S. Tarantola, D. Gatelli, S. Kucherenko, W. Mauntz, Estimating the approximation error when fixing unessential factors in global sensitivity analysis, Reliability Engineering & System Safety 92, (2007), 957–960.

[21] S.L. Zaharieva, I.R. Georgiev, V.A. Mutkov, Y.B. Neikov, Arima approach for forecasting temperature in a residential premises part 2, in 2021 20^{th} International Symposium INFOTEH-JAHORINA (INFOTEH), IEEE, (2021), 1–5.

[22] Z. Zlatev, Computer Treatment of Large Air Pollution Models, KLUWER Academic Publishers, Dorsrecht-Boston-London, (1995).

[23] Z. Zlatev, I. Dimov and K. Georgiev, Relations between Climatic Changes and High Pollution Levels in Bulgaria, Open Journal of Applied Sciences, Vol. 6 (2016), pp. 386-401, http://dx.doi.org/10.4236/ojapps.2016.67040.

[24] Z. Zlatev, I.T. Dimov, K. Georgiev, Three-dimensional version of the Danish Eulerian model, Z. Angew. Math. Mech. 76(S4), (1996), 473–476.

[25] Z. Zlatev, I.T. Dimov, Computational and Numerical Challenges in Environmental Modelling, Elsevier, Amsterdam, (2006).

[26] Z. Zlatev, I. Faragó and Á. Havasi, Impact of Climatic Changes on Pollution Levels, in Mathematical Problems in Meteorological Modelling, Springer Series on "Mathematics in Industry", Vol. 24 (2016), pp. 129-161, A. Bátkai, P. Csomós, I. Faragó, A. Horányi and G. Szépszó, eds., Springer, https://doi.org/10.3390/atmos2030201.

[27] Z. Zlatev, K. Georgiev and I. Dimov, Influence of Climatic Changes on Pollution Levels in the Balkan Peninsula, Computers and Mathematics with Applications, Vol. 65 (2011), pp. 544-562, https://doi.org/10.1016/j.camwa.2012.07.006.

[28] Z. Zlatev, Á. Havasi and I. Faragó: "Influence of Climatic Changes on Pollution Levels in Hungary and Its Surrounding Countries", Atmosphere, Vol. 2 (2011), pp. 201-221, https://doi.org/10.3390/atmos2030201.

[29] Z. Zlatev and L. Moseholm, Impact of Climatic Changes on Pollution Levels in Denmark, Ecological Modelling, Vol. 217 (2008), pp. 305-319, http://dx.doi.org/10.1016/j.ecolmodel.2008.06.030.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1173–1178
DOI: 10.15439/2023F1191
ISSN 2300-5963 ACSIS, Vol. 35

Extremal algebraic graphs, quadratic multivariate public keys and temporal rules

Vasyl Ustymenko
0000-0002-2138-2357
Royal Holloway University of London
United Kingdom
Email: Vasyl.Ustymenko@rhul.ac.uk

Aneta Wróblewska
0000-0001-9724-4586
University of Maria Curie-Skłodowska,
Lublin, Poland
Email: aneta.wroblewska@mail.umcs.pl

***Abstract*—We introduce large groups of quadratic transformations of a vector space over the finite fields defined via symbolic computations with the usage of algebraic constructions of Extremal Graph Theory. They can serve as platforms for the protocols of Noncommutative Cryptography with security based on the complexity of word decomposition problem in noncommutative polynomial transformation group. The modifications of these symbolic computations in the case of large fields of characteristic two allow us to define quadratic bijective multivariate public keys such that the inverses of public maps has a large polynomial degree. Another family of public keys is defined over arbitrary commutative ring with unity. We suggest the usage of constructed protocols for the private delivery of quadratic encryption maps instead of the public usage of these transformations, i.e. the idea of temporal multivariate rules with their periodical change.**

I. On Post Quantum, Multivariate and Noncommutative Cryptography

POST-Quantum Cryptography (PQC) is an answer to a threat coming from a full-scale quantum computer able to execute Shor's algorithm. With this algorithm implemented on a quantum computer, currently used public key schemes, such as RSA and elliptic curve cryptosystems, are no longer secure. PQC is subdivided into Coding based Cryptography, Multivariate Cryptography, Noncommutative Cryptography, Hash based Cryptography, Isogeny based Cryptography and Lattice based Cryptography. Each of these six areas is based on the complexity of certain NP-hard problem. Noteworthy that fundamental assumption of cryptography that there are no polynomial-time algorithms for solving any NP-hard problem remains valid. So all six directions are well justified theoretically.

The tender of US National Institute of Standardisation Technology (NIST, 2017) is dedicated to the standardisation process of possible real life Post-Qantum Public keys. Already selected in July of 2022 four cryptosystems are developed via methods of Lattice based Cryptography. This fact motivates researchers from other four core areas of Post Quantum Cryptography to continue design of new cryptographical primitives. Noteworthy that during the NIST project an interesting results on cryptanalysis of Unbalanced Rainbow Oil and Vinegar digital signatures schemes were found (see [1], [2], [3]). This scheme is defined via quadratic multivariate public rule, which refers to MiniRank problem. Examples of previously known multivariate quadratic public keys a reader can find in classical monographs [4], [5], [6].

Graph based multivariate public keys with bijective encryption maps generated via special walks on incidence graph of projective geometry were proposed in [7] this year. It can be count as attempt to combine methods of Coding based and Multivariate Cryptographies. Classical multivariate public rule is a transformation of n-dimensional vector space over finite field F_q which move vector $(x_1, \dots, x_n)$ to the tuple $(g_1(x_1, \dots, x_n), g_2(x_1, \dots, x_n), \dots, g_n(x_1, \dots, x_n))$, where polynomials g_i are given in their standard forms, i.e. lists of monomial terms in the lexicographical order. The degree of this transformation is the maximal value of $\deg(g_i)$. Traditionally public rule has degree 2 or 3.

We use the known family of graphs $D(n, q)$ and $A(n, q)$ of increasing girth (see [8], [9] and further references) and their analogs $D(n, K)$ and $A(n, K)$ defined over finite commutative ring K with unity for the construction of our public keys. Noteworthy to mention that for each prime power q, $q > 2$ graphs $D(n, q)$, $n = 2, 3, \dots$ form a family of graphs of large girth (see [8]). There is well defined projective limit of these graphs which is a q-regular forest. In fact if K is an integral domain both families $A(n, K)$ and $D(n, K)$ are approximations of infinite dimensional algebraic forests. Cubical transformation groups $GA(n, K)$ and $GD(n, k)$ of K_n (see [10], [11]), were used for the design of key exchange protocols of Noncommutative Cryptography (see [11], [12], [13]), elements of this groups were used for the creation of stream ciphers.

II. On graphs, groups and quadratic maps with the inverses of high degree

Let K be a commutative ring. We define $A(n, K)$ as bipartite graph with the point set $P = K_n$ and line set $L = K_n$ (two copies of a Cartesian power of K are used). We will use brackets and parenthesis to distinguish tuples from P and L. So $(p) = (p_1, p_2, \dots, p_n) \in P_n$ and $[l] = [l_1, l_2, \dots, l_n] \in L_n$. The incidence relation $I = A(n, K)$ (or corresponding bi-

This research is partially supported by British Academy Fellowship for Researchers at Risk 2022

partite graph I) is given by condition pIl if and only if the equations of the following kind hold:

$$\begin{aligned}
&p_2 - l_2 = l_1 p_1,\\
&p_3 - l_3 = p_1 l_2,\\
&p_4 - l_4 = l_1 p_3,\\
&p_5 - l_5 = p_1 l_4,\\
&\dots\\
&p_n - l_n = p_1 l_{n-1} \text{ for odd } n,\\
\text{or } &p_n - l_n = l_1 p_{n-1} \text{ for even } n.
\end{aligned} \tag{1}$$

We can consider an infinite bipartite graph $A(K)$ with points $(p_1, p_2, \dots, p_n, \dots)$ and lines $[l_1, l_2, \dots, l_n, \dots]$. We proved that for each odd n girth indicator of $A(n, K)$ is at least $2n + 2$.

Another incidence relation $I = D(n, K)$ is defined below. The following interpretation of a family of graphs $D(n, K)$ in case of general commutative ring K is convenient for the computations. Let us use the same notations for points and lines as in previous case of graphs $A(n, K)$. Points and lines are elements of two copies of the affine space over K. Point $(p_1, p_2, \dots, p_n)$ is incident with the line $[l_1, l_2, \dots, l_n]$ if the following relations between their coordinates hold:

$$\begin{aligned}
&p_2 - l_2 = l_1 p_1,\\
&p_3 - l_3 = p_1 l_2,\\
&p_4 - l_4 = l_1 p_3,\\
&\dots\\
&l_i - p_i = p_1 l_{i-2} \text{ if } i \text{ congruent to 2 or 3 modulo 4},\\
\text{or } &l_i - p_i = l_1 p_{i-2} \text{ if } i \text{ congruent to 1 or 0 modulo 4}.
\end{aligned} \tag{2}$$

Let $\Gamma(n, K)$ be one of graphs $D(n, K)$ or $A(n, K)$. The graph $\Gamma(n, K)$ has so called linguistic colouring ρ of the set of vertices. We assume that $\rho(x_1, x_2, \dots, x_n) = x_1$ for the vertex x (point or line) given by the tuple with coordinates $x_1, x_2, \dots, x_n$. We refer to x_1 from K as the colour of vertex x. It is easy to see that each vertex has a unique neighbour of the chosen colour. It means that the path in this graph is uniquely determined by initial vertex and the sequence of colours of the vertices. Let N_a and J_a be operators of taking the neighbour with colour a and jump operator changing the original colour of point or line for new colour a from K.

Let $[y_1, y_2, \dots, y_n]$ be the line y of $\Gamma(n, K[y_1, y_2, \dots, y_n])$ and $(\alpha(1), \alpha(2), \dots, \alpha(t))$ and $(\beta(1), \beta(2), \dots, \beta(t))$ are the sequences of colours from $K[y_1]$ of the length at least 2. We consider the sequence ${}^0v = y, {}^1v = J_{\alpha(1)}({}^0v), {}^2v = N_{\beta(1)}({}^1v), {}^3v = N_{\alpha(2)}({}^2v), {}^4v = N_{\beta(2)}({}^3v), \dots, {}^{2t-2}v = N_{\beta(t-1)}({}^{2t-3}v), {}^{2t-1}v = N_{\alpha(t)}({}^{2t-2}v), {}^{2t}v = J_{\beta(t)}({}^{2t-1}v)$. Assume that $v ={}^{2t} v = [v_1, v_2, \dots, v_n]$ where v_i are from $K[y_1, y_2, \dots, y_n]$. We consider polynomial transformation $g(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$, $t \geq 2$ of affine space K_n of kind $y_1 \to y_1 + \beta(t), y_2 \to v_2(y_1, y_2), y_3 \to v_3(y_1, y_2, y_3), \dots, y_n \to v_n(y_1, y_2, \dots, y_n)$.

It is easy to see that:

$$\begin{aligned}
&g(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))\cdot\\
&\cdot g(\gamma(1), \gamma(2), \dots, \gamma(s), \sigma(1), \sigma(2), \dots, \sigma(t)) =\\
&= g(\alpha(1), \alpha(2), \dots, \alpha(t), \gamma(1)(\beta(t)), \gamma(2)(\beta(t)), \dots,\\
&\gamma(s)(\beta(t)), \beta(1), \beta(2), \dots, \beta(s), \sigma(1)(\beta(t)),\\
&\sigma(2)(\beta(t)), \dots, \sigma(s)(\beta(t))).
\end{aligned}$$

Proposition II.1. *[11] Transformations of kind* $g = g(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$*,* $t \geq 2$ *generate a semigroup* $S(\Gamma(n, K))$ *of transformations of* K_n*.*

Lemma II.1. *[11] The degree of transformation* g *of the II.1 is at least* $[\deg(\alpha(1)) + \deg(\alpha(1) - \alpha(2)) + \deg(\alpha(2) - \alpha(3)) + \dots + \deg((\alpha(t-1) - \alpha(t))] + [\deg(\beta(1) + (\deg(\beta(1) - \beta(2)) + (\deg(\beta(2) - \beta(3)) + \dots + (\deg(\beta(t-2) - \beta(t-1))]$.

Lemma II.2. *[11] Transformation* g *as in the II.1 is bijective if and only if* $\beta(t)(x) = a$ *has a unique solution for each* a *from* K*.*

Proposition II.2. *[11] Transformations of kind* ${}^n g = g(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$*,* $t \geq 2$ *such that* $\deg(\alpha(i)) = 0$ *and* $\beta(i) = y_1 + c(i)$*,* $c(i) \in K$ *generate a subgroup* ${}^2G(\Gamma(n, K))$ *of transformation of maximal degree* 2*.*

Remark II.1. *The inverse element of* ${}^n g = g(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$*,* $t \geq 2$ *as in the II.2 can be written as* ${}^n g(\alpha(t), \alpha(t-1), \dots, \alpha(1), \beta(t-1)(\beta(t) - 1), \beta(t-2)(\beta(t)^{-1}, \dots, \beta(1)(\beta(t)^{-1}), \beta(t)^{-1})$.

Remark II.2. *In the case of two quadratic transformations of* K_n *of "general position" their composition will have degree* 4*.*

We associate with the sequence $\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t)$ of II.2 another quadratic transformation $h = H(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$ constructed via the sequence of vertices ${}^0v, {}^1v, {}^2v, \dots, {}^{2t-2}v = N_{\beta\times(t-1)}({}^{2t-3}v), {}^{2t-1}v = N_\alpha(t)({}^{2t-2}v)$. We compute ${}^{2t}v = J_{a(t)}({}^{2t-1}v) = v$ where $a(t) = (y_1)^2 + \beta(t)$ and define h as the quadratic map $y_i \to v_i$, $i = 1, 2, \dots, n$.

Theorem II.1. *(see [26], [11]) Let* K *be the finite field* F_q*,* $q = 2^r$*. Then transformation* $h = h(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$ *is a quadratic transformation of the vector space* $(F_q)^n$*. The polynomial degree of its inverse transformation is at least* 2^{r-1}*.*

Let us consider the linear projection $\tau : K_n + d \to K_n$ of deleting last d coordinates of the tuple.

The map $(p) \to (\tau(p)), [l] \to [\tau(l)]$ is an automorphism of the graph $\Gamma(n + d, K)$ onto $\Gamma(n, K)$. It induces the homomorphism θ of $S(\Gamma(n + d, K))$ onto $S(\Gamma(n, K))$ such that $\theta({}^2G(\Gamma(n + d, K)) = {}^2G(\Gamma(n, K))$.

Tame Homomorphism (TH) protocol (see [14]).

Alice selects ring K of kind F_q or Z_q where q is a prime power > 2 , parameters n and d, $d > 3$. She takes tuples of elements of K of kind $a(t_i) = ({}^i\alpha(1), {}^i\alpha(2), \dots, {}^i\alpha(t_i))$ and $b(t_i) = ({}^ib(1), {}^ib(2), \dots, {}^ib(t_i))$, $i = 1, 2, \dots, t$, $t \geq 2$ such that ${}^i\alpha(j) \neq {}^i\alpha(j + 1)$ and ${}^ib(j) \neq {}^ib(j + 1)$, $j = 1, 2, \dots, t_{i-1}$ together with affine transformation T from $AGL_{n+d}(F_q)$ and Y from $AGL_n(F_q)$.

Alice computes the standard forms of elements $a_i = T^{n+d}g({}^i\alpha(1), {}^i\alpha(2), \dots, {}^i\alpha(t_i), y_1 + {}^ib(1), y_1 + {}^ib(2), \dots y_1 + {}^ib(t_i))T^{-1}$ and $b_i = Y^n g({}^i\alpha(1), {}^i\alpha(2), \dots, {}^i\alpha(t_i), y_1 + {}^ib(1), y_1 + {}^ib(2), \dots y_1 + {}^ib(t_i))Y^{-1}$. She sends pairs (a_i, b_i), $i = 1, 2, \dots, t$ to Bob. Bob writes word $w(z_1, z_2, \dots, z_t)$ in formal alphabet $z_1, z_2, \dots, z_t$ of length at least t which uses each letter z_i. He computes the specialisations $w_A = w(a_1, a_2, \dots, a_t)$ and $c = w(a_1, a_2, \dots, a_t)$ in the groups of polynomial transformations of vector spaces $K^{n+d}$and K^n. Bob sends w_A to Alice and keeps c for himself. Alice computes $T^{-1}w_AT = {}^1c$, uses the homomorphism θ for getting $\theta({}^1c) = {}^2c$. She computes the collision map as Y^2cY^{-1}. Noteworthy that c is a quadratic map from the group of kind $y_1 \to c_1(y_1, y_2, \dots, y_n)$, $y_2 \to c_2(y_1, y_2, \dots, y_n)$,$\dots$, $y_n \to c_n(y_1, y_2, \dots, y_n)$.

Remark II.3. *Adversary has to decompose the standard form w_A into the word in the alphabet of generators $a_1, a_2, \dots, a_t$. Solution of this task in a polynomial time even with usage of Quantum Computer is unknown. So this is NP hard problem of Postquantum Cryptography.*

Remark II.4. *The complexity is determined by the complexity of computation of composition of two polynomial maps of degree 2 written in their standard forms. It is $O(n^7)$.*

Inverse TH protocol (see [14])

Alice selects the same data as in presented above protocol. She computes the standard forms of elements $a_i = T^{n+d}g({}^i\alpha(1), {}^i\alpha(2), \dots, {}^i\alpha(t_i), y_1 + {}^ib(1), y_1 + {}^ib(2), \dots y_1 + {}^ib(t_i))T^{-1}$. Instead of b_i Alice computes their inverses $c_i = b_i^{-1}$ and sends pairs (a_i, c_i) to Bob. He selects $j(1), j(2), \dots, j(r)$, $1 \le j(i) \le t$ and forms $w_A = a_{j(1)}a_{j(2)} \dots a_{j(r)}$ for Alice. Bob keeps $b = c_{j(r)}c_{j(r-1)}, \dots, c_{j(1)}$ for himself. Alice computes $T^{-1}w_AT = {}^1c$, uses the homomorphism θ for getting $\theta({}^1c) = {}^2c$. She computes the element a as Y^2cY^{-1}. It is easy to see that a and b are mutually inverse quadratic transformations of K^n.

Remark II.5. *Correspondents can use the protocol as a cryptosystem working with plaintexts from K_n. Alice can convert her message x to ciphertext $a(x) = y$. Bob decrypts y via the usage of his quadratic map b. After the usage of up to $[n^2/2]$ sessions they renovate their encryption/decryption tools via the new session of the inverse TH protocol.*

III. Cryptosystems with Quadratic Multivariate Rules

A. On the public key over F_q and its temporal form

Alice selects finite field F_q, $q = 2^r$, dimension n of the vector space over F_q, ${}^1T$ and ${}^2T$ from $AGL_n(F_q)$ defined by matrices with most entries distinct from zero.

She chooses parameter $t = O(n)$, elements $\alpha(1), \alpha(2), \dots, \alpha(t)$, $\beta(1), \beta(2), \dots, \beta(t)$ for which $\alpha(i) \ne \alpha(i)$, $\beta(i) \ne \beta(i+1)$, $i = 1, 2, \dots, n$ and compute the standard form of $F = {}^1Th(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))^2T$. She presents F of kind $y_i \to f(y_1, y_2, \dots, y_n)$, $i = 1, 2, \dots, n$ as public map. Public user Bob use this transformation to encrypt his plaintext p in time $O(n^3)$. Alice knows the decomposition ${}^1Th^2T$ and sequences $\alpha(i)$ and $\beta(i)$, $i = 1, 2, \dots, t$. It allows her to decrypt in time $O(n^2)$.

Remark III.1. *II.1 insures that multivariate map ${}^1Th^2T$ has inverse of polynomial degree at least 2^{r-1}. So if $r \ge 16$ then the cryptosystem is resistant to a differential linearisation attacks. We implement the case with $r = 32$. We suggest this classical type multivariate public key as the object for standardisation studies.*

Remark III.2. ***Temporal** TH **public rule.** Alice creates bijective F according presented above method. Together with Bob she executes TH protocol to elaborate the collision map and sends C+F to his partner. So correspondents can use "public key rule" F in a private mode. The usage of F just $t(n) = [n^2/2]$ times for the message encryption or electronic signatures times does not allow adversary to make the restoration of F. After the exchange of $t(n)$ vectors correspondents can start the new session.*

B. On temporal multivariate public rules

Correspondents can execute the inverse TH protocol and get mutually inverse outputs a and b acting on the vector space. Alice generates the quadratic map F as it described in unit 3.1 with ${}^1T = Y$. She sends the composition Y of a and H to Bob. He restores F as bY. They can make $O(1)$ sessions of the inverse protocol and get several outputs ${}^1a, {}^2a, \dots, {}^sa$ and ${}^1b, {}^2b, \dots, {}^sb$. After that Alice or Bob can renovate their initial public key F via the following procedure. One of correspondents sends the the word $(i(1), i(2), \dots., i(t))$, $1 \le i(k) \le s$ to his/her partner. Bob uses $b^{i(t)}b^{i(t-1)} \dots b^{i(1)}F$ for the encryption. Alice gets $b^{i(t)}b^{i(t-1)} \dots b^{i(1)}F(p) = c$ from Bob. She computes $a^{i(1)}a^{i(2)} \dots a^{i(t)}(c) = d$ and solves the equation $F(x) = d$ with the usage of her knowledge on $\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))$ and affine transformations ${}^1T$ and ${}^2T$ of degree 1. Noteworthy that correspondents do not need to compute compositions of generators ${}^ia$ or ${}^ib$, they will apply them consecutively.

C. Modification with direct TH protocol

Correspondents can use s-times direct TH protocol with outputs ${}^1c, {}^2c, \dots, {}^sc$. Alice computes the standard form of kind $g_i = Yg(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))Y^{-1}$, $i = 1, 2, \dots, s$ from $Y^2G(\Gamma(n, K)Y^{-1}$ and sends $c^i + g_i$ to Bob. Bob restores g_i in their standard forms. After the agreement on the word $(i(1), i(2), \dots, i(t))$, $1 \le i(k) \le s$ via open channel he encrypts with the consecutive usage of $g_{i(1)}, g_{i(2)}, \dots, g_{i(s)}$ and F. Recommended period of usage of words is $[n^2/2]$. It does not allow adversary to approximate the quadratic encryption transformation.

D. Remark on the implementation

We use computer simulation to generate maps of kind $y = \tau_1 h = h(\alpha(1), \alpha(2), \dots, \alpha(t), \beta(1), \beta(2), \dots, \beta(t))\tau_2(x)$

related to graphs $A(n, K)$ and $D(n, K)$. K is one of the commutative rings: Boolean ring $B(32)$, modular ring Z_2^{32} and finite field F_2^{32}. We have implemented three cases of invertible affine transformations. Tables and figures presenting simulation in all cases for F_2^{32} can be found in extended reprint version of this paper. The third case is presented in the following table.

1) τ_1 and τ_2 are identities, its just evaluation of time execution of core quadratic transformation,
2) τ_1 and τ_2 are of kind $x_1 \to x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n$ (linear time of computing execution of τ_1 and τ_2),
3) $\tau_1 = A_1x + b_1$ and $\tau_2 = A_2x + b_2$, nonsingular matrices A_1, A_2 have nonzero entries and vectors b_1, b_2 with mostly all coordinates differ from zero standard forms of the maps in the cases 2 and 3.

The program is written in C++ and compiled with the gcc compiler. We used an average PC with processor Pentium 3.00 GHz, 2GB memory RAM and system Windows 7. Table I present the time of encryption with symmetric algorithm for commutative ring $F_{2^{32}}$.

IV. Trees of infinite forest $D(F_q)$ and obfuscations of quadratic multivariate rules

We suggest modification quadratic $D(n, K)$ transformations presented before which is based on the descriptions of the connected components of these graphs. The description uses the following alternative definition of them.

The family of graphs $D(n, K)$, $n = 2, 3, \ldots$ where K is arbitrary commutative ring defines the projective limit $D(K)$ with points

$$\begin{aligned}(p) = (&p_{10}, p_{11}, p_{12}, p_{21}, p_{22}, p'_{22}, \ldots,\\ &p'_{ii}, p_{i,i+1}, p_{i+1,i}, p_{i+1,i+1}, \ldots),\end{aligned} \tag{3}$$

and lines

$$\begin{aligned}[l] = [&l_{01}, l_{11}, l_{12}, l_{21}, l_{22}, l'_{22}, \ldots,\\ &l'_{ii}, l_{ii+1}, l_{i+1,i}, l_{i+1,i+1}, \ldots].\end{aligned} \tag{4}$$

which can be thought as infinite sequences of elements in K such that only finitely many components are nonzero.

A point (p) of this incidence structure I is incident with a line $[l]$, i.e. $(p)I[l]$, if their coordinates obey the following relations:

$$\begin{aligned} p_{i,i} - l_{i,i} &= p_{1,0}l_{i-1,i},\\ p'_{i,i} - l'_{i,i} &= p_{i,i-1}l_{0,1},\\ p_{i,i+1} - l_{i,i+1} &= p_{i,i}l_{0,1},\\ p_{i+1,i} - l_{i+1,i} &= p_{1,0}l'_{i,i}, \end{aligned} \tag{5}$$

These four relations are well defined for $i > 1$, $p_{1,1} = p'_{1,1}$, $l_{1,1} = l'_{1,1}$.

Let D be the list of indices of the point of the graph $D(K)$ written in their natural order, i. e. sequence $(1, 0), (1, 1), (1, 2), (2, 1), (2, 2), (2, 2)' \ldots$. Let ${}^kD$ be the list of k first elements of D. The procedure of deleting coordinates of points and lines of $D(k, K)$ indexed by elements of $D - {}^kD$ defines the homomorphism of $D(K)$ onto graph $D(k, K)$ with the partition sets isomorphic to the variety K^n and defined by the first $k - 1$ equations from the list (5).

Let $k \geq 6$, $t = [(k + 2)/4]$, and let $u = (u_i, u_{11}, \ldots, u_{tt}, u'_{tt}, u_{t,t+1}, u_{t+1,t}, \ldots)$ be a vertex of $D(k, K)$. We assume that $u_1 = u_{1,0}$ $(u_{0,1})$ if u be a point (a line, respectively). It does not matter whether u is a point or a line. For every r, $2 \leq r \leq t$, let $a_r = a_r(u) = \Sigma_{i=0,r}(u_{ii}u'_{r-i,r-i} - u_{i,i+1}u_{r-i,r-i-1})$ and $a = a(u) = (a_2, a_3, \ldots, a_t)$.

The following statement was proved in [17] for the case $K = F_q$. Its generalization on arbitrary commutative rings is straightforward, see [18].

Proposition IV.1. *Let K be a commutative ring with unity and u and v be vertices from the same connected component of $D(k, K)$. Then $a(u) = a(v)$. Moreover, for any $t - 1$ ring elements $x_i \in K$, $2 \leq i \leq \lfloor (k + 2)/4 \rfloor = t$, there exists a vertex v of $D(k, K)$ for which $a(v) = (x_2, x_3, \ldots, x_t) = (x)$.*

So the classes of equivalence for the relation $\tau = \{(u, v) \mid a(u) = a(v)\}$ on the vertices of the graph $D(n, K)$ are unions of connected components.

Theorem IV.1. *[18] For each commutative ring with unity, the graph $D(k, K)$ is edge transitive.*

Equivalences classes of τ form an imprimitivity systems of automorphism group of $D(k, K)$. Graph $C(n, K)$ was introduced in [9] as the restriction of incidence relation of $D(k, K)$ on a solution set of system of homogeneous equations $a_2(x) = 0$, $a_3(x) = 0$, $\ldots$, $a_t(x) = 0$. The dimension of this algebraic variety is $n - t = d$. Thus $d = [4/3n] + 1$ for $n = 0, 2, 3$ mod 4, $d = [4/3n] + 2$ for $n = 1$ mod 4. For convenience we assume that $C(n, K) = C_d(K)$. Symbol $CD(k, K)$ stands for the connected component of graph $D(k, K)$. The following statement holds.

Theorem IV.2. *(see [11] and further references).*

The diameter of the graph $C_m(K)$, $m \geq 2$, K is a commutative ring with unity of odd characteristic, is bounded by parameter $f(m)$ which does not depend on K.

Corollary IV.1. *If K is a commutative ring with unity of odd characteristics then $CD(n, K) = C(n, K)$.*

Let us rename coordinates $y_{1,0}, y_{1,1}, y_{1,2}, y_{2,1}, \ldots$ of symbolic line y of $D(n, K)$ accordingly to the natural order on them as $y_1, y_2, \ldots, y_n$ and write equations of the graph in the form 5. It allows as to write connectivity invariants of the line $y = [y_1, y_2, \ldots, y_n]$ as $a_i([y]) = a_i(y_1, y_2, \ldots, y_n)$ where $i = 2, 3, \ldots, t$. Similar notations we will use in the case of points. For the nonlinear map F of K^n with bounded degree given in its standard form we define trapdoor accelerator $F = {}^1TG_A{}^2T$ as the triple ${}^1T$, ${}^2T$, G_A of transformations of K^n, where ${}^iT$, $i = 1, 2$ are elements of $AGL_n(K)$, $G = G_A$ is nonlinear map on K^n depending on the piece of information A which allows to compute the reimage for nonlinear G in time $O(n^2)$ (see [20]). In this paper we assume that A is

TABLE I
Generation time for the map (ms) $D(n, F_{2^{32}})$,length of the word, case III

n	**16**	**32**	**64**	**128**	**256**
16	71	136	263	518	1030
32	1220	2324	4535	8962	17824
64	21884	40412	77476	151587	299839
128	453793	812136	152678	2946017	5792884

given as a tuple of characters $(d(1), d(2), \ldots, d(m))$ in the alphabet K.

We use graphs $D(n, K)$ and $D(n, K[y_1, y_2, \ldots, y_n])$ to define family of quadratic multivariate maps F of kind $y_1 \to f_1(y_1, y_2, \ldots, y_n)$, $y_2 \to f_2(y_1, y_2, \ldots, y_n)$, $\ldots$, $y_n \to f_n(y_1, y_2, \ldots, y_n)$ with trapdoor accelerator $F = T_1 G_A T_2$, $T_1, T_2 \in AGL_n(K)$.

We take the line $[y_1, y_2, \ldots, y_n]$ of the graph $D(n, K[y_1, y_2, \ldots, y_n]$ for the colour α_1 from K we compute $[z] = J_{\alpha_1}([y]) = [\alpha_1 y_1, y_2, \ldots, y_n] = [z_1, z_2, \ldots, z_n]$ and compute $a_r = a_r([z]) = a_r(\alpha_1, y_2, \ldots, y_n)$, for $r = 2, 3, \ldots$. We form the quadratic expression $B = (y_1^s + C(y_2, y_3, \ldots, y_n)$ where $C(y_2, y_3, \ldots, y_n) = \lambda_2 a_2 + \lambda_3 a_3 + \cdots + \lambda_t a_t + \lambda_1$ with nonzero λ_i from K and $s = 2$ if the order of K^* is odd and $s = 1$ in all other cases. We form the walk in the graph $D(n, K[y_1, y_2, \ldots, y_n])$ starting from the line $[z]$ of colour α_1 and consecutive vertices of colours $y_1 + \beta_1, \alpha_2, y_1 + \beta_1, \alpha_3, \ldots, \alpha_{l-1}, y_1 + \beta l - 1, \alpha_l$ such that $\alpha_i \neq \alpha_{i+1}$, $\beta_i \neq \beta_{i+1}$ for $i = 1, 2, \ldots, l-1$.

We form the path with the starting line $v_1 = J_{\alpha_1}([y])$, $v_2 = N_{y_1+\beta_1}(v_1)$, $v_3 = N_{\alpha_2}(v_2)$, $\ldots$, $v_{2t-1} = N_{\alpha_t}(v_{2t-2}$ and consider $v_t = J_B(v_{2t-1}) = u$. The vertex u allows us to define the following transformation $G = G_A$, $A = (\alpha_1, \alpha_2, \ldots, \alpha_l; \beta_1, \beta_2, \ldots, \beta_{l-1}, B(y_1, y_2, \ldots, y_n))$ of K^n to itself

$y_1 \to (y_1)^s + C(y_1), y_2, \ldots, y_n)$,
$y_2 \to u_2(y_1, y_2)$,
$\ldots$
$y_n \to u_2(y_1, y_2, \ldots, y_n)$.

We identify $A = {}^1A$ with the array $(\alpha_1, \alpha_2, \ldots, \alpha_l; \beta_1, \beta_2, \ldots, \beta_{l-1}, \lambda_1, \lambda_2, \lambda_r, B(y_1, y_2, \ldots, y_n))$

Proposition IV.2. *Let T_1 and T_2 are bijective transformations from $AGL_n(K)$ and K is arbitrary commutative ring with unity. Then the standard form of $F = T_1 G_{l^A} T_2$, $l = O(n)$ has a trapdoor accelerator given by coefficients of T_1 and T_2 together with the array A described above.*

Proof. We have to justify that the reimage x of $v = G_A(x)$ can be computed in time $O(n^2)$. The procedure of its computation is the following:

1) Let the value v of G_A is given. We have to compute the connectivity invariants $a_2(u)$, $a_3(u)$, $\ldots$, $a_r(u)$ of the line $u = [\alpha_l, v_2, v_3, \ldots, v_n]$.
2) The computation of linear combination $b = \lambda_2 a_2(u) + \lambda_3 a_3(u) + \cdots + \lambda_r a_r(u) + \lambda_1$.
3) The computation of the solution $y_1 = c$ of the equation ${y_1}^2 + b = v_1$.
4) We form the parameters $d_1 = c + \beta_{l-1}$, $d_2 = \alpha_{l-1}$, $d_3 = c + \beta_{l-2}$, $d_4 = \alpha_{l-2}$, $\ldots$, $d_{2l-2} = \alpha_1$, of "reverse path" with the starting line $[u]$.
5) Conducting recurrent computations $N_{d_1}(u) = {}^1u$, $N_{d_2}({}^2u)$, $\ldots$, , $N_{d_{2l-1}}({}^{2l-2}u)$.
6) Computing of the reimage $J_c({}^{2l-2}u)$. The complexity of the algorithm is $O(n^2)$. So the map has a trapdoor accelerator.

The standard forms of transformations $F = T_1 G_A T_2$ can be used as a public keys. In fact this family is an obfuscation of quadratic multivariate public keys suggested in [15].

The idea of $D(n, K)$ based encryption with the usage of connectivity invariants was suggested in [16].

□

V. Conclusion

Multivariate Cryptography in wide sense is about constructions and investigations of Public Keys in a form of nonlinear Multivariate rule defined over some finite commutative ring K. These rule F has to be written as transformation $x_i \to f_i$, $i = 1, 2, \ldots, n$, $f_i \in K[x_1, x_2, \ldots, x_n]$ over commutative ring K. Bijective F can be used for the encryption of tuples (plaintexts) from the affine space K^n. Multivariate rules can serve as instruments for creation of digital signatures. In the case of bijective transformation decryption process can be thought as application of inverse rule G. The degree of G can be defined as maximum of degrees of polynomials $G(x_i)$, $i = 1, 2, \ldots, n$. For the usage of given publicly F as efficient and secure instrument its degree of has to be bounded by some constant c (traditionally $c = 2$) but the polynomial degree of the inverse G has to be high.

The key owner (Alice) suppose to have some additional piece T of private information about pair (F, G) to decrypt ciphertext obtained from the public user (Bob). Recall that family the family F_n, $n = 2, 3, \ldots$ has trapdoor accelerator ${}^nT$ if the knowledge of the piece of information ${}^nT$ allows to compute reimage x of $y = F_n(x)$ in time $O(n^2)$. Of course the concept of trapdoor accelerator is just instrument to search for practical trapdoor functions. As you know that the existence of theoretical trapdoor functions is just a conjecture. In fact it is closely connected to Main Conjecture of Cryptography about the fact that $P \neq NP$. Without the knowledge of T_n one has to solve nonlinear system of equations which generally is NP-hard problem. Finding of the inverse for F_n is an NP-hard problem if these maps are in so called "general position".

In the case of specific maps additional argumentation of the complexity to find inverses G_n can be useful.

We present such heuristic arguments in the case of $D(n,K)$ based encryption defined for arbitrary commutative ring K with unity with at least 3 elements and presented in previous section. Graphs $D(n,K)$ have partition sets K^n (set of points and set of lines) and incidence relation between points and lines is given by system of linear equations over K.

To define trapdoor accelerator for standard forms F_n, $n = 2, 3, \ldots$ we use special walks on graphs $(D(n,K)$ and and $D(n, K[x_1, x_2, \ldots, x_n])$.The constructed map F_n acts on the selected partition set K^n. In the case of trivial affine transformations T_1 and T_2 the relation $F_n(x) = y$ for $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ vertices x and $y = (f(y_1, y_2, \ldots, y_n), y_2, y_3, \ldots, y_n)$ are joint in the graph $D(n,K)$ by the path of length $> cn$, where c is positive constant and $f \in K[y_1, y_2, \ldots, y_n]$ is known quadratic expression. Finding the path will give us the trapdoor accelerator for the computation of preimages. This can be done by Dijkstra algorithm of complexity $v \log(v)$ where v is the order of graphs. It could not be done in polynomial time because of $v = 2|K|^n$ and $|K| \geq 3$. Noteworthy that the usage of nontrivial T_1 and T_2 will complicate the cryptanalysis.

We presented $D(n,K)$ based platform $H(n,K)$ of quadratic transformations. So correspondents Alice and Bob can use $H(n,K)$ protocols and elaborate collision map C, $C \in H(n,K)$. So Alice can create F_n and send $C + F_n$ to Bob instead of public announcement of this multivariate transformation. It gives the option to change the encryption tool periodically.

Alternatively Alice and Bob use the inverse $H(n,K)$ protocol to elaborate mutually inverse elements H and H^{-1} in their possessions. So Bob can change the rule F_n for the quadratic $H^{-1}F_n$ via left multiplication. These actions form a basis for algorithms with temporal public rules presented in the paper.

References

[1] A. Canteaut and F. X. Standaert (Eds.), *40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Eurocrypt 2021, LNCS 12696, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I, Springer, 2021, 839p, https://doi.org/10.1007/978-3-030-77886-6

[2] J. Ding, J. Deaton, , Vishakha, B.-Y. Yang, *The nested subset differential attack a practical direct attack against LUOV which forges a signature within 210 minutes*, Eurocrypt 2021, Part 1, pp. 329-347, https://doi.org/10.1007/978-3-030-77870-5_12

[3] W. Beullens, *Improved cryptanalysis of UOV and Rainbow*, Eurocrypt 2021, Part 1, pp. 348-373, https://doi.org/10.1007/978-3-030-77870-5_13

[4] L. Goubin, J. Patarin, B.-Y. Yang, *Multivariate cryptography, Encyclopedia of Cryptography and Security* (2nd Ed.) 2011, pp. 824-828, https://doi.org/10-.1007/978-1-4419-5906-5_421

[5] J. Ding, A. Petzolt and D. S. Schmidt, *Multivariate public key cryptosystems*, Springer, ADIS, vol. 80, 2020, https://doi.org/10.1007/978-1-0716-0987-3_2

[6] N. Koblitz, *Algebraic aspects of cryptography*, Springer, 1998, p. 206, https://doi.org/10.1007/978-3-662-03642-6_1

[7] V. Ustimenko, *Linear codes of Schubert type and quadratic public keys of Multivariate Cryptography*, IACR e-print archive, 2023/175, https://eprint.iacr.org/2023/175

[8] F.Lazebnik , V. Ustimenko and A.J.Woldar, *A new series of dense graphs of high girth*, Bulletin of the AMS 32 (1) (1995), pp. 73-79, https://doi.org/10.1090/s0273-0979-1995-00569-0

[9] V. Ustimenko, *On the extremal graph theory and symbolic computations*, Dopovidi National Academy of Sci, Ukraine, 2013, No. 2, pp. 42-49.

[10] V. Ustimenko, M. Klisowski, *On Noncommutative Cryptography with cubical multivariate maps of predictable density*, In "Intelligent Computing", Proceedings of the 2019 Computing Conference, Volume 2, Part of Advances in Intelligent Systems and Computing(AISC), volume 99, pp. 654-674, https://doi.org/10.1007/978-3-030-22868-2_47

[11] V. Ustimenko, *Graphs in terms of Algebraic Geometry, symbolic computations and secure communications in Post-Quantum world*, University of Maria Curie Sklodowska Editorial House, Lublin, 2022, 198 p.

[12] A. G. Myasnikov, V. Shpilrain and A. Ushakov, *Non-commutative cryptography and complexity of group-theoretic problems*, American Mathematical Society, 2011, https://doi.org/10.1090/surv/177/05

[13] B. Tsaban, *Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography* , J. Cryptol. 28, No. 3 (2015), pp. 601-622, https://doi.org/10.1007/s00145-013-9170-9

[14] V. Ustimenko, *On new symbolic key exchange protocols and cryptosystems based on a hidden tame homomorphism*, Dopovidi National Academy of Sci, Ukraine, 2018, n10, pp. 26-36, https://doi.org/10.15407/dopovidi2018.10.026

[15] V. Ustimenko and A. Wroblewska, *Dynamical systems as the main instrument for the constructions of new quadratic families and their usage in cryptography*, Annales UMCS Informatica AI XII, 3 (2012) pp. 65–74, https://doi.org/10.2478/v10065-012-0030-2

[16] V. A. Ustimenko, *Graphs with special arcs and cryptography*, Acta Applicandae Mathematicae, vol. 71, N2, November 2002, pp. 117-153, https://doi.org/10.1023/a:1020686216463

[17] Lazebnik, F., Ustimenko, V.A. and A.J. Woldar, *A characterisation of the components of the graph $D(k,q)$*, Discrete Mathematics, 157 (1996), pp. 271-283, https://doi.org/10.1016/s0012-365x(96)83019-6

[18] V. Ustimenko, *Linguistic dynamical systems, graphs of large girth and cryptography*, Journal of Mathematical Sciences, Springer, vol.140, N3 (2007) pp. 461-471, https://doi.org/10.1007/s10958-007-0453-2

[19] Anshel, M. Anshel and D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res.Lett. 6(3–4), pp. 287–291 (1999), 169 https://doi.org/10.4310/mrl.1999.v6.n3.a3

[20] S.R. Blackburn and S.D. Galbraith, *Cryptanalysis of two cryptosystems based on group actions*, In: Advances in Cryptology—ASIACRYPT '99. Lecture Notes in Computer Science, vol. 1716, pp. 52–61. Springer, Berlin (1999), https://doi.org/10.1007/978-3-540-48000-6_6

[21] K.H. Ko, S.J. Lee, J.H. Cheon, J.W. Han, J.S. Kang and C. Park, *New public-key cryptosystem using braid groups*, In: Advances in Cryptology—CRYPTO 2000, Santa Barbara, CA. Lecture Notes in Computer Science, vol. 1880, pp. 166-183. Springer, Berlin (2000), https://doi.org/10.1007/3-540-44598-6_10

[22] G. Maze, C. Monico and J. Rosenthal, *Public key cryptography based on semigroup actions*, Adv.Math. Commun. 1(4), pp. 489–507 (2007), https://doi.org/10.3934/amc.2007.1.489

[23] P.H. Kropholler, S.J. Pride , W.A.M. Othman K.B. Wong and P.C. Wong, *Properties of certain semigroups and their potential as platforms for cryptosystems*, Semigroup Forum (2010) 81: pp. 172–186, https://doi.org/10.1007/s00233-010-9248-8

[24] J.A. Lopez Ramos, J. Rosenthal, D. Schipani and R. Schnyder, *Group key management based on semigroup actions*, Journal of Algebra and its applications, 2017, vol.16 (08):1750148, https://doi.org/10.1142/s0219498817501481

[25] G.Kumar and H. Saini, *Novel noncommutative cryptography scheme using extra special group*, Security and Communication Networks ,Volume 2017, Article ID 9036382, 21 pages, https://doi.org/10.1155/2017/9036382

[26] A. Wroblewska, *Linguistic dynamical systems based on algebraic graphs and their application in cryptography*, PhD Thesis, Institute of Fundamental Technological Research Polish Academy of Sciences, Warsaw, Poland, 2017, https://oldwww.ippt.pan.pl/_download/doktoraty/2016wroblewska_a_doktorat.pdf

[27] V. Ustimenko, A. Wroblewska, *Extremal algebraic graphs, quadratic multivariate public keys and temporal rules*, https://eprint.iacr.org/2023/738

DOI: 10.15439/2023F7763
ISSN 2300-5963 ACSIS, Vol. 35

On the implementations of new graph based cubic Multivariate Public Keys

Vasyl Ustimenko
0000-0002-2138-2357
Royal Holloway University of London
Institute of Telecommunication and Global
Information Space, Kyiv, Ukraine
Email: Vasyl.Ustymenko@rhul.ac.uk

Tymoteusz Chojecki, Michal Klisowski
0000-0002-3294-2794
0000-0002-2817-8404
Maria Curie-Sklodowska University,
Lublin, Poland.
Email: {tymoteusz.chojecki@umcs.pl, mklisow@hektor.umcs.lublin.pl}

***Abstract*—Algebraic Constructions of Extremal Graph Theory were efficiently used for the construction of Low Density Parity Check Codes for satellite communication, constructions of stream ciphers and Postquantum Protocols of Noncommutative cryptography and corresponding El Gamal type cryptosystems. We shortly observe some results in these applications and present idea of the usage of algebraic graphs for the development of Multivariate Public Keys (MPK). Some MPK schemes are presented at theoretical level, implementation of one of them is discussed. Extended version of this article is available online at [31].**

I. Introduction

EXTREMAL algebraic graphs were traditionally used for the construction of stream ciphers of multivariate nature (see [19], [8] and further references). We introduce the first graph based multivariate public keys with bijective encryption maps. We hope that new recent results on algebraic constructions of Extremal Graph Theory [16] will lead to many applications in Algebraic Cryptography which includes Multivariate cryptography and Noncommutative Cryptography. Some graph based algebraic asymmetrical algorithms will be presented in this paper.

NIST 2017 tender starts the standardisation process of possible Post-Quantum Public keys aimed for purposes to be (i) encryption tools, (ii) tools for digital signatures (see [28]).

In July 2020 the Third Round of the competition started. In the category of Multivariate Cryptography (MC) remaining candidates are easy to observe. For the task (i) multivariate algorithm was not selected, single multivariate candidate is ”The Rainbow Like Unbalanced Oil and Vinegar” (RUOV) digital signature method. As you see RUOV algorithm is investigated as appropriate instrument for the task (ii). During Third Round some cryptanalytic instruments to deal with ROUV were found (see [20] and further references]). That is why different algorithms were chosen at the final stage. In July 2022 first four winners of NIST standardisation competition were chosen. They all are lattice based algorithms. They all are not the algorithms of Multivariate Cryptography.

Noteworthy that all considered multivariate NIST candidates were presented by multivariate rule of degree bounded by constant (2 or 3) of kind

$x_1 \to f_1(x_1, x_2, \ldots, x_n)$,
$x_2 \to f_2(x_1, x_2, \ldots, x_n)$,
. . .,
$x_n \to f_n(x_1, x_2, \ldots, x_n)$.

Classical results of Multivariate Cryptography can find in [25], [26] and [27].

We think that NIST outcomes motivate investigations of alternative options in Multivariate Cryptography oriented on encryption tools for

(a) the work with the space of plaintexts ${F_q}^n$ and its transformation G of linear degree cn, $c > 0$ on the level of stream ciphers or public keys

(b) the usage of protocols of Noncommutative Cryptography with platforms of multivariate transformations for the secure elaboration of multivariate map G from $End(F_q[x_1, x_2, \ldots, x_n])$ of linear or superlinear degree and density bounded below by function of kind cn^r, where $c > 0$ and $r > 1$.

Some ideas in directions of (a) and (b) are presented in [17].

We hope that classical multivariate public key approach i. e. usage of multivariate rules of degree 2 or 3 is still able to bring reliable encryption algorithms. In this paper we suggest new cubic multivariate public rules.

Recall that the density is the number of all monomial terms in a standard form $x_i \to g_i(x_1, x_2, \ldots, x_n)$, $i = 1, 2, \ldots, n$ of multivariate map G, where polynomials g_i are given via the lists of monomial terms in the lexicographical order.

We use the known family of graphs $D(n, q)$ and $A(n.q)$ of increasing girth (see [1]-[6] and further references) and their analogs $D(n, K)$ and $A(n, K)$ defined over finite commutative ring K with unity for the construction of our public keys. Noteworthy to mention that for each prime power q, $q > 2$ graphs $D(n, q)$, $n = 2, 3, \ldots$ form a family of large girth (see [1]), there is well defined projective limit of these graphs which is a q-regular forest. in fact if K is an integral domain both families $A(n, K)$ and $D(n, K)$ are approximations of infinitedimensional algebraic forests. The definitions of such

This research is partially supported by British Academy Fellowship for Researchers at Risk 2022 and by UMCS program UMCS Mini-Grants.

approximations are given in Section 3 together with short survey of their applications.

In Section 2 we present the known mathematical definitions of algebraic geometry for further usage of them as instruments of Multivariate Cryptography. In particular definition of affine Cremona semigroup of endomorphisms of multivariate ring $K[x_1, x_2, \ldots, x_n]$ defined over commutative ring K and affine Cremona group ${}^nCG(K)$ are presented there.

The concept of *trapdoor accelerator* of the transformation from affine Cremona semigroup ${}^nCS(K)$ is presented there as a piece of information which allows computation of reimage of the map in time $O(n^2)$.

This is a weaker version of the definition of trapdoor one way function. The definition of the trapdoor accelerator is independent from the conjecture $P \neq NP$ of the Complexity theory. Section 2 also contains some statements on the existence of the trapdoor accelerator with the restrictions on the degrees on maps and their inverses for families of elements of the affine Cremona group ${}^nCG(K)$.

Section 3 is dedicated to infinite forests approximations and their connections with Algebraic Geometry and Extremal Graph Theory.

The description of linguistic graphs $D(n, K)$ and $A(n, K)$ and some their properties are presented in Section 4, 5. These sections contain the descriptions of subgroups and subsemigroups of ${}^nCS(K)$ defined via walks in graphs $D(n, K)$ and their extensions $D(n, K[x_1, x_2, \ldots, x_n])$ and graphs $A(n, K)$ and $A(n, K[x_1, x_2, \ldots, x_n])$ respectively. Some statements about degrees of elements of these semigroups are given.

Section 6 contains examples of cryptographic applications of graph based trapdoor accelerators in the form of cubic multivariate public key.

Detailed description of multivariate public key related to one of presented families is presented in in the Section 7.

Remarks on security level connected with girth studies of tree approximations reader can find in section 8. Last Section 9 presents short conclusions.

II. On elements of Algebraic Geometry and trapdoor accelerators

Let K be a commutative ring with a unity. We consider the ring $K' = K[x_1,$ $x_2, \ldots, x_n]$ of multivariate polynomials over K. Endomorphisms δ of K' can be given via the values of $\delta(x_i) = f_i(x_1, x_2, \ldots, x_n)$, $f_i \in K'$. They form the semigroup $End(K[x_1, x_2, \ldots, x_n]) = {}^nCS(K)$ of K' known also as affine Cremona semigroup named after the famous Luigi Cremona (see [29]). The map $\tilde{\delta} : (x_1, x_2, \ldots, x_n) \to (f_1(x_1, x_2, \ldots, x_n), f_2(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n))$ is polynomial transformation of affine space K^n. These transformations generate transformation semigroup $CS(K^n)$. Note that the kernel of homomorphism of ${}^nCS(K)$ to $CS(K^n)$ sending δ to $\tilde{\delta}$ depends on the choice of commutative ring K.

Affine Cremona Group ${}^nCG(K) = Aut(K[x_1, x_2, \ldots, x_n])$ acts bijectively on K^n. Noteworthy that some elements of ${}^nCS(K)$ can act bijectively on K^n but do not belong to ${}^nCG(K)$. For instance endomorphism $x \to x^3$ of $R[x]$ acts bijectively on set R of real number but the inverse $x \to x^{1/3}$ of this map is birational element outside of ${}^1CG(R)$.

Recall that degree of δ is the maximal degree of polynomials $\delta(x_i)$, $i = 1, 2, \ldots, n$. The density of δ is a total number of monomial terms in all $\delta(x_i)$.

Assume that automorphism F from ${}^nCG(K)$ has constant degree d, $d \geq 2$. It is given in its standard form written as $x_1 \to f_1(x_1, x_2, \ldots, x_n)$, $x_2 \to f_2(x_1, x_2, \ldots, x_n)$, …, $x_n \to f_n(x_1, x_2, \ldots, x_n)$ where f_i, $i = 1, 2, \ldots, n$ are elements of $K[x_1, x_2, \ldots, x_n]$ and used as public rule to encrypt plaintexts from K^n.

The following definition was motivated by the idea to have a weaker version of trapdoor one way function.

We say that family $F_n \in^n CG(K)$ of bijective nonlinear polynomial transformations of affine space K^n of degree ≤ 3 has *trapdoor accelerator* ${}^nT$ of level $\geq d$ if

(i) the knowledge of piece information ${}^nT$ ("trapdoor accelerator") allows to compute the reimage x for F_n in time $O(n^2)$

(ii) the degree of ${F_n}^{-1}$ is at least d, $d \geq 3$.

Notice that if F_n are given by their standard forms and degrees of ${F_n}^{-1}$ are equal to d then the inverse can be approximated in polynomial time $f(n, d) = O(n^{d^2+1})$ via linearisation technique. One can see that the approximation task becomes unfeasible if d is "sufficiently large" like $d = 100$. Examples of cubic families F_n with trapdoor accelerator of high level t are given in the case of special finite fields F_q in the section 3.

III. On algebraic forest approximations and their applications

We define thick forest as simple graph without cycles such that each of its vertex has degree at least 3. In probability theory branching process is a special stochastic process corresponding to a random walk on a thick forest. A genealogy of single vertex is a tree. One of the basic properties of finite tree is the existence of a leaf, i. e. vertex of degree 1. Thus each thick tree is an infinite simple graph.

Let K be a commutative ring and K^n be an affine space of dimension n over K (free module in other terminology). A subset M in K^n is an algebraic set over K if it is a solution set for the system of algebraic equations of kind $f = 0$ or inequalities of kind $g \neq 0$ where f and g are elements of $K[x_1, x_2, \ldots, x_n]$. There are several alternative approaches to define dimension of M. In the case when K is a field these approaches are equivalent and dimension of M can be computed with the usage of Groőbner basis technique (see [21], [22], [23]).

We say that graph Γ is algebraic over K if its vertex and edge sets are algebraic sets over K

We investigate a possibility to define thick forest F by system of equations over some commutative ring K, i.e. construct F as a projective limit of algebraic over K bipartite graphs Γ_i, $i = 1, 2, \ldots$. Noteworthy that the girth $g_i = g(\Gamma_i)$, which is the length of minimal cycle in Γ_i tends to infinity when i is growing. In this situation we refer to F as algebraic forest over K.

We say that the family Γ_i is an algebraic forest approximation over the ring K. In the case $g_i \geq cn_i$, where n_i are dimensions of the algebraic sets $V(\Gamma_i)$ of vertices of the graph Γ_i and c is some positive constant we use term *algebraic forest approximation of large girth.* Note that algebraic forest approximations of large girth over finite field F_q, $q > 2$ are *families of graphs of large girth* in sense of P. Erdős'(see [15] and further references). The first algebraic forest approximation of a large girth was introduced by F. Lazebnik and V. Ustimenko (see [1], [2]) in the case of $K = F_q$.

The properties of trees of this algebraic forest and their approximations over F_q were investigated in the paper[30].

In 1998 more general algebraic graphs $D(n, K)$ defined over arbitrary commutative ring K were introduced [4]. It was stated that a girth of $D(n, K)$ is $\geq n+5$ in the case of arbitrary integrity domain K. This inequality insures that $D(n, K)$, $n = 2, 3, \ldots$ is algebraic forest approximation of large girth. The prove of the inequality reader can find in [5], simpler prove of this fact the reader can find in [18].

Noteworthy that in the case of integrity domain K together with $D(n, K)$, $n = 2, 3, \ldots$ one can consider another thick forest approximation $D(n, K[x_1,$
$x_2, \ldots, x_m])$ for each parameter m. Thus paper [5] opened a possibility to use extremal properties of these graphs in the Theory of Symbolic Computations and its various applications to Cryptography.

The paths of even length t on trees and their approximations can be used to induce multivariate transformations on varieties P_i and L_i of points and lines of $V(\Gamma_i)$. These transformations can serve as encryption maps acting on the potentially infinite space P_i of plaintexts (see [7], [19], [8] and further references). They form a group $G_i = G(\Gamma_i)$ which can be a platform for the protocols of Noncommutative Cryptography (see [9]-[14]). Noteworthy that if t is at most half of the girth of Γ_i then different paths produce distinct transformations. So, forest approximations of large girth are preferable for cryptographic applications.

Other tree approximation over the integrity domain K is formed by graphs $A(n, K)$ defined in [6]. In fact these graphs were defined earlier [5] as homomorphic images $E(n, K)$ of graphs $D(n, K)$ or their connected components $CD(n, K)$. As it was stated recently in short paper [24] for each integrity domain K, $K \neq F_2$ graphs $A(n, K)$ form a tree approximation of large girth.

Some encryption algorithms (stream ciphers) based on $A(n, K)$ and $D(n, K)$ were already introduced (see [7], [19], [8], [16]).

IV. On Linguistic Graphs $A(n, K)$, Related Semigroups and Groups and Symmetric Ciphers

Regular algebraic graph $A(n, q) = A(n, F_q)$ is an important object of Extremal Graph Theory. In fact we can consider more general graphs $A(n, K)$ defined over arbitrary commutative ring K.

This graph is a bipartite graph with the point set $P = K^n$ and line set $L = K^n$ (two copies of Cartesian power of K are used). It is convenient to use brackets and parenthesis to distinguish tuples from P and L.

So, (p) $= (p_1, p_2, \ldots, p_n) \in P_n)$ and [l] $= [l_1, l_2, \ldots, l_n] \in L_n$. The incidence relation $I = A(n, K)$ (or corresponding bipartite graph I) is given by the following condition.

pIl if and only if the equations

$p_2 - l_2 = l_1 p_1, p_3 - l_3 = p_1 l_2, p_4 - l_4 = l_1 p_3, p_5 - l_5 = p_1 l_4,$ $\ldots$, $p_n - l_n = p_1 ln - 1$ hold for odd n and $p_n - l_n = l_1 p_{n-1}$ for even n.

In the case of $K = F_q$, $q > 2$ of odd characteristic graphs $A(n, F_q)$, $n > 1$ form a family of small world graphs because their diameter is bounded by linear function in variable n (see [6]).

Recall that the girth of the graph is the length of its minimal cycle. We can consider an infinite bipartite graph $A(K)$ with points $(p_1, p_2, \ldots, p_n, \ldots)$ and lines $[l_1, l_2, \ldots, l_n, \ldots]$ which is a projective limit of graphs $A(n, K)$ when n tends to infinity. If K, $|K| > 2$ is an integrity domain then $A(K)$ is a tree and the girth g_n of $A(n, K)$, $n = 2, 3, \ldots$ is bounded below by linear function cn for some positive constant c [24].

As a byproduct of this result we get that $A(n, q)$, $n = 2, 3, \ldots$ for each fixed q, $q > 2$ form a family of large girth in sense of Erdős'. In fact graphs $A(n, K)$ were obtained in [5] as homomorphism images of known graphs $CD(n, K)$ of large girth (see [1], [2], [3]).

Let K be a commutative ring with a unity. Graphs $A(n, K)$ belong to the class of linguitic graphs of type $(1, 1, n - 1)$ [19], i.e. bipartite graphs with partition sets $P = K^n$ (points of kind $(x_1, x_2, \ldots, x_n)$, $x_i \in K$) and $L = K^n$ (lines $[l_1, l_2, \ldots, l_n]$, $l_i \in K$) and incidence relation $I = I(n, K)$ such that $(x_1, x_2, \ldots, x_n)I[y_1, y_2, \ldots, y_n]$ if and only if $a_2 x_2 + b_2 x_2 = f_2(x_1, y_1)$, $a_3 x_3 + b_3 x_3 = f_3(x_1, x_2, y_1, y_2)$, $\ldots$, $a_n x_n + b_n x_n = f_n(x_1, x_2, \ldots, x_n)$, where a_i and b_i are elements of multiplicative group K^* of K and f_i are multivariate polynomials from $K[x_1, x_2, \ldots, x_{i-1}, y_1, y_2, \ldots, y_{i-1}]$ for $i = 2, 3, \ldots, n$.

The colour of $\rho(v)$ of vertex v of graph $I(K)$ is defined as x_1 for point $(x_1, x_2, \ldots, x_n)$ and y_1 for line $[y_1, y_2, \ldots, y_n]$.

The definition of linguistic graph insures that there is a unique neighbour with the chosen colour for each vertex of the graph. Thus we define operator $u = N_a(v)$ of taking neighbour u with colour a of the vertex v of the graph. Additionally we consider operator ${}^aC(v)$ of changing colour of vertex v, which moves point $(x_1, x_2, \ldots, x_n)$ to point $(a, x_2, x_3, \ldots, x_n)$ and line $[x_1, x_2, \ldots, x_n]$ to line $[a, x_2, x_3, \ldots, x_n]$.

Let us consider a walk $v, v_1, v_2, \ldots, v_{2s}$ of even length $2s$ in the linguistic graph $I(K)$. The information on the walk is given by v and the sequence of colours $\rho(v_i)$, $i = 1, 2, \dots, 2s$. The walk will not have edge repetitions if $\rho(v_2) \neq \rho(v)$, $\rho(v_i) \neq \rho(v_{i-2})$ for $i = 3, 4, \dots, n$. Notice that v and v_{2s} are elements of the same partition set (P or L). For each vertex v of $I(K)$ we consider a variety of *walks* with jumps, i. e. totality of sequences of kind v, $v_1 = {}^{a_1}C(v)$, $v_2 = N_{a_2}(v_1)$, $v_3 = {}^{a_3}C(v_2)$, $v_4 = N_{a_4}(v_3)$, …, $v_5 = {}^{a_5}C(v_4)$, …, $v_{4s} = N_{a_{4s}}(v_{4s-1})$, $v_{4s+1} = {}^{a_{4s+1}}C(v_{4s})$. Note that for each s , $s \geq 0$ vertices v, v_1, v_{4s}, v_{4s+1} are elements of the same partition. Let $u = (a_1, a_2, \dots, a_{4s}, a_{4s+1})$ be the colours of the walk with jumps.

We introduce the following polynomial transformations of partition sets P and L. Firstly we consider the pair of linguistic graphs $I(K)$ and $I(K[x_1,$
$x_2, \dots, x_n])$. These graphs are defined by the same equations with coefficients from the commutative ring K. We look at sequences of walks with jumps of length $4s+1$ where $s \geq 0$ starting in the point $v = (x_1, x_2, \dots, x_n)$ (or line $[x_1, x_2, \dots, x_n]$) of the graph $K[x_1, x_2, \dots, x_n]$ which uses colors $a_1(x_1)$, $a_2(x_1)$, …, $a_{4s+1}(x_1)$ from $K[x_1]$. The final vertex of this walk is v_{4s+1} with coordinates $a_{4s+1}(x_1)$, $f_2(x_1, x_2)$, $f_3(x_1, x_2, x_3)$, …, $f_n(x_1, x_2, \dots, x_n)$). Let us consider the transformations ${}^uT_P$ and ${}^uT_L$ sending starting vertex to the destination point of the walk with jumps acting via the rule $x_1 \to a_{4s+1}(x_1)$, $x_2 \to f_2(x_1, x_2)$, …, $x_n \to f_n(x_1, x_2, \dots, x_n)$ on the partition sets P and L isomorphic to K^n. It is easy to see that transformations of kind ${}^uT_P$ (or ${}^uT_L$) form the semigroup $LS_P(I(K))$ ($LS_L(I(K))$ respectively). We refer to this transformation semigroup as *linguistic semigroup* of graph $I(K)$.

Let us consider an algebraic formalism for the introduction of linguistic semigroups. We take the totality of words $F(K[x])$ in the alphabet $K[x]$ and define the product of $u = (a_1(x), a_2(x), \dots, a_k(x))$ and $w = (b_1(x), b_2(x), \dots, b_s(x))$ as word $= (a_1(x)$, $a_2(x)$, …, $a_k(x)) \times (b_1(x)$, $b_2(x)$, …, $b_t(x)) = (a_1(x)$, $a_2(x)$, …, $a_{k-1}(x)$, $b_1(a_k(x))$, $b_2(a_k(x))$, …, $b_t(a(x)))$.

Obtained semigroup $F(K[x])$ is slightly modified free product of $End(K[x])$ with itself. Note that we can identify $a(x)$ from $K[x]$ with the map $x \to a(x)$ from $End(K[x])$.

Let F_K be a subsemigroup of words of length of kind $4s+1$, $s \geq 0$.

PROPOSITION 1.

Let $I(K)$ be a linguistic graph defined over commutative ring K with unity. The map ${}^{I(K)}\eta_P : F_K \to End(K[x_1, x_2, \dots, x_n])$ such that ${}^{I(K)}\eta(u) ={}^u T_P$ (or $\eta(u)_L ={}^u T_L$) is a semigroup homomorphism.

It is easy to see that ${}^{I(K)}\eta_P(F_K) = LS_P(I(K)$ and ${}^{I(K)}\eta_L(F_K) = LS_L(I(K)$.

PROPOSITION 2. (see [19] and further references)

The image of $u = (a_1(x), a_2(x), \dots, a_k(x))$ from F_K under the map ${}^I(K)\eta_P$ (or ${}^I(K)\eta_P$ is invertible element of $LS_P(I(K)$ (or $LS_L(I(K)$ if and only if the map $x \to a_k(x)$ is an element of $\mathrm{Aut}(K[x])$.

REMARK 1.

The transformations $({}^{I(K)}\eta_P(u), P)$ and $({}^{I(K)}\eta_L(u), L)$ are bijective if and only if the map $x \to b(x)$ is bijective.

ILLUSTRATIVE EXAMPLE.

Let $K = R$ (real numbers) or K be algebraically closed field of characteristic 0 and $b(x) = x^3$. The inverse map for $x \to x^3$ is birational automorphism $x \to x^{1/3}$ of $K[x]$. Thus $g_P ={}^{I(K)} \eta_P(u)$ and $g_L^{I(K)}\eta_L(u)$ do not have inverses in $End(K[x])$. They have bijective birational inverses. Noteworthy that g_P and g_L are transformations of infinite order. Degree of polynomial transformations of ${g_P}^s$ and ${g_L}^s$ are at least 3^s.

So we have an algorithm of generation bijective polynomial maps of arbitrary large degree on variety K^n.

We refer to subgroups $G_P(I(K))$ and $G_L(I(K))$ of invertible elements of $LS_P(I(K))$ and $LS_L(I(K))$ as groups of linguistic graphs $I(K)$. They are different from automorphism group of $I(K)$.

Let us consider semigroup $\tilde{F}_K$ of words of kind $u = (x, f_1, f_1, f_2, \dots, f_s, f_s)$. It is easy to see that for each linguistic graph $I(K)$ the transformations $g_P(u) = {}^I(K)\eta_P(u)$ and ${g_L}^I(K)\eta_L(u)$ are computed via consecutive usage of N_{f_i} in the linguistic graph. Thus we refer to $SW_P(I(K) = \{g_P(u) | u \in \tilde{F}_K\}$ and $SW_L(I(K) = \{g_L(u) | u \in \tilde{F}_K\}$ as semigroups of symbolic walks on partition sets of $I(K)$. We refer to $GW_P(I(K) = SW_P(I(K) \cup G_P(I(K))$ and $GW_L(I(K) = SW_L(I(K) \cap G_L(I(K))$ as groups of symbolic walks.

Finally we consider the semigroup $St(K)$ of words $u = (x+\alpha_1, x+\alpha_2, \dots, x+\alpha_k)$ where α_i are elements of K. We consider $F_K = F_K \cap St_K$ $\tilde{F}_K = \tilde{F}_K \cap St_K = \Sigma_K$ and introduce groups ${}^{I(K)}\eta_P(F_K) = \tilde{H}_P(I(K))$, ${}^{I(K)|}\eta_P(F'_K) = \tilde{H}_P(I(K))$, ${}^{I(K)|}\eta_P(\Sigma_K) = H_P(I(K))$, ${}^{I(K)|}\eta_P(\Sigma_K) = H_P(I(K))$.

We can change set P for the line set L and introduce ${}^{I(K)|}\eta_L(\Sigma_K) = H_L(I(K))$.

We refer to groups $H_P(I(K))$, $H_L(I(K))$ as groups of walks on partition sets of linguistic graph $I(K)$.

PROPOSITION 3.

If a linguistic graph $I(K)$ is connected then groups $H_P(I(K))$ and $H_L(I(K))$ are acting transitively on K^n.

THEOREM 1. (see [19])

For each commutative ring K groups $H_P(A(n,K)) = GA(n,K)$ and $H_L(A(n,K)) = {}^*GA(n,K)$ are totalities of cubical automorphisms of $K[x_1, x_2, \dots, x_n]$.

COROLLARY 1.

Let us consider element $u = (x, x+a_1, x+a_1, x+a_2, x+a_2, \dots, x+a_{k-1}, x+a_{k-1}x+a_k, x^t$ of F_K for commutative ring K with unity with finite multiplicative group of order d, $d > 2$ where $t = 2$ or $t = 3$ and $(d,t) = 1$. Then transformation ${}^{A(n,K)}\eta(u)$ is a cubical one.

THEOREM 2. (see [19]). *For each commutative ring K groups $H_P(D(n,K)) = GD(n,K)$ are totalities of cubical automorphisms of $K[x_1, x_2, \dots, x_n]$.*

COROLLARY 2. *Let us consider element $u = (x, x+a_1, x+a_1, x+a_2, x+a_2, \dots, x+a_{k-1}, x+a_{k-1}, x+a_k, x^t)$ of F_K for commutative ring K with unity with finite multi-*

plicative group of order d, $d > 2$ where $t = 2$ or $t = 3$ and $(d, t) = 1$. Then transformation ${}^{D(n,K)}\eta(u)$ is a cubical one.

V. Explicit constructions of trapdoor accelerators and their applications

EXAMPLE 1

Let us consider general commutative ring K with unity and $F_n = T_1^{A(n,K)}\eta(u)T_2$, where T_1, T_2 are elements of $AGL_n(K)$ and the tuple (x, $x+\alpha_1$, $x+\alpha_1$, $x+\alpha_2$, $x+\alpha_2$, …, $x+\alpha_2$,$\ldots$, $x+\alpha_s$, $x+\alpha_s$) such that $cn < s < n$ for some constant $c > 0$. According to Theorem 2 the transformations F_n and ${F_n}^{-1}$ are of degree 3. So $T = \{T_1, T_2, u\}$ is a trapdoor accelerator of F_n of degree 3 and level 3.

The following two constructions give families of cubic multivariate map with trapdoor accelerator of rather large level.

Let us consider the implementation of public key based on the trapdoor accelerator of Example 1.

As usually name Alice corresponds to owner of the public key and name Bob corresponds to public user of the cryptosystem. Alice has to select size of finite field and dimension of the space V of plaintexts. Assume that she takes field $F_{2^{3}2}$ and dimension $n = 256$. Additionally Alice has to identify vector space V with point set P or line set L. Assume that she select L. It means that her plaintext is the tuple [$x_{0,1}$, $x_{1,1}$, x_{12}, x_{22}, $\ldots$, $x_{127,128}$, $x_{128,128}$]. Additionally Alice has to select parameter s corresponding to length of the path in the graph $A(256, F_{2^{3}2})$. For proper selection of this parameter one can investigate cycle indicator $Cind(v)$ of the vertex v of the graph, i. e minimal length of the cycle through v and evaluate maximal value of $Cind(v)$ via all possible vertexes v (cycle indicator $A(256, F_{2^{32}})$ of the graph). Accordingly [Archive] cycle indicator of the graph $A(n, F_q)$ is at least $2n+2$. In fact $Cind(A(n, F_q)) = 2n + 2$ for infinitely many special parameters q. There are $q^{[n/2]}$ lines $[l] \in L$ such that $Cind([l]) \geq 2n + 2$. Let $[l] = [x_{01}, x_{11}, \ldots, x_{[n/2],[n/2]}]$ be one of the lines with written above property where parameter n is even integer. The trapdoor accelerator uses path $p(t_1, t_2, \ldots, t_s)$ of even length s starting in $[l]$ given by colours of vertexes x_{01}, $x_{01} + t_1$, $x_{0,1} + t_2$, …, $x_{0,1} + t_s$ where $t_2 \neq 0$, $t_i \neq t_{i-2}$, for $i = 3, 4, \ldots, s$. Let us assume that $s \leq n$ and u be the last vertex of the path. Lower bound for $Cind([l])$ insures that destination lines of $p(t_1, t_2, \ldots, t_s)$ and $p(t'_1, t'_2, \ldots, t'_s)$, $t_1 \neq t'_1$ are different. The accelerator uses destination line $[y]$ of path of $A(n, F_q[x_{01}, x_{11}, \ldots, x_{n,n}]$ with colours x_{01}, $x_{01} + t_1$, $x_{0,1} + t_2$, … $x_{0,1} + t_s$ starting in $[l]$. Assume that $[y] = [x_{01} + t_s$, g_{11}, $g_{1,2}$, $g_{2,2}$, …, $g_{n,n}]$, where g_{11} , $g_{1,2}$, …, $g_{n,n}$ are cubical or quadratic multivariate polynomials in variables x_{01}, x_{11}, …, $x_{n,n}$. The trapdoor accelerator uses cubical transformation $F(t_1, t_2, \ldots, t_s)$ of $L = {F_q}^n$ of kind $x_{01} \to x_{1,0} + t_s$,

$x_{1,1} \to g_{1,1}$,

…,

$x_{nn} \to g_{n,n}$.

It is important that the map $F(t_1, t_2, \ldots, t_s)$ differs from each of $(q-1)^s$ transformations $F(t'_1, t'_2, \ldots, t'_s)$, $t'_1 \neq t_1$ if $s \leq n$. So Alice can take $s = 256$ and select one of $q(q-1)^{255}$ sequence t_1, t_2, …, t_{256}.

To construct trapdoor accelerator Alice has to generate two bijective linear transformations ${}^1T$ and ${}^2T$ of L of kind

$x_{01} \to^i l_{01}(x_{01}, x_{11}, \ldots, x_{128,128})$

$x_{11} \to^i l_{11}(x_{01}, x_{11}, \ldots, x_{128,128})$

$x_{128,128} \to^i l_{11}(x_{01}, x_{11}, \ldots, x_{128,128})$ where $i = 1, 2$. In a spirit of LU factorisation Alice can generate each ${}^iT$ as a composition of lower triangular matrix ${}^iL$, $i = 1, 2$ with nonzero entries on diagonal and upper triangular matrices ${}^iU$ with unity elements on diagonal. For selection of the tuple t_i, $i = 1, 2, \ldots, 256$, ${}^iL$ and ${}^iU$, $i = 1, 2$ Alice can use pseudorandom generators of field elements or some methods of generating genuinely random sequences (usage of existing implementation the quantum computer, other Probabilistic modifications of Turing machine, quasi-stellar radio sources (quasars) and etc).

Alice takes tuple of variables $[x] = (x_{0,1}, x_{11}, \ldots, x_{128,128})$ and conducts the following steps.

Step 1.

She compute a product of $[x]$ and ${}^1T$. The output is a string $[{}^1l_{01}(x_{0,1}, x_{11}, \ldots,$ $x_{128,128})$, ${}^1l_{11}(x_{0,1}, x_{11}, \ldots, x_{128,128})$, $\ldots$ ${}^1l_{128,128}(x_{0,1}, x_{11}, \ldots, x_{128,128})] = [{}^1u]$. Alice treats the output as the line of graph $A(256, F_{2^{32}}[x_{01}, x_{11}, \ldots, x_{128,128}])$

Step 2.

She computes the destination line $[{}^2u]$ of path with starting line $[{}^1u]$ and colours ${}^1u_{0,1}$, ${}^1u_{0,1} + t_1$, ${}^1u_{0,1} + t_2$, …, ${}^1u_{0,1} + t_{256}$.

Step 3.

Alice takes the tuple $[{}^2u] = [{}^1u_{0,1} + t_{256}, {}^2u_{1,1}, {}^2u_{1,2}, \ldots, {}^2u_{128,128}]$ of elements $F_{2^{32}}[x_{01}, x_{11}, \ldots, x_{128,128}]$ and forms the line ${}^3u = [{}^1u_{0,1})^2$, ${}^2u_{1,1}$, $\ldots$ ${}^2u_{128,128}]$ of the vector space L.

Step 4.

She computes the composition of the tuple ${}^3u$ and the matrix of linear map ${}^2T$. So Alice has the tuple of cubic multivariate polynomials ${}^4u = (f_{01}, f_{11}, \ldots,$ $f_{128,128})$. She presents coordinates of ${}^4u$ via their standard forms, i. e sums of monomial terms taken in the lexicographical order and writes the public rule F $x_{0,1} \to f_{0,1}(x_{01}, x_{11}, \ldots, x_{128,128})$, $x_{1,1} \to f_{1,1}(x_{01}, x_{11}, \ldots, x_{128,128})$, $x_{1,2} \to f_{1,2}(x_{01}, x_{11}, \ldots, x_{128,128})$, $\ldots$ $x_{128,128} \to f_{128,128}(x_{01}, x_{11}, \ldots, x_{128,128})$.

Finally Alice announces this multivariate rule for public users. Noteworthy that for the development of this private key Alice use only operations of addition and multiplication in the commutative ring $F_{2^{32}}[x_{01}, x_{11}, x_{1,2}, \ldots, x_{128,128}]$.

ENCRYPTION PROCESS.

Public user Bob creates her message p $=$ ($p_{0,1}$, from the space $(F_{2^{32}})^m$, $m = 256$. He computes tuple $(f_{0,1}(p_{01}, p_{11}, \ldots, p_{128,128})$, $f_{1,1}(p_{01}, p_{11}, \ldots, p_{128,128})$, $f_{1,2}(p_{01}, p_{11}, \ldots, p_{128,128})$, …, $f_{128,128}(x_{01}, x_{11}, \ldots x_{128,128}))$ of the ciphertext c. Theoretical estimation of the execution time is $O(m^4)$. Let

$D(m)$ be the density of the public rule F, which is a total number of monomial terms in all multivariate polynomials f_{01}, f_{11}, f_{12}, $\ldots$. Execution time is $cD(m)$ where constant c is time of the computation of single cubic monomial term. This constant depends on the choice of the computer. The following parameters can be useful. $D(16) = 5623$, $D(32) = 62252$, $D(64) = 781087$, $D(128) = 10826616$, $D(256) = 138266164$.

We can speed up the encryption process via reduction of parameter s. If we take twice shorter of the path of the graph, i.e. select $s = m/2$ then the values of $D(m)$ would be the following. $D(32) = 5623$, $D(64) = 62252$, $D(128) = 781087$, $D(256) = 10826616$.

This numbers disclose an interesting remarkable coincidences.

We can encode each character of $F_{2^{32}}$ by four symbols of F_{2^8}. Thus we can identify plaintext and the ciphertext with the tuple of binary symbols of length 1024. So we can encrypt files with extensions .doc, .jpg, .avi, .tif, .pdf and etc.

DECRYPTION PROCEDURE.

Alice has the private key which consists of the sequence t_1, t_2, $\ldots$, t_{256} and matrices ${}^1T$ and ${}^2T$. Assume that she got a ciphertext c from Bob. She computes ${}^2T^{-1} \times c ={}^1 c$ and treats this vector as line $[{}^1l] = [c_{01}$, c_{11}, c_{12}, $\ldots$, $c_{128,128}]$. Alice computes parameter $d = c_{01}{}^{31}$. She changes the colour of $[{}^1l]$ for $d + t_{256}$ and gets the line $[l] = [d + t_{256}, c_{11}, c_{12}, \ldots, c_{128,128}]$. Alice has to form the path in the graph $A(256, F_{2^{32}})$ with the starting line $[l]$ and further elements defined by colours $d + t_{255}$, $d + t_{254}$, $d + t_{253}$, $\ldots$, $d + t_1$ and d. So she computes the destination line $[{}^1l] = [d$, $d_{1,1}$, d_{12}, $\ldots$, $d_{128,128}]$. Finally Alice computes the plaintext p as $[1^l] \times^2 T^{-1}$.

VI. Conclusions

In [31] we describe several trapdoor accelerators defined with described above approach in selected cases of finite fields and arithmetical rings Z_m, where m is a prime power. They can be used for the constructions of multivariate public keys which is able to serve as tools for the encryption or construction of digital signatures. In this paper we consider the important case of finite fields of characteristic 2. Computer simulations of several variants of implementation of this public keys are presented in [31] where time evaluation and numbers of monomial terms are given. In [31] the reader can find heuristic arguments on security of suggested public rules.

References

[1] F. Lazebnik, V.Ustimenko, *Some Algebraic Constructions of Dense Graphs of Large Girth and of Large Size*, DIMACS series in Discrete Mathematics and Theoretical Computer Science , v.10, (1993) 75 – 93.

[2] F. Lazebnik, V.Ustimenko, *Some Algebraic Constractions of Dense Graphs of Large Girth and ofLarge Size*, DIMACS series in Discrete Mathematics and Theoretical Computer Science , v.10, (1993) 75 - 93.

[3] F.Lazebnik V. Ustimenko and A.J.Woldar, *A new series of dense graphs of high girth*, Bulletin of the AMS 32 (1) (1995), 73-79.

[4] V. Ustimenko, *Coordinatisation of Trees and their Quotients*, in the Voronoj's Impact on Modern Science, Kiev, Institute of Mathematics, 1998, vol. 2, 125-152.

[5] V. Ustimenko, *Linguistic Dynamical Systems, Graphs of Large Girth and Cryptography*, Journal of Mathematical Sciences.- Springer.- vol.140.- N3 .- 2007 .- P. 412-434.

[6] V. A. Ustimenko *On the extremal graph theory and symbolic computations*, Dopovidi National Academy of Sci, Ukraine, 2013, No. 2, p. 42-49.

[7] V. Ustimenko, *CRYPTIM: Graphs as Tools for Symmetric Encryption*, Lecture Notes in Computer Science, Springer, LNCS 2227, Proceedings of AAECC-14 Symposium on Applied Algebra, Algebraic Algorithms and Error Correction Codes, November 2001, pp. 278-286.

[8] V. Ustimenko, U. Romanczuk-Polubiec, A. Wroblewska, M. Polak, E. Zhupa, *On the constructions of new symmetric ciphers based on non-bijective multivariate maps of prescribed degree*, Security and Communication Networks, Volume 2019, Article ID 213756.

[9] Alexei G. Myasnikov, Vladimir Shpilrain, Alexander Ushakov. *Noncommutative Cryptography and Complexity of Group-theoretic Problems*. American Mathematical Society, 2011.

[10] A. G. Myasnikov, A. Roman'kov, *A linear decomposition attack*, Groups Complex. Cryptol. 7, No. 1 (2015), 81-94.

[11] V. A. Roman'kov, *A nonlinear decomposition attack*, Groups Complex. Cryptol. 8, No. 2 (2016), 197-207.

[12] V. Roman'kov, *An improved version of the AAG cryptographic protocol*, Groups, Complex., Cryptol, 11, No. 1 (2019), 35-42.

[13] A. Ben-Zvi, A. Kalka and B. Tsaban, *Cryptanalysis via algebraic span*, In: Shacham H. and Boldyreva A. (eds.) Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I, Vol. 10991, 255-274, Springer, Cham (2018).

[14] B. Tsaban, *Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography*, J. Cryptol. 28, No. 3 (2015), 601-622.

[15] B. Bolloba's', *Extremal Graph Theory*, Academic Press 1978, Dover, 2004.

[16] Tymoteusz Chojecki, Vasyl Ustimenko, *On fast computations of numerical parameters of homogeneous algebraic graphs of large girth and small diameter and encryption of large files*, IACR e-print archive, 2022/908.

[17] Vasyl Ustimenko, *On Extremal Algebraic Graphs and Multivariate Cryptosystems* IACR e-print archive, 2022/1537.

[18] Vasyl Ustimenko, *On the families of algebraic graphs with the fastest growth of cycle indicator and their applications*, IACR e-print archive, 022/1668(PDF)

[19] V. Ustimenko, *Graphs in terms of Algebraic Geometry, symbolic computations and secure communications in Post-Quantum world*, UMCS Editorial House, Lublin, 2022, 198 p.

[20] Anne Canteaut, François-Xavier Standaert (Eds.), *Eurocrypt 2021*, LNCS 12696, 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I, Springer, 2021, 839p.

[21] O. Zariski, P. Samuel, *Commutative algebra*, 2, Springer (1975).

[22] I.R. Shafarevich, *Basic algebraic geometry*, Springer (1977) (Translated from Russian).

[23] R. Hartshorne, *Algebraic geometry*, Springer (1977).

[24] V. Ustimenko, *On new results on Extremal Graph Theory, Theory of Algebraic Graphs and their applications in Cryptography and Coding The ory*, Reports of Nath. Acad. of Sci. of Ukraine, 2022, No. 4, P. 42-49.

[25] J. Ding, J. E. Gower, D. S. Schmidt, *Multivariate Public Key Cryptosystems*, 260. Springer, Advances in Information Security, v. 25, (2006).

[26] N. Koblitz, *Algebraic aspects of cryptography*, Springer (1998), 206 P.

[27] L. Goubin, J.Patarin, Bo-Yin Yang, *Multivariate Cryptography, Encyclopedia of Cryptography and Security*, (2nd Ed.) 2011, 824-828.

[28] $https : //csrc.nist.gov/pubs/pd/2021/08/04/migration - to - postquantum - cryptography/final$

[29] M. Noether, *Luigi Cremona*, Mathematische Annalen, 59 (1904), pp. 1-19.

[30] Lazebnik, F., Ustimenko, V.A. and A.J. Woldar, *A characterisation of the components of the graph $D(k, q)$*, Discrete Mathematics, 157 (1996), pp. 271-283.

[31] V. A. Ustimenko, T. Chojecki, M. Klisowski, *On Extremal Algebraic Graphs and implementations of new cubic Multivariate Public Keys*, https://eprint.iacr.org/2023/744

DOI: 10.15439/2023F3594
ISSN 2300-5963 ACSIS, Vol. 35

Comparing Performance of Machine Learning Tools across Computing Platforms

Pedro Vicente[*†], Pedro M. Santos[*†], Barikisu Asulba[§†], Nuno Martins[‡], Joana Sousa[‡], Luis Almeida[§†]
[*] *Instituto Superior de Engenharia do Porto (ISEP)*
[§] *Universidade do Porto – Faculdade de Engenharia*
[†] *CISTER Research Center in Real-Time & Embedded Computing Systems, Portugal*
[‡] *NOS Inovação, Portugal*
{1180558, pss}@isep.ipp.pt, up202103270@fe.up.pt, {nuno.mmartins, joana.sousa}@nos.pt, lda@fe.up.pt

***Abstract*—Embedded systems (ES) are wide-spread in our world and responsible for many critical systems. More recently, machine learning (ML) tools have become a well-established solution for data-intensive tasks, but their application in embedded systems is still gaining traction and their real-time performance is often unclear. We provide a (non-extensive) review of the ML tools that may be suited for deployment in ES, from which we selected two representative tools – the well-established Python-based Scikit-Learn, and the interoperability-oriented ONNX Runtime – to compare their response time. Using archetypal datasets and four pre-trained ML models, we measure the prediction time for each sample, for each model, in Scikit-Learn and ONNX Runtime in a standard desktop (to compare performance of the tools in the same platform), and for ONNX Runtime in a representative ES, a Raspberry Pi v4 (to compare performance of the same tool across platforms). We report that ONNX considerably improves over Scikit-Learn, and experiences a negligible performance degradation when ported to the RPi.**

***Index Terms*—Machine Learning, Embedded Systems, Prediction Time, Scikit-Learn, ONNX Runtime**

I. Introduction

Artificial intelligence (AI) and machine learning (ML) have grown dramatically in recent years, to the point where AI & ML is becoming a core technological component in many modern systems. In turn, embedded systems (ES) are a well-established technology that has enjoyed widespread use in our world for decades now, inconspicuously ensuring the efficient execution of a plethora of everyday operations. Many applications of ES are critical and time-sensitive ones [1]; for example, the timely detection and mitigation of cyberattacks, that is crucial for the integrity and dependability of many modern-world digital systems (e.g., banking sector).

The use of ML in embedded systems has garnered substantial interest, with the topic often being referred to as *TinyML*. The challenge is that embedded systems are typically resource-constrained platforms (ranging from micro-controllers to ARM or small-scale x86 platforms) and, while there is a plethora of ML libraries, not all provide the small memory footprint and stand-alone operation (i.e., sufficiently stripped-down from external dependencies) necessary for operation in embedded systems. Furthermore, a common strategy is to carry out training at the cloud (due to the higher processing capabilities available), whereas the embedded device only performs prediction. This raises the need for interoperability, as possibly the ML tool used for training can be different than the one available at the embedded device. Finally, characterization of response time is important to design real-time systems. Reports of execution time and/or speed-up against baselines can be found (e.g., [2], [3]), but typically for single models and not considering potential response time variability.

A noteworthy category of solutions are intermediate description languages, such as *Open Neural Network Exchange* (ONNX), and associated runtime environments (RTE), notably *ONNX Runtime* and *Tensorflow Lite*. Intermediate description languages describe a (trained) ML model using a (small) set of operators that the RTE is able to execute. This reduces computational requirements as it suffices that the RTE implements that set of operators to produce predictions from a given model. Downsides are that training may not be available and that the set of models at disposal may be limited.

In this work we report the performance of two selected libraries, *Scikit-Learn* and *ONNX Runtime*, in two platforms: a standard desktop and an archetypal embedded system, a Raspberry Pi v4. We deploy four one-class ML models – Isolation Forest (iForest), Local Outlier Factor (LOF), One Class Support Vector Machine (OC-SVM) and Stochastic Gradient Descent OC-SVM (SGD-SVM) –, that were pre-trained with network traffic data sets (legitimate and malicious) to detect cyberattack-related traffic. We show that ONNX Runtime can offer a speed-up of at least ≈ 14x with respect to Scikit-Learn for most models when both are executed in the desktop, and that ONNX Runtime running in the Raspberry Pi produces speed-ups of at least ≈ 8x against Scikit-Learn running in the desktop.

The structure of this paper is as follows. Section II portrays a motivating use-case and relevant ML models. An overview on ML for ES is provided in Section III. Section IV reports response times for selected ML libraries and computing platforms. Section V draws final remarks.

II. Motivating Use-Case and Selected ML Tools

A. Cybersecurity Use-Case

Cybersecurity is a domain of notable technological and societal impact in the modern world. The exposure surface for cyberattacks, and for recruiting devices that can be commandeered to participate in those attacks, increases everyday

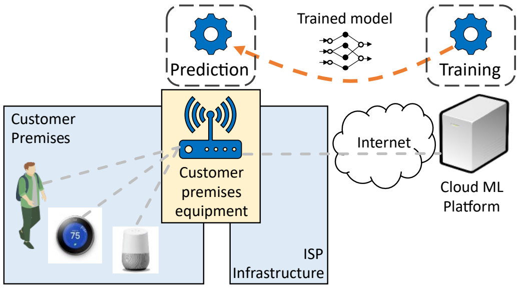


Fig. 1: Architecture of a cloud-edge system.

as the number of low-security IoT devices grows. This has been a driver for the increase of Distributed Denial of Service (DDoS) attacks, that aim at disrupting the servers of high-profile online services (e.g., Amazon, Google or Netflix) by having a very large number of infected devices (typically vulnerable IoT devices) issuing dummy requests to those servers. Internet Service Providers (ISP), that enable Internet service at customer premises through a Customer-Premises Equipment (CPE), are interested in mitigating the involvement of their customers' devices in cyberattacks through the use of Intrusion Detection Systems (IDS). The fastest response time is attained by deploying the IDS the closest possible to the targeted (or involved) nodes; for the ISP, this is the CPE.

Machine learning, whose successful application to cybersecurity is well documented [4]–[6], can help addressing this issue by learning network traffic patterns that are legitimate, and apply that knowledge to identify anomalous patterns that may concern malicious traffic. However, the CPE is often an embedded system with relatively few computational resources; while it can perform model prediction, it lacks the power to perform training, that ends up taking place in the cloud. The question arises of how to transfer models trained in the cloud, often with a state-of-the-art ML library, to the embedded system, that often will support only a limited set of ML libraries. Figure 1 presents the architecture of a cloud/edge system, and how can an ML-based IDS be deployed by leveraging the resource-rich cloud for training and transferring trained models to the resource-constrained CPE. Additional details on this use-case can be found in [7].

B. Datasets & ML Tools

To enable the presented use-case, we prepared datasets of network traffic (both legitimate and malicious) from publicly available sources, and trained four models to produce a ML mechanism that detect anomalous (potentially malicious) traffic. The details are described in [6]. We focus on One-Class (OC) models, i.e., models trained with samples of a single class to create a boundary around these, against which outliers can be detected. This semi-supervised approach allows the models to learn the regular (legitimate) traffic at a customer's network, and report anomalies that can potentially reveal themselves to be malicious traffic. All models were trained using Scikit-Learn [8], a free Python library that enjoys widespread use in the ML community.

The four selected models are reviewed next for convenience:

Isolation forest [9]: an unsupervised mechanism based on decision trees. It leverages the assumption that an anomalous sample requires less partitioning steps to be isolated. Thus, an isolation forest work by recursively generating partitions, by randomly splitting an attribute's value between the minimum and maximum values allowed for that attribute, until a target sample is contained in its own partition. Anomalies will require less partitions to be isolated.

Local Outlier Factor (LOF) [10]: LOF identifies local outliers by measuring the deviation of the density of a data point to its neighbors. The k-Nearest Neighbors is used to compute the reachability distance and local reachability density of each data point. The associated LOF score is calculated as the ratio of its local reachability density to the densities of its k-nearest neighbors. Points with high LOF scores are considered outliers. The value k (number of nearest neighbors) must be chosen carefully to avoid overfitting or underfitting.

One-Class Support Vector Machines [11]: traditional Support Vector Machines (SVM) select a decision boundary for which the margin between data points of different classes is maximized. Other interpretation is that SVMs maximize the distance between the convex hulls of points belonging to each class. One-Class SVM (OC-SVM) applies the same boundary-based mechanism for semi-supervised learning. It uses a hypervolume to encompass all of the instances; points outside the hypervolume are classified as anomalies.

Stochastic Gradient Descent [One Class] SVM (SGD-SVM) [12]: an online linear version of One-Class SVM, using a stochastic gradient descent (SDG). SDG algorithms are suited for applications where the number of data points and the problem dimensionality are both very large.

III. Overview of ML Tools for Embedded Systems

We review (not exhaustively) ML libraries targetting embedded systems and tools for interoperability and transpilation.

A. ML Libraries for Embedded Systems

TensorFlow (TF) [1] is an open-source library for AI/ML, composed of datasets and pre-trained models developed and released by the Tensorflow Community. Colaboratory (Colab) for instance, is a free Jupyter notebook environment and runs in the cloud so the user doesn't need to setup anything in his local machine. This library is supported in Haskell, C#, Julia, R, Ruby, Scala and Javascript.

Armadillo [2] is a library in C++ for linear algebra and scientific computing. It can use Open Multi-Processing (OpenMP), a free easy-to-use library for parallel computing.

mlpack [3] is a C++ ML library focused in providing fast and extensible implementations of ML models. This library is the combination of *Armadillo*, *ensmallen*, a library for numerical optimization and *cereal*, a serialization library.

[1] https://www.tensorflow.org/ (Note: all links last accessed on 2023-07-31)
[2] https://arma.sourceforge.net/
[3] https://mlpack.org/

Shogun [4] is an open-source library in C++ for machine learning development. It provides interfaces for C++, Python, Octave, R, Java, Lua, C#, Ruby and implements all the standard ML algorithms and some advanced as well. It is available for most operating systems.

SHARK [5] is an open-source machine learning library implemented in C++. It provides neural networks, kernel-based learning algorithms, linear and nonlinear optimization methods and is available for the most common operating systems.

A notable mention also goes to **CAFFE** [6], that focus on deep learning, thus supporting mostly neural networks (e.g., CNN, RCNN, LSTM).

There are also efforts focusing on deploying specific ML models in resource-scarce devices. The authors of [13] present *ProtoNN*, an algorithm that replicates k-Nearest Neighbor (k-NN) but has several orders lower storage and prediction complexity, and *ProtoNN* models can be deployed in very scarce plaforms (e.g. an Arduino UNO with 2kB RAM). The authors of [3] presents *SeeDot*, a domain-specific language to express ML inference algorithms and a compiler that compiles SeeDot programs to fixed-point code that can efficiently run on constrained IoT devices. In [2] *CMSIS-NN* is presented, which is essentially efficient kernels to maximize the performance and minimize memory footprint of neural network applications on Arm Cortex-M processors.

B. Interoperability of ML models

The following options, rather than tools, are standards to provide a common description of ML models, therefore enabling porting between libraries.

Open Neural Network Exchange (ONNX) [7] is an open specification with the following components: a definition of an extensible computation graph model; definitions of standard data types; and definitions of built-in operators. The first two make up the ONNX Intermediate Representation (or IR). In ONNX IR, each computation dataflow graph is structured as a list of nodes that form an acyclic graph. Each node is a call to an operator, and they have one or more inputs and outputs. Built-in operators are divided into a set of primitive operators and functions (the latter being, essentially, sub-graphs using primitive operators and/or other functions). Operators are implemented externally to the graph, but the set of built-in operators is portable across frameworks. Every framework supporting ONNX will provide implementations of these operators on the applicable data types. ONNX is compatible with at least 29 frameworks and converters and 30 inference runtimes.

Predictive Model Markup Language (PMML) [8] is a document format based on the Extensible Markup Language (XML) that can be used to described machine learning algorithms. It enables ML model porting between existing supporting libraries; these exist for C++, such as *cPMML* [9], and for Python, notably with the Scikit-Learn library *sklearn2pmml* [10], among others.

C. Transpilers

Transpilers translate a source code into a language different than the original one. The resulting code is described natively in the target language.

Sklearn-porter [11] is a Python library specifically developed to transpile ML models built with Scikit-Learn to other programming languages such as C, GO and JavaScript.

Model 2 Code Generator (m2cgen) [12] is a free, open-source library mainly developed in Python, that transpiles trained statistical models (trained, e.g., with Scikit-Learn or *lightning* libraries) into a native code for at least 16 different programming languages (R, Visual Basic, Haskell, C#, etc.).

D. Runtime Environments

A third dimension discussed here are tools that offer runtime environments (or simply runtime). Some of the aforementioned ML libraries leverage mechanisms for intermediate model representation that can be compiled or interpreted by a runtime environment. This solution avoids the need to deploy the entire library at the target device, thus resulting in a lightweight version of the initial library.

ONNX Runtime [13] is a cross-platform machine-learning model accelerator, used to deploy ONNX format models into production. It is meant to enable acceleration of machine learning inferencing across a variety of target hardware.

Tensorflow Lite [14] is a TF-variant tailored for resource-constrained systems that also uses a runtime. Using Tensorflow Lite, the target devices do not require the full TF library installation, but solely the *tflite_runtime* to perform inference. This tool eases the computational requirements of the target system, but its accuracy can be compromised if it uses operations not supported by the Tensorflow Lite. A recent paper reports TensorFlow Lite Micro [14], that adopts an interpreter-based approach to address ML efficiency and fragmentation in ES.

IV. Prediction Time Comparison of Selected Tools

A. Selected Tools & Experimental Setup

We have picked ONNX Runtime as the target ML tool to evaluate, and Scikit-Learn as the baseline reference. The option for Scikit-Learn was straightforward, as it is one of the most widely-used ML tools. It is also the tool used to train the models used in these measurements. As for the tools for deployment in embedded systems, we opted for ONNX Runtime based on a mix of our own requirements (that, when crossed against the available documentation, lead us to eliminate the remaining candidate tools), and impressions

[4] https://github.com/Kolkir/mlcpp/tree/master/classification_shogun

[5] https://www.shark-ml.org/

[6] https://caffe.berkeleyvision.org/

[7] https://onnx.ai/about.html

[8] https://dmg.org/pmml/v4-1/GeneralStructure.html

[9] https://amadeusitgroup.github.io/cPMML/

[10] https://github.com/jpmml/sklearn2pmml

[11] https://github.com/nok/sklearn-porter

[12] https://github.com/BayesWitnesses/m2cgen

[13] https://onnxruntime.ai/

[14] https://www.tensorflow.org/lite

TABLE I: Dataset descriptions.

Dataset	Traffic type	# samples	# Features
IOT23 [15]	IoT devices	487	26
Botnet [16]	Data theft	196	26

TABLE II: Selected platforms.

	Desktop	Raspberry Pi
Number of cores	4	4
Frequency utilized	2.00 GHz	600.00 MHz
RAM memory	9.64 GB	1.91 GB
Operating System	Ubuntu 20.04.6 LTS	Debian GNU/Linux 11
Python version	3.8.10	3.9.2
ONNX version	1.13.1	N/A
ONNXRuntime	1.14.1	1.14.1

acquired from experimenting with the other high-potential candidates. We lay down next the authors' impressions of the reviewed tools; this should not be interpreted, in any way, as a methodical and criterious analysis of these tools.

ML libraries: *Tensorflow* proved to be a collection of disperse, pre-trained models, making it hard to train new models from scratch. *Armadillo*, *mlpack*, *Shogun*, *SHARK* and *Caffe*, despite being described in C/C++, do not seem tailored for deployment in resource-constrained devices.

Interoperability: ONNX provides a clear and well document specification of how to convert models between tools, with extensive software support. PMML enables model porting between supporting libraries but, as aforementioned, we found no library to be a suitable candidate.

Transpilers: *sklearn-porter* is still under development and the range of models that can be transpiled to C is small (SVM and Decision Trees/Random Forest). Regarding *m2cgen*, even though transpiled models were able to perform closely to the original model, the tool offers very little documentation, making it hard to interpret the tool's output or even understand how the transpilation process actually occurs.

Runtimes: *ONNX Runtime* showed up as the best option. *TensorFlow Lite* was not explored, as usage of standard *TensorFlow* was also not straightforward.

Table I describes the data sets used in this performance analysis; more details in [6]. Table II presents the characteristics of the selected computing platforms. The models were converted to the ONNX specification using the *sklearn-onnx* library. A variant named *ONNX Runtime Optimized*, that optimizes the ONNX graphs describing the models, was also evaluated. Model accuracy obtained with ONNX Runtime and its Optimized variant was similar to that of Scikit-Learn.

B. Results

Figure 2 presents the average prediction time (over all input samples) of the four ML models across the three tools in the desktop equipment. Presented values are the average time of prediction for each new sample. We observe that ONNX produces an acceleration for most models, notably of ≈ 16x for Isolation Forest, ≈ 14x for OC-SVM, and ≈ 49x for SGD-SVM. In all this cases, the performance of the ONNX Runtime and its Optimized version do not differ substantially from each other. The same is not true, however, for the Local Outlier Factor (LOF), as shown in Figure 2 (top-right). We observed that ONNX underperforms in this model, taking longer than the Scikit-Learn. This leaves the door open for a more efficient implementation of LOF using the ONNX operators.

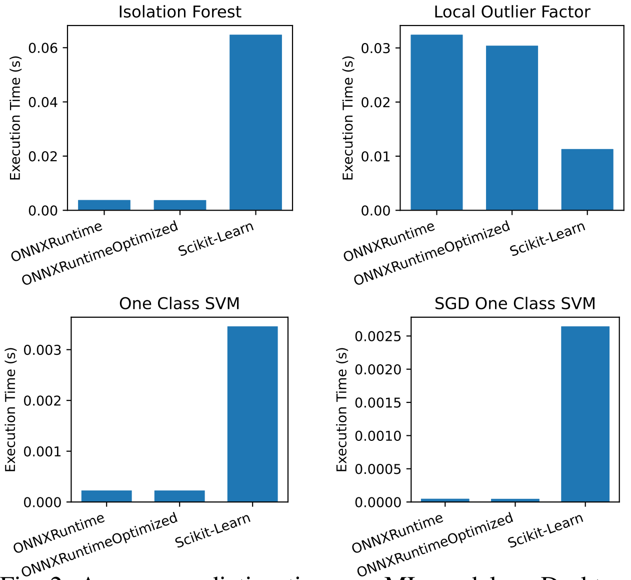


Fig. 2: Average prediction time per ML model on Desktop.

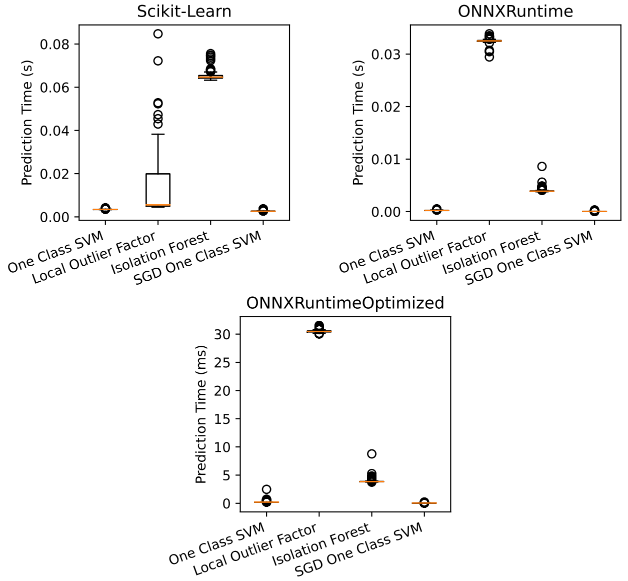


Fig. 3: Prediction time distribution per library on Desktop.

Figure 3 depicts the distribution of the prediction time of the various models per tool when executed in the Desktop. It is noteworthy for that, for ONNX Runtime (vanilla and Optimized), LOF presents the highest prediction time whereas, for Scikit-Learn, it is iForest that takes up the most time. Regarding the distribution of the samples, this is limited in the case of ONNX Runtime and Optimized to a few occasional outliers of additional time. For Scikit-Learn, LOF experiences considerable variability in prediction time. This may be a trade-off of the Scikit-Learn LOF implementation to achieve a lower average time for this concrete model.

Figure 4 exhibits the same analysis as Figure 2 for the

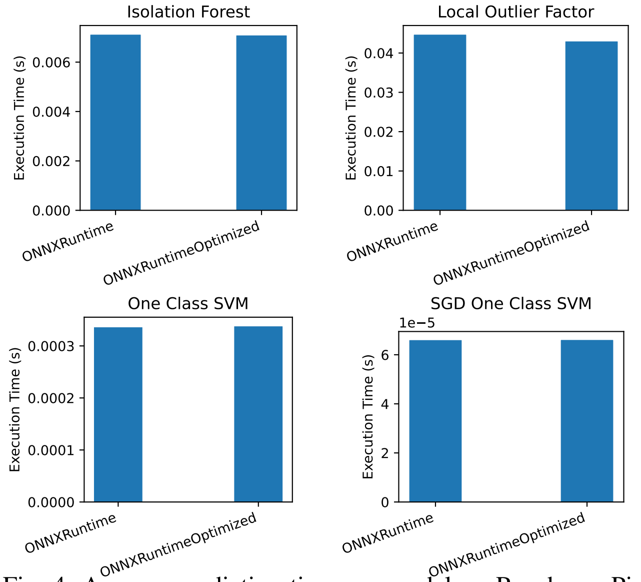


Fig. 4: Average prediction time per model on Raspberry Pi.

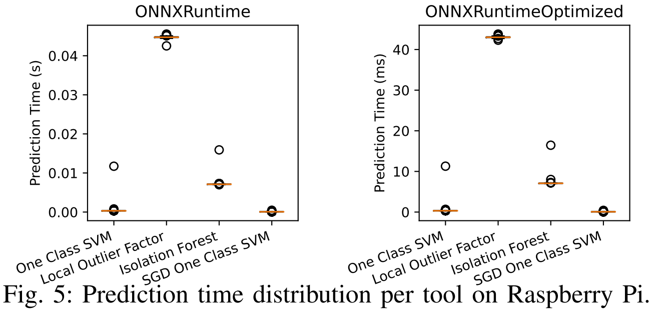


Fig. 5: Prediction time distribution per tool on Raspberry Pi.

second platform. The average of prediction time for the four ML models in the Raspberry Pi is superior to that of the Desktop response time; in detail, for Isolation Forest by $\approx 81\%$, for LOF by $\approx 37\%$; and for OC-SVM by $\approx 43\%$. However, when comparing with the Scikit-Learn running in the desktop, we obtain speedups of $\approx$ 8x for Isolation Forest, $\approx$ 9x for OC-SVM, and $\approx$ 39x for SGD-SVM. Results in Figure 5 presents little differences to Figure 3 (right and bottom) where it applies, apart from the generally higher median values in the Raspberry Pi.

V. Conclusion

We reviewed Machine Learning (ML) tools according to their potential for embedded system. We selected a particular tool, ONNX Runtime, for comparing prediction time against the well-established Python-based Scikit-Learn. ONNX Runtime is capable of running models described in the ONNX format; the models were trained in Scikit-Learn and exported to ONNX. The prediction time was measured in two platforms – a standard desktop and a target embedded system, a Raspberry Pi v4 – for four pre-trained ML models and datasets. We observe that ONNX Runtime considerably improves over the prediction time of Scikit-Learn, and experiences a negligible performance degradation when ported to the RPi. Future work will evaluate performance on more ML tools and platforms and investigate trade-offs with model target accuracy.

Acknowledgements

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020), and by the Portuguese National Innovation Agency (ANI) through the Operational Competitiveness Programme and Internationalization (COMPETE 2020) under the PT2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project(s) grant POCI-01-0247-FEDER-069522 (MIRAI).

References

[1] A. Hristoskova, N. González-Deleito, S. Klein, J. Sousa, N. Martins, J. Tagaio, J. Serra, C. Silva, J. Ferreira, P. M. Santos, R. Morla, L. Almeida, B. Bulut, and S. Sultanoğlu, "An Initial Analysis of the Shortcomings of Conventional AI and the Benefits of Distributed AI Approaches in Industrial Use Cases," in *IFIP AIAI 2021 Workshops*. Springer International Publishing, 2021, vol. 628, pp. 281–292.

[2] L. Lai, N. Suda, and V. Chandra, "CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs," *arXiv:1801.06601 [cs]*, Jan. 2018.

[3] S. Gopinath, N. Ghanathe, V. Seshadri, and R. Sharma, "Compiling KB-sized machine learning models to tiny IoT devices," in *40th ACM SIGPLAN Conference*. Phoenix AZ USA: ACM, Jun. 2019, pp. 79–95.

[4] R.Vinayakumar, M.Alazab, K.Soman, P.Poornachandran, A.Al-Nemrat, and S.Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[5] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices," *IEEE IoT Journal*, vol. 7, no. 8, pp. 6882–6897, Aug. 2020.

[6] N. Schumacher, P. M. Santos, P. F. Souto, N. Martins, J. Sousa, J. M. Ferreira, and L. Almeida, "One-Class Models for Intrusion Detection at ISP Customer Networks," in *IFIP AIAI 2023*, León, Spain, Jun. 2023.

[7] P. M. Santos, J. Sousa, R. Morla, N. Martins, J. Tagaio, J. Serra, C. Silva, M. Sousa, P. Souto, L. L. Ferreira, J. Ferreira, and L. Almeida, "Towards a Distributed Learning Architecture for Securing ISP Home Customers," in *IFIP AIAI 2021 Workshops*. Springer International Publishing, 2021, vol. 628, pp. 311–322.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *2008 IEEE Int'l Conference on Data Mining*, Pisa, Italy, Dec. 2008, pp. 413–422.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers."

[11] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support Vector Method for Novelty Detection," p. 7.

[12] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *21st Int'l Conference on Machine Learning*, 2004, p. 116.

[13] C. Gupta, A. S. Suggala, A. Goyal, H. V. Simhadri, B. Paranjape, A. Kumar, S. Goyal, R. Udupa, M. Varma, and P. Jain, "ProtoNN: Compressed and Accurate kNN for Resource-scarce Devices," in *34th Int'l Conference on Machine Learning*, Sydney, Australia, 2017, p. 16.

[14] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang, and P. Warden, "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems," *arXiv:2010.08678 [cs]*, Mar. 2021.

[15] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set]," *Stratosphere Lab., Praha, Czech Republic, Tech. Rep*, 2020.

[16] N. Koroniotis, "Designing an effective network forensic framework for the investigation of botnets in the Internet of Things," Ph.D. dissertation, UNSW Sydney, 2020.

DOI: 10.15439/2023F944
ISSN 2300-5963 ACSIS, Vol. 35

Deep Neural Networks application for Cup-to-Disc ratio estimation in eye fundus images

Sandra Virbukaitė
0000-0002-8758-8294
Institute of Data Science and Digital Technologies,
Vilnius University
Akademijos str. 4, Vilnius, Lithuania
Email: sandra.virbukaite@mif.vu.lt

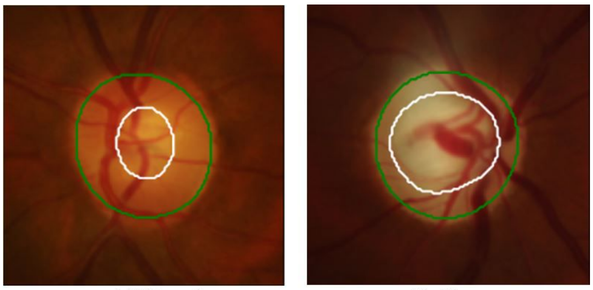
Jolita Bernatavičienė
0000-0001-5435-8348
Institute of Data Science and Digital Technologies,
Vilnius University
Akademijos str. 4, Vilnius, Lithuania
Email: jolita.bernataviciene@mif.vu.lt

Abstract—**Glaucoma is the second eye disease causing blindness worldwide. Optic Cup-to-Disc ratio (CDR) is a commonly applied method in glaucoma detection. The CDR is calculated based on Optic Disc (OD) and Optic Cup (OC) in eye fundus image screening. Therefore, the accurate segmentation of these two parameters is very important. Lately, Deep Neural Networks have demonstrated great effort in automated Optic Disc and Optic Cup segmentation but the overlapping between regions of OC and OD cause the challenge to obtain CDR automatically with high accuracy. In this paper, we assess the performance of CDR evaluation on three modifications of the Convolutional Neural Network (CNN) U-Net, namely Attention U-Net, Residual Attention U-Net (RAUNet), and U-Net++ applied on publicly available datasets RIM-ONE, DRISHTI, and REFUGE. We calculated the ground truth CDR value of testing eye fundus images of these datasets and compared it with the CDR value obtained by trained CNNs. Our results show that Attention U-net obtains the closest CDR to the ground truth CDR value but the identification of early-stage glaucoma needs an improvement.**

I. Introduction

GLAUCOMA is a progressive eye disease caused by damage to the optic nerve which is critical to vision. Usually, there are no symptoms in its early stages, and without proper treatment, glaucoma can lead to blindness. The evaluation for glaucoma starts by evaluating the cup-to-disc ratio (CDR) which is the ratio of the vertical optic cup diameter (VCD) to the vertical optic disc diameter (VDD) of a fundus image [1]. Fig. 1. presents an example of an eye fundus image. Depending on this ratio, several stages of glaucoma are distinguished. The cup-to-disc ratio of 0.4, 0.5 – 0.7, and above 0.7 indicate early-stage glaucoma, moderate-stage glaucoma, and severe-stage glaucoma respectively. A healthy eye has a CDR of 0.3 [2]. The CDR calculation is based on the segmented optic disc and optic cup regions. OD appears as a bright oval region, and OC takes place as the brighter oval region in the center of the optic disc (Fig. 1).

Addressing the limitation of medical resources in many areas worldwide [3], deep learning methods become successful in medical image segmentation [22]. Especially convolutional neural networks (CNN) demonstrated powerful representation and generalization abilities [4][5][6]. In automated glaucoma identification, the CNNs are trained on eye fundus images having the ground truth labels of OD and OC prepared by the ophthalmologists. Therefore, precise segmentation of the optic disc and optic cup is of the essence. However, accurate calculation of cup-to-disc ratio value is still in the development stage and faces challenges [7]. Here, the complexity occurs due to overlapping in the optic disc and optic cup areas. The automated deep learning-based algorithms fail in differing the boundaries of OC in the eye fundus image. The publically available fundus image datasets have insufficient images and segmentation masks to learn CNN for OD and OC segmentation, and CDR calculation.


(a) Normal

(b) Glaucoma

Fig. 1. Eye fundus image. The Optic Disc (Green) and the Optic Cup (White).

In this research, we aim to evaluate CNN's ability to accurate segmentation of OD and OC used for further CDR calculation. As the stage of glaucoma is identified by the CDR value, our experiment seeks to verify the equivalence of the CDR calculated with the help of CNN with the CDR calculated by ophthalmologists. Therefore, the datasets consisted of eye fundus images with the ground truth of OD and OC will be used.

II. Related work

The resultant challenges in CDR estimation prompt researchers to seek the improvements in presence of deep learning-based methods and new proposals.

Zhao et al. in [7] introduced a direct CDR estimation method based on a semi-supervised learning scheme. The proposed method is directly regressing the CDR value based on

This work was not supported by any organization

TABLE I
SUMMARY OF RELATED WORKS

Work	Dataset	The task to be solved in CDR evaluation and the results
[7]	Direct-CSU and ORIGA	Regression. AUC - 0.90 on Direct-CSU, AUC - 0.88 on ORIGA
[8]	Moorfields Eye Hospital in London	Classification. Sen - 0.92, Spec - 0.88
[9]	DRIONS-DB, RIM-ONE, IDRiD	Segmentation. IoU - 0.983 on DRIONS-DB IoU - 0.979 on RIM-ONE IoU - 0.976 on IDRiD
[10]	RIM-ONE, DRISHTI-GS, DRIONS-DB	Segmentation. Dice - 0.94 and IoU - 0.88 on RIM-ONE Dice - 0.94 and IoU - 0.88 on DRISHTI-GS Dice - 0.85 and IoU - 0.75 on DRIONS-DB

the OD feature using a Deep learning (DL) technique, bypassing intermediate segmentation. The approach is of two stages incorporating CDR value regression by random forest regressor and unsupervised feature representation of fundus image with a CNN, named MFPPNet. Alghmdi et al. in [8] proposed an approach by utilizing linear iterative clustering (SLIC) and a feed-forward neural network classifier. The classifier was used to classify the superpixels to detect the boundaries of OD and OC. The final detection and segmentation of OD and OC were completed by applying morphological operations and an elliptical estimation. In [9] the two-stage deep learning-based approach for CDR estimation was proposed. In the initial stage of optic disc segmentation, the U-Net was adopted. At a later stage, image-processing algorithms are used to estimate the CDR. In [10] a modified U-Net model was presented to locate OD. The CDR was calculated by segmented OD and OC incorporating the adaptive thresholding.

The discussion above shows the variety of methods applied in CDR estimation and it is difficult to compare their effectiveness. The summary is provided in Table I. The metrics used in the performance evaluation of the proposed approaches are dice coefficient (Dice), Jaccard Index (IoU), sensitivity (Sen), specificity (Spec), and area under the curve (AUC). In this paper, the OD and OC segmentation task will be solved by using CNNs to calculate CDR and to compare the glaucoma stage by CDR value obtained using CNNs with the glaucoma stage provided by the experts. Therefore, the Dice measure was chosen to evaluate the obtained results.

III. Methodology

A detailed description of the applied methods is presented in four sub-sections. The sub-section *3.1.* presents the datasets. In the sub-section *3.2.* the applied image preprocessing techniques are described. The sub-section *3.3.* presents the convolutional neural networks used in our experiments. The sub-section *3.4.* provides the details of metrics used for convolutional neural networks performance evaluation and cup-to-disc ratio calculation.

A. Dataset description

The public dataset DRISHTI-GS [11] contains 101 images with ground truth divided into 50 training and 51 testing images. All the images have been marked by four eye experts. All images were taken centered on an optic disc with a Field-Of-View (FOV) of 30 degrees and saved in the PNG uncompressed image format with a resolution of 2045 x 1752. Ground truths were collected from data experts.

RIM-ONE v.3 [12] is a public dataset consisting of 159 annotated stereo eye fundus images. The images were taken by a Nidek AFC-210 camera and saved in JPEG image format with a resolution of 2144 x 1424. The OD of each image has been segmented by two experts in ophthalmology to create the ground truth.

REFUGE [13] is a public dataset containing 1200 fundus images, with ground truth and clinical glaucoma labels. The dataset is split 1:1:1 into 3 subsets equally for training, validation, and testing. The training set with a total of 400 color fundus images taken by a Zeiss Visucam 500 fundus camera of size 2124 x 2056 is provided with the corresponding glaucoma status and the unified manual pixel-wise ground truths. The testing dataset contains 800 color fundus images taken by a Canon CR-2 camera of size 1634 x 1634 and is split into 400 testing images and 400 validation images. The images of validation and testing subsets were used in this paper only.

B. Preprocessing

For the purpose of image diversity increasing, various image augmentation techniques, namely image zooming by 20%, rotation by an angle of rotation from 0° to 45°, and horizontal and vertical flipping were applied. With this approach, the number of images in each dataset was extended to 1000. The region of interest (ROI) with the double size of OD area was extracted automatically by cropping the area around the centroid of optic disc and optic cup accordingly. The ROI images were resized to a size of 512 x 512 pixels by bicubic interpolation [14].

C. Convolutional neural networks

The three CNNs, namely UNet++ [15], Attention U-Net [16], and Residual Attention U-Net (RAUNet) [17] with significant improvement in image segmentation have been chosen to be trained for optic disc and cup segmentation. During the training of these CNNs, the binary cross-entropy loss function [1] and Adam optimizer [5] have been used. The parameters such as batch size, learning rate, and dropout rate for each CNN were searched by applying the KerasTuner framework.

UNet++ [15] is a nested and dense skip connections-based method. In the encoder part, the feature maps incur a dense convolution block. Here, the pyramid level causes the number of convolution layers. Due to the nested skip pathways, the proposed network generates full-resolution feature maps at multiple semantic levels.

Attention U-Net [16] contains the encoder, decoder, and attention gate at the skip connection of each level. The pre-trained network ResNet50 takes a place as an encoder, which consists of residual blocks with skip connections overcoming the vanishing gradient problem. The decoder contains up-sampling, and concatenation. Each convolution layer is followed by a rectified linear units (ReLU) activation function and batch normalization.

Residual Attention U-Net (RAUNet) [17] is an encoder-decoder-based network, where the encoder is constructed of pre-trained ResNet34 for semantic features extraction. The decoder contains a new augmented attention module (AAM) for multi-level features fusion and global context capturing.

D. Metrics

The evaluation metrics such as the Dice coefficient (Dice) [19], [20] and the cup-to-disc ratio (CDR) [21] are used in this paper.

The cup-to-disc ratio is calculated by dividing the OC diameter by the OD diameter [10].

$$CDR \quad = \quad \frac{vertical\,cup\,diameter}{vertical\,disc\,diameter} \tag{1}$$

Dice, which describes the similarity between the two images, is applied to evaluate the performance of trained CNNs in OD and OC segmentation.

$$Dice \quad = \quad \frac{2\left|S\bigcap L\right|}{\left|S\right|+\left|L\right|} \tag{2}$$

where, S – the predicted output map by segmentation, L – the ground truth binary map.

IV. Experiment and Results

The experiment was run by training the three different CNNs, namely UNet++, Residual Attention U-Net, and Attention U-Net on a dataset consisting of combined eye fundus images and their binary labels of DRISHTI-GS, REFUGE, and RIM-ONE and named as a combined dataset. The training of convolutional neural networks was performed on a single GPU machine [18] with 1TB of RAM using Keras included in TensorFlow version 2.9.1. An early stopping technique seeking a minimum for validation loss was applied to reduce unnecessary training time. The Adam optimizer and binary cross-entropy loss function were used during the training. The KerasTuner framework was applied to search for parameters, namely the learning rate, batch size, and dropout rate of each network. The trained CNNs were tested on 50 testing images of each dataset, REFUGE, RIM-ONE, and DRISHTI-GS separately to evaluate the Dice and calculate the CDR for the predicted OD and OC by each CNN. The CDR values were grouped into ranges of (0.3-0.4], (0.4-0.7], and above 0.7 according to glaucoma stages.

TABLE II
Dice of OD and OC segmentation by different CNN

	Attention U-Net		RAUNet		UNet++	
Test dataset	**OD**	**OC**	**OD**	**OC**	**OD**	**OC**
DRISHTI-GS	**0.979**	**0.877**	0.956	0.853	0.969	0.873
REFUGE	**0.973**	**0.874**	0.951	0.846	0.964	0.862
RIM-ONE	**0.977**	**0.855**	0.949	0.830	0.965	0.846

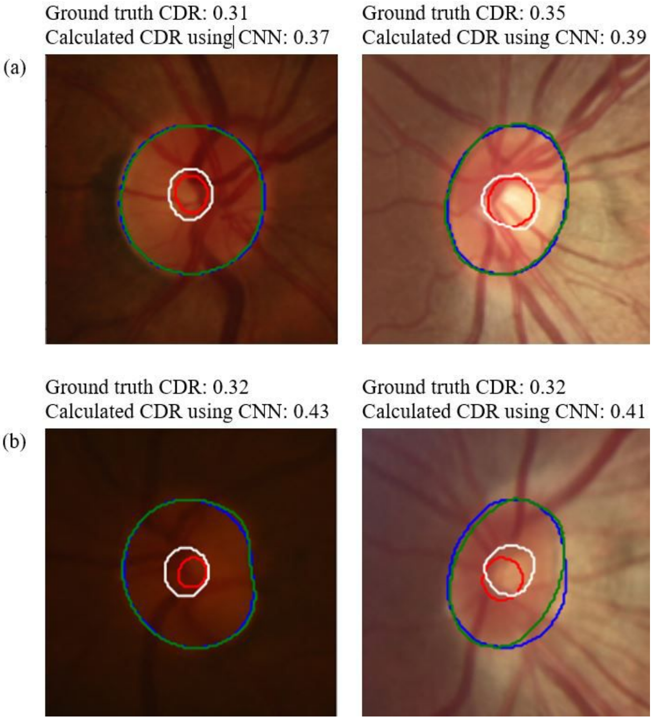

Ground truth CDR: 0.31
Calculated CDR using CNN: 0.37
Ground truth CDR: 0.35
Calculated CDR using CNN: 0.39
(a)
Ground truth CDR: 0.32
Calculated CDR using CNN: 0.43
Ground truth CDR: 0.32
Calculated CDR using CNN: 0.41
(b)

Fig. 2. Early-stage glaucoma. (a) The correct CDR value. (b) The wrong CDR value. The ground truth of OD and OC is indicated by green and red boundaries respectively. The segmented OD and OC are indicated by blue and white boundaries respectively.

Table II provides the evaluated Dice of the optic disc and cup segmentation testing the trained convolutional neural networks on REFUGE, RIM-ONE, and DRISHTI-GS test datasets. Here, comparing the performance of CNNs in OD and OC segmentation, the Attention U-Net demonstrates the highest Dice value of 0.9789, 0.9732, and 0.9770 for OD segmentation and 0.8769, 0.8742, and 0.8549 for OC segmentation on DRISHTI-GS, REFUGE, and RIM-ONE test datasets respectively. This leads to results in Table III presenting the mean and variance of ground truth CDR obtained on images

TABLE III
GROUND TRUTH CDR OF EYE FUNDUS IMAGE AND CALCULATED CDR USING CNN ON EACH TEST DATASET SEPARATELY

CDR by glaucoma stages	Test dataset	mean±variance of ground truth CDR	mean±variance of calculated CDR using CNN		
			Attention U-Net	RAUNet	UNet++
(0.3–0.4]	DRISHTI	–	–	–	–
	RIM-ONE	0.34±0.001	0.36±0.002	0.41±0.007	0.38±0.005
	REFUGE	0.35±0.001	0.42±0.002	0.52±0.010	0.46±0.002
(0.4–0.7]	DRISHTI	0.57±0.006	0.56±0.005	0.50±0.007	0.52±0.006
	RIM-ONE	0.54±0.008	0.54±0.013	0.62±0.018	0.58±0.027
	REFUGE	0.51±0.006	0.54±0.006	0.61±0.006	0.55±0.008
Above 0.7	DRISHTI	0.85±0.007	0.84±0.008	0.79±0.009	0.81±0.009
	RIM-ONE	0.83±0.002	0.84±0.004	0.73±0.026	0.78±0.003
	REFUGE	0.76±0.002	0.78±0.002	0.72±0.005	0.78±0.007

TABLE IV
% OF TRUTH CDR AND CDR CALUCLATED USING CNN ON EACH TEST DATASET SEPARATELY

CDR by glaucoma stages	Test dataset	Amount of images	% of correctly calculated CDR using CNN		
			Attention U-Net	RAUNet	UNet++
(0.3–0.4]	DRISHTI	–	–	–	–
	RIM-ONE	8	50	13	38
	REFUGE	14	21	14	14
(0.4–0.7]	DRISHTI	9	89	67	78
	RIM-ONE	27	85	67	70
	REFUGE	33	91	85	88
Above 0.7	DRISHTI	41	66	20	46
	RIM-ONE	15	93	53	80
	REFUGE	3	100	67	100

TABLE V
GROUND TRUTH CDR OF NON-GLAUCOMA EYE FUNDUS IMAGE AND CALCULATED CDR USING CNN

CDR by glaucoma stages	Test dataset	mean±variance of ground truth CDR	mean±variance of CDR calculated using CNN		
			Attention U-Net	RAUNet	UNet++
≤0.3	RIM-ONE	0.29±0.002	0.28±0.002	0.30±0.008	0.27±0.004

of each test dataset and the mean and variance of CDR calculated by Unet++, Attention U-Net, and Residual Attention U-Net on testing images of each test dataset. Assessing the mean and variance of ground truth CDR and the CDR calculated using the segmented OD and OC by CNNs, the best result was obtained by Attention U-Net.

Using the same approach, the amount of eye fundus images in each test dataset was calculated and evaluated the percentage of how many images each CNN is able to calculate the correct CDR in comparison with ground truth. The results are shown in Table IV. The obtained percentage of truth CDR and CDR calculated using segmented OD and OC by CNNs indicates that the Convolutional Neural Networks better identify moderate-stage glaucoma and severe-stage glaucoma cases but the identification of early-stage glaucoma is quite poor. For example, using Attention U-Net for REFUGE dataset images, the CDR, compared to the truth CDR, was calculated correctly for 91% of moderate-stage glaucoma images and 100% of severe-stage glaucoma images. Meanwhile, the correct CDR was calculated only for 21% of early-stage glaucoma images. The showcase examples of the optic disc and cup segmentation in the images of early-stage glaucoma by Attention U-Net and the ground truth are provided in Fig. 2. Fig. 2 (a) shows the cases when the predicted CDR by CNN is near the ground truth value of CDR. Fig. 2 (b) shows the cases when the CNN is wrong in the optic disc and cup segmentation and predicts the CDR value of moderate-stage glaucoma for early-stage glaucoma images. This can be caused by a noticeable difference in image quality. The images, for which the values of CDR were predicted correctly, are brighter and indicate more clear boundaries of OD and OC. Meanwhile, the boundaries of OD and OC in the wrongly predicted value of CDR are blurry. As the only RIM-ONE dataset has images of non-glaucoma cases, these have been tested separately and the results of the mean and variance of CDR are shown in Table V. The obtained CDR results indicate CNN's ability to segment non-glaucoma cases quite accurately. This can be influenced by clear boundaries of the optic disc and cup in images of healthy eyes.

V. Conclusion

The three Convolutional Neural Networks, namely U-Net++, Attention U-Net, and Residual Attention U-Net were applied in this paper for the evaluation of cup-to-disc ratio. The experiments show that the non-glaucoma cases were identified quite accurately by all three CNNs. However, evaluating the ability of CNNs in identifying the different glaucoma stages, it is noticed that CNNs perform better in identifying moderate-stage glaucoma and severe-stage glaucoma, but the early-stage glaucomatous cases are poorly identified. Attention U-net was able to identify 50% of early-stage glaucoma cases in RIM-ONE, and 13% and 20% early-stage glaucoma cases were identified by Residual Attention U-Net and U-Net++ respectively. In the REFUGE dataset, only 21%, 14%, and 13% of early-stage glaucoma cases were identified by Attention U-NET, U-Net ++, and Residual Attention U-net respectively. CNNs misidentify cases of early-stage glaucoma by classifying them as intermediate-stage glaucoma. Which is not so bad as such cases will be noticed by the doctors. However, further research and the refining of CNNs are needed.

Acknowledgment

We would like to thank the IT Research Center of Vilnius University [18] for the provided HPC resources.

References

[1] P. Yi, Y. Xu, J. Zhu, J. Liu, Ch. Yi, H. Huang, and Q. Wu, "Deep level set learning for optic disc and cup segmentation," *Neurocomputing,* 2021, pp. 330–341, https://doi.org/10.1016/j.neucom.2021.08.102.

[2] S. Virbukaitė and J. Bernatavičienė, "Deep Learning Methods for Glaucoma Identification Using Digital Fundus Images," *Baltic J. Modern Computing,* 2020, pp. 520–530, https://doi.org/10.22364/bjmc.2020.8.4.03.

[3] M. N. Cheema, A. Nazir, B. Sheng, P. Li, J. Qin, J. Kim and D. D. Feng, "Image-Aligned Dynamic Liver Reconstruction Using Intra-Operative Field of Views for Minimal Invasive Surgery," *IEEE Trans Biomed Eng.,* 2019, pp. 2163–2173, https://doi.org/10.1109/TBME.2018.2884319.

[4] H. Xiong, S. Liu, R. V. Sharan, E. Coiera, S. Berkovsky, "Weak label based Bayesian U-Net for optic disc segmentation in fundus images," *Artificial Intelligence In Medicine,* vol. 126, 2022, https://doi.org/10.1016/j.artmed.2022.102261.

[5] Y. Jiang, F. Wang, J. Gao and S. Cao, "Multi-Path Recurrent U-Net Segmentation of Retinal Fundus Image," *Applied Sciences,* vol. 10, 2020, https://doi.org/10.3390/app10113777.

[6] J. Civit-Masot, F. Luna-Perejon, S. Vicente-Diaz, J. M. Rodriguez Corral and A. Civit, "TPU Cloud-Based Generalized U-Net for Eye Fundus Image Segmentation," *IEEE Access,* vol. 7, 2019, pp. 142379–142387, https://doi.org/10.1109/ACCESS.2019.2944692.

[7] R. Zhao, X. Chen, X. Liu, Z. Chen, F. Guo and S. Li, "Direct Cup-to-Disc Ratio Estimation for Glaucoma Screening via Semi-Supervised Learning," *IEEE Journal of Biomedical and health informatics,* 2020, pp. 1104–1113, https://doi.org/10.1109/JBHI.2019.2934477.

[8] H. Alghmdi, H. Lilian Tang, M. Hansen, A. O'Shea, L. Al Turk and T. Peto, "Measurement of optical cup-to-disc ratio in fundus images for glaucoma screening," *2015 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM),* 2015, pp. 1–5, https://doi.org/10.1109/IWCIM.2015.7347097.

[9] E. Sudheer Kumar and C. Shoba Bindu, "Two-stage framework for optic disc segmentation and estimation of cup-to-disc ratio using deep learning technique," *Journal of Ambient Intelligence and Humanized Computing,* 2021, https://doi.org/10.1007/s12652-021-02977-5.

[10] G. Rakesh and V. Rajamanickam, "A Novel Deep Learning Algorithm for Optical Disc Segmentation for Glaucoma Diagnosis," *Traitement du Signal,* vol. 39, 2022, pp. 305–311, https://doi.org/10.18280/ts.390132.

[11] J. Sivaswamy, S. R. Krishnadas, G. D. Joshi, M. Jain and A. U. S. Tabish, "Drishti-GS: Retinal image dataset for optic nerve head(ONH) segmentation," *Proceedings of 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI),* 2014, pp. 53–56, https://doi.org/10.1109/ISBI.2014.6867807.

[12] F. J. F. Batista, T. Diaz-Aleman, J. Sigut, S. Alayon, R. Arnay, and D. Angel-Pereira, "RIM-ONE DL: A Unified Retinal Image Database for Assessing Glaucoma Using Deep Learning," *Image Analysis and Stereology,* vol. 39, 2020, pp. 161-–167, https://doi.org/10.5566/ias.2346.

[13] J. I. Orlando, H. Fu, J. B. Breda, K. van Keer, D. R. Bathula, A. Diaz-Pinto, R. Fang, P. Heng, J. Kim, J. Lee, J. Lee, X. Li, P. Liu, S. Lu, B. Murugesan, V. Naranjo, S. S. R. Phaye, S. M. Shankaranarayana, A. Sikka, J. Son, A. van den Hengel, S. Wang, J. Wu, Z. Wu, G. Xu, Y. Xu, P. Yin, F. Li, X. Zhang, Y. Xu and H. Bogunovic, "REFUGE Challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs," *Medical Image Analysis,* vol. 59, 2020, pp. 101570, https://doi.org/10.1016/j.media.2019.101570.

[14] S. Virbukaitė and J. Bernatavičienė, "Image Resizing Level Impact on Eye Fundus Optic Disc and Optic Cup Segmentation," *Proceedings of the 30th Jubilee International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,* 2022, https://www.doi.org/10.24132/CSRN.3201.39.

[15] Z. Zhou, Md M. R. Siddiquee, N. Tajbakhsh and J. Liang, "UNet++: A Nested U-Net Architecture for Medical Image Segmentation," *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2018. Lecture Notes in Computer Science,* 2018, https://doi.org/10.48550/arXiv.1807.10165.

[16] T. Shyamalee and D. Meedeniya, "Attention U-Net for Glaucoma Identification Using Fundus Image Segmentation," *2022 International Conference on Decision Aid Sciences and Applications (DASA),* 2022, https://doi.org/10.1109/DASA54658.2022.9765303.

[17] Z. Ni, G. Bian, X. Zhou, Z. Hou, X. Xie, C. Wang, Y. Zhou, R. Li and Z. Li, "RAUNet: Residual Attention U-Net for Semantic Segmentation of Cataract Surgical Instruments," *Neural Information Processing. ICONIP 2019. Lecture Notes in Computer Science,* 2019, https://doi.org/10.48550/arXiv.1909.10360.

[18] ITAPC, "Information technology research center of Vilnius University," *Energy Economics,* 2023.

[19] T. Akshat, P. Kumar and S. Pathan, "Automated segmentation of optic disc and optic cup for glaucoma assessment using improved UNET++ architecture," *Biocybernetics and Biomedical Engineering,* vol. 41, 2021, pp. 819–832, https://doi.org/10.1016/j.bbe.2021.05.011.

[20] Y. Jiang, F. Wang, J. Gao and S. Cao, "Multi-Path Recurrent U-Net Segmentation of Retinal Fundus Image," *Applied Sciences,* vol. 10, 2020, https://doi.org/10.3390/app10113777.

[21] J. Carrillo, L. Bautista, J. Villamizar, J. Rueda, M. Sanchez, and D. Rueda, "Glaucoma detection using fundus images of the eye," *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA),* 2019, pp. 1–4, https://doi.org/10.1109/STSIVA.2019.8730250.

[22] D.M.S. Barros, J.C.C. Moura, C.R. Freire, A.C. Taleb, R.A.M. Valentim, P.S.G. Morais, "Machine learning applied to retinal image processing for glaucoma detection: review and perspective," *BioMed Eng OnLine,* 2020, https://doi.org/10.1186/s12938-020-00767-2.

[23] D.M.S. Barros, J.C.C. Moura, C.R. Freire, A.C. Taleb, R.A.M. Valentim, P.S.G. Morais, "Machine learning applied to retinal image processing for glaucoma detection: review and perspective," *BioMed Eng OnLine,* 2020, https://doi.org/10.1186/s12938-020-00767-2.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1197–1203
DOI: 10.15439/2023F3892
ISSN 2300-5963 ACSIS, Vol. 35

On some concept lattice of social choice functions

Piotr Wasilewski
0000-0003-0027-1102
Systems Research Institute,
Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
Faculty of Computer Science,
Dalhousie University
6050 University Avenue,
Halifax, Nova Scotia, Canada
Email: pwasilew@ibspan.waw.pl

Janusz Kacprzyk
0000-0003-4187-5877
Systems Research Institute,
Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
WIT – Warsaw School
of Information Technology
Newelska 6, 01-447, Warsaw, Poland
Email: kacprzyk@ibspan.waw.pl

Sławomir Zadrożny
0000-0002-6642-0927
Systems Research Institute,
Polish Academy of Sciences
Newelska 6, 01-447 Warsaw, Poland
WIT – Warsaw School
of Information Technology
Newelska 6, 01-447, Warsaw, Poland
Email: zadrozny@ibspan.waw.pl

***Abstract*—Social choice function or voting procedure is one of the crucial concepts in the domain of political sciences. It maps individuals' preferences over a set of candidates to some subset (possibly one-element) of the candidates who can be thought as the winners of an election procedure. The paper is aimed at applications of formal concept analysis methods to study of social choice functions. We will construct concept lattices over selected set of social choice functions characterized by possessing some properties deemed as important from the point of view of political sciences. We will discuss issues connected with reducibility of both objects and attributes, irreducibility of object concepts as well as attribute concepts and attribute implications. We will discuss also the shape of the constructed concept lattice of social choice functions which in some part is exceptionally regular from the perspective of the lattice theory.**

I. Introduction

THIS paper is aimed at some applications of formal concept analysis (FCA) methods [32], [6] in social choice theory [16], [4], [19], one of the most important research domains of political sciences. We concentrate on social choice functions or voting procedures which are concepts of a crucial importance in the theory of social choice [4], [5], [18]. Our aim is to offer a non-standard approach to studying and comparing popular social choice functions. The FCA has been used in broadly meant social choice theory (cf., e.g., [27], [27], [8] or even [6]) but not with such a specific goal as here. Also, various non-classical approaches has been proposed in this area. For example, fuzzy logic has been applied with success to model various aspects of social choice (cf., e.g.[9], [21], [10], [22], [12], [23], [11]) Another line of non-classical research in this area, which is relevant for our purposes, is based on the rough-sets theory (RST) [25], [24], [26]. In particular, the research presented in [3], [13] concentrated on the issue of reduction of voting criteria and on measuring of similarity and dissimilarity of different social choice functions, ideas and methods used in comparison of voting procedures.

Actually, rough set theory can be viewed as a similar theory to formal concept analysis in the domain of data mining and knowledge discovery [31], [30] and such view drove our attention to the idea of application of formal concept analysis methods in the area of social choice functions.

The rest of the paper is organized as follows. In Section 2 some basic concepts of theory of social choice functions are introduced and discussed including voting procedures and criteria used for comparisons of different voting procedures. In Section 3 formal context of social choice functions and its concept lattice are introduced together with an investigation of the structure of concept lattice of these social choice functions. Section 4 is devoted to analysis of information provided by concept lattice of social choice functions, including attribute independency, reduction of information and attribute implications holding in the analyzed context of voting procedures. Section 4 is followed by Conclusions discussing results and presenting directions for further research.

II. Social choice functions

We consider social choice problem in a general setting which may be characterized as follows. There is a set of experts $E = \{e_j\}_{j \in J}$ and a set of options (alternatives) $O = \{o_i\}_{i \in I}$. Each expert e_j is assumed to represent his or her testimonies over the set of options O in the form of a binary preference relation $R_j \subset O \times O$ where $R_j(o_{i_1}, o_{i_2})$ is meant to represent preference of the expert e_j for the option o_{i_1} over the option o_{i_2}, i.e., that in his or her opinion option o_{i_1} is better than the option o_{i_2}. Preference relations R_j may be assumed to exhibit various properties. Often, the transitivity ($R_j(o_{i_1}, o_{i_2}) \wedge R_j(o_{i_2}, o_{i_3})$ implies $R_j(o_{i_1}, o_{i_3})$), completeness ($\forall i_1, i_2 \in I$ either $R_j(o_1, o_2)$ or $R_j(o_2, o_1)$ holds), and some form of anti-symmetry (e.g., $R_j(o_1, o_2) \implies \neg R_j(o_2, o_1)$), is assumed.

In such a setting, the *social choice function* F may be defined as follows:

$$F(X, \{R_j\}) = Y \tag{1}$$

where $X, Y \subseteq O$ are sets of options such that $Y \subseteq X$, and $\{R_j\}$ is a set of preference relations on O.

Thus, a social choice function determines which options Y are to be selected from a set of options X in view of the preference relations $\{R_j\}$ of a group of experts.

Voting procedures used in the *elections* may be interpreted as social choice functions. Often, the voting procedure is required to indicate as Y exactly one element subset of X (cf. (1)), i.e., $Y = \{o_i\}$ and the option (candidate) o_i is then called the *winner* of the election. In case of voting procedures we will usually refer to experts and options as voters and candidates, respectively.

Particular voting procedures differ in that how the winner is selected. For example, some arrive at the decision in an iterative way and the voters are requested to express their preferences several times, often with respect to a changing set of candidates. Often, the *agenda* is established which determines in which order the candidates are voted for. Most of voting procedures do not require voters to express their whole preference relations, at least not at the very beginning, but assuming existence of such a complete preference relation (*ranking* of the candidates) makes it possible to derive the winner of the election (assuming they always vote in accordance with their complete preference relation).

There are many postulated properties which are desired to be met by a fair voting procedure properly reflecting the preferences of the voters. However, it turns out that it is impossible to find one possessing all desired properties. Thus, satisfaction of such desired properties may be treated as criteria in evaluation of particular voting procedures.

In our approach, based on our previous work [3], [11], [13], [23], our point of departure is the following list of desired properties (criteria) of voting procedures:

A - Condorcet winner If each time a candidate is preferred by the majority of voters when compared to any other candidate then it has to be the winner.

B - Condorcet loser If all other candidates are preferred by the majority of voters when compared to a given candidate then the latter candidate cannot be the winner.

C - majority winner if a candidate is top-ranked in the rankings of the majority (more than 50%) of voters then this candidate have to be the winner.

D - monotonicity If a candidate is the winner then if it is ranked higher by a voter then it has still to be the winner and if a candidate is not the winner then if ranked lower by a voter cannot become the winner.

E - weak Pareto winner If for a given candidate o_1 there exists another candidate o_2 which is ranked higher than o_1 by all voters then o_1 cannot be the winner.

F - consistency If the set of voters E is divided in two groups ($E = E_1 \cup E_2$), in any possible way, and a candidate is the winner both for E_1 and E_2 then it has to be the winner for E.

G - heritage If a candidate $o_i \in O$ is the winner then it has to be the winner also when any subset of candidates $O_1 \subseteq O$ is considered such that $o_i \in O_1$

In the paper we will consider some popular voting procedures which are briefly characterized below.

Amendment Candidates are voted individually, in some order, and if a candidate gets the majority of votes it becomes the winner; otherwise the next candidate is voted.

Copeland the winner is a candidate for which the highest is the difference between the numbers of pairwise comparisons with other candidates in which it is voted by majority, respectively, as better and as worse.

Dodgson the winner is the candidate for which the minimum number of changes in voters rankings is needed to make it a Condorcet winner.

Schwartz if there is a Condorcet winner it is the winner; otherwise the set $O_S \subseteq O$ of all candidates who are voted as better by majority of voters in pairwise comparison with all candidates belonging to the set $O \setminus O_S$ are the winners.

Max-min/Egalitarian The winner is the candidate whose worst position over the rankings of all voters is the highest.

Plurality Only top-ranked candidates for each voter are taken into account and the winner is the one which is most often among them.

Borda Each position in the ranking is assigned a score, highest for the top position and lowest for the last one and the winner is a candidate for which the sum of scores of the positions it takes in rankings of particular voters (the Borda count) is the highest.

Approval Each voter points out a subset of preferred candidates and the winner is the option which is present in the highest numer of these subsets.

Black The winner is the Condorcet winner, if it exists; otherwise the Borda voting procedure is used.

Runoff Works like Plurality but two best candidates are selected and then Plurality voting is repeated for just two of them.

Nanson The Borda voting procedure is iteratively repeated and in each iteration a candidate with the lowest Borda count is excluded from the voting in the following iteration.

Hare The Plurality voting procedure is iteratively repeated and in each iteration candidates with the lowest number of top positions in the rankings are excluded from the voting in the following iteration.

Coombs The winner is a candidate which is top-ranked by the majority of voters, if it exists. Otherwise, the procedure is iteratively repeated but in each iteration the candidate which is most often ranked as the last one is eliminated.

III. Formal Concept Analysis

Formal concept analysis (FCA) was introduced by Wille in [32]. FCA is founded on lattice theory and aimed at data analysis and representation. FCA uses tabular-type data representations called *formal contexts* where objects are characterized by mono-valued attributes[1]. In FCA data are represented and analyzed by concept lattices using algebraic, order and logical methods based on concept lattices. A construction of concept lattices is based on Galois connections determined by formal contexts. Here we present basic notions of FCA. For a detailed

[1] In the process of development, FCA was broadened also for multi-valued attributes by means of conceptual scaling [6].

presentation of formal concept analysis see the first monograph on FCA by Ganter and Wille [6] and for elements of lattice theory see an excellent textbook freely available on-line by Burris and Sankappanavar [2].

A *formal context* is defined as a triple of the form (G, M, I), where G and M are sets, while I is a binary relation $I \subseteq G \times M$. Elements of set G and M are called *objects* and *attributes* respectively as well as *an extent* and *an intent*, respectively, of the context (G, M, I). The fact that $a \in G$ and $m \in M$ are in relation I will be denoted as $a\ I\ m$, and will be described as that *object a possesses attribute m*, or that *attribute m is possessed by object a*.

For context (G, M, I) two different operators between power sets $\wp(G)$ and $\wp(M)$ are defined:

$$X \mapsto X^i = \{m \in M : a\ I\ m,\ \forall\ a \in X\},$$

$$Y \mapsto Y^e = \{a \in G : a\ I\ m,\ \forall\ m \in Y\},$$

for each $X \subseteq G$, $Y \subseteq M$. Operator $\cdot^i$ is called *an intension operator* and operator $\cdot^e$ is called *an extension operator*. One can note that operators $\cdot^i$ and $\cdot^e$ are perfectly dual in the sense of order theory, thus in FCA there is commonly used practice to denote these operators by the same prime symbol $\cdot'$ [6]. This practice is justified by the formal properties of extension and intension operators presented in Table I and makes calculation easier. It also does not lead into confusion: in Table I for example, since $Y \subseteq M$, then formula (3b) $Y' = Y'''$ can be rewritten as $Y^e = Y^{eie}$.

TABLE I
BASIC PROPERTIES OF INTENSION AND EXTENSION OPERATORS FOR FORMAL CONTEXT (G, M, I) AND SETS $X, X_1, X_2 \subseteq G$ AND $Y, Y_1, Y_2 \subseteq M$ [6].

(1a) $X_1 \subseteq X_2 \Rightarrow X_2' \subseteq X_1'$	(1b) $Y_1 \subseteq Y_2 \Rightarrow Y_2' \subseteq Y_1'$
(2a) $X \subseteq X''$	(2b) $Y \subseteq Y''$
(3a) $X' = X'''$	(3b) $Y' = Y'''$
(4a) $(X_1 \cup X_2)' = X_1' \cap X_2'$	(4b) $(Y_1 \cup Y_2)' = Y_1' \cap Y_2'$

A *formal concept* of context (G, M, I) is pair (A, B) with $A \subseteq G$, $B \subseteq M$, such that $A = B'$ and $B = A'$. A and B are called *the extent* and *the intent* of the concept (A, B) respectively. The family of all formal concepts of context (G, M, I) is denoted by $\mathfrak{B}(G, M, I)$. If $(A, B) \in \mathfrak{B}(G, M, I)$ and $g \in A$, then g is *an object from the concept* (A, B). Using properties form Table I one can show that for any object $g \in G$ and any attribute $m \in M$, the following equations hold: $(\{g\}'', \{g\}'), (\{m\}', \{m\}'') \in \mathfrak{B}(G, M, I)$. Concept $(\{g\}'', \{g\}')$ is called *an object concept of object g* whereas concept $(\{m\}', \{m\}'')$ is *an attribute concept of attribute m*. The object concept of any object $g \in G$ we denote by $\tilde{\gamma}(g)$ and the attribute concept of any attribute $m \in M$ we denote by $\tilde{\mu}(m)$. If $(A, B) = (\{g\}'', \{g\}')$, then object g is called *an own object* of concept (A, B), i.e. g posses only those attributes which are contained in B. Analogically, If $(A, B) = (\{m\}', \{m\}'')$, then attribute m is called *an own attribute of concept* (A, B), i.e. m is possessed only by those objects which are contained in A.

Let (G, M, I) be a formal context. On family $\mathfrak{B}(G, M, I)$ we define relation $\preccurlyeq$ in the following way:

$$(A_1, B_1) \preccurlyeq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2 (:\Leftrightarrow B_2 \subseteq B_1).$$

where $(A_1, B_1), (A_2, B_2) \in \mathfrak{B}(G, M, I)$. In this case (A_1, B_1) is called *a subconcept* of (A_2, B_2) and (A_2, B_2) is called *a superconcept* of (A_1, B_1). The relation $\preccurlyeq$ is a partial order on the family $\mathfrak{B}(G, M, I)$ and it is called *the hierarchical order* (or simply *order*). One can show that the family $\mathfrak{B}(G, M, I)$ ordered by the relation $\preccurlyeq$ is a complete lattice called the *concept lattice* of the context (G, M, I). We denote that lattice by $\underline{\mathfrak{B}}(G, M, I)$.

Having the relation $\preccurlyeq$ defined for the concept lattice one can equivalently consider two binary operations denoted as $\wedge$ and $\vee$ which can be expressed in terms, respectively, of the *infimum* and *supremum* with respect to relation $\preccurlyeq$. Namely, $a \wedge b = \inf\{a, b\}$ and $a \vee b = \sup\{a, b\}$. Thanks to the semantics of the infimum and supremum, these operations may be easily extended for arbitrary sets of arguments. The Basic Theorem on Concept Lattices [32], [6] shows that in the case of the concept lattice $\underline{\mathfrak{B}}(G, M, I)$ these operations and their generalizations for arbitrary sets of concepts are given by the following equations respectively:

$$\bigwedge_{i \in I}(A_i, B_i) = (\bigcap_{i \in I} A_i, (\bigcup_{i \in I} B_i)''),$$

$$\bigvee_{i \in I}(A_i, B_i) = ((\bigcup_{i \in I} A_i)'', \bigcap_{i \in I} B_i).$$

Therefore concept lattices can be viewed as hierarchical conceptual structures equipped with some operations on concepts and representing data stored in formal contexts. When the number of objects or the number of attributes in formal contexts are relatively small, then concept lattices can be used also for visualization of information stored in their formal contexts. In the next section we present a relatively small concept lattice representing a selection of social choice functions characterized by selected voting criteria, briefly introduced in section II.

IV. CONCEPT LATTICE OF SOCIAL CHOICE FUNCTIONS

This section is devoted to construction and structural analysis of proposed lattice of social choice functions. We start with definition of formal context of social choice functions on the basis of consideration conducted in the previous section. Let

$$\mathbb{SCF} := (\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$$

be a formal context where set $\mathrm{G}_{\mathbb{SCF}}$ comprises all voting procedures presented in Section II, set $\mathrm{M}_{\mathbb{SCF}}$ consists of selected criteria denoted by letters $A, ..., G$ in Section II, while set of pairs $\mathrm{I}_{\mathbb{SCF}}$ is the incidence relation presented in Table II.

Now, on the basis of formal context $\mathbb{SCF}$ we construct the concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ (Fig. 1) and study its properties[2].

[2] Diagrams of concept lattices are generated with usage of ConExp software by Serhiy A. Yevtushenko.

TABLE II
A FORMAL CONTEXT OF SELECTED SOCIAL CHOICE FUNCTIONS. ROWS CORRESPOND TO FORMAL OBJECTS WHICH ARE SOCIAL CHOICE FUNCTIONS AND COLUMNS CORRESPOND TO FORMAL ATTRIBUTES WHICH ARE SOME CRITERIA INTRODUCED IN SECTION II.

Voting procedures	Criteria						
	A	B	C	D	E	F	G
Amendment	×	×	×	×			
Copeland	×	×	×	×	×		
Dodgson	×		×		×		
Schwartz	×	×	×	×			
Max-min	×		×	×	×		
Plurality			×	×	×	×	
Borda		×		×	×	×	
Approval				×		×	×
Black	×	×	×	×	×		
Runoff		×	×		×		
Nanson	×	×	×		×		
Hare		×	×		×		
Coombs		×	×		×		

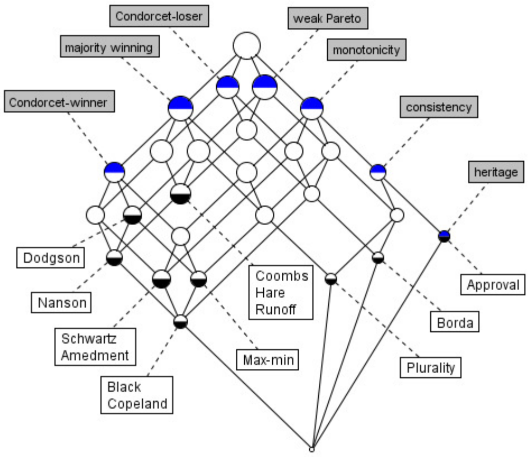


Fig. 1. Concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$. Half-black nodes represent object concepts while half-blue nodes represent attribute concepts .

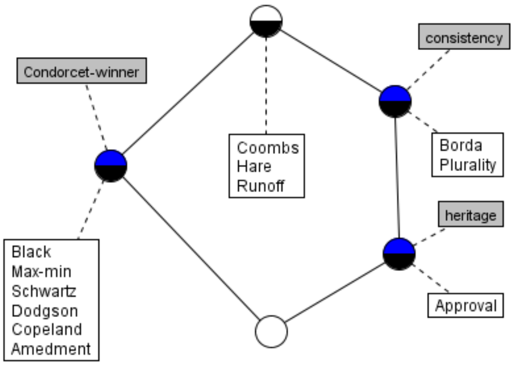


Fig. 2. Concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_1, \mathrm{M}_1, \mathrm{I}_1)$.

Fact 1: Concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ is nondistributive, i.e., it is not the case that $\forall C_1, C_2, C_3 \in \mathfrak{B}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ $C_1 \wedge (C_2 \vee C_3) = (C_1 \wedge C_2) \vee (C_1 \wedge C_3)$ nor $C_1 \vee (C_2 \wedge C_3) = (C_1 \vee C_2) \wedge (C_1 \vee C_3)$.

In order to show this one can consider the following formal context:

$$\mathbb{K}_1 := (\mathrm{G}_1, \mathrm{M}_1, \mathrm{I}_1),$$

where $\mathrm{G}_1 := \mathrm{G}_{\mathbb{SCF}}$, $\mathrm{M}_1 := \{Condorcet-winner, \; consistency, \; heritage\}$, and $\mathrm{I}_1 := \mathrm{I}_{\mathbb{SCF}} \; \cap (G_1 \times M_1)$. Now one can note that its concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_1, \mathrm{M}_1, \mathrm{I}_1)$, which is presented in Fig. 2, is a famous N5 lattice [2] and can be embedded into concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$. This by famous theorems by Dedekind and by Birkhoff implies that concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ is neither modular nor distributive (see e.g. [2]).

The fact that concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ lacks such regular property as distibutivity is not surprising. In fact it is quite rare phenomenon that real, non-manipulated data generate concept lattice possessing some regular properties. For example, one can consult concept lattice presented in [17] and generated from Threats Matrix (in German *Gefahrenmatrix*) used in commanding of tactical actions by German Fire Service [1], [7]. However, looking at the concept lattice presented in Fig. 1 one can note that the left part of this lattice diagram reveals some regularity. Namely, the concept lattice of social choice functions $\underline{\mathfrak{B}}(\mathrm{G}_1, \mathrm{M}_1, \mathrm{I}_1)$ contains as sublattices some distributive lattices or some Boolean algebras. Moreover, some subcontexts generated from context $\mathbb{SCF}$ of social choice functions generate lattices possessing some regularity.

Let

$$\mathbb{K}_2 := (\mathrm{G}_2, \mathrm{M}_2, \mathrm{I}_2),$$

be a formal context, where $\mathrm{G}_2 := \mathrm{G}_{\mathbb{SCF}}$, $\mathrm{M}_2 := \{Condorcet-winner, majority\ winning, \; weak\ Pareto, \; monotonicity\}$, and $\mathrm{I}_2 := \mathrm{I}_{\mathbb{SCF}} \; \cap (G_2 \times M_2)$, thus $\mathbb{K}_2$ is subcontext of social choice functions context $\mathbb{SCF}$. Then concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_2, \mathrm{M}_2, \mathrm{I}_2)$ is presented in Figure 3. One can note that concept lattice $\underline{\mathfrak{B}}(\mathbb{K}_2)$ is distributive.

For another example let us consider the following subcontext of social choice functions context $\mathbb{SCF}$:

$$\mathbb{K}_3 := (\mathrm{G}_3, \mathrm{M}_3, \mathrm{I}_3),$$

where $\mathrm{G}_3 := \mathrm{G}_{\mathbb{SCF}}$, $\mathrm{M}_3 := \{Condorcet-loser, \; majority\ winning, \; weak\ Pareto, monotonicity\}$, and $\mathrm{I}_3 := \mathrm{I}_{\mathbb{SCF}} \cap (G_3 \times M_3)$. One can note that concept lattice $\underline{\mathfrak{B}}(\mathbb{K}_3)$ presented in Figure 4 is Boolean lattice isomorphic to the power set algebra of a four-element set.

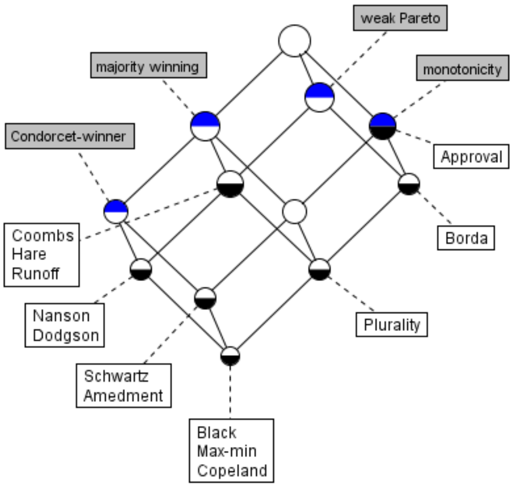


Fig. 3. Distributive concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_2, \mathrm{M}_2, \mathrm{I}_2)$.

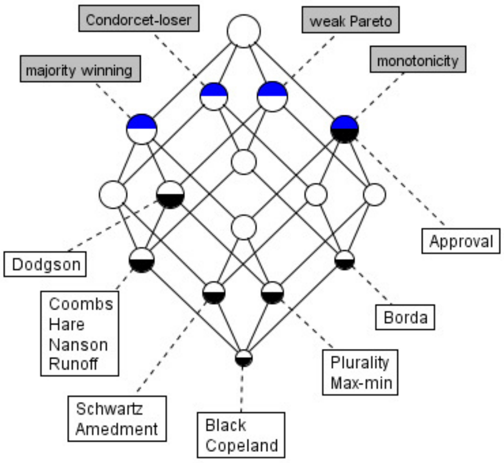


Fig. 4. Boolean concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_3, \mathrm{M}_3, \mathrm{I}_3)$.

V. Information Provided by Lattice of Social Choice Functions

This section is devoted to the analysis of social choice functions listed in Section II by means of the FCA. We will concentrate on issues of attribute independency, reduction of information and on attribute implications derived from proposed formal context of social choice functions by means of FCA.

A. Attribute Independency

Let us start our consideration with the independency of attributes in the formal context $\mathbb{SCF}$ of social choice functions (voting procedures). Let (G, M, I) be the formal context. Attributes in $X \subseteq M$ are independent if there are no trivial dependencies between them i.e. functional (or ordinal) dependencies where set of attributes Y is functionally (ordinally) dependent on set of attributes X and $Y \subseteq X$. Following [29] we recall:

Lemma 1: Attributes are independent if they span a hypercube in a concept lattice.

For example concept lattice of social choice functions has four coatoms and these formal concepts as coatoms are also attribute concepts, namely these are $\tilde{\mu}(majority\ winning)$, $\tilde{\mu}(Condorcet - loser)$, $\tilde{\mu}(weak\ Paretto)$ and $\tilde{\mu}(monotonicity)$. These attributes are independent since every three attributes from this set (by their attribute concepts) span a hypercube in the concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$. In fact, in concept lattice $\underline{\mathfrak{B}}(\mathrm{G}_{\mathbb{SCF}}, \mathrm{M}_{\mathbb{SCF}}, \mathrm{I}_{\mathbb{SCF}})$ it is easier to characterize sets of attributes which are not independent: in concept lattice $\underline{\mathfrak{B}}(\mathbb{SCF})$ there are four two-element chains of attribute concepts, namely:

- $\{\tilde{\mu}(Condorcet - winner), \tilde{\mu}(majority\ winning)\}$,
- $\{\tilde{\mu}(heritage),\ \tilde{\mu}(consistency)\}$,
- $\{\tilde{\mu}(heritage),\ \tilde{\mu}(monotonicity)\}$,
- $\{\tilde{\mu}(consistency),\ \tilde{\mu}(monotonicity)\}$.

Sets of attributes which are not independent are exactly sets of attributes containing at least one pair of attributes such that their attribute concepts are contained in one of the above two-element chains.

B. Reducibility of Information

Reduction of information is one of the main advantages of formal concept analysis. Here we describe reduction of information within concept lattice of social choice functions.

Let us recall that context (G, M, I) is called *clarified* if for any objects $g, h \in G$, $g' = h'$ implies $g = h$ and for any attributes $m, n \in M$, $m' = n'$ implies $m = n$. Now let us note the following facts:

Fact 2: Context $\mathbb{SCF}$ is not clarified.
It is so since, e.g., $\{Coombs\}' = \{Runoff\}'$ but obviously $Coombs \neq Runoff$. However, for the set of all attributes of context $\mathbb{SCF}$ (denoted by $\mathrm{M}_{\mathbb{SCF}}$) one of the necessary conditions for a clarified context holds, i.e.:

Fact 3: For all attributes (criteria) $m, n \in \mathrm{M}_{\mathbb{SCF}}$ the following implication holds:

$$\{m\}' = \{n\}' \Rightarrow m = n.$$

It is easily seen in Fig. 1 where there are no two criteria determining the same attribute concept.

Let us recall that for any formal context (G, M, I), object $g \in G$ is *reducible* if its object concept $\tilde{\gamma}(g)$is *supremum-reducible*, i.e., can be represented as the supremum of strictly smaller concepts what implies that concept $\tilde{\gamma}(g)$ has no unique lower neighbour in concept lattice $\underline{\mathfrak{B}}(G, M, I)$. Analogously, for any context (G, M, I), attribute $m \in M$ is *reducible* if its attribute concept $\tilde{\mu}(m)$ is *infimum-reducible*, i.e., can be represented as infimum of strictly greater concepts, i.e. concept $\tilde{\gamma}(m)$ has no unique upper neighbour in concept lattice $\underline{\mathfrak{B}}(G, M, I)$. Now one can note that:

Fact 4: The social choice function Dodgson (more formally: the object representing this social function in the lattice) is reducible. The rest of social choice functions from the context $\mathbb{SCF}$ are irreducible.

One can note that social choice function $Dodgson$ as a formal object is reducible since its object concept is a supremum of two different concepts:

$$\tilde{\gamma}(Dodgson) = \tilde{\gamma}(Nanson) \vee \tilde{\gamma}(Max - min),$$

in $\underline{\mathfrak{B}}(G, M, I)$, the concept lattice of social choice functions. Namely the object concept $\tilde{\gamma}(Dodgson)$ is the lattice union of the object concepts determined by voting procedures Nanson and the object concept determined by the voting procedure Max-min. This observation may be expressed in a different way by saying that the Dodgson social choice function is Pareto-dominated by the Nanson and Max-min functions. Such a statement is justified as the attributes of the functions express their desired properties and thus the aforementioned dominance is here well-defined.

Concerning the rest of voting procedures from the social choice functions context $\mathbb{SCF}$, their object concepts have exactly one lower neighbour in the concept lattice of social choice functions, thus by Proposition 2 of [6] these object concepts are irreducible.

Fact 5: All attributes are irreducible.

One can note that every attribute concept determined by a criterion from the context $\mathbb{SCF}$ has exactly one upper neighbour what in the light of Proposition 2 of [6] shows that all attribute concepts in the social choice context $\mathbb{SCF}$ are infimum-irreducible.

Fact 6: In the formal context $\mathbb{SCF}$ of social choice functions there is only one concept which is both object concept and attribute concept.

In order to show this one can note that:

$$\tilde{\gamma}(Approval) = \tilde{\mu}(heritage),$$

i.e. social choice function (voting procedure)$Approval$ and voting criterion $heritage$ determine the same concept in the concept lattice of social choice functions. It stems from the fact, that the property (attribute) $heritage$ distinguishes the voting procedure $Approval$ from the other procedures and, at the same time, property $heritage$ is satisfied only by $Approval$.

C. Implications holding in the Context of Social Choice Functions

The FCA based analysis of voting procedures brings in another potentially interesting insight into their functioning. Namely, *implications* holding in the context of social choice functions may provide social choice theorists with valuable information. Those implications are not laws derived by theoretical considerations directly from knowledge gathered in the framework of the social choice theory but they are derived from the description of the voting procedures created by politicians and social choice theorist and expressed in terms of different properties postulated by social choice theorists.

Let us recall the notion of attribute implication. Informally, implications between attributes are the statements of the following form "Every object with the attributes a, b, c, ... also has the attributes x, y, z, ... " [6]. Formally speaking, an implication between attributes in context (G, M, I) is a pair of subsets of the attribute set M. If $A, B \subseteq M$, then implication between A and B is denoted by $A \rightarrow B$. An implication between attributes may or may not hold in a given formal context. Instead of formal definition of implication which holds in a given formal context we recall a transparent characterization of this notion given in Proposition 19 in [6]: an implication $A \rightarrow B$ holds in (G, M, I) if and only if $B \subseteq A''$.

Looking at concept lattice of social choice functions presented in Fig. 1 one can note relatively large number of nontrivial implications between singular attributes which are enlisted below:

- $\{Condorcet - winner\} \rightarrow \{majority\ winning\}$
- $\{heritage\} \rightarrow \{consistency\}$
- $\{consistency\} \rightarrow \{monotonicity\}$

One of the formal reasons for that is the fact that five of seven attribute concepts are involved into two chains maximal with respect to the property that they consist only of attribute concepts, namely the following two chains:

- $\{\tilde{\mu}(Condorcet - winner), \tilde{\mu}(majority\ winning)\}$
- $\{\tilde{\mu}(heritage),\ \tilde{\mu}(consistency),\ \tilde{\mu}(monotonicity)\}$.

Finally, one can note that maximal antichains consisting only of attribute concepts have four elements which seems to be a relatively high number compared to the fact that maximal antichains in concept lattice of social choice functions analyzed within this paper have seven elements, i.e. the width of the concept lattice of social choice functions is 6.

VI. Conclusions

The concept lattice of social choice functions constructed and analyzed within this paper has an interesting and pretty regular structure. Despite the fact that itself it is nondistributive lattice it contains quite a few regular sublattices, including Boolean, distributive and modular lattices of a quite large size compared to the size of the whole lattice.

From the perspective of social choice theory interesting is a comparison of the applicability of formal concept analysis methods and rough sets theory methods. The latter has been already reported in the literature [3], [13]. One of particular dimensions of such comparison will be the issue of information reduction in both approaches.

And last but not least, an interesting issue worth of further research from the perspective of FCA is to find out whether observations reported in the paper can be interpreted in a deeper way in the language of the social choice theory. Thus, the further research in this direction can be focused on one task: to understand the observed phenomena presented in this paper in terms of social choice theory.

References

[1] Bundesant für Bevölkerungsschutz und Katastrophenhilfe: Feuerwehr - Dienstvorschrift 100 Führung und Leitung im Einsatz: Fuhrungssystem, FwDV 100 Stand: 10 März 1999.

[2] S. Burris and S. Sankappanavar, "A Course in Universal Algebra". The Millennium Edition updated in 2012 is freely available on-line at the web page University of Waterloo, https://www.math.uwaterloo.ca/~snburris/htdocs/UALG/univ-algebra2012.pdf.

[3] M. Fedrizzi, J. Kacprzyk, and H. Nurmi, "How different are social choice functions: a rough sets approach", *Quality & Quantity: International Journal of Methodology*, vol. 30(1), 1996, pp. 87–99.

[4] P. C.Fishburn, *The Theory of Social Choice Functions*, Princeton University Press, Princeton, 1973.

[5] P. C. Fishburn, "Social choice functions", *Society for Industrial and Applied Mathematics Review* vol. 16(1), 1974, pp. 63–90.

[6] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Heidelberg, 1999.

[7] A. Graeger, U. Cimolino, H. de Vries, and J. Sümersen, *Einsatz- und Abschnittsleitung: Das Einsatz-Füchrungs-System (EFS)*, Ecomed Sicherheit, 2009.

[8] D. I. Ignatov and L. Kwuida, "On Shapley value interpretability in concept-based learning with formal concept analysis", *Annals of Mathematics and Artificial Intelligence* vol. 99, 2022, pp. 1197–1222.

[9] J. Kacprzyk, "Group decision making with a fuzzy majority", *Fuzzy Sets and Systems* vol. 18, 1986, pp. 105—118.

[10] J. Kacprzyk, M. Fedrizzi, and H. Nurmi, "Group decision making and consensus under fuzzy preferences and fuzzy majority", *Fuzzy Sets Systems* vol. 49 1992, pp. 21–31.

[11] J. Kacprzyk, J. Merigó, H. Nurmi, and S. Zadrożny, "Multi-agent systems and voting: how similar are voting procedures", in *19th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMU 2020*, Springer, Cham, 2020, pp. 172–184.

[12] J. Kacprzyk, H. Nurmi, and S. Zadrożny, "Reason vs. rationality: from rankings to tournaments in individual choice", in *Transaction on Computational Collective Intelligence, LNCS*, vol. 10480, 27, 2017, pp. 28–39.

[13] J. Kacprzyk, H. Nurmi, and S. Zadrożny, "Towards a comprehensive similarity analysis of voting procedures using rough sets and similarity measures", in: *Rough Sets and Intelligent Systems - Professor Zdzislaw Pawlak in Memoriam*, vol. 1, Springer, 2013, pp. 359–380.

[14] J. Kacprzyk and S. Zadrożny, "Towards a general and unified characterization of individual and collective choice functions under fuzzy and nonfuzzy preferences and majority via the ordered weighted average operators" *International Journal of Intelligent Systems* vol. 24, 2009, pp. 4–26.

[15] J. Kacprzyk and S. Zadrożny, "Towards human consistent data driven decision support systems using verbalization of data mining results via linguistic data summaries", *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 58(3), 2010, pp. 359–370.

[16] J. S. Kelly, *Social choice theory*, Springer, Berlin, 1988.

[17] A. Krasuski and P. Wasilewski, Outlier Detection by Interaction with Domain Experts. *Fundamenta Informaticae* vol. 127(1-5), 2013, pp. 529–544.

[18] H. Nurmi, *Comparing Voting Systems*, D. Reidel, Dordrecht, 1987.

[19] H. Nurmi, *Voting Paradoxes and How to Deal With Them*, Springer, Heidelberg, 1999.

[20] H. Nurmi, "The choice of voting rules based on preferences over criteria", in *Outlooks and Insights on Group Decision and Negotiation*, Springer, Cham, 2015, pp. 241–252.

[21] H. Nurmi and J. Kacprzyk, "On fuzzy tournaments and their solution concepts in group decision making", *European Journal of Operational Research*, vol. 51(2), 1991, pp. 223–232.

[22] H. Nurmi, J. Kacprzyk and M. Fedrizzi, "Probabilistic, fuzzy and rough concepts in social choice", *European Journal of Operational Research*, vol. 95, 1996, 264–277.

[23] H. Nurmi, J. Kacprzyk, and S. Zadrożny, "Voting systems in Theory and Practice", in *Colllective Decisions: Theory, Algorithms and Decision Support Systems*, Studies in Systems, Decision and Control, vol. 392, Springer, Cham, 2022, pp.3–16.

[24] Z. Pawlak, "Information Systems – theoretical foundations", *Information systems*, 6, 1981, pp. 205–218.

[25] Z. Pawlak, "Rough sets", *International Journal of Computing and Information Sciences* 18, 1982, pp. 341–356.

[26] Z. Pawlak, *Rough sets. Theoretical Aspects of Reasoning About Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[27] A. Revenko and S. O. Kuznetsov, "Attribute Exploration of Properties of Functions on Ordered Sets", in *7th International Conference on Concept Lattices and Their Applications*, 2010, pp. 313–324.

[28] A. Revenko and S. O. Kuznetsov, "Attribute Exploration of Properties of Functions on Sets", *Fundamenta Informaticae* vol. 115(4), 2012, pp. 377-394.

[29] G. Stumme, "Conceptual knowledge discovery and data mining with formal concept analysis", Tutorial slides at the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML/PKDD'2002, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f9a8f4529aca992903823150dda77d9a89d193a2, 2002.

[30] P. Wasilewski, "Concept lattices vs. Approximation spaces", in *10th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Lecture Notes in Artificial Intelligence 3641, 2005, pp. 114–123.

[31] P. Wasilewski, "Algebras of Definable Sets vs. Concept Lattices", *Fundamenta Informaticae*, 167(3), 2019, pp. 235–256.

[32] R. Wille, "Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts", in: *Ordered Sets*. NATO Advanced Study Institutes Series, vol. 83, Reidel, Dordrecht, 1982, pp. 445–470.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1205–1210

DOI: 10.15439/2023F9901
ISSN 2300-5963 ACSIS, Vol. 35

Postquantum symmetric cryptography inspired by neural networks

Wojciech Węgrzynek*, Paweł Topa†
Faculty of Computer Science, Electronics and Telecommunications,
AGH University of Kraków
Kraków, Poland
Email: *wegrzynek@student.agh.edu.pl,
†topa@agh.edu.pl

***Abstract*—We introduce a novel approach to postquantum symmetric encryption that allows us to modify and continue to use any encryption scheme. By composing the encryption and decryption functions with the evaluation of arbitrarily wide neural networks we are able to verify that anyone performing these functions has access to at least a certain amount of memory. Since the number of qubits in quantum computers has been relatively slow-growing, this provides us security from the Grover's search attack, and any attack utilizing a similar oracle circuit.**

***Index Terms*—post-quantum cryptography, symmetric key cryptography, encryption, neural networks**

I. Introduction

THE development of quantum computers is likely to lead to major breakthroughs in many areas of science and engineering. Cryptography is one area where this breakthrough is already evident. Although we do not yet have a quantum computer capable of cracking the 2048-bit RSA key, the world is preparing for that moment. In 2016, the US NIST announced a competition for a post-quantum public key algorithm. In June 2022, after three rounds of review, four algorithms implementing key encapsulation (Crystals-Kyber) and digital signature functionality (Crystals-Dilithium, Falcon, Sphincs+) were selected.

Secret key cryptography is much less threatened by quantum computers. The Grover algorithm is only able to halve the security strength of the AES algorithm. This means that AES with a 256-bit key will be as secure as AES with a 128-bit key is today.

This is still enough of a security margin not to change the cipher, which is the "workhorse" of the Internet, too quickly.

However, it is worth considering all possible directions for post-quantum secret key cryptography. Here, we present an idea of making Grover's search attack more challenging in terms of qubits of memory needed on a quantum computer.

II. State-of-the-art

Currently, the most widely used symmetric cipher is AES [1], which is susceptible to the Grover's search attack. AES exists in three variants: AES-128, AES-196, and AES-256 named after the length of the binary key string. The key spaces are then of size 2^{128}, 2^{196}, and 2^{256} respectively. Since Grover's search attack effectively halves the exponent of the key space size, AES-256 would be reduced to the security level of AES-128 and the remaining two variants would become insecure by the previous standard. Since AES-128 is currently considered secure, switching to AES-256 is the solution provided in [2] and [3]. Additionally, for purposes requiring AES-256 level security, [3] extends AES to include a 512 key length variant.

It is first worth noting that there already exist solutions that are secure against the Grover's search attack, in particular AES-QPP, which is a variant of AES in which the SubBytes and AddRoundKey are replaced by a Quantum Permutation Pad operation, granting quantum safety [4], or Saturnin, which is a block cipher that has been specifically designed for the purpose of being quantum-safe while also maintaining lightweight properties making it more suitable for IoT applications [5]. In comparison to them, however, AES has the benefit of having been exposed to extensive analysis by the public.

Using neural networks cryptographic solution, at least academically, is not a novel concept either. In [6] the authors utilize recurrent neural networks of a specific shape to define a symmetric cipher. In [7] a cryptographic hash function is defined based on the evaluation of a neural network with randomized matrices.

III. Preliminaries and Definitions

In this section, we introduce the terms and notation we will use throughout this article. Since this work exist at the intersection of a few different subfields within computer science, we divide it into subsections

A. Algebraic notation

For the sake of simplicity of notations in this subsection, we will introduce some shorthands that we will use throughout this article.

Let $v \in \mathcal{F}^n$ be an n-dimensional vector over some field $\mathcal{F}$, by $v[i]$ for $0 \leq i < n$ we will denote the i-th element of the vector.

A **Galois field** is synonymous to a finite field and we use the notation $GF(q)$ to denote a Galois field of size q. By a well-known algebraic theorem when q is of the form $q = p^n$ for some prime p and some $n \in \mathbb{N}_+$ then $GF(q)$ exists and is unique in the sense of isomorphisms. For any other size a finite field does not exist, thus $GF(q)$ is well-defined only for powers of a prime number.

B. Cryptography

Here we attempt to formalize some notions present in cryptography. Note that it is particularly difficult to reflect the practical nature of this field. We nonetheless make the attempt in order to provide a mathematical argument for the correctness of our claims.

Definition 1. *A **symmetric cipher** is a tuple $(e, d, \mathcal{P}, \mathcal{C}, \mathcal{K})$ such that:*

- *$\mathcal{P}$ is a set of plaintexts,*
- *$\mathcal{C}$ is a set of ciphertexts,*
- *$\mathcal{K}$ is a key space,*
- *$e : \mathcal{P} \times \mathcal{K} \to \mathcal{C}$,*
- *$d : \mathcal{C} \times \mathcal{K} \to \mathcal{P}$,*
- *for any $p \in \mathcal{P}$, $k \in \mathcal{K}$: $d(e(p, k), k) = p$.*

A cipher $\sigma = (e, d, \mathcal{P}, \mathcal{C}, \mathcal{K})$ is going to be **secure** if the following conditions are met:

1) Given only the value of $e(p, k)$ it is impossible to reliably compute p faster than the naive approach of iterating through all of the key space.
2) Given only the values of $e(p, k)$ and p it is impossible to reliably compute k faster than the naive approach of iterating through all of the key space.

Any algorithm or process that proves a cipher to be insecure is called an attack. Note that this is one place where there is a discrepancy between this formal definition and the practical notion of security — many ciphers are still considered practically secure despite existing attacks because those attacks are proved to be impractical.

An attack that violates the first property will be called a **ciphertext-only attack** and an attack that violates the second (but not the first) will be called a **known-plaintext attack**.

C. Neural networks

Since the main result of this work is heavily inspired by neural networks, we feel the need to define some notions from that field of study. Let us start with defining, arguably, the simplest type of neural network — the multilayer perceptron. This will be the only type of neural network we will refer to in this work, so we will sometimes use the term neural network as a synonym for a multilayer perceptron, but we note that in the wider topic such equivalence would be false.

Definition 2. *A **multilayer perceptron** over the field $\mathcal{F}$ is a function $f : \mathcal{F}^{n_1} \to \mathcal{F}^{n_{d+1}}$ of form $f = l_1 \circ l_2 \circ \cdots \circ l_d$, where $d \in \mathbb{N}_+$ is the **depth** of the neural network. Each function $l_i : \mathcal{F}^{n_i} \to \mathcal{F}^{n_{i+1}}$ for $1 \le i \le d$ (which we will call a **layer**) must be in the form $l_i = m_i \circ a_i$ where:*

- *$m_i : \mathcal{F}^{n_i} \to \mathcal{F}^{n_{i+1}}$ is a linear transformation using the matrix $M_i \in \mathcal{F}^{n_i \times n_{i+1}}$,*
- *$a_i : \mathcal{F}^{n_{i+1}} \to \mathcal{F}^{n_{i+1}}$ is a non-linear transformation (we will sometimes call a_i the **activation** function).*

Note that the typical definition is usually constrained to $\mathcal{F} = \mathbb{R}$ and also requires the activation function to be differentiable and monotonic on each element. This is a conscious choice on our part, since we need to generalize this concept to other fields, because of the impracticality of representing real numbers on computers.

D. Quantum computing

A **qubit** is the most fundamental unit of quantum information. The state of a qubit is any vector $\psi \in \mathbb{C}^2$ with its norm equal to 1. We traditionally denote the state of a qubit as a ket in bra-ket notation (also called Dirac notation), like so $|\psi\rangle$. Two special states, forming an orthogonal basis, are usually distinguished:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},\ |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

If more than one qubit exists, the state of such a system is the Hadamard product of the states of each qubit. In bra-ket donation, the Hadamard product of two states $|\psi\rangle$ and $|\phi\rangle$ is denoted by $|\psi\rangle |\phi\rangle$, or sometimes $|\psi\phi\rangle$. For an n-qubit system we also define the orthogonal basis $\{|0\rangle_n, |1\rangle_n, \cdots, |2^n - 1\rangle_n\}$ where $|i\rangle_n$ is a vector such that $|i\rangle_n [i] = 1$.

A quantum gate O, acting on an n-qubit state, is any invertible, unitary $2^n \times 2^n$ matrix over complex numbers. The application of O on the state $|\phi\rangle$ is the product of the matrix and vector, and is denoted as $O |\psi\rangle$.

Notice that because of the definition of a quantum gate, to uniquely identify a quantum gate, it suffices to define the results of applying it to some basis. Below we use this fact to define some gates that will be referenced throughout this article.

The NOT gate, or the X gate is a quantum gate acting on a single qubit, and is defined as follows:

- $X |0\rangle = |1\rangle$,
- $X |1\rangle = |0\rangle$.

The Toffoli or the CCX gate is a quantum gate acting on a 3-qubit state. The Toffoli gate acts according to the following rules:

- $CCX |11a\rangle = |11\rangle (X |a\rangle)$,
- $CCX |abc\rangle = |abc\rangle$, if $|ab\rangle \neq |11\rangle$.

IV. Cryptanalysis with quantum computers

The development of quantum computing technology poses a threat to our existing, widely used, ciphers. In the field of public-key cryptography, there is, for example, the famous Shor's algorithm, the usage of which can break RSA (Diffie-Hellman, ElGamal and elliptic curve cryptography too) in polynomial time. For private-key encryption the known quantum attacks are much less spectacular, nonetheless, they do exist and are worth investigating.

Grover's search algorithm is a quantum computing algorithm that, given an oracle circuit Q, over $n + 1$ qubits, with the property that for any $x \in \{0, 1, \cdots, N-1\}$, and for any $a \in \{0, 1\}$

$$Q|x\rangle|a\rangle = \begin{cases} |x\rangle X|a\rangle, \text{ iff } x = a \\ |x\rangle|a\rangle, \text{ otherwise} \end{cases}$$

is able to find a with only $O(\sqrt{N})$ invocations of Q. This is an obvious improvement over an analogous situation in classical computing where the fastest such algorithm needs $O(N)$ invocations.

For any symmetric cipher $\sigma = (e, d, \mathcal{P}, \mathcal{C}, \mathcal{K})$ we may then define the following known plaintext attack:

1) Define an oracle Q that given k_1 as input, returns $e(p, k) = e(p, k_1)$.
2) Use Grover's search algorithm to compute k.

This takes $O(\sqrt{|\mathcal{K}|})$ invocations of Q, making it an attack on σ. Note that this attack is only possible if the attacker is able to execute the oracle circuit — which is what we make use of in this work.

V. The Proposed Solution

We introduce a novel approach that allows one to secure any private key cipher against a Grover attack, provided that both communicating sides have more memory bits on their machines than an attacker might have on their quantum computer. The algorithm works by modifying a scheme σ to form a scheme σ', such that the decryption function σ' requires at least a set number of bits of memory to compute — thus preventing the formation of the Grover oracle. In this section, we will detail how we achieve this.

Let's start with an arbitrary encryption scheme $\sigma := (e, d, \mathcal{P}, \mathcal{C}, \mathcal{K})$ and a bijection $f : \mathcal{C}^n \to \mathcal{P}^n$, for some n, the details of which we outline in Section VI. The scheme $\sigma' := (e', d', \mathcal{P}^n, \mathcal{C}^n, \mathcal{K})$ is then defined such that for a key $k \in \mathcal{K}$:

- $e'((p_1, p_2, \cdots, p_n), k) := (e(f(e(p_1, k)), k), \cdots, e(f(e(p_n, k)), k))$,
- $d'((c_1, c_2, \cdots, c_n), k) := (d(f^{-1}(d(c_1, k)), k), \cdots, d(f^{-1}(d(c_n, k)), k))$.

In other words, to encrypt a message, we concatenate n messages encrypted with σ, pass the output through f, split it back into n messages, and encrypt them again. To decrypt a message we then: decrypt the ciphertexts using σ, concatenate them, pass the output through f^{-1}, split them, and decrypt them using σ.

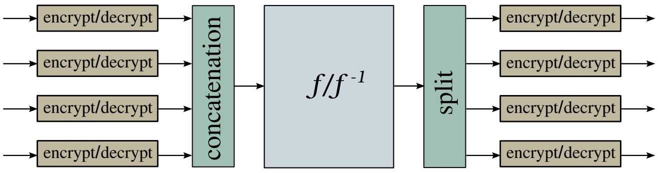


Figure 1: The proposed solution: securing cipher against Grover's attack with nonlinear bijective function f

Let us define a k-mixing function:

Definition 3. *A function $g : GF(2)^n \to GF(2)^n$ is k-**mixing** if, it is impossible to compute values for any subset of output bits, given only k of input bit values.*

Let us then state the following observation

Observation 1. *The proposed above solution produces a scheme that is secure against the Grover attack on machines with at most k qubits if the function f is k-mixing.*

Proof. Since the Grover attack requires the attacker to formulate an oracle circuit, and the attacker has at most k qubits available, this oracle function would have to compute values of some output bits given only k of the input bits, since that is all the attacker could store in memory. □

All that is now needed is to propose a family of bijective functions that will contain k-mixing functions for arbitrarily large k-s. At first glance, traditional multilayer perceptrons could be utilized. Indeed, a sigmoid activation function and invertible weight matrices would guarantee bijectivity, and the expressive power of neural networks should, at least intuitively, correlate to the mixing property, if random weights were used. However, with this approach, a practical issue arises - the naive implementation of such a function utilizing floating point arithmetic, would almost certainly not be able to produce an exact mathematical inverse. Instead, in the following section, we introduce a structure that operates on the finite field $GF(2)$ in a similar way that traditional multilayer perceptrons operate on $\mathbb{R}$, which we then may use instead.

VI. Definition of the f Function

Let us define a Galois neural network:

Definition 4. *A **Galois neural network** is a multilayer perceptron over a Galois field.*

Let's observe that such a Galois network may serve as our f function, as long as we design it to be a bijection. To do so, it suffices to design a layer that is bijective. For the linear part of each layer, since we have total control of the values of the matrices, it suffices to find an invertible matrix M. For the activation function, this is less trivial, partially since, unlike traditional neural networks, all bijections over the field $GF(2)$ are linear, and so this activation function must in some way require interaction between elements of a vector. Below, we detail the design of the family of functions that fit these requirements.

We define a family of functions that work analogously to the iterative application of the Toffoli gate in quantum computers.

Definition 5. *A function $t_{m,k,l} : GF(2)^n \to GF(2)^n$ will be called a **Toffoli function** acting on bit m with control bits k, l (where m, k, l are pairwise different and $k < l$) if for all $v \in GF(2)^n$:*

1) $\forall_{i \neq m} t_{m,k,l}(v)[i] = v[i]$,
2) $(t_{m,k,l}(v)[m] \neq v[m]) \iff (v[k] = v[l] = 1)$.

The following lemma is then true:

Lemma 1. *Let $f : GF(2)^n \to GF(2)^n$ be a composition of at least one Toffoli function $f = t_{m_1,k_1,l_1} \circ t_{m_2,k_2,l_2} \circ \cdots \circ t_{m_i,k_i,l_i}$, such that for any $1 \leq \alpha, \beta \leq i$ where $\alpha \neq \beta$, at least one of the following is true:*

1) $m_\alpha \neq m_\beta$,
2) $k_\alpha \neq k_\beta$ *or* $l_\alpha \neq l_\beta$.

Then f is

1) *bijective,*
2) *nonlinear.*

Proof. The function is trivially bijective since $t_{m_1,k_1,l_1} = t^{-1}_{m_1,k_1,l_1}$.

Let us then prove nonlinearity, and suppose, by contradiction f is linear. There have to then exist $A \in GF(2)^{n \times n}, b \in GF(2)^n$ such that $f(v) = Av + b$ for any $v \in GF(2)^n$. Take t_{m_1,k_1,l_1} and consider a vector v_0, that is defined as follows:

- $v_0[i] = 0$ for all $i \notin \{k_1, l_1\}$,
- $v_0[i] = 1$ for all $i \in \{k_1, l_1\}$.

We know, from the definition of f that $v_0[m_1] = 1$, on the other hand $v_0[m_1] = (Av_0 + b)[m_1] = A[m_1]v_0 + b[m_1] = A[m_1][k_1] + A[m_1][l_1] + b[m_1]$. Notice that $b[m_1]$ must equal 0 since $f(0) = 0$, by definition of a Toffoli function. That means exactly one of $A[m_1][k_1], A[m_1][l_1]$ must equal one. Suppose, without loss of generality it is $A[m_1][k_1]$. Consider then a vector v_1, that only has a one as its k_1-th element. By definition of a Toffoli function, $f(v_1) = v_1$, but since $A[m_1][k_1] = 1$ and $b[k_1] = 0$, $f(v_1)[m_1] = 1 \neq v_1[m_1]$, thus we have a contradiction. □

Notice how Lemma 1 guarantees exactly the requirements for an activation function for a bijective Galois neural network.

We now formulate a conjecture which, if true, would guarantee security by Observation 1.

Conjecture 1. *For each $w \in \mathbb{N}_+$ there exists a w-mixing Galois neural network.*

We will try to argue for the validity of this conjecture by applying some statistical tests to randomly generated Galois neural networks in Section VIII.

VII. Performance Test

In this section, we evaluate the applicability of this approach through performance test results. We note, however, that the implementation used for these tests has likely yet to be fully optimized, one can expect enhancement in that regard with further development.

The Galois neural networks have been implemented using bitwise logical operations in Numpy [8] and the correctness of this approach was validated against the Galois package [9] for Python.

We will contain ourselves to single-layer NNs for the purposes of this evaluation, since increasing the number of layers since increasing the number of layers results in almost exactly linear growth of computation time.

To establish a frame of reference, we will contrast these results with the AES-256 implementation found in the PyCryptodome [10] package, note however that while we choose to compare to AES, because of its popularity, our solution is designed to work with any cipher, in particular less performant ones, where the performance gap might be less noticeable.

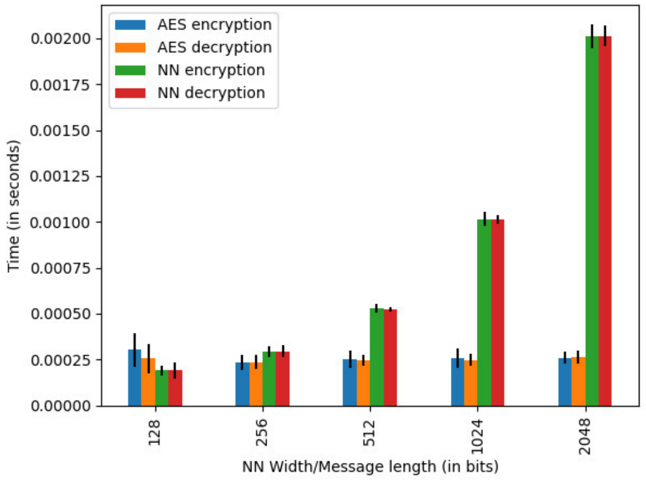


Figure 2: Results of performance tests. Encryption/decryption time contrasted with that of AES.

We compare the performance of a Galois neural network of a certain width on a single input with the performance of AES on that same input. For each of the lengths 128, 256, 512, 1024, and 2048 we ran 20 randomized messages, encrypted and decrypted them, and recorded the means and standard deviation. The tests were run on a mid-range laptop.

The results of these tests can be seen in Figure 2. Unsurprisingly, as Galois neural network evaluation has cubic complexity with respect to the width, the neural network execution time quickly trumps that of AES. However, this level of increased execution time might still be acceptable for applications where speed is not of high priority. Nonetheless, for most usecases, this data indicates a need for optimization, especially past the 512 mark.

Note that the results for AES do not seem to increase as the length of the messages increases. This is likely due to some kind of parallel execution.

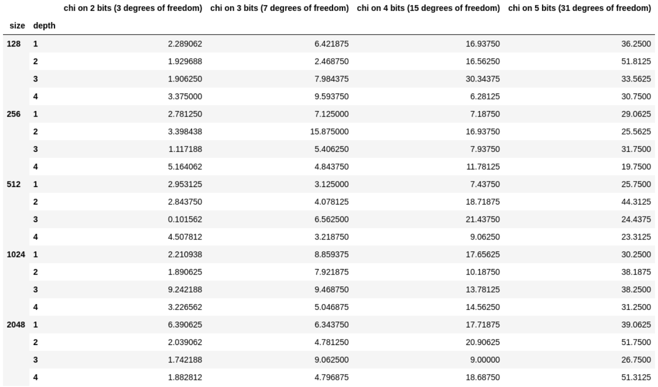
VIII. Statistical Tests

To investigate the validity of Conjecture 1 we performed a suite of tests to examine some properties of Galois neural networks with randomized but constant weights.

A. *Uniform distribution test*

First, we propose the following test. Let w be the (even) width of a Galois neural network, randomly select the first $w/2$ bits of the input. Then repeatedly randomly select the remaining $w/2$ bits, concatenate all w bits together, compute the output of the network, and record the first few bits of the output. For a $w/2$-mixing function, we would expect a roughly uniform distribution of output values.

We performed this test, recording the counts for each output value, we then used the chi-squared test to assess how alike



		chi on 2 bits (3 degrees of freedom)	chi on 3 bits (7 degrees of freedom)	chi on 4 bits (15 degrees of freedom)	chi on 5 bits (31 degrees of freedom)
size	depth				
128	1	2.289062	6.421875	16.93750	36.2500
	2	1.929688	2.468750	16.56250	51.8125
	3	1.906250	7.984375	30.34375	33.5625
	4	3.375000	9.593750	6.28125	30.7500
256	1	2.781250	7.125000	7.18750	29.0625
	2	3.398438	15.875000	16.93750	25.5625
	3	1.117188	5.406250	7.93750	31.7500
	4	5.164062	4.843750	11.78125	19.7500
512	1	2.953125	3.125000	7.43750	25.7500
	2	2.843750	4.078125	18.71875	44.3125
	3	0.101562	6.562500	21.43750	24.4375
	4	4.507812	3.218750	9.06250	23.3125
1024	1	2.210938	8.859375	17.65625	30.2500
	2	1.890625	7.921875	10.18750	38.1875
	3	9.242188	9.468750	13.78125	38.2500
	4	3.226562	5.046875	14.56250	31.2500
2048	1	6.390625	6.343750	17.71875	39.0625
	2	2.039062	4.781250	20.90625	51.7500
	3	1.742188	9.062500	9.00000	26.7500
	4	1.882812	4.796875	18.68750	51.3125

Figure 3: χ^2 values from 1024 iterations of the uniform distribution tests.

these outputs are to those from a uniform distribution. The results values of those tests can be viewed in figure 3. The critical values for significance level 0.01 and the given degrees of freedom are (approx.) 11.345, 18.475, 30.578, 52.191 respectively. Thus all those tests failed to reject the null-hypothesis with a significance level of at least 0.01.

B. The bit flip test

Another property a mixing function should have, is that a small change in the input should have a large impact on the output. In [11] this property is tested in the following way:

- Consider x a random input to the function, record $f(x)$.
- Change a random bit of x, call it x', record $f(x')$.
- Compute the number of bits that are different between $f(x)$ and $f(x')$.

If f were a random permutation, we would expect the number of differing bits to follow a binomial distribution. Since we expect f to behave like a random permutation, we may use the Student's t-test to check if that is the case.

We performed 1000 samples of this test for every combination of $1, 2, 3$, and 4 deep and 128, 256, 512, 1024, and 2048 wide Galois neural networks. We then computed the Student's t test z statistic for each of those combinations and found the highest of these values to be less than 0.002, since for significance level 0.01 and 999 degrees of freedom, the passing z value is around 2.58, this indicates passing of the Student's t test.

C. Nonlinearity measurement

One of the fundamental properties of neural networks is their lack of linearity, in traditional applications this is useful because it lets them express and approximate nonlinear functions. Here, we would also benefit from a similar property, as if our function f were to be linear, it would be easily recognizable from a random permutation. We had already proven in Lemma 1 that, at least a single layer, Galois neural network would be nonlinear. In this section, we investigate the extent of nonlinearity, as depth increases.

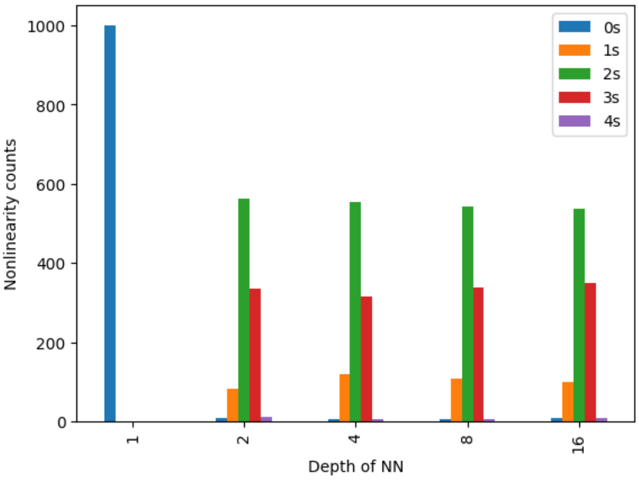


Figure 4: The distribution of nonlinearity for $w = 128$, $t = 5$, exponentially growing GNN depth. 1000 samples.

In [12] the nonlinearity of a function $f : \mathcal{F}^n \rightarrow \mathcal{F}^m$ where $m, n \in \mathbb{N}_+$, and $\mathcal{F}$ is a finite field, is defined in the following fashion.

Definition 6. *The **nonlinearity** $\mathcal{N}$ of a function $f : \mathcal{F}^n \rightarrow \mathcal{F}^m$ is given as:*

$$\mathcal{N}(f) = \min_{(u,w,v) \in \mathcal{F}^n \times \mathcal{F}^m \times \mathcal{F}} \#\{x \in \mathcal{F}^n | w^t f(x) \neq u^t x + v\}$$

In other words, we look at how closely we may approximate a projection of the output to a scalar, with an affine transformation of the input. We may use this definition to apply a measure to the notion of nonlinearity and express how nonlinear each depth of a neural network is.

Unfortunately, the naive computation of $\mathcal{N}(g)$, where $g : GF(2)^n \rightarrow GF(2)^m$ requires $\mathcal{O}(2^{n+m})$ time complexity, and $\mathcal{O}(2^n)$ invocations of g. This is too steep to compute for a wide Galois neural network. Instead, we may propose the following experiment:

1) Define a Galois neural network of width w, let $f : GF(2)^n \rightarrow GF(2)^n$ signify its computation.
2) Take a small number t.
3) Randomize the last $w - t$ bits and call them v_s.
4) Consider a function $f_t : GF(2)^t \rightarrow GF(2)^t$ that for input v_p is equal to $f(c(v_p, v_s))$ constrained to the first t bits, where c is a function that concatenates two vectors.
5) Compute the nonlinearity of f_t.

Such an experiment may help us derive some insights into the function f, and the efficiency of the activation function. Note that unlike measuring the nonlinearity of f, this is a random process so we will have to repeat it to gather reliable data.

In figures 4 and 5 we can see the results of such experiments. We may notice that for only one layer, and therefore only one activation function the nonlinearity values seem low and constant. This might be the result of the exact

depth	0	13	14	15	16	17	18	19	20
1	100	0	0	0	0	0	0	0	0
2	0	1	2	6	13	42	31	4	1
4	0	0	0	7	17	31	37	8	0
8	0	0	0	8	14	35	40	3	0
16	0	1	1	5	21	35	31	6	0

(a) $t = 6$

depth	32	36	37	38	39	40	41	42	43	44
1	100	0	0	0	0	0	0	0	0	0
2	0	4	0	8	0	38	0	45	0	5
4	0	1	2	7	10	17	25	28	10	0
8	0	1	2	3	12	22	20	26	13	1
16	0	1	1	4	8	23	24	18	16	5

(b) $t = 7$

depth	64	86	87	88	89	90	91	92	93	94	95	96	97
1	100	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	2	0	5	0	26	0	51	0	15	0
4	0	2	0	3	2	5	14	20	17	19	11	7	0
8	0	2	0	4	2	9	8	18	15	23	13	5	1
16	0	1	1	2	2	8	14	21	14	17	18	2	0

(c) $t = 8$

Figure 5: The distribution of nonlinearity values for larger t-s, $w = 128$, exponentially growing GNN depth. 100 samples for each.

implementation of the activation function, but nonetheless, this seems like a strong argument not to use such a low-depth GNN.

It is also noteworthy, that, with the exception of depth equal to 1, while the GNN depth increases exponentially it fails to have a significant impact on the magnitude of nonlinearity (at least in these experiments). However, the results for depth 2 still show some non-random behavior, like the results for t equal to 7 and 8 seemingly avoiding odd values. While a depth of 2 seems to be enough for the purposes of nonlinearity, practically it may be wise to air on the side of caution and recommend larger values.

IX. Future Work

It seems that the biggest area for improvement with this algorithm is performance. In the current form, it seems unlikely that the algorithm would see widespread use.

One idea to deal with that limitation would be to see if sparse matrices could be utilized to improve performance without loss of the statistical properties. One could also propose a different architecture for the underlying Galois neural network.

A different approach would be to seek improvements in the technical implementation of the algorithm, and either find improvements on the side of the algorithm or try to achieve better hardware support.

The second area for potential further work is trying to further verify and work towards proving Conjecture 1. One could for example try to replicate existing and to design new statistical tests perhaps on a bigger scale in order to show the guarantees required for widespread usage.

Acknowledgment

The research presented in this paper was realized with funds from the Polish Ministry of Science and Higher Education assigned to AGH University of Krakow.

References

[1] J. Daemen and V. Rijmen, “Aes proposal: Rijndael,” 1999.

[2] D. J. Bernstein and T. Lange, “Post-quantum cryptography,” *Nature*, vol. 549, no. 7671, pp. 188–194, Sep. 2017. doi: 10.1038/nature23461. [Online]. Available: https://doi.org/10.1038/nature23461

[3] X. Bogomolec, J. G. Underhill, and S. A. Kovac, “Towards post-quantum secure symmetric cryptography: A mathematical perspective,” 2019, https://eprint.iacr.org/2019/1208. [Online]. Available: https://eprint.iacr.org/2019/1208

[4] R. Kuang, D. Lou, A. He, and A. Conlon, “Quantum safe lightweight cryptography with quantum permutation pad,” *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pp. 790–795, 2021.

[5] A. Canteaut, S. Duval, G. Leurent, M. Naya-Plasencia, L. Perrin, T. Pornin, and A. Schrottenloher, “Saturnin: a suite of lightweight symmetric algorithms for post-quantum security,” Mar. 2019, soumission à la compétition ”Lightweight Cryptography” du NIST. [Online]. Available: https://hal.inria.fr/hal-02436763

[6] M. Arvandi, S. Wu, A. Sadeghian, W. Melek, and I. Woungang, “Symmetric cipher design using recurrent neural networks,” in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006. doi: 10.1109/IJCNN.2006.246972 pp. 2039–2046.

[7] J. Tchórzewski and A. Byrski, “Performance of computing hash-codes with chaotically-trained artificial neural networks,” in *Computational Science – ICCS 2022*, D. Groen, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2022. ISBN 978-3-031-08754-7 pp. 408–421.

[8] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. doi: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[9] M. Hostetter, “Galois: A performant NumPy extension for Galois fields,” 11 2020. [Online]. Available: https://github.com/mhostetter/galois

[10] “Pycryptodome,” https://www.pycryptodome.org/, accessed: 2023-05-22.

[11] J. Tchórzewski, “Application of artificial neural networks as hashing functions,” Ph.D. dissertation, AGH University of Technology, 2022.

[12] K. Nyberg, “On the construction of highly nonlinear permutations,” in *Advances in Cryptology - EUROCRYPT ’92, Workshop on the Theory and Application of of Cryptographic Techniques, Balatonfüred, Hungary, May 24-28, 1992, Proceedings*, ser. Lecture Notes in Computer Science, vol. 658. Springer, 1992. doi: 10.1007/3-540-47555-9_8 pp. 92–98.

DOI: 10.15439/2023F2076
ISSN 2300-5963 ACSIS, Vol. 35

Adding Linguistic Information to Transformer Models Improves Biomedical Event Detection?

1st Laura Zanella
LORIA (Université de Lorraine, CNRS, Inria)
Nancy, France
laura-alejandra.zanella-calzada@loria.fr

2nd Yannick Toussaint
LORIA (Université de Lorraine, CNRS, Inria)
Nancy, France
yannick.toussaint@loria.fr

***Abstract*—Biomedical event detection is an essential subtask of event extraction that identifies and classifies event triggers, indicating the possible construction of events. In this work we propose the comparison of BERT and four of its variants for the detection of biomedical events to evaluate and analyze the differences in their performance. The models are learned using seven manually annotated corpora in different biomedical subdomains and fine-tuned by adding a linear layer and a Bi-LSTM layer on top of the models. The evaluation is done by comparing the behavior of the original models and by adding a lexical and a syntactic features. SciBERT emerged as the highest performing model when the fine-tuning is done using a Bi-LSTM layer, without need of extra features. This result suggests that the use of a transformer model that is pretrained from scratch and uses biomedical and general data for its pretraining, allows to detect event triggers in the biomedical domain covering different subdomains.**

***Index Terms*—Biomedical Event Extraction, Event Detection, Transformer Language Models, Named Entity Recognition**

I. Introduction

BIOMEDICAL event extraction is a complex information extraction task that identifies key information from large sets of textual data for further applications, such as the study of biomolecular mechanisms or epigenetic changes. A biomedical event is constructed from an event trigger and one or more arguments that orbit around the trigger. Event triggers generally refer to nouns or verbs that express an action, circumstance or eventuality, while the arguments refer either to biomedical entities or to other events, called nested events. Fig. 1 shows the example of a sentence annotated with two biomedical events, '-Reg' (which stands for 'Negative regulation') and 'Locl' (which stands for 'Localization'). The event 'Locl' (the event is given the same type as the trigger) that is constructed from the trigger word 'excretion' presents as argument the biomedical entity of the type 'D/C' (which stands for 'Drug or compound'), who plays the role 'Th' (which stands for 'Theme'). This role allows answering the question 'What is excreted?'. While the event '-Reg', constructed from the trigger word 'reduces', presents two arguments. The first argument is a biomedical entity of the type 'Drug or compound', who plays the role 'Cause'. This role allows answering the question 'What causes the reduction?'. The second argument is the nested event 'Locl' described before, who plays the role 'Theme', answering the question 'What is reduced?'.

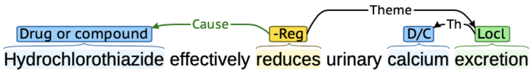

Drug or compound
Cause
-Reg
Theme
D/C
Th
Locl
Hydrochlorothiazide effectively reduces urinary calcium excretion

Fig. 1. Example of event extraction; the '-Reg' (negative regulation) event has the 'Locl' (localization) nested event as argument.

Event extraction is usually divided into three main sub-tasks, event detection, argument identification and event construction. Event detection identifies and classifies the trigger words into a set of predefined types of event triggers, while argument identification identifies and classifies the corresponding event arguments and their respective roles [1]. Event construction refers to the merging of the relations that correspond to the same event. This work focuses on event detection, which has a fundamental role in the construction of events, since the triggers are the targets that allow to know that an event may exist [2]. Difficulty for trigger detection comes from the sensitivity to the domain or subdomain (text can present specialized language), linguistic forms (triggers can be single words, multi-words, discontinuous markers) and ambiguity on the trigger class (a trigger word can be given different trigger classes) [3]. According to different works, such as in [1], solutions to address these issues may include additional features to provide lexical, syntactic and semantic information about text, which have proven to be useful for detecting event triggers. Transformers models have been adopted for event detection due to their positive achievements in performance for solving different Natural Language Processing (NLP) tasks [4], [5]. BERT [6], which stands for Bidirectional Encoder Representations from Transformers, is pretrained to generate bidirectional representations of the words, taking into account the semantics by considering both left and right directions of the text. From this pretraining, BERT can be fine-tuned by including additional layers on top of the model to solve new specific tasks. Furthermore, a series of variants from BERT have been developed for specific domains by being trained on large corpus with the same context, such as the biomedical domain.

In this work we compare BERT and four of its variants pretrained in the biomedical domain for the detection of biomedical event triggers to analyze their performance and identify which model is the most appropriate to address this task. For this purpose, BERT, BioBERT, SciBERT, Pub-

MedBERT, and BioMedRoBERTa are fine-tuned using two different classifiers, a linear layer and a Bidirectional Long Short Term Memory (Bi-LSTM) layer, to detect biomedical event triggers. These BERT variants have been chosen for comparison because they share the same BERT architecture but have previously been pretrained using different data in the biomedical and/or general domain [7]–[9]. The models are learned using seven manually annotated data sets merged together. These corpora were originally developed for the event extraction task in different biomedical subdomains. In addition to these data, two features are included as lexical and syntactical extra-information to the models, the stems and the parts-of-speech (POS) tags, respectively. SciBERT presented the highest performance when the fine-tuning is done using a Bi-LSTM classifier without adding any extra-features. This result suggests that using a transformer model that is pretrained from scratch using biomedical and general domain data, allows to detect biomedical event triggers addressing different biomedical subdomains.

Our main contributions refer to the (1) comparison of the capability of different pretrained transformer models to detect biomedical events, (2) evaluation of the performance of two different classifiers for the fine-tuning of event detection, (3) analysis of the impact of manually annotated corpora on different biomedical subdomains to detect event triggers, and (4) assessment of whether adding lexical and syntactic information improves biomedical event detection.

II. Related Work

Current SOTA systems for event detection use neural network models due to their robust event extraction capabilities.

P. V. Rahul et al. [10] used Recurrent Neural Networks (RNN) to extract higher level features through the hidden state of the network to identify biomedical event triggers. They also used the word and the entity type embeddings as features, demonstrating positive results in the MLEE [11] corpus. S. Duan et al. [12] and Y. Zhao et al. [13] explored an augmentation of the semantic information by integrating the full document representation. Both proposed the use of RNNs to extract cross-sentence features without the use of external resources. T. H. Nguyen and R. Grishman [14] presented a Graph Convolution Network (GCN) model to exploit syntactic dependency relations. They used dependency trees to link words to their informative context for event detection. H. Yan et al. [15] also proposed a GCN model, integrating aggregative attention to model and aggregate multi-order syntactic representations of the sentences, while in the case of S. Cui et al. [2], they extended the GCN by adding the relation aware concept, which exploits the syntactic relation labels and models the relation between words. DeepEventMine [16] is an end-to-end system for event extraction that consists on four main modules; BERT model, trigger and entity detection and classification, relation extraction and event identification. For each of the modules, BERT is used as base model and a linear layer is added. One of the main objectives of this system is improving the extraction of nested events, where it has achieved the new SOTA performance on seven biomedical nested event extraction tasks. B. Portelli et al. [17] compared BERT and five of its variants for the identification of Adverse Drugs and Events (ADEs). They showed that span-based pretraining, from spanBERT, provides an improvement in the recognition of ADEs, and that the pretraining of the models in the specific domain is particularly useful in comparison to train the models from scratch. A. Ramponi et al. [18] developed BEESL, a neural network model based on a sequence labeling system for the extraction of events. The system converts the event structures into a format of sequence labeling, and uses BERT as language model. Y. Chen [19] proposed the Multi-Source Transfer Learning-based Trigger Recognizer system, which is an extension on transfer learning using multiple source domains. All the datasets from the different domains are used for jointly train the neural network, achieving a higher recognition performance on the biomedical domain, having a wide coverage of events.

According to these works, transformer architectures have achieved positive results for detecting event triggers, and the use of pretrained language models has shown an improvement in the performance of this task. However, these works have been developed in a specific biomedical subdomain or in the general domain, not allowing a generalization to different biomedical subdomains. This may present a limitation in the detection of biomedical triggers because the language in biomedical texts is usually specialized and very specific. In addition, an analysis on how the pretrained language models used were selected over the other existing models is not described. Besides, according to A. Ramponi et al. [18], the detection of triggers continues to be the most important source of errors in event extraction, where around 31 % of the errors correspond to non-detection of triggers and 28 % to over-detection of triggers.

III. Materials and Methods

Fig. 2 shows the approach followed in this work. The annotated data is given as input to the pretrained transformer models to calculate the embeddings. The models used are BERT and four of its variants, who have achieved state-of-the-art performance in different NLP tasks without requiring major architectural modifications according to the specific tasks. In addition, the embeddings of a lexical and a syntactic features are also calculated. Then, a classification layer is added on top of the models for fine-tuning to detect event triggers.

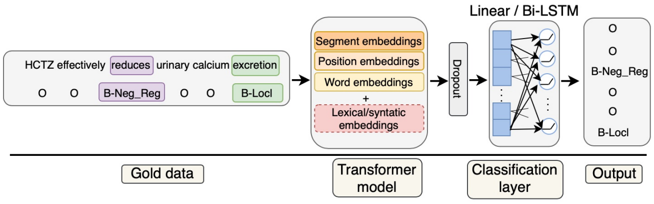


Fig. 2. Overview of the approach proposed to detect event triggers.

A. Transformer Model: BERT

BERT [6] is a contextualized word representation model based on a masked language model pretrained with bidirectional transformers [7]. In BERT, the sequence of input tokens (words or sub-words) is constituted with initial vectors that are the combination of the token embeddings, the (token) position embeddings and the segment embeddings (text segment to which the token corresponds) through element-wise summation. The embeddings of extra features can be computed and included in this summation, such as the POS embeddings (token function in meaning and grammar within the sentence), which has demonstrated to be helpful in detecting event triggers [1]. The embeddings are then passed to a set of layers of transformer modules. Each transformer layer generates a contextual representation of every token by summing the non-linear transformation of the tokens' representations from the previous layer. This representation is weighted by the attentions calculated using the representations of the previous layer as query. The last layer generates the contextual representations for all the tokens, where the information of the whole text span is combined [20]. Following the BERT principle, other transformer models have been developed being pretrained with data from specific domains, e.g. biomedical data, presenting better adaptation for solving in-domain tasks. BioBERT [7] and BioMedRoBERTa [21] are some examples of BERT variants pretrained in the biomedical domain.

B. Fine-Tuning Transformer Models for Event Detection

Various downstream text mining tasks can be performed by making minimal modifications to the BERT architecture through a process of fine-tuning. Here, the transformer models are fine-tuned following the Named Entity Recognition (NER) task. NER is one of the main tasks of biomedical text mining, which aims to recognize domain-specific nouns in a biomedical corpus by giving each word s_i in a sentence $S = s_1, s_2, ..., s_n$ (n refers to the number of words in the sentence) a predefined class $l \in L$ (where L refers to the predefined collection of entity types including the no-entity class). In this work, NER is adapted to identify triggers, which implies not only identifying nouns, but also verbs and in some cases adjectives. Two different classification layers, a linear layer and a Bi-LSTM layer, are used separately for comparison. The output labels are obtained following the IOB (Inside-Outside-Beginning) tagging to identify and classify the triggers into the predefined trigger categories (in the case of the I and B tags).

IV. Experimental Settings

A. Corpus

Table I presents the seven datasets [1] (all publicly available) used for fine-tuning the transformer models. These corpora were manually or semi-manually annotated by experts and released to be used in the development and improvement of event extraction models.

TABLE I
STATISTICS OF THE CORPUS USED

Dataset	No. Triggers	Trig Classes	Documents	Train/Dev/Test
CG 2013	9,790	35	PubMed abstracts	300/100/200
EPI 2011	2,035	14	PubMed abstracts	600/200/400
GENIA 2011	10,210	10	MEDLINE abstracts	1,000 (total)
GENIA 2013	4,676	12	PMC full-text	34 (total)
ID 2011	2,155	10	PMC full-text	15/5/10
PC 2013	6,220	22	PubMed abstracts	260/90/175
MLEE	5,554	15	PubMed abstracts	131/44/87

For the development of the experiments, the training and development datasets of all the corpora are initially merged into one single dataset and split into sentences, obtaining a total of 24,819 sentences. The original test sets are not used since the annotation are not released. Then, a random data partition into 80/20 is applied to obtain the training and testing sets, containing 19,855 and 4,964 sentences, respectively. Each sentence is further split into words by spaces and then, each word into sub-words or tokens following the setting of the BERT tokenization. These tokens are then given as input to the transformer model. All the trigger classes from each corpus are considered for the final trigger classification, presenting a final set of 58 classes (some classes overlap among the different corpora).

B. Pretrained Transformer Models

The transformer model, BERT [6], and four BERT variants pretrained in the biomedical domain, BioBERT [7], SciBERT [8], PubMedBERT [20], and BioMedRoBERTa [21], are used and compared for the detection of event triggers. These models differ from each other by the corpora in which they were pretrained (all in English), the type of pretraining and the size of the vocabulary. SciBERT and PubMedBERT, were pretrained from scratch, meaning that they use a unique vocabulary on the pretraining corpus and include embeddings that are specific for in-domain words. BioBERT and BioMedRoBERTa were pretrained starting from the BERT checkpoints, which means that the vocabularies are built with general-domain texts (similar to BERT) as well as the initialization of the embeddings.

C. Lexical and Syntactic features

The embeddings of stems and POS tags are also computed and added as extra-features. Stems provide lexical information that correspond to the words reduced to their word roots, without needing to be an existing word in the dictionary. Stems are obtained by applying a set of rules to remove attached suffixes and prefixes (affixes) from terms without considering the POS or the context of the word occurrence [27]. POS tags represent syntactic information that provides the categorical differences of the words according to their functions in meaning and grammatically within the sentence. POS tagging consists on automatically obtaining the POS tag of each word among the different POS categories corresponding to their

[1] Cancer Genetics (CG) 2013 [22], Epigenetics and Post-translational Modifications (EPI) 2011 [23], GENIA 2011 [24], GENIA 2013 [25], Infectious Diseases (ID) 2011 [26], Pathway Curation (PC) 2013 [22], Multi-Level Event Extraction (MLEE) [11]

syntactical role [28]. For this work, the stems of the words are obtained using the 'Snowball Stemmer' module from NLTK-3.4.5 [2], while the POS were obtained using spaCy-3.0.0 [3], using 'en_core_web_sm', a pipeline developed for biomedical data. The embeddings of the stems and POS tags are summed to the rest of the embeddings (token, position and segment) calculated by the transformer models.

D. Parameters Settings

All the experiments are done with PyTorch, using the Transformers [4] library and the models were taken from Hugging Face [5]. The transformer models are trained using the original parameters from BERT, presenting a dropout probability for the attention heads and hidden layers of 0.1, a hidden size of 768, an initializer range of 0.02, a max position embeddings of 512 and an intermediate size of 3,072. The number of attention heads and hidden layers was 12 for both. 'Adam' was used as optimizer and 'gelu' as activation function. The training parameters of the classification layers, both linear and Bi-LSTM, were set as follows; batch size of training and testing sets of 16, learning rate of 1e-05 and max gradient norm of 10, since gradient clipping was included. The maximum length of the sentences was set to 256. All the models were trained during 100 epochs on the training set without applying early stopping, and evaluated by measuring the precision (P), recall (R) and F1-score.

V. Results and Discussion

The evaluation results of the fine-tuning of the models for event detection are shown in Table II. The approximate time in hours for the fine-tuning of each model is presented in the last column of the table. The highest results obtained in epochs 10, 30 and 100 are presented in bold, and the highest overall results of all epochs are presented in bold and underlined. First, we observe that SciBERT, which was pretrained from scratch using biomedical and general data, obtained the best results for each number of epochs and overall, in P, R and F1. It presented higher values when Bi-LSTM was used as classifier, especially when extra features were not added or when the lexical feature is added in the case of the training for 10 epochs. When the training was done for more than 10 epochs, the performance between SciBERT+POS (syntactic feature) and SciBERT+stem (lexical feature) was very similar. When the fine tuning was done using a linear classifier, SciBERT+POS achieved the best results, having a difference of around 10 % to when the lexical feature (SciBERT+stem) is added. PubMedBERT, a model pretrained from scratch using biomedical data, achieved the second best performance, being below SciBERT by 4 % when the training is done for 30 epochs, using Bi-LSTM as classifier and no adding extra-features (which was the best overall result of SciBERT). When PubMedBERT used Bi-LSTM as classifier, the results were very similar between adding the syntactic or lexical features and not adding them. These results were also similar to when a linear classifier was used and the extra features are added, noticing that the result was worse when no features were added. In the case of BERT, which was trained from scratch using data from the general domain, it presented lower results than PubMedBERT by around 5 %. The best results of BERT were obtained using a linear classifier and not adding extra features, noticing that the results of BERT+POS and BERT+stem were slightly lower and very similar between each other. This same behavior can be noticed when Bi-LSTM was used as classifier. These three last transformer models, SciBERT, PubMedBERT and BERT, presented some similarities in that they were trained from scratch, used very comparable text sizes for their pretraining and had similar vocabulary sizes. The two models that presented the lowest performance are BioBERT and BioMedRoBERTa, both pretrained from the BERT weights, using biomedical and, biomedical and general data, respectively, presenting the largest text sizes of all the models. BioBERT used the smallest vocabulary for its pretraining, while BioMedRoBERTa used the largest in comparison to the rest of the models. In both models it was observed that there was not significant change when adding the extra features, although there was an improvement of around 7 % when using a Bi-LSTM classifier compared to a linear classifier. In general, what can be noticed in all the models is that adding the syntactic and lexical features does not improve the performance for detecting biomedical events.

Fig. 3 shows the performance of fine-tuning SciBERT during 30 epochs using a Bi-LSTM classifier on the seven datasets separately. The F1-scores obtained using EPI, CG, ID, GE'13 and PC were similar between each other, obtaining values between 0.70 and 0.80. When GE'11 was used, the F1-score reached a value of around 0.65 and when MLEE was used, the model completely failed the detection of triggers. In Fig. 4 it is observed the effect of fine-tuning SciBERT over 30 epochs using a Bi-LSTM classifier without adding extra-features by cumulatively adding each corpus one by one. Below each corpus is shown the total number of classes by adding each corpus. Recall was improved when CG and EPI were used together, and then reduced as the rest of the corpora were added. Precision was affected when EPI and GE'11 were added. The behavior of recall and precision varied differently depending on the added corpus, although when GE'13 was added both values were comparable, and as might be expected according to the observed on Fig. 3, when MLEE was added the values were negatively affected. This behavior may be due to the fact that when adding a new corpus for the fine-tuning of the models, some classes may overlap between the corpora while other classes do not, causing to probably have less samples in the new classes and, therefore, affecting the balance of the data. In addition, the context of the different biomedical subdomains may also affect the performance, since BERT and its variants compute embeddings considering the semantics.

[2] https://www.nltk.org/_modules/nltk/stem/snowball.html

[3] https://spacy.io/

[4] https://github.com/huggingface/transformers

[5] https://huggingface.co/

TABLE II
RESULTS OF THE MODELS' FINE-TUNING FOR EVENT DETECTION

Classifier	Model	10 epochs			30 epochs			100 epochs			Time
		P	R	F1	P	R	F1	P	R	F1	(h)
Linear	BERT	0.57	0.67	0.62	0.60	0.68	0.64	0.62	0.68	0.65	13
	BERT+POS	0.58	0.61	0.59	0.62	0.63	0.62	0.64	0.64	0.64	14
	BERT+stem	0.62	0.58	0.59	0.67	0.57	0.61	0.66	0.62	0.63	18
Bi-LSTM	BERT	0.59	0.57	0.57	0.67	0.58	0.62	0.65	0.64	0.64	19
	BERT+POS	0.46	0.59	0.51	0.58	0.62	0.60	0.61	0.63	0.62	21
	BERT+stem	0.57	0.59	0.57	0.63	0.61	0.62	0.67	0.60	0.63	15
Linear	BioBERT	0.49	0.49	0.48	0.52	0.50	0.50	0.56	0.49	0.51	19
	BioBERT+POS	0.54	0.44	0.47	0.49	0.51	0.49	0.51	0.51	0.51	16
	BioBERT+stem	0.48	0.50	0.47	0.52	0.46	0.49	0.53	0.48	0.50	18
Bi-LSTM	BioBERT	0.60	0.39	0.45	0.60	0.56	0.58	0.64	0.56	0.59	14
	BioBERT+POS	0.57	0.39	0.44	0.59	0.55	0.57	0.61	0.55	0.58	15
	BioBERT+stem	0.54	0.50	0.50	0.61	0.52	0.56	0.59	0.57	0.58	20
Linear	SciBERT	0.59	0.64	0.61	0.61	0.65	0.63	0.70	0.70	0.70	11
	SciBERT+POS	**0.67**	**0.72**	**0.69**	0.69	0.71	0.70	0.72	**0.73**	**0.72**	16
	SciBERT+stem	0.56	0.62	0.58	0.61	0.62	0.61	0.64	0.62	0.63	13
Bi-LSTM	SciBERT	0.65	0.71	0.68	0.71	**0.73**	**0.72**	0.74	0.71	**0.72**	19
	SciBERT+POS	0.55	0.56	0.54	0.70	0.71	0.70	0.73	0.70	0.71	22
	SciBERT+stem	**0.67**	0.68	0.67	**0.72**	0.68	0.70	**0.75**	0.68	0.71	16
Linear	PubMedBERT	0.49	0.61	0.54	0.58	0.66	0.61	0.58	0.62	0.60	14
	PubMedBERT+POS	0.63	0.68	0.65	0.64	0.68	0.66	0.68	0.67	0.67	16
	PubMedBERT+stem	0.62	0.66	0.64	0.66	0.67	0.66	0.70	0.67	0.68	18
Bi-LSTM	PubMedBERT	0.57	0.65	0.61	0.66	0.69	0.67	0.67	0.69	0.68	19
	PubMedBERT+POS	0.58	0.65	0.61	0.67	0.66	0.66	0.69	0.67	0.68	17
	PubMedBERT+stem	0.59	0.66	0.61	0.66	0.69	0.67	0.70	0.66	0.68	18
Linear	BioMedRoBERTa	0.48	0.49	0.47	0.52	0.52	0.51	0.55	0.50	0.52	14
	BioMedRoBERTa+POS	0.52	0.56	0.53	0.55	0.51	0.52	0.55	0.53	0.54	13
	BioMedRoBERTa+stem	0.50	0.53	0.51	0.51	0.51	0.51	0.53	0.54	0.53	18
Bi-LSTM	BioMedRoBERTa	0.58	0.50	0.53	0.60	0.57	0.58	0.69	0.53	0.59	19
	BioMedRoBERTa+POS	0.51	0.56	0.52	0.61	0.53	0.56	0.62	0.56	0.58	15
	BioMedRoBERTa+stem	0.51	0.54	0.52	0.57	0.59	0.57	0.60	0.59	0.59	15

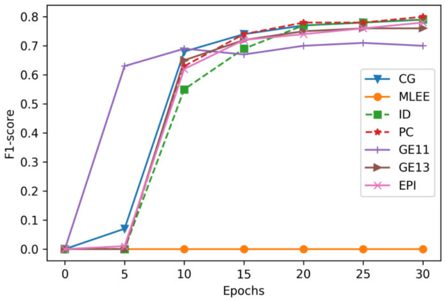


Fig. 3. Fine-tuning SciBERT on the different corpus (Bi-LSTM classifier).

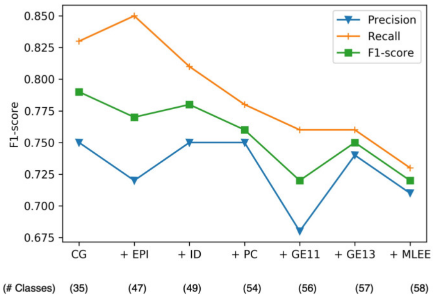


Fig. 4. Fine-tuning SciBERT by cumulatively adding the corpus one by one (Bi-LSTM classifier).

VI. Conclusions and Limitations

In this work, we analyze BERT and four of its variants for biomedical event detection using corpora of different biomedical subdomains. By comparing the performance of the models and by adding a lexical and syntactic features, we found that fine-tuning SciBERT during 30 epochs using a Bi-LSTM classifier is the best strategy to detect biomedical events, especially if the additional features are not included. Furthermore, it is shown that fine-tuning the models for 10 to 30 epochs achieves most of the model learning, while training for more epochs can only achieve a slightly better result. One of the limitations of this work is the imbalance of the data. Since some classes of the different corpora overlap, the samples for those classes are increased, while the unique classes for each corpora present fewer samples. This can negatively affect the behavior of the models between the different subdomains. Also, using external tools to get POS tags and stems can lead to errors that are learned by the models and may be one of the reasons why performance without additional features achieves better results.

References

[1] C. Shen, H. Lin, X. Fan, Y. Chu, Z. Yang, J. Wang, and S. Zhang, "Biomedical event trigger detection with convolutional highway neural network and extreme learning machine," *Applied Soft Computing*, vol. 84, p. 105661, 2019. doi: 10.1016/j.asoc.2019.105661

[2] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang, and J. Shi, "Event detection with relation-aware graph convolutional neural networks," *arXiv e-prints*, pp. arXiv–2002, 2020.

[3] C. Zerva and S. Ananiadou, "Event extraction in pieces: Tackling the partial event identification problem on unseen corpora," in *Proceedings of BioNLP 15*, 2015. doi: 10.18653/v1/W15-3804 pp. 31–41.

[4] R. Hanslo, "Deep learning transformer architecture for named-entity recognition on low-resourced languages: State of the art results," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 53–60.

[5] K. Kaczmarek, J. Pokrywka, and F. Graliński, "Using transformer models for gender attribution in polish," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022, pp. 73–77.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. doi: 10.18653/v1/n19-1423

[7] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020. doi: 10.1093/bioinformatics/btz682

[8] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019. doi: 10.18653/v1/D19-1371

[9] A. Erdengasileng, Q. Han, T. Zhao, S. Tian, X. Sui, K. Li, W. Wang, J. Wang, T. Hu, F. Pan *et al.*, "Pre-trained models, data augmentation, and ensemble learning for biomedical information extraction and document classification," *Database*, vol. 2022, 2022. doi: 10.1093/database/baac066

[10] P. V. Rahul, S. K. Sahu, and A. Anand, "Biomedical event trigger identification using bidirectional recurrent neural network based models," *arXiv preprint arXiv:1705.09516*, 2017. doi: 10.18653/v1/W17-2340

[11] S. Pyysalo, T. Ohta, M. Miwa, H.-C. Cho, J. Tsujii, and S. Ananiadou, "Event extraction across multiple levels of biological organization," *Bioinformatics*, vol. 28, no. 18, pp. i575–i581, 2012. doi: 10.1093/bioinformatics/bts407

[12] S. Duan, R. He, and W. Zhao, "Exploiting document level information to improve event detection via recurrent neural networks," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2017, pp. 352–361.

[13] Y. Zhao, X. Jin, Y. Wang, and X. Cheng, "Document embedding enhanced event detection with hierarchical and supervised attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018. doi: 10.18653/v1/P18-2066 pp. 414–419.

[14] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Thirty-second AAAI conference on artificial intelligence*, 2018. doi: 10.1609/aaai.v32i1.12039

[15] H. Yan, X. Jin, X. Meng, J. Guo, and X. Cheng, "Event detection with multi-order graph convolution and aggregated attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. doi: 10.18653/v1/D19-1582 pp. 5766–5770.

[16] H.-L. Trieu, T. T. Tran, K. N. Duong, A. Nguyen, M. Miwa, and S. Ananiadou, "Deepeventmine: end-to-end neural nested event extraction from biomedical texts," *Bioinformatics*, vol. 36, no. 19, pp. 4910–4917, 2020. doi: 10.1093/bioinformatics/btaa540

[17] B. Portelli, E. Lenzi, E. Chersoni, G. Serra, and E. Santus, "Bert prescriptions to avoid unwanted headaches: A comparison of transformer architectures for adverse drug event detection," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021. doi: 10.18653/v1/2021.eacl-main.149 pp. 1740–1747.

[18] A. Ramponi, R. van der Goot, R. Lombardo, and B. Plank, "Biomedical event extraction as sequence labeling," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. doi: 10.18653/v1/2020.emnlp-main.431 pp. 5357–5367.

[19] Y. Chen, "A transfer learning model with multi-source domains for biomedical event trigger extraction," *BMC genomics*, vol. 22, no. 1, pp. 1–18, 2021. doi: 10.1186/s12864-020-07315-1

[20] Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021. doi: 10.1145/3458754

[21] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, and N. A. Smith, "Don't stop pretraining: Adapt language models to domains and tasks," in *Proceedings of ACL*, 2020. doi: 10.18653/v1/2020.acl-main.740

[22] C. Nédellec, R. Bossy, J.-D. Kim, J.-J. Kim, T. Ohta, S. Pyysalo, and P. Zweigenbaum, "Overview of bionlp shared task 2013," in *Proceedings of the BioNLP shared task 2013 workshop*, 2013, pp. 1–7.

[23] T. Ohta, S. Pyysalo, and J. Tsujii, "Overview of the epigenetics and post-translational modifications (epi) task of bionlp shared task 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, 2011, pp. 16–25.

[24] J.-D. Kim, Y. Wang, T. Takagi, and A. Yonezawa, "Overview of genia event task in bionlp shared task 2011," in *Proceedings of BioNLP shared task 2011 workshop*, 2011, pp. 7–15.

[25] J.-D. Kim, Y. Wang, and Y. Yasunori, "The genia event extraction shared task, 2013 edition-overview," in *Proceedings of the BioNLP Shared Task 2013 Workshop*, 2013, pp. 8–15.

[26] S. Pyysalo, T. Ohta, R. Rak, D. Sullivan, C. Mao, C. Wang, B. Sobral, J. Tsujii, and S. Ananiadou, "Overview of the infectious diseases (id) task of bionlp shared task 2011," in *Proceedings of BioNLP Shared Task 2011 Workshop*, 2011, pp. 26–35.

[27] A. G. Jivani *et al.*, "A comparative study of stemming algorithms," *Int. J. Comp. Tech. Appl*, vol. 2, no. 6, pp. 1930–1938, 2011.

[28] S. Petrov, D. Das, and R. McDonald, "A universal part-of-speech tagset," *arXiv preprint arXiv:1104.2086*, 2011.

DOI: 10.15439/2023F3745
ISSN 2300-5963 ACSIS, Vol. 35

Sensitivity analysis of the criteria weights used in selected MCDA methods in the multi-criteria assessment of banking services in Poland in 2022

Marek Zborowski
0000-0003-4762-127X
University of Warsaw
in Warsaw
ul. Krakowskie Przedmieście 26/28, 00-927 Warsaw, Poland
Email: mzborowski@wz.uw.edu.pl

Witold Chmielarz
0000-0002-9189-1675
University of Warsaw
in Warsaw
ul. Krakowskie Przedmieście 26/28, 00-927 Warsaw, Poland
Email: witek@wz.uw.edu.pl

Abstract—**The main purpose of this article is to compare the sensitivity of the weights of individual criteria in the assessment of the most popular banks' websites and the impact of the MCDA methods applied on the results of these assessments. The analysis was carried out for three selected, the most popular and different assessment methods: TOPSIS, VIKOR and PROMETHEE II. The evaluation of the websites was made on a sample of 350 bank customers, whose opinions were obtained using the CAWI method using a survey form. The survey made it possible to distinguish the 16 most popular banking services in this group, and only these banks were then evaluated. The survey questionnaire was obtained after verification of the pilot version created on the basis of previous research. The websites most known to the respondents were tested using three MDCA methods: TOPSIS, VIKOR, PROMETHEE II. The sensitivity of the results in each of the banks to the development of weights for 18 attributes (service evaluation criteria) was examined. The obtained results indicate the possibility of interchangeable use of the distinguished assessment methods, which may be helpful for business practitioners when analyzing and designing banking services.**

I. Introduction

TO PREPAREhe main purpose of this article is to determine the sensitivity of the position of the websites of the rated banks (C1, ..., C16) in the respondents' rankings depending on the weights of the distinguished assessment attributes (A1, ..., A18). Evaluations of websites, [1] assessed by respondents at the end of 2022, allowed for the construction of a common data set. Based on this set, three experiments were performed:

- comparison of the results obtained from the simple point method with the new, original MDCA method [2], [3] - the Conversion method [4],
- banking services were assessed and the five most popular MDCA [5], [6], [7] methods: TOPSIS [7], COMET [8], VIKOR [9], PROSA-C and PROMETHE II [10] were compared, in terms of convergence of results, to see if the situation observed in the first experiment also occurs among the methods already widely known and recognized . The obtained results indicate significant deviations of the applied VIKOR [11] method from other methods. Thus, the discrepancies in the results of the Conversion method in this light did not seem to be something exceptional after this study,
- the most important factor determining the assessment, and thus the position of individual banking services in the ranking, are their attributes, a priori determined at the beginning of the study. Therefore, the sensitivity of the position in the ranking of banking services depending on the change in the weights of these attributes was examined.

This article presents the results of a recent experiment. It is a continuation of research undertaken on the application of multi-criteria methods [12], [13] for evaluating the latest IT solutions in the economy [14]. These studies are both methodical and practical. Because, as it turned out, there is no single, universal method of assessing usability [15]. The results depend primarily on the selection of parameters (attributes) - evaluating the studied phenomenon. These parameters vary depending on the studied phenomenon, industry, end user, purpose of the results, etc. [16].

This study uses a constantly modified set of criteria agreed in 2008 with the best specialists in Poland dealing with research in the banking industry. During subsequent crises, it was modified arbitrarily by the authors (external factors - high inflation, pandemic) and by end users (bank customers) for reasons of comprehensibility and preferences. External factors extended the list of attributes, which in turn, after verification by a pilot group of users, was reduced to the most understandable attributes (after possible corrections) and important from the client's point of view [17].

The problem directly resulting from the selection of attributes are the significance weights assigned to them, which in a sense reduce the subjectivity of the final methods [18]. The simplest method to solve this problem is, of course, to ask end users about the level of significance of a given criterion,

This work was not supported by any organization

and to take the average of their answers.

However, a different approach is often used that simulates the results depending on the level of weights assigned to them. Sometimes it is assigning weights to specific types of attributes (in the case of banking services: economic, technical and anti-crisis), and sometimes to each assigned attribute and observing the results [19]. The simplest method to solve this problem is, of course, to ask end users about the level of significance of a given criterion, and to take the average of their answers.

However, a different approach is often used that simulates the results depending on the level of weights assigned to them. Sometimes it is assigning weights to specific types of attributes (in the case of banking services: economic, technical and anti-crisis), and sometimes to each assigned attribute and observing the results.

II. Preliminaries

A. Research procedure

The research procedure in this case included the following steps:

- bibliographic analysis of website evaluation using MCDA methods,
- construction of a pilot version of the survey questionnaire,
- verification of the survey form and preparation of its final version,
- random selection of groups of respondents and inviting them to complete a questionnaire using the CAWI (Computer Associated Web Interview) method,
- obtaining data and initial verification of their correctness,
- selection and justification of methods for evaluating banking services in order to make calculations and obtain results as well as to make comparisons between them,
- analysis, discussion and comparison of results
- drawing conclusions, defining limitations and further directions of research.

Due to the research conducted earlier and the popularity of the analyzed methods, two of them were initially selected in the first experiment: the simple point method and the Conversion method [20]. Interpretation and comparison of the results obtained with these methods and their differentiation by means of the Euclidean distance were performed. Then, for five consecutive MDCA [14], [21], [22] methods: TOPSIS [23], COMET [24], VIKOR [25], PROSA-C [26] and PROMETHE II [27], calculations were made, ranking lists were prepared and the results were compared. In the analysis of the results, the calculated values of the preference function for subjective weights were taken into account. Then, the correlations between the rankings obtained with different methods and the Euclidean distances were calculated in order to examine the level of differentiation of the results obtained between the individual pairs of the methods used. The above analyzes were the content of the previous two articles.

However, this paper presents a sensitivity simulation. It consisted in the fact that for each method (TOPSIS, VIKOR, PROMETHEE II) [28], [29], [30] the values of the weights of individual evaluation criteria (A1,. . . , A18) of the existing selection variants (C1 - C16) were successively modified. The value of the weights was changed successively, and the weights of the remaining criteria were adjusted - in equal proportions, so that the weights of all the criteria add up to 1.

B. Sample characteristic

The rankings of websites were based on data collected using the CAWI method in autumn 2022. They covered 356 people, with over 48% survey response. A five-point, simplified, standardized Likert [31] scale was adopted to assess individual criteria:

- 1.0 - fully meets the requirements of the criterion,
- 0.75 - almost completely meets the requirements of the criterion,
- 0.50 - meets the requirements of the criterion on average,
- 0.25 - meets the requirements of the criterion at least,
- 0.00 - does not meet the requirements of the criterion.

The original form of the questionnaire was verified on a pilot sample of 50 people, conducted in an academic environment. Individual criteria - attributes of banking services - were examined in terms of their comprehensibility and importance for an average website user. After verification, corrections and removal of the least important criteria, 18 attributes were taken into account for the assessment of each website, divided into the following three groups: economic, technological and anti-crisis. A detailed list of attributes is included in Table 1.

The evaluation was conditional upon evaluating the websites of a well-known electronic bank in comparison with the website of another banking website. This condition resulted from the desire to obtain answers from experienced respondents dealing with various electronic banking services. Thus, a total of 712 full banking service ratings were received, as some respondents rated two or three banks.

Respondents rated the sixteen (A1, A2, . . . , A16) the most frequently used banking websites. They can be found in Table 2. All banking services that received less than five ratings are not included in this list - the ratings of 16 banks were rejected.

The research sample was selected in a diversified way: on purpose - the research was carried out in the academic environment in randomly selected didactic groups and using a link to the online survey [32].

The survey was mainly aimed at young people. The age range therefore ranged from 19 to 35 years). This choice could have influenced the results of the survey (41 million people in Poland are potential customers of online and mobile banking, over 54% of registered customers are active users of online banking and 44% of active users of mobile banking). The surveyed age group constitutes over 65% of users) . Among the surveyed respondents there were over 70% women and nearly 30% men. 19% had bachelor's and incomplete higher education, and 80% had secondary education. The largest group of people came from large cities (over 200,000 inhabitants), and 19% from rural areas. One fourth came from small, medium and large towns - up to 200,000. inhabitants.

TABLE I
LIST OF ATTRIBUTES OF BANKING WEBSITES ASSESSMENT; SOURCE: OWN STUDY

No	**Attributes**
A1	**Nominal annual interest rate on personal accounts**
A2	**Keeping an account in PLN/month**
A3	**Fee for transfer to the parent bank**
A4	**Fee for transfer to another bank**
A5	**Direct Debit**
A6	**Fee for issuing a debit card**
A7	**Monthly fee for the card PLN/month**
A8	**Additional services**
A9	**Account access channels**
A10	**Security**
A11	**Visualization**
A12	**Navigation**
A13	**Readability and ease of use**
A14	**Scope of functionality**
A15	**Interest rates on savings accounts**
A16	**The interest rate on deposits is 10,000.**
A17	**Interest rate on loans 10 thousand.**
A18	**Anti-crisis activities**

TABLE II
LIST OF ATTRIBUTES OF BANKING WEBSITES ASSESSMENT; SOURCE: OWN STUDY

No	**Bank**
C1	**Alior Bank SA**
C2	**Bank Handlowy w Warszawie SA**
C3	**Bank Millenium SA**
C4	**Bank Pocztowy SA**
C5	**Bank Polska Kasa Opieki**
C6	**Bank Polskiej Spółdzielczości**
C7	**BNP Paribas SA**
C8	**Credit Agricole Bank Polska SA**
C9	**Getin Noble Bank (obecnie Velo Bank)**
C10	**ING - Bank ŚLąski SA**
C11	**mBank SA**
C12	**Nest Bank SA**
C13	**PKO Bank Polski SA**
C14	**Santander Bank Polska SA**
C15	**Santander Consumer Bank SA**
C16	**TOYOTA Bank Polska SA**

Among the respondents, there were 52% of students, 31% of people working under a contract for specific work, mandate or running their own business and 17% working under an employment contract. The most frequently performed occupations are: office workers (63%), service workers (16%), specialists (8%) and workers employed for simple technical work (7%). Most of them describe their financial situation as good (61%), very good (22%), average (16%) and sufficient (2%).

Data on the assessment of banking services are generally relatively homogeneous and consistent. After obtaining them, a reliability test in the form of Cronbach's alpha coefficient was applied. For all attributes, the Cronbach's alpha coefficient indicating the internal consistency and reliability of the sample [68] was greater than 0.75. The measure of internal consistency of the 16 dependent variables, based on Cronbach's alpha coefficient, was 0.85 (0.94 for Cronbach's alpha calculated on the basis of standardized items), for 18 items in total.

III. ANALYSIS OF RESULTS AND DISCUSSION

Sensitivity analysis is a technique that studies the effect of changes in one of the independent variables that make it up on the dependent variable of any model. In order to study in detail how the changes in weights affect the final ranking, a sensitivity analysis was performed. For each criterion, nine evaluations were performed, where the criterion in question was given weight 0.1, 0.2, ..., 0.9 while all other criteria were set to 0.5. This allowed to study the effect of each particular criterion on the overall ranking. A total of 162 evaluations was therefore performed. The results of these evaluations were then plotted and are presented on `figure1` and `figure2`. Here are shown results of the sensitivity test to changes in the weights for individual attributes (A1, A2, ..., A18) successively, for all sixteen analyzed banking services (C1, C2, ..., C16). Place - position in the ranking (on the y-axis) 1 - is the best, 18 - is the worst. On the x-axis, there are weights of the selected criterion (its symbol is in the chart title). The weight values in all cases changed every 0.1. Examination of the figure below shows that although sometimes very minor,

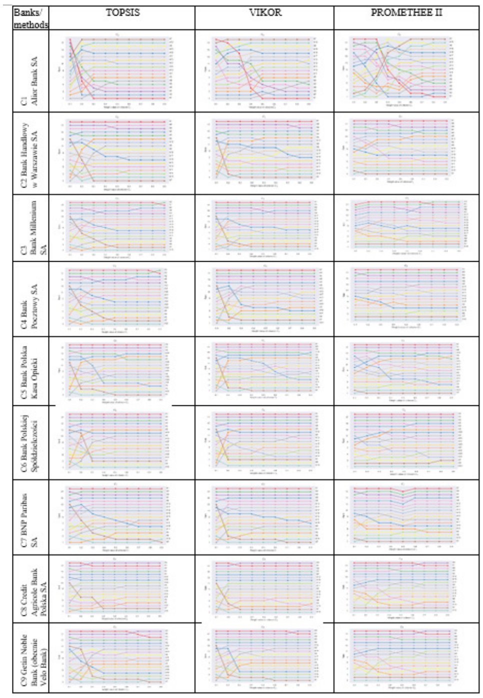


Fig. 1. Sensitivity to simulated changes in the weights of individual attributes in the analyzed banks, banks C1-C9; Source: own study

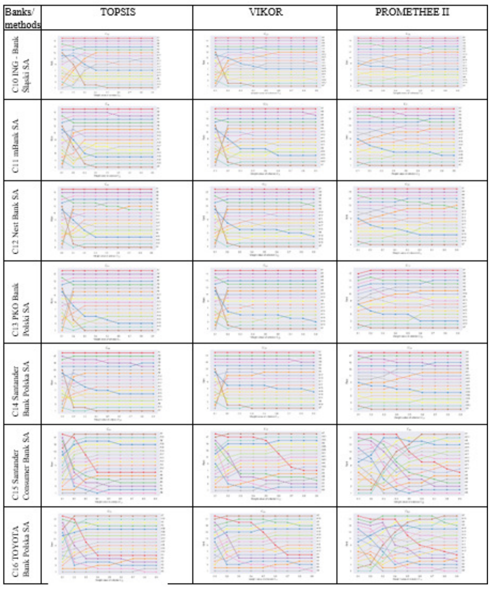


Fig. 2. Sensitivity to simulated changes in the weights of individual attributes in the analyzed banks, banks C10-C16; Source: own study

there are some changes in ranking positions if the weights are manipulated in procedurę of simulation. Sensitivity analysis was carried out for three popular MDPI methods: TOPSIS, VIKOR and PROMETHEE II in order to compare how the positions in the ranking of the analyzed banks react to changes in the weights of individual attributes (based on the same data and the same attributes).

The analysis of the sensitivity of the position in the ranking of individual banks to changes in attributes leads to the following assessments. For C1, the key attribute in the TOPSIS method is A7, which, when raised to the value of 0.3, shifts the significance of this parameter to the first position. It reacts similarly to changes in parameters A2, A3 and A8. Similar behavior can be observed in the VIKOR method, but it requires a value of 0.5. However, the ranking positions for C1 behave separately and completely chaotically. For C2 - the key attribute is A7, a slight increase in its value moves its position from 13th to 1st in the TOPSIS and VIKOR methods. In the PROMETHEE II method, this change, as in the case of other parameters, increased the position by only one position. For the C3 bank - the most important feature is A10, after raising the weight to the value of 0.2-0.3, it moves to the first position after using the TOPSIS and VIKOR methods, in the PROMETHEE II method, the A10 attribute remains firmly in the first place. For the A4 bank, such a key attribute is also A10, after increasing the weight of this parameter (using the TOPSIS method), it moves to the first place in the ranking, the attribute A9, which previously occupied it, moves to the thirteenth place. It is similar in the other two methods, in PROMETHEE II, it is still the most important attribute, regardless of the weight assigned to it.

The situation is slightly different for the C5 bank. Here, the most important parameter is A7, which with the increase of the weight value moves from the twelfth position first to the third, and then with the weight equal to 0.5 - to the first place. A similar "jump" can be observed in the VIKOR method, while in the PROMETHEE II method this process is more stable. For the C6 bank, the most important attribute is A10, while the most sensitive to weight change is A7, which was moved from thirteenth to third position in the TOPSIS method. Such a tendency is shown by the attributes A10 and A7 in the VIKOR method, while in the PROMETHEE II method they always stop at the first and third position, regardless of the values of the adopted weights. A similar situation occurs for the C7 bank, where A7 moves from the twelfth to the first position in the ranking, with weight=0.9, dealing with the A10 parameter (TOPSIS method). When using the VIKOR method, this scheme is almost duplicated, the application of the PROMETHEE II method shows a low sensitivity of these attributes to the change of weights and leaves them respectively on A7 in the first position, A10 in the second position. For banks C8-C14, the situation almost repeats itself, as the weight increases, the attribute A7 from the further position is moved to the first position, replacing the parameter A10 (except for C9, where A10 remains in the first position all the time as the weight increases). For all the cases mentioned so far, the attributes A1-A4 and A8 are relatively stable and independent of the weights and the assessment method adopted, sometimes only changing places in the ranking. That is, for these banks the most important attributes are: security and the amount of the bank card fee (if any), and the least important: nominal annual interest rate on personal accounts, keeping an account in PLN/month, fee for transfer to the parent bank, and fee for transfer to another bank and additional services.

The situation is completely different for the C15 bank. In the TOPSIS method, A8 moves from position 15 to the first position and A2 from the twelfth position to the second position. As the weights increase, parameters A9 and A5 are lost. Finally placing respectively in: fifth and sixth position. This phenomenon is repeated for the VIKOR and PROMETHEE II methods, where there is also the strongest so far "shaking" of the attributes' positions due to the height of the simulated weights (all of them change their place in the ranking). The sensitivity of the attributes to the change of weights for the C16 bank is also different. The first two positions A8 and A9 are relatively stable, regardless of the calculation method. In the TOPSIS and VIKOR methods, the remaining attributes undergo significant changes in the ranking position. In the PROMETHEE II method, the remaining attributes undergo far-reaching changes in position. An interesting phenomenon is the fact that the parameters A7 and A10 for the last two analyzed cases are moved to the last positions in the ranking as the weights increase. The analysis shows that for the customers of these two banks the most important features may be: additional services and account access channels, and much less important attributes related to fees. Technical parameters turned out to be the least sensitive to weight changes in all cases.

IV. Conclusions

This work was aimed at comparing:

- sensitivity of the results of the multi-criteria evaluation of websites of the most popular banks in Poland to changes in the weights of the attributes used for this evaluation,
- the results of changes in attribute weights in three selected MDCA methods: TOPSIS, VIKOR and PROMETHEE II.

The results obtained as a result of three experiments allow us to conclude that for the proper evaluation of a multi-criteria problem, certain conditions must be met:

- firstly, it is necessary to select the evaluation criteria (attributes) characterizing the most important features of the analyzed issue from the user's point of view,
- secondly - the selection of a method that will guarantee that the collected data will be properly used,
- thirdly - the method of comparing the results obtained,
- fourthly - it is also recommended to select the appropriate criteria weights, the structure of which may reflect the preferences of the decision maker or the client.

The current research shows that the most comparable rankings were obtained using the TOPSIS and COMET methods, while the greatest differences in relation to the results obtained from other methods were observed using the VIKOR method. On the other hand, the results obtained using the VIKOR method were closest to the results obtained using the TOPSIS method.

When testing the sensitivity of the results to changes in the criteria weights during the experiments, the results obtained with the TOPSIS and VIKOR methods behaved similarly. Separate results were obtained for the results obtained using the PROMETHEE II method.

It seems that it is difficult to judge the optimality of the methods used or their objectivity even after this series of experiments. However, it cannot be ruled out that the combination of several methods, e.g. simple methods for preliminary analyzes and complex methods such as VIKOR or PROMEYHEE II for more comprehensive results, would give positive results.

The main barriers of this work were the limitation of obtaining data to academic representatives of generation Z, which on the one hand indicates the main, future users of banking services, but on the other hand makes it difficult to generalize conclusions and use a limited number of the five most popular MCDA complex methods.

Nevertheless, the results of the conducted experiments encourage to continue research in order to expand the set of MCDA methods enabling the pursuit of convergence of results and thus the objectification of assessments in the analyzed area. Also, due to the importance of the banking sphere and the importance of the tool of communication with the user, which is the website, this direction will dominate in future research.

References

[1] W. Chmielarz and M. Zborowski, 'Comparative Analysis of Electronic Banking Websites in Poland in 2014 and 2015', in Information Technology for Management, E. Ziemba, Ed., in Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2016, pp. 147–161. doi: 10.1007/978-3-319-30528-8_9.

[2] C.-L. Hwang and K. Yoon, 'Methods for Multiple Attribute Decision Making', in Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey, C.-L. Hwang and K. Yoon, Eds., in Lecture Notes in Economics and Mathematical Systems. Berlin, Heidelberg: Springer, 1981, pp. 58–191. doi: 10.1007/978-3-642-48318-9_3.

[3] M. Cinelli, M. Kadziński, M. Gonzalez, and R. Słowiński, 'How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy', Omega, vol. 96, pp. 240–261, Oct. 2020, doi: 10.1016/j.omega.2020.102261.

[4] W. Chmielarz and M. Zborowski, 'Conversion Method in Comparative Analysis of e-Banking Services in Poland', in Perspectives in Business Informatics Research, A. Kobyliński and A. Sobczak, Eds., in Lecture Notes in Business Information Processing. Berlin, Heidelberg: Springer, 2013, pp. 227–240. doi: 10.1007/978-3-642-40823-6_18.

[5] A. Piegat and W. Sałabun, Comparative Analysis of MCDM Methods for Assessing the Severity of Chronic Liver Disease, vol. 9119. 2015, p. 238. doi: 10.1007/978-3-319-19324-3-21.

[6] W. Sałabun and A. Piegat, 'Comparative analysis of MCDM methods for the assessment of mortality in patients with acute coronary syndrome', Artif Intell Rev, vol. 48, no. 4, pp. 557–571, Dec. 2017, doi: 10.1007/s10462-016-9511-9.

[7] W. Sałabun, 'The Characteristic Objects Method: A New Distance-based Approach to Multicriteria Decision-making Problems', Journal of Multi-Criteria Decision Analysis, vol. 22, no. 1–2, pp. 37–50, 2015, doi: 10.1002/mcda.1525.

[8] A. Karczmarczyk, J. Wątróbski, and J. Jankowski, 'Comparative Study of Different MCDA-Based Approaches in Sustainable Supplier Selection Problem', in Information Technology for Management: Emerging Research and Applications, E. Ziemba, Ed., in Lecture Notes in Business Information Processing. Cham: Springer International Publishing, 2019, pp. 176–193. doi: 10.1007/978-3-030-15154-6_10.

[9] M. Kumar and C. Samuel, 'Selection of Best Renewable Energy Source by Using VIKOR Method', Technol Econ Smart Grids Sustain Energy, vol. 2, no. 1, p. 8, Apr. 2017, doi: 10.1007/s40866-017-0024-7.

[10] J.-M. Martel and B. Matarazzo, ‘Other Outranking Approaches’, in Multiple Criteria Decision Analysis: State of the Art Surveys, J. Figueira, S. Greco, and M. Ehrogott, Eds., in International Series in Operations Research & Management Science. New York, NY: Springer, 2005, pp. 197–259. doi: 10.1007/0-387-23081-5_6.

[11] J. Papathanasiou and N. Ploskas, ‘VIKOR’, in Multiple Criteria Decision Aid: Methods, Examples and Python Implementations, J. Papathanasiou and N. Ploskas, Eds., in Springer Optimization and Its Applications. Cham: Springer International Publishing, 2018, pp. 31–55. doi: 10.1007/978-3-319-91648-4_2.

[12] J. Jankowski, W. Sałabun, and J. Wątróbski, ‘Identification of a Multicriteria Assessment Model of Relation Between Editorial and Commercial Content in Web Systems’, 2017, pp. 295–305. doi: 10.1007/978-3-319-43982-2_26.

[13] ‘Identification of a Multi-criteria Model of Location Assessment for Renewable Energy Sources | SpringerLink’. https://link.springer.com/chapter/10.1007/978-3-319-39378-0_28 (accessed May 26, 2023).

[14] B. Kizielewicz, J. Wątróbski, and W. Sałabun, ‘Identification of Relevant Criteria Set in the MCDA Process-Wind Farm Location Case Study’, Energies, vol. 13, p. 6548, Dec. 2020, doi: 10.3390/en13246548.

[15] N. Tsotsolas and S. Alexopoulos, ‘MCDA Approaches for Efficient Strategic Decision Making’, 2018, pp. 17–58. doi: 10.1007/978-3-319-90599-0_2.

[16] J. Wątróbski, J. Jankowski, P. Ziemba, A. Karczmarczyk, and M. Zioło, ‘Generalised framework for multi-criteria method selection’, Omega, vol. 86, pp. 107–124, Jul. 2019, doi: 10.1016/j.omega.2018.07.004.

[17] W. Chmielarz and M. Zborowski, ‘Analysis of e-Banking Websites’ Quality with the Application of the TOPSIS Method – A Practical Study’, Procedia Computer Science, vol. 126, pp. 1964–1976, Jan. 2018, doi: 10.1016/j.procs.2018.07.256.

[18] A. Papapostolou, F. D. Mexis, E. Sarmas, C. Karakosta, and J. Psarras, ‘Web-based Application for Screening Energy Efficiency Investments: A MCDA Approach’, in 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA, Jul. 2020, pp. 1–7. doi: 10.1109/IISA50023.2020.9284403.

[19] W. Chmielarz and M. Zborowski, ‘Towards Sustainability in E-Banking Website Assessment Methods’, Sustainability, vol. 12, no. 17, Art. no. 17, Jan. 2020, doi: 10.3390/su12177000.

[20] W. Chmielarz and M. Zborowski, ‘Towards VES Function for Creating a Sustainable Method for Evaluating e-Banking Websites Quality’, Procedia Computer Science, vol. 192, pp. 5139–5155, Jan. 2021, doi: 10.1016/j.procs.2021.09.292.

[21] H. Voogd, ‘Multicriteria Evaluation with Mixed Qualitative and Quantitative Data’, Environment and Planning B, vol. 9, no. 2, pp. 221–236, 1982.

[22] P. Fortemps, S. Greco, and R. Słowiński, ‘Multicriteria Choice and Ranking Using Decision Rules Induced from Rough Approximation of Graded Preference Relations’, in Rough Sets and Current Trends in Computing, S. Tsumoto, R. Słowiński, J. Komorowski, and J. W. Grzymała-Busse, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 510–522. doi: 10.1007/978-3-540-25929-9_62.

[23] W. Sałabun, J. Wątróbski, and A. Shekhovtsov, ‘Are MCDA methods benchmarkable? A comparative study of TOPSIS, VIKOR, COPRAS, and PROMETHEE II methods’, Symmetry, vol. 12, pp. 1–55, Sep. 2020, doi: 10.3390/sym12091549.

[24] A. Shekhovtsov, J. Więckowski, B. Kizielewicz, and W. Sałabun, ‘Effect of Criteria Range on the Similarity of Results in the COMET Method’, Sep. 2021, pp. 453–457. doi: 10.15439/2021F44.

[25] D. Liang, Y. Zhang, Z. Xu, and A. Jamaldeen, ‘Pythagorean fuzzy VIKOR approaches based on TODIM for evaluating internet banking website quality of Ghanaian banking industry’, Applied Soft Computing, vol. 78, pp. 583–594, May 2019, doi: 10.1016/j.asoc.2019.03.006.

[26] P. Ziemba, ‘Towards Strong Sustainability Management—A Generalized PROSA Method’, Sustainability, vol. 11, no. 6, Art. no. 6, Jan. 2019, doi: 10.3390/su11061555.

[27] J.-P. Brans and B. Mareschal, ‘Promethee Methods’, in Multiple Criteria Decision Analysis: State of the Art Surveys, J. Figueira, S. Greco, and M. Ehrogott, Eds., in International Series in Operations Research & Management Science. New York, NY: Springer, 2005, pp. 163–186. doi: 10.1007/0-387-23081-5_5.

[28] C. Beaudrie, C. Corbett J, T. A. Lewandowski, T. Malloy, and X. Zhou, ‘Evaluating the Application of Decision Analysis Methods in Simulated Alternatives Assessment Case Studies: Potential Benefits and Challenges of Using MCDA’, Integrated Environmental Assessment and Management, vol. 17, no. 1, pp. 27–41, Feb. 2020, doi: 10.1002/ieam.4316.

[29] A. Shekhovtsov, J. Kołodziejczyk, and W. Sałabun, ‘Fuzzy Model Identification Using Monolithic and Structured Approaches in Decision Problems with Partially Incomplete Data’, Symmetry, vol. 12, no. 9, Art. no. 9, Sep. 2020, doi: 10.3390/sym12091541.

[30] A. Shekhovtsov, V. Kozlov, V. Nosov, and W. Sałabun, ‘Efficiency of Methods for Determining the Relevance of Criteria in Sustainable Transport Problems: A Comparative Case Study’, Sustainability, vol. 12, no. 19, Art. no. 19, Jan. 2020, doi: 10.3390/su12197915.

[31] R. Likert, A Technique for the Measurement of Attitudes. New York: New York University, 1932.

[32] ‘Respondenci do ankiet online’. https://ankieteo.pl/program-do-ankiet/respondenci (accessed May 10, 2023).

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1223–1227

DOI: 10.15439/2023F1144
ISSN 2300-5963 ACSIS, Vol. 35

Let's estimate all parameters as probabilities: Precise estimation using Chebyshev's inequality, Bernoulli distribution, and Monte Carlo simulations

Lubomír Štěpánek[†, ‡], Filip Habarta[†], Ivana Malá[†], Luboš Marek[†]
[†]Department of Statistics and Probability, [‡]Department of Mathematics
Faculty of Informatics and Statistics
Prague University of Economics and Business
W. Churchill's square 4, 130 67 Prague, Czech Republic
{lubomir.stepanek, filip.habarta, malai, marek}@vse.cz

Abstract—**Regarding the parameter estimation task, besides the time effectiveness of the simulation, parameter estimates are required to be precise enough. Usually, the estimates are Monte Carlo-simulated using a prior estimated variability within a small sample. However, the problem with pre-estimated variability is that it can be estimated imprecisely or, even worse, underestimated, resulting in estimation bias. In this work, we address the abovementioned issue and suggest estimating all parameters as probabilities. Since the probability is not only finite but has its theoretical maximum as 1, using outcomes of Bernoulli and binomial distribution's upper-bounded variance and Chebyshev's inequality, the estimator's variability is theoretically upper-bounded within the Monte Carlo simulation and estimation process. It cannot be underestimated or estimated inaccurately; thus, its precision is ensured till a given decimal digit, with very high probability. If there is a known process that treats the parameter of interest in terms of probability, we can estimate how many iterations of the Monte Carlo simulation are needed to ensure parameter estimate on a given level of precision. Also, we analyze the asymptotic time complexity of the proposed estimation strategy and illustrate the approach on a short case study of π constant estimation.**

I. Introduction

In THIS work, we focus on the estimation of parameters that are of a non-probabilistic fashion, e.g., simulated estimates of claim amounts in actuaries [1] or simulated numbers of patients at risk of disease recurrence [2]. Typically, these parameters are hardly analytically derivable, thus estimated using Monte Carlo simulation and the following logic [3]. Firstly, within an initial Monte Carlo simulation, a number of iterations (100, 1000, and so) generating the parameter estimate is run, and the parameter estimates from individual iterations are averaged, and their standard deviation is calculated. Then, applying the central limit theorem, a confidence interval for the parameter is constructed, and the Monte Carlo simulation is repeated so many times that the interval is no wider than a given precision. A problem of the abovementioned approach is in the parameter's variability estimation within the initial Monte Carlo simulation. If the variability is underestimated, the confidence interval is falsely narrower than it should be, and the precision is, in fact, lower than expected. To overcome this issue, we rather refine the simulation logic – firstly, we find a function of the parameter equal to some probability, which is then simulated using Monte Carlo simulation. Since the probability has a theoretically-based upper bound, its variability is upper-bounded. Then, we use the properties of the Bernoulli distribution to estimate the largest possible variability of the parameter as a probability and Chebyshev's inequality to enumerate the number of iterations keeping the parameter estimate's precision. Due to Chebyshev's inequality, we do not need the assumption of the parameter estimate's normality, which makes the proposed approach more robust.

II. A traditional approach to Monte Carlo simulation and estimation of parameters of non-probabilistic fashion

Let us assume a parameter θ of a non-probabilistic fashion that can be estimated n times using point estimates $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_n$. Then, calculation of the estimates' average and standard deviation, i.e.,

$$\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_i \qquad \text{and} \qquad \sigma_\theta = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\hat{\theta}_i - \bar{\theta})^2},$$

is feasible. To estimate parameter θ using Monte Carlo simulation on a given level of precision $1-\varepsilon$, where $\varepsilon \gtrsim 0$, reached with probability $1-\alpha$, one needs to know a number of iterations n of the simulation [3].

A. Principles of the traditional approach to Monte Carlo simulation and estimation of parameters of non-probabilistic fashion

Adopting the mathematical notation from the previous section, typical values of ε and α are, e.g., $\varepsilon = 0.001$ and $\alpha = 0.05$, respectively. A traditional approach to parameter θ estimation follows.

Thematic track: Computer Aspects of Numerical Algorithms

(i) Choose n_0 for initial Monte Carlo simulation that would pre-estimate parameter θ using individual estimates $\hat{\theta}_{0,1}, \hat{\theta}_{0,2}, \ldots, \hat{\theta}_{0,n_0}$. Typically, n_0 is chosen as $n_0 = 100$ or $n_0 = 1000$ or similar.

(ii) Calculate an average and a standard deviation of the pre-estimated parameter as

$$\bar{\theta}_0 = \frac{1}{n_0}\sum_{i=1}^{n_0} \hat{\theta}_{0,i} \text{ and } \sigma_{0,\theta} = \sqrt{\frac{1}{n_0 - 1}\sum_{i=1}^{n_0} (\hat{\theta}_{0,i} - \bar{\theta}_0)^2}.$$

(iii) Applying Ljapunov's central limit theorem [4], parameter θ should lie in an interval of

$$\left\langle \bar{\theta}_0 - u_{1-\alpha/2}\frac{\sigma_{0,\theta}}{\sqrt{n}},\ \bar{\theta}_0 + u_{1-\alpha/2}\frac{\sigma_{0,\theta}}{\sqrt{n}} \right\rangle \quad (1)$$

in $(1-\alpha)n$ cases of n total cases, thus, approximately with a probability $1-\alpha$, where $u_{1-\alpha/2}$ is the $(1-\alpha/2)$-th quantile of the standard normal distribution.

(iv) Number of iterations n of the main Monte Carlo simulation, outputting parameter estimates $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n$, is chosen to keep precision $1-\varepsilon$ with probability $1-\alpha$, so the confidence interval's half length from formula (1) should be less than or equal to ε, thus

$$u_{1-\alpha/2}\frac{\sigma_{0,\theta}}{\sqrt{n}} \leq \varepsilon, \quad (2)$$

and, equivalently, the number of needed iterations is

$$n \geq \left(\frac{u_{1-\alpha/2} \cdot \sigma_{0,\theta}}{\varepsilon}\right)^2. \quad (3)$$

(v) Finally, parameter θ is estimated using $\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}_i$, believed to keep precision $1-\varepsilon$ with probability $1-\alpha$.

B. Limitations of the traditional approach to Monte Carlo simulation and estimation of parameters of non-probabilistic fashion

Although the abovementioned approach works in general and is commonly applied, it can suffer from not meeting the asymptotic properties assumed by Ljapunov's central limit theorem when the confidence interval from formula (1) is constructed. This might happen particularly for low values of n or very high demands on precision, e.g., when $\varepsilon < 10^{-6}$. On a more practical note, inspecting formula (3), if the parameter's standard deviation σ_θ is underestimated by $\sigma_{0,\theta}$, i.e., when $\sigma_{0,\theta} < \sigma_\theta$, then also number n of iterations needed to keep imprecision $\leq \varepsilon$ is underestimated, which may result into imprecise, i.e., wrong (!) decimal digits staring the i-th digit behind (or before) the decimal point, where $i = \lfloor |\log_{10}(\varepsilon)| \rfloor$, if $\varepsilon < 1$ (or $\varepsilon > 1$, respectively).

C. The asymptotic time complexity of the traditional approach to Monte Carlo simulation and estimation of parameters of non-probabilistic fashion

Obviously, if one iteration of the Monte Carlo simulation takes τ units of time, then, since the simulation is repeated two times, firstly with n_0 iterations and secondly with $n \geq \left(\frac{u_{1-\alpha/2} \cdot \sigma_{0,\theta}}{\varepsilon}\right)^2$ iterations as comes from formula (4), the total asymptotic time complexity of the procedure, $\Theta(\dagger)$, is

$$\begin{aligned} \Theta(\dagger) &= \Theta\left((n_0 + n)\tau\right) \geq \\ &\geq \Theta\left(\left(n_0 + \left(\frac{u_{1-\alpha/2} \cdot \sigma_{0,\theta}}{\varepsilon}\right)^2\right)\tau\right), \end{aligned} \quad (4)$$

so, while $\Theta(\dagger)$ is linear in n_0 and n terms, it is quadratic in $\sigma_{0,\theta}$ and $\frac{1}{\varepsilon}$ terms.

III. A proposed approach to Monte Carlo simulation and estimation of parameters of (non-)probabilistic fashion

Let us suppose a parameter θ of non-probabilistic fashion. Besides the traditional approach for θ estimation as introduced above, we may assume a link function $f(\bullet)$ so that $f(\theta)$ has got a dimension of probability, so,

$$f(\theta) = P(\mathcal{T}), \quad (5)$$

where $P(\bullet)$ is a probability function as comes from σ-algebra, and $\mathcal{T}$ is a random event or a proposition consisting of random events. If occasionally θ would be apriori a probability, then the link function $f(\bullet)$ is an identity, i.e., $f(\theta) = \theta$, and the approach below still works. That is why we bound the prefix *non-* into brackets in the section title.

Thus, to estimate parameter θ of the (non-)probabilistic fashion, keeping precision $1-\varepsilon$ with probability $1-\alpha$, let us first assume a random variable X following Bernoulli distribution with an argument $P(\mathcal{T})$, i.e., $f(\theta)$ (probability of success). A sum of n independent Bernoulli trials follows the binomial distribution with arguments n (number of trials) and $f(\theta)$ (probability of success in each trial). After collecting n estimates $\hat{X}_i$ coming from the above mentioned Bernoulli distribution, we calculate $\frac{1}{n}\sum_{i=1}^{n}\hat{X}_i$ to estimate parameter $f(\theta)$. The number of trials n, i.e., a number of iterations of Monte Carlo simulation, is prior estimated using Chebyshev's inequality, also considering the terms of precision, $1-\varepsilon$, and confidence probability, $1-\alpha$.

A. Mathematical and statistical preliminaries of the proposed approach to Monte Carlo simulation and estimation

As we have seen, we need to revisit Bernoulli and binomial distribution and Chebyshev's inequality and their statistical properties. So let's start with Bernoulli and binomial distributions.

Definition 1 (Bernoulli distribution)**.** *A random variable X follows Bernoulli distribution with an argument $0 \leq p \leq 1$ (probability of success), if*

$$X = \begin{cases} 1, & \textit{with probability } p \\ 0, & \textit{with probability } 1-p. \end{cases}$$

Formally, we write $X \sim$ Bernoulli(p).

Lemma 1 (Bernoulli distribution's expected value and variance)**.** *Let random variable X follow Bernoulli distribution with an argument p. Then expected value of X is $\mathbb{E}(X) = p$ and variance of X is $\operatorname{var}(X) = p(1-p)$.*

Proof. According to definition 1, it is $\mathbb{E}(X) = \sum_{i \in \{0,1\}} i \cdot P(X = i) = 0 \cdot (1-p) + 1 \cdot p = p$, and $\mathbb{E}\left(X^2\right) = \sum_{i \in \{0,1\}} i^2 \cdot P(X = i) = 0^2 \cdot (1-p) + 1^2 \cdot p = p$. Since routinely is $\operatorname{var}(X) = \mathbb{E}\left(X^2\right) - (\mathbb{E}(X))^2$, we get $\operatorname{var}(X) = \mathbb{E}\left(X^2\right) - (\mathbb{E}(X))^2 = p - p^2 = p(1-p)$.

□

Definition 2 (Binomial distribution)**.** *A random variable X follows binomial distribution with arguments $n \in \mathbb{N}$ and $0 \leq p \leq 1$, if X is sum of n independent variables following Bernoulli distribution with an argument $0 \leq p \leq 1$ (i.i.d.). Formally, we write $X \sim \text{binomial}(n, p)$.*

Lemma 2 (Binomial distribution's expected value and variance)**.** *Let random variable X follow binomial distribution with arguments n and p. Then expected value of X is $\mathbb{E}(X) = np$ and variance of X is $\operatorname{var}(X) = np(1-p)$.*

Proof. According to definition 2, if $X \sim \text{binomial}(n, p)$, it is $X = Y_1 + Y_2 + \cdots + Y_n = \sum_{i=1}^{n} Y_i$, where $Y_i \sim \text{Bernoulli}(p)$ for $\forall i \in \{1, 2, \ldots, n\}$. So, $\mathbb{E}\left(Y_i\right) = p$ (†) and $\operatorname{var}\left(Y_i\right) = p(1-p)$ (‡). Thus,

$$\mathbb{E}(X) = \mathbb{E}\left(\sum_{i=1}^{n} Y_i\right) \overset{\text{i.i.d.}}{=} \sum_{i=1}^{n} \mathbb{E}\left(Y_i\right) \overset{(\dagger)}{=} \sum_{i=1}^{n} p = np, \tag{6}$$

and also

$$\begin{aligned} \operatorname{var}(X) = \operatorname{var}\left(\sum_{i=1}^{n} Y_i\right) \overset{\text{i.i.d.}}{=} \sum_{i=1}^{n} \operatorname{var}\left(Y_i\right) \overset{(\ddagger)}{=} \sum_{i=1}^{n} p(1-p) = \\ = np(1-p). \end{aligned} \tag{7}$$

□

Lemma 3 (Binomial distribution's maximum variance)**.** *Let random variable X follow binomial distribution with arguments n and p. Then maximum possible variance of X is $\operatorname{var}(X) = \frac{n}{4}$.*

Proof. According to lemma 2 and formula (7), if $X \sim \text{binomial}(n, p)$, it is $\operatorname{var}(X) = np(1-p)$. Let mark $p \equiv \frac{1}{2} + \delta$, where $\delta \in \left\langle -\frac{1}{2}, \frac{1}{2} \right\rangle$. Then, obviously,

$$\begin{aligned} \operatorname{var}(X) = np(1-p) = n\left(\frac{1}{2} + \delta\right)\left(1 - \left(\frac{1}{2} + \delta\right)\right) = \\ = n\left(\frac{1}{2} + \delta\right)\left(\frac{1}{2} - \delta\right) = n\left(\frac{1}{4} - \delta^2\right) \leq \frac{n}{4}. \end{aligned} \tag{8}$$

Thus, generally $\operatorname{var}(X) \leq \frac{n}{4}$, and $\operatorname{var}(X) = \frac{n}{4} = n\left(\frac{1}{4} - 0^2\right)$ for $\delta = 0$, so if and only if $p = \frac{1}{2} + \delta = \frac{1}{2} + 0 = \frac{1}{2}$. □

Finally, let's revisit Markov's and Chebyshev's inequalities [5], that enables us to derive the number of needed iterations of Monte Carlo simulation.

Theorem 1 (Markov's inequality)**.** *Let X be a non-negative random variable with expected value $\mathbb{E}(X)$. For $a > 0$ is*

$$P(X \geq a) \leq \frac{\mathbb{E}(X)}{a}. \tag{9}$$

Proof. Surely, since $a > 0$ and $X \geq 0$, it is $\mathbb{E}(X \mid X < a) \geq 0$ and $\mathbb{E}(X \mid X \geq a) \geq a$ (†). Because $\mathbb{E}(X \mid X < a) \geq 0$ and $P(X < a) \geq 0$, it is also $\mathbb{E}(X \mid X < a)P(X < a) \geq 0$. So, $\mathbb{E}(X) \geq \mathbb{E}(X) - \mathbb{E}(X \mid X < a)P(X < a)$. Also, due to the total expectations theorem, it is $\mathbb{E}(X) = \mathbb{E}(X \mid X < a)P(X < a) + \mathbb{E}(X \mid X \geq a)P(X \geq a)$, and $\mathbb{E}(X) - \mathbb{E}(X \mid X < a)P(X < a) = \mathbb{E}(X \mid X \geq a)P(X \geq a)$. Thus,

$$\begin{aligned} \mathbb{E}(X) \geq \mathbb{E}(X) - \mathbb{E}(X \mid X < a)P(X < a) = \\ = \mathbb{E}(X \mid X \geq a)P(X \geq a) \overset{(\dagger)}{\geq} a \cdot P(X \geq a), \end{aligned}$$

and, finally,

$$P(X \geq a) \leq \frac{\mathbb{E}(X)}{a}.$$

□

Theorem 2 (Chebyshev's inequality)**.** *Let X be a random variable with expected value $\mathbb{E}(X)$, and non-zero and finite variance $0 < \operatorname{var}(X) < \infty$. For $b > 0$ is*

$$P(|X - \mathbb{E}(X)| \geq b) \leq \frac{\operatorname{var}(X)}{b^2}. \tag{10}$$

Proof. If we realize that $\operatorname{var}(X) = \mathbb{E}\left((X - \mathbb{E}(X))^2\right)$ (†) and formally put $X \equiv (X - \mathbb{E}(X))^2$ and $a \equiv b^2$ into formula (9) of Markov's inequality, we directly get Chebyshev's inequality,

$$\begin{aligned} P(X \geq a) &\leq \frac{\mathbb{E}(X)}{a} \\ P\left((X - \mathbb{E}(X))^2 \geq b^2\right) &\leq \frac{\mathbb{E}\left((X - \mathbb{E}(X))^2\right)}{b^2} \\ P\left(|X - \mathbb{E}(X)| \geq b\right) &\overset{(\dagger)}{\leq} \frac{\operatorname{var}(X)}{b^2}. \end{aligned}$$

□

B. Number of needed iterations of Monte Carlo simulation for parameter estimation keeping the estimate's given precision

Let's assume a random variable X following Bernoulli distribution with an argument $P(\mathcal{T})$, i.e., $X \sim \text{Bernoulli}(P(\mathcal{T}))$. Thus, probability of a success in each Bernoulli trial is $P(\mathcal{T}) = f(\theta)$. If we repeat Bernoulli trials n times, based on definition 1, we can get a random variable $\sum_{i=1}^{n} X_i$, where $X_i \sim \text{Bernoulli}(P(\mathcal{T}))$ for $i \in \{1, 2, \ldots, n\}$. Since lemma 2, it is $\mathbb{E}\left(\sum_{i=1}^{n} X_i\right) = nP(\mathcal{T}) = nf(\theta)$ and $\operatorname{var}\left(\sum_{i=1}^{n} X_i\right) = nP(\mathcal{T})(1 - P(\mathcal{T})) = nf(\theta)(1 - f(\theta))$ (†). Using Chebyshev's inequality from formula (10), we get

$$P\left(\left|\sum_{i=1}^{n} X_i - \mathbb{E}\left(\sum_{i=1}^{n} X_i\right)\right| \geq b\right) \leq \frac{\operatorname{var}\left(\sum_{i=1}^{n} X_i\right)}{b^2}$$
$$P\left(\left|\sum_{i=1}^{n} X_i - nf(\theta)\right| \geq b\right) \overset{(\dagger)}{\leq} \frac{nf(\theta)(1-f(\theta))}{b^2}$$

We can simplify the right-hand side using lemma 3, i.e., $\operatorname{var}\left(\sum_{i=1}^{n} X_i\right) \leq \frac{n}{4}$, so

$$P\left(\left|\sum_{i=1}^{n} X_i - nf(\theta)\right| \geq b\right) \leq \frac{nf(\theta)(1-f(\theta))}{b^2} \overset{(8)}{\leq} \frac{n}{4b^2}$$
$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \frac{nf(\theta)}{n}\right| \geq \frac{b}{n}\right) \leq \frac{n}{4b^2}$$
$$P\left(\left|\bar{X} - f(\theta)\right| \geq \frac{b}{n}\right) \leq \frac{n}{4b^2},$$

let's set $b \equiv n\varepsilon$, then we get

$$P\left(\left|\bar{X} - f(\theta)\right| \geq \frac{n\varepsilon}{n}\right) \leq \frac{n}{4(n\varepsilon)^2}$$
$$P\left(\left|\bar{X} - f(\theta)\right| \geq \varepsilon\right) \leq \frac{1}{4n\varepsilon^2},$$

and by setting the probability's uncertainty as $\frac{1}{4n\varepsilon^2} \leq \alpha$ is

$$P\left(\left|\bar{X} - f(\theta)\right| \geq \varepsilon\right) \leq \frac{1}{4n\varepsilon^2} \leq \alpha. \tag{11}$$

Formula (11) tells us that a probability of getting a distance between the parameter $f(\theta)$ and its estimate $\bar{X}$ greater than ε, is lower than α. Thus, to keep imprecision $\leq \epsilon$, i.e., to keep $\frac{1}{4n\varepsilon^2} \leq \alpha$, we need n iterations of the Monte Carlo simulation,

$$n \geq \frac{1}{4\alpha\varepsilon^2}, \tag{12}$$

and unlike (3), formula (12) does not include a stochastic term.

C. A scheme of the proposed approach to Monte Carlo simulation and estimation of parameters

The previous paragraphs and particularly formulas (11) and (12) suggest Monte Carlo simulation for not only probabilistic-like parameters, keeping non-underestimated precision, that consists of the following steps.

(i) Setting the tuning parameters of the simulation – precision $1-\varepsilon$ and probability $1-\alpha$.

(ii) Assuming formula (5), constructing a generative Bernoulli process $X \sim \text{Bernoulli}(P(\mathcal{T}))$. We want to estimate parameter θ's value since we don't know it using link function f and a different random process, known from theory, with outcome $P(\mathcal{T})$ where $P(\mathcal{T}) = f(\theta)$.

(iii) Repeating Bernoulli process n times, where $n \geq \frac{1}{4\alpha\varepsilon^2}$, and collecting the outcomes $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n$.

(iv) Finally, averaging the outcomes, $\bar{X} = \frac{1}{n}\sum_{i=1}^{n}\hat{X}_i$, by getting estimate $f(\theta)$ on precision level $1-\varepsilon$ with probability $1-\alpha$.

(v) Parameter θ's estimate is $f^{-1}(f(\theta)) = f^{-1}\left(\bar{X}\right)$. While parameter $f(\theta)$ is estimated with imprecision ε, parameter $\theta = f^{-1}\left(\bar{X}\right)$ is estimated with imprecision $f^{-1}(\varepsilon)$.

An algorithm for the proposed Monte Carlo simulation is in Algorithm 1.

Algorithm 1: The proposed approach to Monte Carlo simulation and estimation of parameters of not only probabilistic fashion

```
Data: generative Bernoulli process with probability of
      success $P(\mathcal{T})$, link function $f$ ensuring that
      $P(\mathcal{T}) = f(\theta)$, precision $1-\varepsilon$, probability $1-\alpha$
Result: parameter $\theta$'s estimate
$\boldsymbol{X} = \{\emptyset\}$                // a vector for;
                                 // estimates saving;
$n_{\min} = \frac{1}{4\alpha\varepsilon^2}$            // # of iterations;
for $i = 1 : n_{\min}$ do
    generate outcome $\hat{X}_i$ from $X \sim \text{Bernoulli}(P(\mathcal{T}))$;
    $\boldsymbol{X} = \{\boldsymbol{X}, \hat{X}_i\}$     // update the vector;
end
estimate $\theta$ as $\theta = f^{-1}\left(\bar{\boldsymbol{X}}\right) = f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\hat{X}_i\right)$;
```

D. The asymptotic time complexity of the proposed approach to Monte Carlo simulation and estimation of parameters

The simulation is repeated n times where $n \geq \frac{1}{4\alpha\varepsilon^2}$, as comes from formula (12). Assuming one iteration of the Monte Carlo simulation takes τ time units, the total asymptotic time complexity of the procedure, $\Theta(\ddagger)$, is

$$\Theta(\ddagger) = \Theta(n\tau) \geq \Theta\left(\frac{1}{4\alpha\varepsilon^2}\tau\right), \tag{13}$$

so, while $\Theta(\ddagger)$ is linear in n and $\frac{1}{\alpha}$ terms, it is quadratic in $\frac{1}{\varepsilon}$ term. To compare asymptotic time complexity $\Theta(\dagger)$ from formula (4) for the traditional estimation procedure and $\Theta(\ddagger)$ from formula (13) for the proposed one, assuming that $n_0 \ll n$, it is $\Theta(\ddagger) > \Theta(\dagger)$, since, in general, is $\frac{1}{\alpha} > u_{1-\alpha/2}$. Both functions $\frac{1}{\alpha}$ and $u_{1-\alpha/2}$ are monotonous and decreasing while α increases, but for $\alpha \leq 0.05$ is $u_{1-\alpha/2} \gtrsim 2$ while $\frac{1}{\alpha} \geq 20$. So, while the traditional approach is "faster" in terms of time complexity, it may suffer from false underestimating of the parameter estimate's variability.

E. Keeping the first k decimal digits precise in the proposed approach to Monte Carlo simulation and estimation

Due to avoiding the issue of variability coming from lemma 3 and Chebyshev's inequality (10), an appropriate setting of precision level $1-\varepsilon$ can ensure the first k decimal digits are correctly estimated within the proposed simulation and estimation approach. Inspecting formula (11), we can realize that $\bar{X} - \varepsilon \leq f(\theta) \leq \bar{X} + \varepsilon$ with probability $1-\alpha$. On a given probability level $1-\alpha$, to avoid rounding error up to k-th decimal digit, we should set ε as

$$\varepsilon < 0.5 \cdot 10^{-k}. \tag{14}$$

Moreover, if the inversion function to link function f is of an additive (linear) form, i.e., $\forall \xi, \eta \in \mathbb{R}$ is $f^{-1}(\xi+\eta) = f^{-1}(\xi) + f^{-1}(\eta)$, it is also $f^{-1}(\bar{X}-\varepsilon) \leq f^{-1}(f(\theta)) \leq f^{-1}(\bar{X}+\varepsilon)$, so, $f^{-1}(\bar{X}) - f^{-1}(\varepsilon) \leq \theta \leq f^{-1}(\bar{X}) + f^{-1}(\varepsilon)$, and we can estimate also the *real* imprecision level ε_θ for parameter θ, i.e., not only $f(\theta)$, as

$$f^{-1}(\varepsilon_\theta) < 0.5 \cdot 10^{-k}, \quad \text{or,} \quad \varepsilon_\theta < f(0.5 \cdot 10^{-k}). \tag{15}$$

IV. The proposed approach to Monte Carlo simulation and estimation applied: π estimation

Revisiting the well-known example of π constant estimation using Monte Carlo simulation, let us assume a quarter circle with a radius of 1 as in Fig. 1. For a random point $A = [x, y]$ in the unit square around the quarter circle, where $[x, y] \in \langle 0, 1\rangle^2$, the generative Bernoulli process $X \sim \text{Bernoulli}(P(\mathcal{T}))$ here returns number 1 if A lies in the quarter circle (in gray color in Fig. 1), otherwise it returns 0. Thus, the random event is $\mathcal{T} = \{A \in \text{quarter circle} \mid A \in \text{unit square}\}$ and $P(\mathcal{T}) = \frac{S_{\text{quarter circle}}}{S_{\text{unit square}}} = \frac{\pi \cdot 1^2}{4}/1 = \frac{\pi}{4}$. So, the Bernoulli process enables us to estimate $f(\theta) = \frac{\pi}{4}$, which implies the link function f as $f(\eta) = \frac{\eta}{4}$.

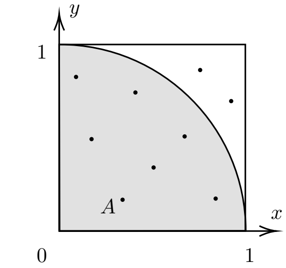


Fig. 1. A quarter circle in a unit square enabling estimation of $\frac{\pi}{4}$ parameter using Monte Carlo simulation of many points such as $A = [x, y] \in \langle 0, 1\rangle^2$.

Both for traditional and the proposed approach, we repeated Monte Carlo simulation $m = 100$ times to evaluate how likely the k-th decimal digit is not correct, with $k \in \{1, 2, 3\}$. We set probability level $\alpha = 0.05$ and real imprecision $\varepsilon_\theta = f(0.4 \cdot 10^{-k}) = \frac{0.4 \cdot 10^{-k}}{4} = 0.1 \cdot 10^{-k}$ using formula (15). The number of simulation iterations was estimated using formulas (3) and (12). The initial number of iterations for the traditional approach needed for pre-estimating the estimate's standard deviation $\sigma_{0,\theta}$, was $n_0 = 100$. We used R programming language and environment [6] for the Monte-Carlo simulations. There are more numerical applications of R language to various fields in [7]–[9].

Results are in Table I. While the traditional approach did not always ensure the precise k-th digit, particularly (but rarely, in $\leq \alpha = 0.05 = 5$ % of all cases) for $k = 2$ and $k = 3$, the proposed approach kept the k-th digit's precision every time. Unlike the proposed method not considering a stochastic term (see formula (12)), the traditional one may suffer from a possible underestimate of initial estimate's variability $\sigma_{0,\theta}$ and needed number n of Monte Carlo iterations (see formula (3)).

TABLE I
Proportions of cases when the k-th digit was incorrect out of $m = 100$ repetitions (marked as r) of Monte Carlo simulation.

		traditional approach		proposed approach	
k	ε_θ	$n_{\min}$	r	$n_{\min}$	r
1	0.01	1,028	0.00	50,000	0.00
2	0.001	102,765	0.01	5,000,000	0.00
3	0.0001	10,276,423	0.01	500,000,000	0.00

V. Conclusions remarks

We introduced an alternative approach to Monte Carlo estimation, using refining all estimated parameters as probabilities. That enables us to apply Bernoulli trials with upper-bounded variability of the estimate and Chebyshev's inequality for a robust estimate of the number of iterations needed to ensure the estimate's precision on a given probability level.

VI. Acknowledgment

This research is supported by grant IG410023 IGA no. 50/2023 provided by Internal Grant Agency of Prague University of Economics and Business.

References

[1] Søren Asmussen. "Conditional Monte Carlo for sums, with applications to insurance and finance". In: *Annals of Actuarial Science* 12.2 (Jan. 2018), pp. 455–478. DOI: 10.1017/s1748499517000252.

[2] Eleni G. Elia, Shirley Ge, Lisa Bergersen, et al. "A Monte Carlo Simulation Approach to Optimizing Capacity in a High-Volume Congenital Heart Pediatric Surgical Center". In: *Frontiers in Health Services* 1 (Feb. 2022). DOI: 10.3389/frhs.2021.787358.

[3] Christopher Mooney. *Monte Carlo Simulation*. SAGE Publications, Inc., 1997. DOI: 10.4135/9781412985116.

[4] Patrick Billingsley. *Probability and Measure*. en. 3rd ed. Wiley Series in Probability & Mathematical Statistics: Probability & Mathematical Statistics. Nashville, TN: John Wiley & Sons, May 1995.

[5] Gerold Alsmeyer. "Chebyshev's Inequality". In: *International Encyclopedia of Statistical Science*. Springer Berlin Heidelberg, 2011, pp. 239–240. DOI: 10.1007/978-3-642-04898-2_167.

[6] R Core Team. *R: A language and environment for statistical computing*. manual. Vienna, Austria, 2021. URL: https://www.R-project.org/.

[7] Patrícia Martinková, Lubomír Štěpánek, Adéla Drabinová, et al. "Semi-real-time analyses of item characteristics for medical school admission tests". In: *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*. IEEE, Sept. 2017. DOI: 10.15439/2017f380.

[8] Lubomír Štěpánek, Filip Habarta, Ivana Malá, et al. "Analysis of asymptotic time complexity of an assumption-free alternative to the log-rank test". In: *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*. IEEE, Sept. 2020. DOI: 10.15439/2020f198.

[9] Lubomír Štěpánek, Filip Habarta, Ivana Malá, et al. "Machine Learning at the Service of Survival Analysis: Predictions Using Time-to-Event Decomposition and Classification Applied to a Decrease of Blood Antibodies against COVID-19". In: *Mathematics* 11.4 (Feb. 2023), p. 819. DOI: 10.3390/math11040819.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1229–1233
DOI: 10.15439/2023F4335
ISSN 2300-5963 ACSIS, Vol. 35

A lower bound for proportion of visibility polygon's surface to entire polygon's surface: Estimated by Art Gallery Problem and proven that cannot be greatly improved

Lubomír Štěpánek[†, ‡], Filip Habarta[†], Ivana Malá[†], Luboš Marek[†]
[†]Department of Statistics and Probability
[‡]Department of Mathematics
Faculty of Informatics and Statistics
Prague University of Economics and Business
W. Churchill's square 4, 130 67 Prague, Czech Republic
{lubomir.stepanek, filip.habarta, malai, marek}@vse.cz

***Abstract*—Assuming a bounded polygon and a point inside the polygon or on its boundary, the visibility polygon, also called the visibility region, is a polygon reachable, i.e., visible by straight lines from the point without hitting the polygon's edges or vertices. If the polygon is bounded, then the visibility polygon is bounded, and the proportion of the visibility polygon's surface area to the given polygon's surface area could be enumerated. Many papers investigate applications of the visibility polygons in robotics and computer graphics or focus on computationally effective finding the visibility region for a given polygon. However, surprisingly, there seems to be no work estimating the proportion of a visibility polygon's surface to an entire polygon's surface or its bounds. Thus, in this paper, we search for a lower bound of the surface proportion of a visibility polygon to a given one. Assuming n-sided simple polygon, i.e., a polygon without holes and edge intersections, we apply the well-known art gallery problem and derive there is always a point inside the polygon or on its boundary that guarantees the proportion of the visibility polygon's surface to the entire polygon's surface is at least $\frac{1}{\lfloor n/3 \rfloor}$. We also show that there are n-sided polygons for which the proportion of the visibility polygon's surface to the entire polygon's surface is asymptotically not greater than $\frac{1}{\lfloor n/3 \rfloor}$ for any point inside the polygon or on its boundary. So, the lower bound of the proportion of the visibility polygon's surface to the entire polygon's surface, $\frac{1}{\lfloor n/3 \rfloor}$, cannot be improved in general.**

I. Introduction

THE visibility region of a polygon related to a given polygon's point, i.e., the largest part of the polygon, so that each point of such a polygon's part is directly visible from the given point, has multiple applications in robotics, operational research, and logistics, security, video game creation, and other situations, mainly of optimization character.

In robotics, visibility regions (polygons) are typically important for robotic agents to enable appropriate movement-making and planning [1]. There is a well-known problem called facility location problem where a number of facilities are to be optimally placed to minimize any transportation costs [2]. Besides other approaches, such as clustering, the visibility polygons could help to find the optimal facility setting [3], [4]. Similarly, in security applications, areas of various geometric shapes are often required to be guarded – then, a number and placement of guards watching the area could be researched using visibility polygons [5].

Since most problems are based on specific n-sided polygons, many papers search for an algorithm for building the visibility polygon in the shortest possible asymptotic time complexity. The naive approach takes quadratic time in terms of n – each pair of every two vertices is inspected to determine whether they are visible from a given point. While Asano published a faster sweeping algorithm for the visibility polygon construction, taking $n \log n$ asymptotic time [6], Lee developed the algorithm in linear time [7] by tricky stacking. Afterward, Joe and Simpson made Lee's algorithm even more robust, keeping it still in linear time [8].

In this paper, we go deeper rather into an estimate of a proportion of the visibility region's surface to the polygon's surface, assuming a point inside the polygon or on its boundary. In general, publications on this topic are missing. A guaranteed lower bound of the proportion, if this would be sufficiently high for at least one polygon's point, could, for instance, help in various tasks to decide whether one point with its visibility region is enough to satisfy the task conditions. On the other hand, cases of polygons that would show the lower bound of the proportion could not be greater than, e.g., some constant, might also imply that there is, for example, no optimal solution of the given task using only one chosen point. We research both situations more in detail and apply the outcomes on the *garden-watering problem*.

II. Preliminaries

Firstly, let us define the visibility region and the proportion of a visibility polygon's surface to an entire polygon's surface. In general, we do not assume convexity of polygons.

Definition 1 (The visibility polygon, visibility region)**.** *Let us assume a simple non-empty n-sided polygon $\mathcal{P}$, i.e., a polygon that does not intersect itself and has no holes, with $n \in \mathbb{N}$ and $n \geq 3$. Let A be a point inside polygon $\mathcal{P}$ or on its boundary, i.e., $A \in \mathcal{P}$. A visibility polygon (region) for point A in polygon $\mathcal{P}$ is polygon $\mathcal{V}_{A,\mathcal{P}}$ created by a set of all points $B \in \mathcal{P}$ so that each segment AB lies completely in $\mathcal{P}$, i.e., $AB \in \mathcal{P}$.*

As an example, the visibility polygon (region) of the polygon in Fig. 1 is colored in gray.

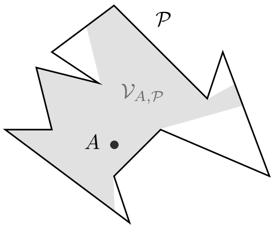


Fig. 1. An example of n-sided polygon $\mathcal{P}$ with $n = 12$. The visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ for point A in polygon $\mathcal{P}$ is in gray color.

Definition 2 (A proportion of a visibility polygon's surface to an entire polygon's surface)**.** *Following definition 1, let us assume $n \in \mathbb{N}$ so that $n \geq 3$, a simple non-empty n-sided polygon $\mathcal{P}$ and a point $A \in \mathcal{P}$. Let $S\left(\mathcal{V}_{A,\mathcal{P}}\right)$ be a surface of the visibility polygon for point A and $S(\mathcal{P})$ be a surface of polygon $\mathcal{P}$. Then, the proportion of the visibility polygon's surface $S\left(\mathcal{V}_{A,\mathcal{P}}\right)$ to the entire polygon's surface $S(\mathcal{P})$ is marked ν and is equal to*

$$\nu = \frac{S\left(\mathcal{V}_{A,\mathcal{P}}\right)}{S(\mathcal{P})}.$$

While trivial upper and lower bounds of the proportion of a visibility polygon's surface to an entire polygon's surface are apparent, as shown in the following lemma, more efficient estimates of the proportion bounds might be tricky, though.

Lemma 1. *Assuming definition 2, the proportion of a visibility polygon's surface $S\left(\mathcal{V}_{A,\mathcal{P}}\right)$ to an entire polygon's surface $S(\mathcal{P})$ is always*

$$0 \leq \nu = \frac{S\left(\mathcal{V}_{A,\mathcal{P}}\right)}{S(\mathcal{P})} \leq 1.$$

Proof. The n-sided polygon $\mathcal{P}$ is non-empty, so $S(\mathcal{P}) > 0$. Since surely $S\left(\mathcal{V}_{A,\mathcal{P}}\right) \geq 0$, it is $\nu = \frac{S(\mathcal{V}_{A,\mathcal{P}})}{S(\mathcal{P})} \geq 0$. The visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ is created by points $B \in \mathcal{P}$, thus $S\left(\mathcal{V}_{A,\mathcal{P}}\right) \leq S(\mathcal{P})$ and $\nu = \frac{S(\mathcal{V}_{A,\mathcal{P}})}{S(\mathcal{P})} \leq 1$. □

Due to the optimization fashion of the tasks using the visibility polygon, it is much more useful to investigate a lower bound of the proportion of a visibility polygon's surface to an entire polygon's surface, which may ensure that for a smart choice of the polygon's point, its visibility region is guaranteed to be sufficiently high. The proportion's upper bound equaled to 1, as shown in lemma 1, is often satisfied, e.g., for convex polygons, as proved below.

Lemma 2. *Assuming definition 2, the proportion of a visibility polygon's surface $S\left(\mathcal{V}_{A,\mathcal{P}}\right)$ to an entire polygon's surface $S(\mathcal{P})$ is equal to 1 if polygon $\mathcal{P}$ is convex.*

Proof. Let us proof that if polygon $\mathcal{P}$ is convex, then for any point $A \in \mathcal{P}$ is $\mathcal{V}_{A,\mathcal{P}} = \mathcal{P}$. Polygon $\mathcal{P}$ is convex, so, for each points $C \in \mathcal{P}$ and $D \in \mathcal{P}$ holds that $CD \in \mathcal{P}$ (†). By contradiction, let's assume there is point $X \in \mathcal{P}$ so that $X \notin \mathcal{V}_{A,\mathcal{P}}$. From definition 1, since the visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ is created by points $B \in \mathcal{P}$, it is surely $\mathcal{V}_{A,\mathcal{P}} \subseteq \mathcal{P}$. If $X \notin \mathcal{V}_{A,\mathcal{P}}$, then $AX \notin \mathcal{P}$, otherwise, due to definition 1, necessarily would be $X \in \mathcal{V}_{A,\mathcal{P}}$. But, if $AX \notin \mathcal{P}$ and both $A \in \mathcal{P}$ and $X \in \mathcal{P}$, this is contrary to (†). Thus, if polygon $\mathcal{P}$ is convex, then for any point $A \in \mathcal{P}$ is $\mathcal{V}_{A,\mathcal{P}} = \mathcal{P}$, so $S\left(\mathcal{V}_{A,\mathcal{P}}\right) = S(\mathcal{P}) > 0$ and $\nu = \frac{S(\mathcal{V}_{A,\mathcal{P}})}{S(\mathcal{P})} = 1$. □

If one could ensure that there always exists a point in a polygon so that the proportion of the surface of the visibility polygon for the point and surface of the entire polygon is greater than or equal to a constant, derivable from the polygon's characteristics, many of the optimization tasks, e.g., in security or logistics could have a plausible solution using only one point (or agent). In another way, there is a class of polygons for which the proportion of a visibility polygon's surface to an entire polygon's surface is not much greater than the constant from the previous case, regardless of the polygon's point choice; it is additional information for the task solution, too – there likely does not exist a satisfying solution using only one point (agent).

III. More on a lower bound of the proportion of a visibility polygon's surface to an entire polygon's surface

In the following sections, we investigate the existence of a simple n-sided polygon's point, i.e., a point in the polygon or on its boundary, so that a proportion of the surface of the visibility polygon for the point and the surface of the entire polygon is always greater than or equal to a constant, related to n. Also, we demonstrate there are n-sided polygons so that each visibility polygon is not much greater than the constant, regardless of the polygon's point selection. Thus, we show the surface proportion that is always greater than or equal to a constant for at least one point in the polygon could not be much more effective.

A. The existence of a polygon's point guaranteeing that the proportion of a visibility polygon's surface to an entire polygon's surface is not lower than a polygon-related constant

Using the popular art gallery problem [9], we prove that, assuming a simple non-empty n-sided polygon, there always exists a point in the polygon so that the proportion of the surface of the visibility polygon for the point, and the surface of the entire n-sided polygon is greater than or equal to $\frac{1}{\lfloor n/3 \rfloor}$.

Let us start with the art gallery problem, first introduced by Chvátal in [9].

Theorem 1 (The art gallery problem)**.** *Assume $n \in \mathbb{N}$ so that $n \geq 3$ and an art gallery of a shape following a simple non-empty n-sided polygon $\mathcal{P}$. Then, $\lfloor n/3 \rfloor$ guards are enough to watch the entire area of the art gallery, i.e., each point in the art gallery's polygon is visible by at least one of the $\lfloor n/3 \rfloor$ guards.*

Proof. The first proof of the classical art gallery problem was published in [9], and the short and elegant one came from [10]. □

In Fig. 2, there is an illustration of the art gallery problem for an art gallery following a shape of n-sided polygon $\mathcal{P}$ with $n = 8$.

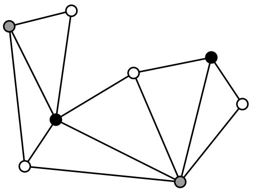


Fig. 2. An example of an art gallery following a shape of n-sided polygon $\mathcal{P}$ with $n = 8$. The n-sided polygon is triangulated, and white, gray, and black colors color the vertices of each triangle. As in theorem 1, $\lfloor n/3 \rfloor = \lfloor 8/3 \rfloor = 2$ guards, placed in vertices of black or gray color, is enough to ensure each point of the polygon is visible by at least one of them. (However, still, in fact, one guard, placed in the left bottom white vertex, is enough to watch the gallery.)

Working out some of the consequences of the art gallery problem, we get the following.

Theorem 2 (Surface of a visibility polygon of a guard in the art gallery)**.** *Let us assume $n \in \mathbb{N}$ and $n \geq 3$, and an art gallery following a simple non-empty n-sided polygon $\mathcal{P}$ with surface $S(\mathcal{P})$. Also, let us assume there are $\lfloor n/3 \rfloor$ guards, placed in vertices of the polygon. There exists a guard, i.e., a point in the polygon or on its boundary, so that a surface of their visibility polygon is greater than or equal to $\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$.*

Proof. Let us prove the theorem by contradiction. For $\forall i \in \{1, 2, \ldots, \lfloor n/3 \rfloor\}$, let $\mathcal{V}_{G_i,\mathcal{P}}$ be a visibility polygon of guard G_i and also let us assume that $S(\mathcal{V}_{G_i,\mathcal{P}}) < \frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$ (†). Due to theorem 1, $\lfloor n/3 \rfloor$ guards are enough to ensure that each point of the polygon is visible for one or more of them. Thus, the polygon created by union of all $\lfloor n/3 \rfloor$ guards' visibility polygons should be of a surface that covers the surface $S(\mathcal{P})$ of polygon $\mathcal{P}$. However, under (†), a surface of the union of all guards' visibility polygons is

$$\begin{aligned} S\left(\bigcup_{i=1}^{\lfloor n/3 \rfloor} \mathcal{V}_{G_i,\mathcal{P}}\right) &\leq \sum_{i=1}^{\lfloor n/3 \rfloor} S(\mathcal{V}_{G_i,\mathcal{P}}) \overset{(\dagger)}{<} \\ &\overset{(\dagger)}{<} \lfloor n/3 \rfloor \cdot \frac{S(\mathcal{P})}{\lfloor n/3 \rfloor} = \\ &= S(\mathcal{P}), \end{aligned}$$

which is contrary to theorem 1's output that $\lfloor n/3 \rfloor$ guards sufficiently secure the polygon. So, (†) cannot be true and the surface of each guard's visibility polygon cannot be lower than $\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$. Thus, there must exist at least one guard, i.e., a point in the polygon or on its boundary, with the visibility polygon of surface greater than or equal to $\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$. □

Finally, now we can prove the main idea of the section.

Theorem 3. *Let us assume $n \in \mathbb{N}$ and $n \geq 3$, and a simple non-empty n-sided polygon $\mathcal{P}$ with surface $S(\mathcal{P})$. There always exists a point A in polygon $\mathcal{P}$ or on its boundary, so that the proportion of surface $S(\mathcal{V}_{A,\mathcal{P}})$ of visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ for the point A, and surface $S(\mathcal{P})$ of polygon $\mathcal{P}$ is greater than or equal to $\frac{1}{\lfloor n/3 \rfloor}$, i.e.,*

$$\nu = \frac{S(\mathcal{V}_{A,\mathcal{P}})}{S(\mathcal{P})} \geq \frac{1}{\lfloor n/3 \rfloor}.$$

Proof. Revisiting theorem 2, the point A is identical to the guard with a visibility polygon of surface at least $\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$. The existence of such a guard is proved by theorem 1 and 2. Since the polygon is non-empty, i.e., $S(\mathcal{P}) > 0$ (‡), and surface of their visibility polygon is greater than or equal to $\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$, it is $S(\mathcal{V}_{A,\mathcal{P}}) \geq \frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}$ and

$$\nu = \frac{S(\mathcal{V}_{A,\mathcal{P}})}{S(\mathcal{P})} \geq \frac{\frac{S(\mathcal{P})}{\lfloor n/3 \rfloor}}{S(\mathcal{P})} \overset{(\ddagger)}{=} \frac{1}{\lfloor n/3 \rfloor}. \tag{1}$$

□

So, we demonstrated that there is always a point in a given simple n-sided polygon (convex or concave) that ensures that a proportion of its visibility polygon's surface and the entire polygon's surface is at least $\frac{1}{\lfloor n/3 \rfloor}$. Such knowledge could be handy in a class of tasks using the visibility polygon based on not necessarily coverage of a given polygon by the visibility polygon.

B. The polygons for which the proportion of a visibility polygon's surface to an entire polygon's surface is asymptotically not greater than a polygon-related constant

Although we showed that there must be a point in a simple n-sided polygon or on its boundary for which a proportion of its visibility polygon's surface and the entire polygon's surface is at least $\frac{1}{\lfloor n/3 \rfloor}$, for each $n \in \mathbb{N}$ where $n \geq 6$, there are polygons that regardless of the point choice, the surface proportion is asymptotically not greater than $\frac{1}{\lfloor n/3 \rfloor}$. Firstly, let's define such polygons and estimate the total surface of their visibility polygons.

Definition 3 (The saw-like polygons)**.** *Let $n \in \mathbb{N}$ so that $n \geq 6$. A saw-like n-sided polygon is a concave polygon consisting of $k = \lfloor n/3 \rfloor$ periodically repeating triple structures: each structure includes a triangle part, a base part and a connection part. While the triangle part is an isosceles triangle with a base of length $a > 0$, the base and connection parts are rectangles with edges of length $a > 0$ and $\varepsilon > 0$, where $a \gg \varepsilon \gtrsim 0$. Thus, the triangle part's surface is much greater than the base or connection part's surface. If $n = 3k$ for $k \in \mathbb{N}$, then the rightmost connection part is missing, and* *if $n = 3k + 1$, then the rightmost connection part is halved into a triangle shape; see Fig. 3 for details and illustration.*

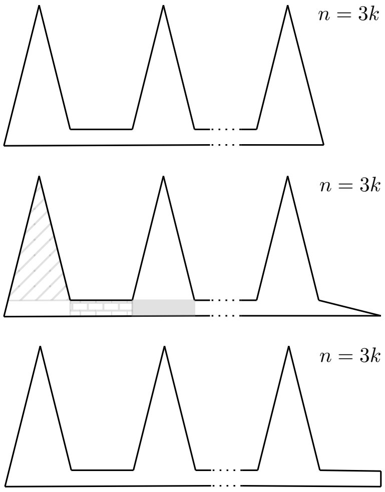


Fig. 3. Examples and patterns of the saw-like n-sided polygons. The triangle part is in a line hatch pattern, the base part is in gray, and the connection part is in a brick hatch pattern. Assuming $k \in \mathbb{N}$, then if $n = 3k$ for $k \in \mathbb{N}$, then the rightmost connection part is missing (top subfigure), if $n = 3k + 1$, the rightmost connection part is halved into a triangle shape (middle subfigure), if $n = 3k + 2$, the rightmost connection part is a rectangle as expected (bottom subfigure).

Lemma 3. *For each $n \in \mathbb{N}$ so that $n \geq 6$, i.e., $n = 3k$, or $n = 3k + 1$, or $n = 3k + 2$, where $k \in \mathbb{N}$, is $k = \lfloor n/3 \rfloor$.*

Proof. If $n = 3k$, then $k = \lfloor n/3 \rfloor = \left\lfloor \frac{3k}{3} \right\rfloor = \lfloor k \rfloor = k$. Else if $n = 3k + 1$, then $k = \lfloor n/3 \rfloor = \left\lfloor \frac{3k+1}{3} \right\rfloor = \left\lfloor k + \frac{1}{3} \right\rfloor = k$. Finally, if $n = 3k + 2$, then $k = \lfloor n/3 \rfloor = \left\lfloor \frac{3k+2}{3} \right\rfloor = \left\lfloor k + \frac{2}{3} \right\rfloor = k$. □

Theorem 4. *Let us assume a saw-like n-sided polygon $\mathcal{P}$ with $n \geq 6$ according to definition 3. For any point $A \in \mathcal{P}$, the proportion of its visibility polygon's surface $S\left(\mathcal{V}_{A,\mathcal{P}}\right)$ to an entire polygon's surface $S\left(\mathcal{P}\right)$ is asymptotically not greater than $\frac{1}{\lfloor n/3 \rfloor}$, i.e.,*

$$\nu = \frac{S\left(\mathcal{V}_{A,\mathcal{P}}\right)}{S(\mathcal{P})} \lesssim \frac{1}{\lfloor n/3 \rfloor}.$$

Proof. Mark $S(\mathcal{T})$ a surface of the triangle part, $S(\mathcal{B})$ a surface of the base part, and $S(\mathcal{C})$ a surface of the connection part. Applying definition 3, obviously, it is $S(\mathcal{B}) = S(\mathcal{C}) = a\varepsilon$ (†). Since lemma 3, polygon $\mathcal{P}$ contains exactly $k = \lfloor n/3 \rfloor$ structures (•) consisting of one triangle, base and connection part (with exception for the rightmost connection part[1], see Fig. 3), the surface of polygon $\mathcal{P}$ is

$$S(\mathcal{P}) \geq k \cdot (S(\mathcal{T}) + S(\mathcal{B}) + S(\mathcal{C})) - S(\mathcal{C}). \qquad (2)$$

For any point $A \in \mathcal{P}$ in any triangle part of polygon $\mathcal{P}$, the visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ includes the triangle part and, at maximum, all base and connection parts; thus, its surface is

$$S\left(\mathcal{V}_{A,\mathcal{P}}\right) \leq S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{T}} = S(\mathcal{T}) + k \cdot S(\mathcal{B}) + k \cdot S(\mathcal{C}), \quad (3)$$

see Fig. 4 for details. Also, for any point $A \in \mathcal{P}$ in any base part of polygon $\mathcal{P}$, the visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ includes the appropriate triangle part and, at maximum, all base and connection parts multiplied by a multiplier ℓ that reflects some fractions of other triangle parts could be partly visible; thus, its surface is

$$S\left(\mathcal{V}_{A,\mathcal{P}}\right) \leq S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{B}} = S(\mathcal{T}) + \ell k \cdot S(\mathcal{B}) + \ell k \cdot S(\mathcal{C}). \quad (4)$$

Finally, for any point $A \in \mathcal{P}$ in any connection part of polygon $\mathcal{P}$, the visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ includes, at maximum, all base and connection parts multiplied by a multiplier ℓ that reflects some fractions of triangle parts could be partly visible; thus, its surface is

$$S\left(\mathcal{V}_{A,\mathcal{P}}\right) \leq S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{C}} = \ell k \cdot S(\mathcal{B}) + \ell k \cdot S(\mathcal{C}). \qquad (5)$$

Putting formulas (3, 4, 5) together, we get

$$\begin{aligned} S\left(\mathcal{V}_{A,\mathcal{P}}\right) &\leq \max\left\{S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{T}}, S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{B}}, S\left(\mathcal{V}_{A,\mathcal{P}}\right)_{\mathcal{C}}\right\} = \\ &= S(\mathcal{T}) + \ell k \cdot S(\mathcal{B}) + \ell k \cdot S(\mathcal{C}). \qquad (6) \end{aligned}$$

Finally, since it is $a \gg \varepsilon \gtrsim 0$, it is also $S(\mathcal{T}) \propto a^2 \gg a\varepsilon = S(\mathcal{B}) = S(\mathcal{C}) \gtrsim 0$ and $1 \gg \frac{a\varepsilon}{S(\mathcal{T})} \gtrsim 0$ (∗). Also, we may expect, that $\ell \approx k$, or, moreover, we may set $\varepsilon \gtrsim 0$ so that $\ell k \cdot \frac{a\varepsilon}{S(\mathcal{T})} \gtrsim 0$ (‡). We get

$$\begin{aligned} \nu = \frac{S\left(\mathcal{V}_{A,\mathcal{P}}\right)}{S(\mathcal{P})} &\overset{(2,6)}{\leq} \frac{S(\mathcal{T}) + \ell k \cdot S(\mathcal{B}) + \ell k \cdot S(\mathcal{C})}{k \cdot (S(\mathcal{T}) + S(\mathcal{B}) + S(\mathcal{C})) - S(\mathcal{C})} \overset{(\dagger)}{=} \\ &\overset{(\dagger)}{=} \frac{S(\mathcal{T}) + \ell k \cdot a\varepsilon + \ell k \cdot a\varepsilon}{k \cdot (S(\mathcal{T}) + a\varepsilon + a\varepsilon) - a\varepsilon} = \\ &= \frac{S(\mathcal{T}) + 2\ell k \cdot a\varepsilon}{kS(\mathcal{T}) + (2k-1) \cdot a\varepsilon} = \\ &= \frac{1 + 2\ell k \cdot \frac{a\varepsilon}{S(\mathcal{T})}}{k + (2k-1) \cdot \frac{a\varepsilon}{S(\mathcal{T})}} \overset{(*)}{=} \\ &\overset{(*)}{=} \lim_{\frac{a\varepsilon}{S(\mathcal{T})} \to 0} \frac{1 + 2\ell k \cdot \frac{a\varepsilon}{S(\mathcal{T})}}{k + (2k-1) \cdot \frac{a\varepsilon}{S(\mathcal{T})}} \overset{(\ddagger)}{=} \\ &\overset{(\ddagger)}{=} \frac{1+0}{k+0} = \frac{1}{k} \overset{(\bullet)}{=} \\ &\overset{(\bullet)}{=} \frac{1}{\lfloor n/3 \rfloor}. \qquad (7) \end{aligned}$$

Thus, we showed that for any point in an n-sided saw-like polygon or on its boundary is the proportion ν of its visibility polygon's surface to an entire polygon's surface asymptotically not greater than $\frac{1}{\lfloor n/3 \rfloor}$, i.e., $\nu \lesssim \frac{1}{\lfloor n/3 \rfloor}$.

□

So, for each simple n-sided polygon, there indeed exists a point in the polygon or on its boundary so that the proportion ν of its visibility polygon's surface to an entire polygon's

[1] Formula (2) could be precised according to whether $n = 3k$, $n = 3k + 1$ or $n = 3k + 2$; however, it has only a low significance for the proof.

surface is $\nu \geq \frac{1}{\lfloor n/3 \rfloor}$. However, the saw-like polygons are examples of n-sided polygons where the proportion ν is upper-bounded, so it is $\nu \lesssim \frac{1}{\lfloor n/3 \rfloor}$, and, finally, $\frac{1}{\lfloor n/3 \rfloor} \leq \nu \lesssim \frac{1}{\lfloor n/3 \rfloor}$, i.e., $\nu \approx \frac{1}{\lfloor n/3 \rfloor}$ for any point in this kind of polygon or on its boundary. So, the lower bound estimate for the surface proportion ν cannot be in general greatly improved.

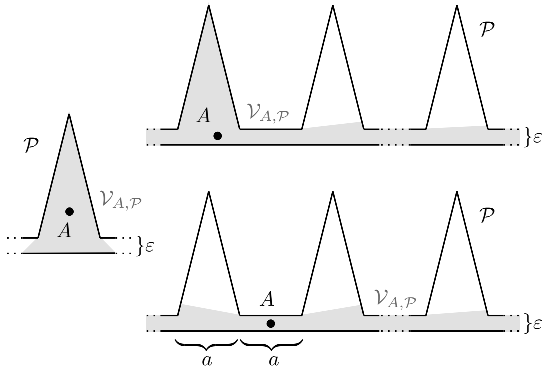


Fig. 4. Visibility polygons (in gray color) for point A in polygon $\mathcal{P}$. For any point $A \in \mathcal{P}$ the visibility polygon $\mathcal{V}_{A,\mathcal{P}}$ includes (i) the triangle part and, at maximum (!), all base and connection parts (left middle subfigure), if $A \in \mathcal{P}$ is in any triangle part of $\mathcal{P}$; (ii) the appropriate triangle part and, at maximum (!), all base and connection parts multiplied by a multiplier ℓ that reflects some fractions of other triangle parts could be partly visible, if $A \in \mathcal{P}$ is in any base part of $\mathcal{P}$ (right top subfigure); (iii) at maximum (!), all base and connection parts multiplied by a multiplier ℓ that reflects some fractions of triangle parts could be partly visible, if $A \in \mathcal{P}$ is in any connection part of $\mathcal{P}$ (right bottom subfigure).

IV. An application of the lower bound of the proportion of a visibility polygon’s surface to an entire polygon’s surface: garden-watering problem

Let us have a garden patch following a shape of a simple n-sided polygon, $n \geq 3$, that needs to be watered using a rotary sprinkler placed in the polygon or on its boundary. Due to the patch’s loose soil and water diffusion, it is enough to water only any η-proportion of the patch’s surface to hydrate the entire patch, where $0 \leq \eta \leq 1$. The sprinkler can water any point of the patch at any distance, but its water stream cannot cross the patch’s boundary. Is it possible to water the patch using only one sprinkler?

Solution. Applying theorem 3, one sprinkler can surely water ν-proportion of the entire patch, where $\nu \geq \frac{1}{\lfloor n/3 \rfloor}$. Thus, if $\frac{1}{\lfloor n/3 \rfloor} \geq \eta$, one sprinkler is sufficient. Moreover, using lemma 2, very likely one sprinkler could water an even greater proportion of the patch, up to $\nu = 1$ if the patch is, e.g., convex. However, if the patch follows a shape of the saw-like polygon as introduced in definition 3 for $n \geq 6$, one sprinkler can water no more than only ν-proportion of the patch, where $\nu \lesssim \frac{1}{\lfloor n/3 \rfloor}$, as proved in theorem 4. As a footnote, the solution is non-constructive, given the garden patch follows a simple n-sided polygon’s shape. So, a sprinkler of the properties mentioned above exists in the polygon or on its boundary. However, the introduced solution does not offer a way to find the exact position of the sprinkler in the polygon or on its boundary, satisfying the demanded properties. Searching for such a sprinkler position could be of high computational complexity. □

V. Conclusion remarks

Having an n-sided polygon without holes and edge intersections for $n \geq 3$, there is always a point inside the polygon or on its boundary that ensures a proportion of the visibility polygon’s surface for the point to the entire polygon’s surface is at least $\frac{1}{\lfloor n/3 \rfloor}$, as derived using the art gallery problem. Also, there are n-sided polygons, e.g., the saw-like ones for $n \geq 6$, so that for any point in the polygon or on its boundary is the proportion of the visibility polygon’s surface for such a point to the entire polygon’s surface not greater than $\frac{1}{\lfloor n/3 \rfloor}$. Thus, the lower bound of the proportion of the visibility polygon’s surface to the entire polygon’s surface, $\frac{1}{\lfloor n/3 \rfloor}$, cannot be generally improved.

VI. Acknowledgement

This research is supported by grant IG410023 IGA no. 50/2023 provided by Internal Grant Agency of Prague University of Economics and Business.

References

[1] D. Bilò, Y. Disser, M. Mihalák, et al. “Reconstructing visibility graphs with simple robots”. In: *Theoretical Computer Science* 444 (July 2012), pp. 52–59. DOI: 10.1016/j.tcs.2012.01.008.

[2] Sudipto Guha and Samir Khuller. “Greedy Strikes Back: Improved Facility Location Algorithms”. In: *Journal of Algorithms* 31.1 (Apr. 1999), pp. 228–248. DOI: 10.1006/jagm.1998.0993.

[3] Ram Pandit and Placid M. Ferreira. “Determination of minimum number of sensors and their locations for an automated facility: An algorithmic approach”. In: *European Journal of Operational Research* 63.2 (Dec. 1992), pp. 231–239. DOI: 10.1016/0377-2217(92)90028-8.

[4] Niall L. Williams, Aniket Bera, and Dinesh Manocha. “Redirected Walking in Static and Dynamic Scenes Using Visibility Polygons”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.11 (Nov. 2021), pp. 4267–4277. DOI: 10.1109/tvcg.2021.3106432.

[5] James King. “Fast vertex guarding for polygons with and without holes”. In: *Computational Geometry* 46.3 (Apr. 2013), pp. 219–231. DOI: 10.1016/j.comgeo.2012.07.004.

[6] Tetsuo Asano and Hiroshi Umeo. “Systolic algorithms for computing the visibility polygon and triangulation of a polygonal region”. In: *Parallel Computing* 6.2 (Feb. 1988), pp. 209–216. DOI: 10.1016/0167-8191(88)90085-3.

[7] D.T Lee. “Visibility of a simple polygon”. In: *Computer Vision, Graphics, and Image Processing* 22.2 (May 1983), pp. 207–221. DOI: 10.1016/0734-189x(83)90065-8.

[8] B. Joe and R. B. Simpson. “Corrections to Lee’s visibility polygon algorithm”. In: *BIT* 27.4 (Dec. 1987), pp. 458–473. DOI: 10.1007/bf01937271.

[9] V Chvátal. “A combinatorial theorem in plane geometry”. In: *Journal of Combinatorial Theory, Series B* 18.1 (Feb. 1975), pp. 39–41. DOI: 10.1016/0095-8956(75)90061-1.

[10] Steve Fisk. “A short proof of Chvátal’s Watchman Theorem”. In: *Journal of Combinatorial Theory, Series B* 24.3 (June 1978), p. 374. DOI: 10.1016/0095-8956(78)90059-x.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1235–1239

DOI: 10.15439/2023F1479
ISSN 2300-5963 ACSIS, Vol. 35

Social Media, Topic Modeling and Sentiment Analysis in Municipal Decision Support

Miloš Švaňa

VSB - Technical Univesity of Ostrava
Faculty of Economics
17. listopadu 2172/15, 708 00 Ostrava, Czechia
Email: milos.svana@vsb.cz

***Abstract*—Many cities around the world are aspiring to become *smart*. However, smart initiatives often give little weight to the opinions of average citizens.**

Social media are one of the most important sources of citizen opinions. This paper presents a prototype of a framework for processing social media posts with municipal decision-making in mind. The framework consists of a sequence of three steps: (1) determining the sentiment polarity of each social media post (2) identifying prevalent topics and mapping these topics to individual posts, and (3) aggregating these two pieces of information into a fuzzy number representing the overall sentiment expressed towards each topic. Optionally, the fuzzy number can be reduced into a tuple of two real numbers indicating the "amount" of positive and negative opinion expressed towards each topic.

The framework is demonstrated on tweets published from Ostrava, Czechia over a period of about two months. This application illustrates how fuzzy numbers represent sentiment in a richer way and capture the diversity of opinions expressed on social media.

I. Introduction

MANY CITIES around the world are aspiring to become *smart*. A *Smart City* is characterized by an extensive use of information technologies to support municipal decision-makers in effective resource utilization [17]. This ICT support can be applied in many areas, including traffic and transportation, waste management, accommodation or culture.

Ideally, municipal planning involves multiple stakeholder groups: local authorities, businesses, average citizens and commuters, or environmental activists. In practice, however, decisions are mostly based on an interplay between authorities and businesses offering Smart City technologies. Opinions of average citizens are often given little weight [9, 18].

Some cities use surveys to gather opinions on at least some issues and projects. Although useful, surveys have many disadvantages. Most importantly, by asking only a predetermined set of questions and letting the respondent choose only from a low number of possible answers they limit the respondent's ability to fully express their thoughts and opinions. This prevents decision makers from serendipitously discovering unexpected issues and ideas.

Social media represent an alternative source of citizen opinions. They offer much greater freedom of expression as users can create new content whenever they want and instead of being limited by a set of questions, they can use free-form text, images, videos or audio. This freedom together with significant content creation velocity might also be a disadvantage. Relatively simple statistical methods for processing survey data are inadequate for social media. Instead, advanced techniques from fields such as machine learning, computer vision, or natural language processing have to be used.

This paper presents a prototype of a framework for extracting information from social media with the aim to support municipal decision-making. The framework combines topic modeling techniques with sentiment analysis. First, the system detects topics discussed on social media at a specific location. Then it evaluates sentiment towards each topic. To capture uncertainty arising from different people having different opinions on the same topic, this sentiment is modelled as a triangular fuzzy number (TFN). The TFN representation, however, might not be understood by people without background knowledge. Therefore a calculation of a "degree of conformity" with fuzzy sets representing the concepts of positive and negative opinion follows. The result can be interpreted as an "amount" of positive/negative opinion expressed towards a specific topic.

The remainder of the paper is organized as follows: In section II I review literature related to topic modeling, sentiment analysis and their applications in the context of Smart Cities. Section III discusses the framework design. Section IV then describes data used in experiments and preprocessing procedures. Demonstration of the framework application on test data can be found in section V. Finally, section VI discusses the limitations of the framework and future research.

II. Literature review

A. Topic Modeling

Topic modeling can be understood as (1) a "statistical technique for revealing the underlying semantic structure in a large collection of documents", or (2) "a technique comes with group of algorithms that reveal, discover and annotate thematic structure in collection of documents" [12].

In practice, topic modeling involves taking a corpora of text documents and discovering various topics discussed in these documents. Topics are usually represented by a set of most relevant terms [4]. Once topics are identified, a mapping between topics and individual documents can be created. Depending on the method used, one document can be assigned to one or multiple topics.

There is a plethora of topic modeling methods. In [12], the authors distinguish between two categories: (1) algebraic methods, usually based on some form of word-document matrix factorization and (2) statistical methods. On the other hand, [4] provides a chronological overview of the development of topic modeling methods. Other reviews, e.g., [16] then focus on selecting the best method for a given dataset or application.

One of the most popular topic modeling methods is Latent Dirichlet Allocation (LDA). This method is based on the idea that documents in a given corpora are generated by a probabilistic process. Each document can be understood as a mixture of topics, with each topic representing a probability distribution over words from some vocabulary [3]. However, the framework presented in this paper relies on a more recent method called BERTopic [8]. It outperforms LDA on multiple benchmark datasets both in coherence and topic diversity – two common topic modeling evaluation metrics [4].

B. Sentiment analysis

Sentiment analysis is "an approach that uses Natural Language Processing (NLP) to extract, convert and interpret opinion from a text and classify them into positive, negative or natural sentiment" [7]. There are two main groups of sentiment analysis methods (1) lexicon-based and (2) methods based on supervised learning models.

Lexicon-based methods use sentiment lexicons containing information about sentiment polarity of different words. Individual word polarities in a given document are looked up and then aggregated into an overall document polarity.

Methods based on supervised learning models rely on various machine learning techniques to train a classification model for predicting document sentiment. Naive Bayes is one of the most popular approaches in this category. Supervised methods require a training dataset that contains a ground truth sentiment for each document. Many product and service review websites let their users combine textual reviews with some sort of a numerical rating scale, e.g., a 5-star rating system. This data can be easily used to train a sentiment classifier.

One of the contributions of this paper lies in the application of fuzzy modelling methods in sentiment analysis. Several researchers have already explored this path. In [11] authors describe a fuzzy expert system for sentiment analysis. There have also been attempts at designing a sentiment analysis method that utilizes a fuzzy thesaurus [10]. However, to our best knowledge, fuzzy methods have not been used to aggregate sentiment expressed towards multiple documents.

There have been multiple applications of sentiment analysis in the context of Smart Cities and urban planning [5, 14]. There are many opportunities and challenges related to the use of sentiment analysis in urban planning, including visualization, multilingual audiovisual opinion mining, or peer-to-peer opinion mining tools for citizens [2].

III. Framework Design

A. Topic Modeling

As mentioned, the framework prototype uses a topic modeling method called BERTopic. Originally proposed in [8], BERTopic solves the problem of topic modeling by combining word embeddings with hierarchical clustering. The procedure consists of the following steps:

1) **Create an embedding for each document**. One of the downsides of traditional methods such as LDA is that they represent documents in an bag-of-words fashion. This representation ignores both order of words in a document and their semantic relationships. BERTopic relies on an embedding representation. Embeddings are vectors that are able to somewhat capture the semantics of words or documents. Documents with similar meaning should be represented by similar vectors. As the name suggests, BERTopic uses embeddings based on BERT [6].
2) **Reduce embedding dimensionality**. Embeddings can have hundreds or even thousands of dimensions. When it comes to clustering, high dimensionality causes multiple issues. [8]. First, the difference between the distance of the nearest point to a cluster centre and the distance between the furthest point from a cluster centre shrinks [1]. Second, lower number of dimensions leads to better performance both in terms of time and clustering accuracy. BERTopic therefore uses UMAP [15] to reduce the number of dimensions.
3) **Use a clustering algorithm to create document clusters**. BERTopic uses HDBSCAN with document embeddings reduced by UMAP as input. HDBSCAN is a hierarchical algorithm able to create a tree of cluster-subcluster structures. BERTopic user can set the number of clusters/topics to be generated by the algorithm. To provide the desired number of clusters, small similar topics are merged together.
4) **Create topic representations**. Similarly to other topic modeling methods, BERTopic represents each topic as a list of words. To find words best describing a given topic BERTopic uses a modified TF-IDF score [8] calculated as:
$$W_{t,c} = tf_{t,c} \ \log(1 + \frac{A}{tf_t}) \tag{1}$$
where $tf_{t,c}$ represents the frequency of term t in cluster c. A is the average number of words per cluster and tf_t is the total frequency of term t across all clusters.

B. Sentiment Analysis

The TextBlob[1] Python library was selected for the framework prototype. This choice was made mainly for pragmatic

[1] https://github.com/sloria/TextBlob

reasons - ease of installation, use and integration with other parts of the framework.

TextBlob offers two sentiment analysis methods: a lexicon and pattern-based one and a pretrained Naive Bayes model. In the former, TextBlob uses a polarity lexicon and structural patterns to determine both the degree of polarity and the degree of subjectivity. The Naive Bayes model was trained to classify movie reviews as positive or negative. Instead of providing degrees of polarity and subjectivity as output, this method returns the probabilities of a given text being positive and negative. Being the default, the lexicon and pattern-based method was also used in presented experiments. The output of this model carries a certain degree of uncertainty which can be exploited when aggregating sentiment analysis results with fuzzy methods.

C. *Fuzzy Aggregation*

Once sentiment polarity and topics are extracted from a set of social media posts, both pieces of information are combined to provide an overall view of what is being discussed in a given municipality and whether the population perceives a given topic positively or negatively.

Simple aggregation metrics such as arithmetic mean would lead to a loss of information. For example, there might be a controversial topic with many positive, but also some negative opinions. Arithmetic mean might present the topic sentiment as slightly positive. When presented with this information, the user cannot tell whether the aggregated opinion is slightly positive because a majority of individual opinions is slightly positive or because there are many positive opinions counterbalanced by a small number of negative opinions.

To address this issue, topic sentiment is modeled as a triangular fuzzy number (TFN). Assuming we know both the sentiment polarity of each social media post, and their topic distribution, the TFN core can be determined as a weighted mean with topic distribution percentages serving as weights. For instance, consider 3 topics with polarities and topic distributions depicted in table I. The core of the TFN representing the sentiment towards topic 1 can be calculated as:

$$m_{t1} = \frac{0.5 \cdot 0.5 + 0.35 \cdot 0.3 - 0.2 \cdot 0.2}{0.5 + 0.3 + 0.2} = 0.315 \tag{2}$$

Weighted standard deviation is then used to determine the TFN support interval. As in the case of determining the TFN core, the degree to which a given post belongs to a given topic should determine the strength of its influence on the shape of the support interval. Therefore a weighted variant of standard deviation is used, the general formula of which being:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{N} w_i (x_i - x^*)^2}{\frac{M-1}{M} \sum_{i=1}^{N} w_i}} \tag{3}$$

with N representing the total sample size, x^* representing weighted mean, and M the number of non-zero weights.

Once the weighed standard deviation is known, the TFN support interval is calculated as:

$$[m_{t1} - s\sigma; m_{t1} + s\sigma] \tag{4}$$

where s is a positive real number that determines the scaling between the length of the support interval and weighted standard deviation. Its influence on the aggregation result is a subject of further research.

TABLE I
EXAMPLE OF AN INPUT FOR THE AGGREGATION PROCEDURE

	Polarity	**Topic distribution**		
		Topic 1	**Topic 2**	**Topic 3**
Post 1	0.5	0.5	0.3	0.2
Post 2	0.35	0.3	0.4	0.3
Post 3	-0.2	0.2	0.2	0.6

Taking inspiration from [19], the next step of the aggregation procedure is the calculation of a degree of conformity between the TFN describing topic sentiment and fuzzy sets representing the concepts of a positive and negative opinion. The result of this step is a tuple of two numbers from interval $[0; 1]$ that could be interpreted as the overall "amount" of positive and negative opinion expressed towards a given topic. Although there is some loss of information when compared to the original TFN, the two numbers are more likely to be understood by a layperson with no knowledge of fuzzy set theory.

The fuzzy metric of possibility [13] is used to calculate the degree of conformity, e.g.:

$$Pos(\tilde{A}, \tilde{PO}) = \sup_{x \in X} \min\left(\mu_{\tilde{A}}(x), \mu_{\tilde{PO}}(x)\right) \tag{5}$$

A denotes aggregated topic sentiment with $\mu_{\tilde{A}}(x)$ being its membership function. Similarly PO represents the concept of a positive opinion with a membership function $\mu_{\tilde{PO}}(x)$.

IV. DATA AND PREPROCESSING

Social media posts published on Twitter were used to demonstrate the proposed framework. This social network was chosen specifically because it lets researchers easily access its data through an API. Using this API, a dataset of approximately 3000 tweets published by users located in Ostrava, Czechia was created. These tweets cover the period of January and February of 2023.

Given the location, most tweets in the dataset are written in Czech. There are, however, several tweets written in other languages, such as English, Slovak or Polish. To address the issue of multilingualism, as well as the fact that TextBlob uses an English lexicon when analyzing sentiment, the DeepL Translator[2] API was used to automatically translate each post to English.

Several preprocessing steps were then applied on translated tweets:

[2] https://www.deepl.com/translator

- removal of tweets that are shorter than 60 characters under the assumption that it might be difficult to reliably extract sentiment and topics from short tweets,
- removal of non-letter characters,
- removal of hashtags, usernames and URLs,
- turning all text into lowercase,
- removal of stop words.

V. Framework Demonstration

The framework prototype was implemented as a Jupyter notebook. To summarize, after the input data was preprocessed, the following sequence of steps was applied:

1) TextBlob was used to determine the sentiment polarity of each post. Polarity of each post is represented as a real number from a $[-1, 1]$ interval with -1 representing absolutely negative polarity and positive number representing absolutely positive polarity.
2) BERTopic was used to identify topics and create a mapping between topics and individual tweets. Provided output follows the format in table I.
3) A set of TFNs representing aggregated topic sentiment was constructed. The scaling constant s in equation 4 was set to 1.
4) Degrees of conformity between each topic's TFN and the concepts "positive opinion" and "negative opinion" were determined.

Overall, BERTopic identified 42 different topics. Topics with the highest overall "mass" calculated as the sum of the topic's probability across all tweets are listed in table II.

Presented topic distribution intuitively makes sense. During given time period, one of the most dominant topics in public discourse was the upcoming presidential election which corresponds to the first topic in table II. One can also see that not all topics are necessarily relevant to municipal decision making, for example topic no. 3 in the table. At the same time, topics such as no. 9 are difficult to interpret.

TFNs representing the several aggregated topic sentiments are depicted in figure V. It can be deduced that the topic *vote, election, politics, party* is perceived quite positively and compared to topic *pay, wage, pension, live* it also has a narrower support. This indicates a lower level of opinion diversity. The latter topic is also perceived most negatively, at least among the topics displayed in the figure.

Table II also contains information about the conformity with the concepts of positive and negative opinion. The membership function of the fuzzy set representing positive opinion has a value of 0 until polarity value 0, then grows linearly to 1 until polarity reaches 0.2 and then has a value of 1. The "negative opinion" fuzzy set is its mirror image.

It can be seen that the information provided by these numbers tells a story similar to the TFN visualization. For example, when comparing topics *school, education, teacher, class* and *area, city, building, ostrava*, one could conclude that the former is not only perceived more positively, but that it is also less controversial given the lower "amount" of negative opinion.

VI. Discussion

As demonstrated, the framework prototype can be used to create a representation of opinions expressed towards different topics on social media posted by citizens of a specific municipality. The prototype, however, employs certain simplifications that should be addressed in future research.

The TFNs representing topic sentiments are currently symmetric. This is might not be the best reflection of reality. Ways of making the TFNs asymmetric should be explored. Metrics such as weighted skewness or semivariance could be used. Next, Twitter users have an option to "like" or "retweet" a post created by someone else. These actions can represent approval and could be therefore used as aggregation weights. Finally, other topic-modeling and sentiment-analysis methods should be tested and compared to the existing TextBlob+BERTopic stack. Moreover, TextBlob provides additional information beyond sentiment polarity: the subjectivity degree of each document. This metric could again be used as weight.

Framework output can be used directly by municipal decision makers to make more informed decisions. However, there are other possible applications. One of them is comparison of different municipalities. It should be possible to compare the TFNs of a specific topic across multiple cities. If additional information such as municipal budgets is available, one could also deploy methods such as Data Envelopment Analysis to evaluate their efficiency. Topic sentiments could be also aggregated into an overall sentiment expressed towards everything happening in a given municipality.

However, using the proposed framework as a replacement for other methods of gathering citizen opinion might not be the best course of action. Instead, the framework should play a complementary role, as social media users might not accurately represent the overall population. Groups such as the elderly might be underrepresented. At the same time, methods such as surveys can provide biased information too. Combining the proposed framework with surveys might lead to a better overall representation of citizen opinion than either of these methods separately.

VII. Acknowledgment

This paper was supported by the SGS project No. SP2023/078. This support is gratefully acknowledged.

References

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In J. Van den Bussche and V. Vianu, editors, *Database Theory — ICDT 2001*, pages 420–434, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[2] K. B. Ahmed, A. Radenski, M. Bouhorma, and M. B. Ahmed. Sentiment analysis for smart cities : State of the art and opportunities. In *Int'l Conf. Internet Computing and Internet of Things (ICOMP'16)*, 2016.

[3] D. M. Blei, A. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2001.

TABLE II
MOST PREVALENT TOPICS IDENTIFIED BY BERTOPIC IN TWEETS PUBLISHED FROM OSTRAVA.

No.	Topic	Prevalence	Positivity	Negativity
1	’vote’, ’election’, ’politics’, ’party’, ’ano’, ’government’, ’people’	41.98	0.997	0.190
2	’czech’, ’republic’, ’czech republic’, ’czechs’, ’czk’, ’inflation’	33.68	0.810	0.434
3	’boy’, ’date’, ’girl’, ’crying’, ’apologize’, ’shes’, ’ive’	27.19	0.812	0.390
4	’school’, ’education’, ’teacher’, ’class’, ’educated’, ’academics’, ’students’	25.93	0.897	0.287
5	’game’, ’field’, ’leader’, ’go’, ’games’, ’team’, ’match’	25.89	0.839	0.390
6	’area’, ’city’, ’building’, ’ostrava’, ’construction’, ’also’, ’mappa’	25.32	0.745	0.473
7	’hydrogen’, ’gt’, ’car’, ’anymore’, ’hydrogen bike’, ’product’, ’hydrogen technology’	19.47	0.741	0.459
8	’something nothing’, ’number’, ’nothing’, ’something’, ’old figure’, ’number thumbs’	18.81	0.760	0.272
9	’miracles’, ’ribbons’, ’expect miracles’, ’political responsibility’, ’imagine’	18.72	0.757	0.344

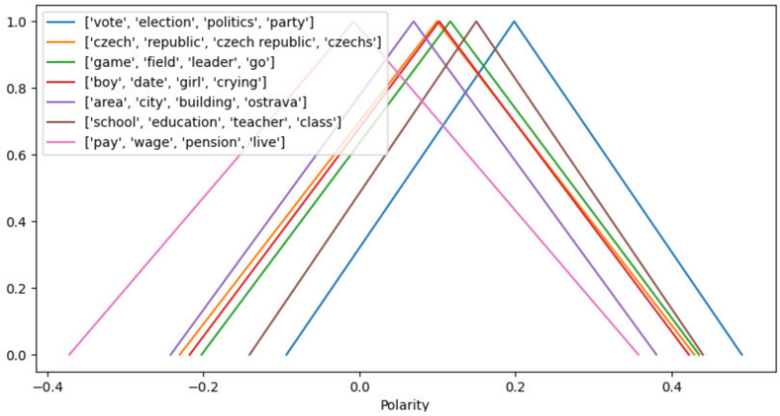


Fig. 1. Depiction of TFNs representing overall sentiment towards selected topics extracted from tweets published from Ostrava
.

[4] R. Churchill and L. Singh. The Evolution of Topic Modeling. *ACM Computing Surveys*, 54(10s):1–35, Jan. 2022.

[5] M. Dahbi, R. Saadane, and S. Mbarki. Social media sentiment monitoring in smart cities: an application to Moroccan dialects. In *Proceedings of the 4th International Conference on Smart City Applications*, pages 1–6, Casablanca Morocco, Oct. 2019. ACM.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[7] Z. Drus and H. Khalid. Sentiment Analysis in Social Media and Its Application: Systematic Literature Review. *Procedia Computer Science*, 161:707–714, 2019.

[8] M. Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure, 2022.

[9] G. R. Halegoua. *Smart cities*. The MIT Press, 2020.

[10] H. M. Ismail, B. Belkhouche, and N. Zaki. Semantic Twitter sentiment analysis based on a fuzzy thesaurus. *Soft Computing*, 22(18):6011–6024, Sept. 2018.

[11] C. Jefferson, H. Liu, and M. Cocea. Fuzzy approach for sentiment analysis. In *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, Naples, Italy, July 2017. IEEE.

[12] P. Kherwa and P. Bansal. Topic Modeling: A Comprehensive Review. *ICST Transactions on Scalable Information Systems*, 0(0):159623, July 2018.

[13] G. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall PTR, 1995.

[14] M. Li, E. Ch’ng, A. Chong, and S. See. The new eye of smart city: Novel citizen Sentiment Analysis in Twitter. In *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, pages 557–562, Shanghai, China, July 2016. IEEE.

[15] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[16] I. Vayansky and S. A. Kumar. A review of topic modeling methods. *Information Systems*, 94:101582, Dec. 2020.

[17] T. M. Vinod Kumar, editor. *Smart Economy in Smart Cities: International Collaborative Research: Ottawa, St.Louis, Stuttgart, Bologna, Cape Town, Nairobi, Dakar, Lagos, New Delhi, Varanasi, Vijayawada, Kozhikode, Hong Kong*. Advances in 21st Century Human Settlements. Springer Singapore, Singapore, 2017.

[18] K. S. Willis. Whose Right to the Smart City? In P. Cardullo, C. Di Feliciantonio, and R. Kitchin, editors, *The Right to the Smart City*, pages 27–41. Emerald Publishing Limited, June 2019.

[19] F. Zapletal, M. Hudec, M. Švaňa, and R. Němec. Three-level model for opinion aggregation under hesitance. *Soft Computing*, Feb. 2023.

PolEval

POLEVAL is an annual NLP challenge organized since 2017. The choice of the name of the challenge was deliberate: as most research concentrates on the most popular languages (especially English), the aim of PolEval was to promote work on processing Polish. By focusing on Polish, it actively promotes the creation of new resources in this language, facilitates further research and contributes to creating new and improved methods and models for Polish.

The goal of PolEval is thus to:

- develop established procedures for evaluating systems solving a wide range of tasks in NLP,
- create annotated datasets that can be used for training and evaluation of systems,
- objectively compare systems performing various tasks in the field of natural language processing,
- bring researchers from the scientific and business communities closer together and exchanging knowledge between them,
- facilitate popularization of NLP issues in the context of the Polish language.

To achieve these goals, PolEval proposes a well-formulated task framework, in which the scope, input data, expected output data, evaluation methods, training and test data are prepared by the organizers. This way the challenge aims to be a platform for objective comparison of methods, models and systems for processing Polish.

DOI: 10.15439/2023F5627
ISSN 2300-5963 ACSIS, Vol. 35

PolEval 2022/23 Challenge Tasks and Results

Łukasz Kobyliński*, Maciej Ogrodniczuk*, Piotr Rybak*, Piotr Przybyła*‡, Piotr Pęzik¶‖,
Agnieszka Mikołajczyk‖, Wojciech Janowski‖, Michał Marcińczuk† and Aleksander Smywiński-Pohl§
*Institute of Computer Science, Polish Academy of Sciences
Jana Kazimierza 5, 01-248 Warszawa, Poland
E-mail: {lukasz.kobylinski, maciej.ogrodniczuk, piotr.rybak, piotr.przybyla}@ipipan.waw.pl
†Department of Artificial Intelligence, Wrocław University of Science and Technology,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
E-mail: michal.marcinczuk@pwr.edu.pl
‡Universitat Pompeu Fabra
Barcelona, Spain
E-mail: piotr.przybyla@upf.edu
§AGH University of Krakow
Kraków, Poland
E-mail: apohllo@agh.edu.pl
¶University of Łódź, Poland
E-mail: piotr.pezik@uni.lodz.pl
‖VoiceLab.AI, Gdańsk, Poland
E-mail: agnieszka.mikolajczyk@voicelab.ai

***Abstract*—This paper summarizes the 2022/2023 edition of PolEval — an evaluation campaign for natural language processing tools for Polish. We describe the tasks organized in this edition, which are: Punctuation prediction from conversational language, Abbreviation disambiguation and Passage Retrieval. We also discuss the datasets prepared for each of the tasks, evaluation metrics chosen to rank the submissions and also sum up the approaches chosen by the participants to tackle the tasks.**

I. Introduction

POLEVAL[1] [14] is a SemEval-inspired evaluation campaign for natural language processing tools for Polish. Submitted tools compete against one another within certain tasks selected by organizers, using available data and are evaluated according to pre-established procedures.

The 2022/2023 edition of Poleval was the sixth event in a series of challenges organized since 2017. During this edition three tasks have been proposed:

1) Punctuation prediction from conversational language,
2) Abbreviation disambiguation,
3) Passage Retrieval.

The participants of this edition have been very active, as we have received more than 400 submissions from 23 teams. The submissions were made through our evaluation platform[2], which has been introduced last year.

In the following part of the paper we describe each of the tasks in detail, present the datasets created for the particular challenges, discuss the evaluation metrics and we give the overview of submissions made by the participants.

II. Task 1: Punctuation Prediction from Conversational Language

A. Problem statement

Speech transcripts generated by Automatic Speech Recognition (ASR) systems typically do not contain any punctuation or capitalization. In longer stretches of automatically recognized speech, lack of punctuation affects the general clarity of the output text [24]. The primary purpose of punctuation restoration (PR), punctuation prediction (PP), and capitalization restoration (CR) as a distinct natural language processing (NLP) task is to improve the legibility of ASR-generated text and possibly other types of texts without punctuation. For the purposes of this task, we define PR as restoration of originally available punctuation from read speech transcripts (which was the goal of a separate task in the PolEval 2021 competition) [10] and PP as prediction of possible punctuation in transcripts of spoken/ conversational language. Aside from their intrinsic value, PR, PP, and CR may improve the performance of other NLP aspects such as Named Entity Recognition (NER), part-of-speech (POS), and semantic parsing or spoken dialog segmentation [5], [12].

One of the challenges of developing PP models for conversational language is the availability of consistently annotated datasets. The very nature of naturally-occurring spoken language makes it difficult to identify exact phrase and sentence boundaries [21], [23], which means that dedicated guidelines are required to train and evaluate punctuation models.

The goal of the present task is to provide a solution for predicting punctuation in the test set collated for this task.

[1] http://poleval.pl
[2] https://beta.poleval.pl

B. Task description

The workflow of this task is illustrated in Figure 1 below. Given raw ASR output, the task is to predict punctuation in annotated ASR transcripts of conversational speech.

C. Dataset

The test set consisted of time-aligned ASR dialogue transcriptions from three sources:

1) CBIZ, a subset of DiaBiz [17], a corpus of phone-based customer support line dialogs[3]
2) VC, a subset of transcribed video-communicator recordings, which are included in the SpokesBiz Corpus[4]
3) SPOKES, a subset of the SpokesMix corpus [16].

Table I below summarizes the size of the three subsets in terms of dialogs, words and duration of recordings.

TABLE I
OVERALL STATISTICS OF THE CORPUS

Subset	Corpus and license	Files	Words	Audio (s)	Speakers
CBIZ	DiaBiz (CC-BY-SA-NC-ND)	69	36 250	16 916	14
VC	Video conversations (CC-BY-NC)	8	44 656	17 123	20
Spokes	Casual conversations (CC-BY-NC)	13	42 730	20 583	19

The full dataset has been split into three subsets as summarized in Table II below.

TABLE II
TRAINING / DEVELOPMENT / TEST SET STATISTICS

Set	Files	Words	Audio (s)	License
Train	69	98 095	44 030	CC-BY-SA-NC-ND
Dev	11	12 563	4 718	CC-BY-NC
Test	10	12 978	5 874	CC-BY-NC

The punctuation annotation guidelines were developed in the CLARIN-BIZ project by Karasińska et al. [20].

Participants are encouraged to use both text-based and speech-derived features to identify punctuation symbols (e.g. multimodal framework [22] or to predict casing along with punctuation [15]. We allow using the punctuation dataset available at http://2021.poleval.pl/tasks/task1 [10].

The punctuation marks evaluated as part of the task are listed in Table III below. Blanks are marked as spaces. The distribution of explicit punctuation symbols in the training and development portion of the dataset provided is shown in Tables III–VI.

1) Data format: We provide two types of data: text and audio data. Text data is provided in the TSV format. For Audio data we provide audio files encoded in WAV and transcripts with force-aligned timestamps. The audio files can be downloaded separately from the website of PolEval.

TABLE III
PUNCTUATION FOR RAW TEXT (ALL SUBCORPORA)

	Symbol	Mean	Median	Max	Sum
fullstop	.	111.15	59	1 157	8 892
comma	,	161.51	69	1 738	12 921
question_mark	?	24.36	11	229	1 949
exclamation_mark	!	3.46	4	45	277
hyphen	-	0.64	25	50	51
ellipsis	...	63.28	11	1 833	5 062
words		1 383.23	569	16 528	110 658

TABLE IV
PUNCTUATION FOR RAW TEXT (CBIZ)

	Symbol	Mean	Median	Max	Sum
fullstop	.	58.06	54	213	3 600
comma	,	70.61	59	388	4 378
question_mark	?	11.26	10	35	698
exclamation_mark	!	0.34	1	5	21
hyphen	-	0.02	1	1	1
ellipsis	...	12.29	9	54	762
words		528.74	483	2 180	32 782

TABLE V
PUNCTUATION FOR RAW TEXT (VC)

	Symbol	Mean	Median	Max	Sum
fullstop	.	411.86	384	1 157	2 883
comma	,	737.86	577	1 738	5 165
question_mark	?	85.29	41	229	597
exclamation_mark	!	10.43	5	43	73
hyphen	-	/	/	/	/
ellipsis	...	514.00	365	1 833	3 598
words		5 704.14	4 398	9 469	39 929

TABLE VI
PUNCTUATION FOR RAW TEXT (SPOKES)

	Symbol	Mean	Median	Max	Sum
fullstop	.	219.00	193	607	2 409
comma	,	307.09	313	614	3 378
question_mark	?	59.45	39	150	654
exclamation_mark	!	16.64	10	45	183
hyphen	-	4.55	50	50	50
ellipsis	...	63.82	45	186	702
words		3 449.73	1 966	16 528	37 947

2) Transcriptions and metadata: The datasets are encoded in the TSV format.

Field descriptions:

- column 1: name of the audio file
- column 2: unique segment id
- column 3: segment text, where each word is separated by a single space

The segment text (column 3) format is:

- single word text:word start timestamp in ms-word end timestamp in ms

[3] https://clarin-pl.eu/dspace/handle/11321/887
[4] http://docs.pelcra.pl/doku.php?id=spokesbiz

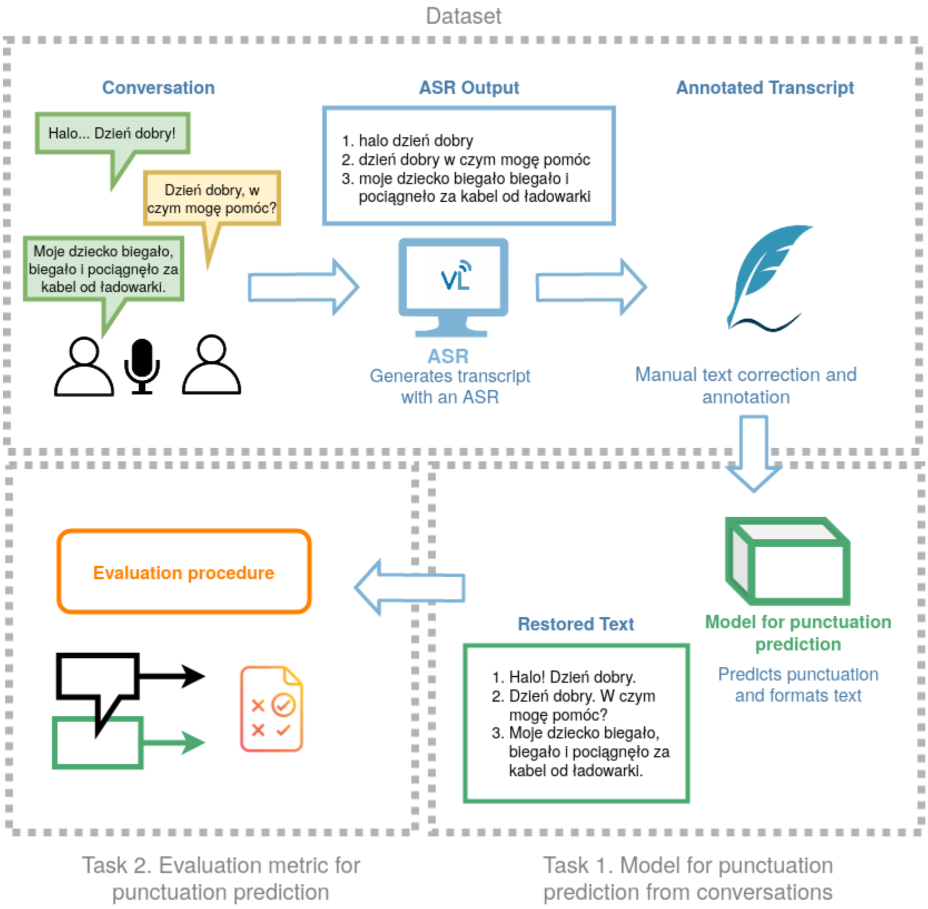

Dataset
Conversation
Halo... Dzień dobry!
Dzień dobry, w czym mogę pomóc?
Moje dziecko biegało, biegało i pociągnęło za kabel od ładowarki.
ASR Output
1. halo dzień dobry
2. dzień dobry w czym mogę pomóc
3. moje dziecko biegało biegało i pociągnęło za kabel od ładowarki
VL
ASR
Generates transcript with an ASR
Annotated Transcript
Manual text correction and annotation
Evaluation procedure
Restored Text
1. Halo! Dzień dobry.
2. Dzień dobry. W czym mogę pomóc?
3. Moje dziecko biegało, biegało i pociągnęło za kabel od ładowarki.
Model for punctuation prediction
Predicts punctuation and formats text
Task 2. Evaluation metric for punctuation prediction
Task 1. Model for punctuation prediction from conversations

Fig. 1. Overview of the punctuation prediction task

D. Evaluation

a) Submission format: Results were to be submitted as plain text file, where each line corresponds to a single segment. The text should include the predicted punctuation marks.

1) Metrics: The final results were evaluated in terms of precision, recall, and F1 scores for predicting each punctuation mark separately. Submissions were compared with respect to the weighted average of F1 scores for each punctuation sign. The method of evaluation was similar to the one used in a PolEval 2021 task named "Punctuation restoration from read text"[5] [10].

2) Per-document score::

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

3) Global score per punctuation sign p::

$$P_p = avg_{micro}Precision(p) = \frac{\sum_{d \in Documents} TP}{\sum_{d \in Documents} TP + FP}$$

$$R_p = avg_{micro}Recall(p) = \frac{\sum_{d \in Documents} TP}{\sum_{d \in Documents} TP + FN}$$

The final scoring metric was calculated as the weighted average of global scores:

$$\frac{1}{N} \sum_{p \in Punctuation} support(p) * avg_{micro}F_1(p)$$

TABLE VII
SUBMISSIONS TO THE PUNCTUATION PREDICTION TASK

Submission	Weighted-F1 score	
	Test-A	Test-B
Oskar Bujacz	79.24	**83.30**
Michał Pogoda	80.47	**82.33**
Jakub Pokrywka	67.30	**71.44**
Filip Graliński	30.88	**35.30**

[5] http://2021.poleval.pl/tasks/task1

E. Results

The winning solution submitted for Task 1 by Oskar Bujacz achieved a weighted F-measure of 83.3 (see Table VII). The author used a token classifier based on the largest variant of the HerBERT model[11] with customized output postprocessing rules.

III. Task 2: Abbreviation disambiguation

A. Problem statement

Abbreviations are often overlooked in many NLP pipelines. However, they are still an important point to tackle, especially in such applications as machine translation, named entity recognition, or text-to-speech systems.

There are at least two practical challenges in processing abbreviations. The first is the ability to find the full, expanded dictionary form of an abbreviation. In many cases, this may be done by a simple dictionary lookup, but: - the use of abbreviations is often unconventional and there is no complete list of all possible abbreviation uses, - many of the abbreviations are ambiguous. That is, the same abbreviation may have more than one meaning, translating to possibly different expanded forms.

As in many other NLP tasks, the disambiguation of abbreviations needs to include context and additional language knowledge to be feasible.

The second challenge, which is specific to languages with rich morphology, such as Polish, is the necessity to produce the expanded form of an abbreviation in correct grammatical form, in concordance with the rest of the sentence.

B. Task description

The task aimed to propose a method of disambiguating Polish abbreviations. The method should recognize if a given phrase is an abbreviation and, if so, produce its expanded form, both base, and inflected ones.

C. Dataset

1) Training data: In this task a (relatively small) training dataset was provided (see example in Figure 2), which included:

- the abbreviation
- an expanded form of the abbreviation
- a base form of the abbreviation
- context of the abbreviation, with the `′ placeholder marking the place where the abbreviation appeared.

The participants were encouraged to collect and use additional training and dictionary data and to publish it after the competition.

2) Test data: The test data consists of only the abbreviation and the context. The system aims to provide the expanded and base forms of the abbreviation.

D. Evaluation

We will calculate two measures of accuracy for each provided submission:

- Af — the accuracy of provided expanded forms of abbreviations (case insensitive string match)
- Ab — the accuracy of provided base forms of abbreviations (case insensitive string match).

Based on these measures, the final score will be calculated using a weighted average:

$$Acc = 0.25 * Af + 0.75 * Ab \tag{4}$$

E. Results

We received five submissions (see Table VIII). The final ranking was calculated based on the weighted accuracy of the Test-B dataset. The scores ranged from 92.01 to 19.09. Krzysztof Wróbel obtained the highest score of 92.01.

TABLE VIII
SUBMISSIONS TO THE ABBREVIATION DISAMBIGUATION TASK

Submission	**Weighted**	**accuracy**
	Test-A	Test-B
Krzysztof Wróbel	92.76	**92.01**
Jakub Karbowski	91.75	**91.27**
Marek Kozlowski	89.00	**88.73**
Jakub Pokrywka	65.48	**66.25**
Rafał Prońko	n/a	**19.09**

Krzysztof Wróbel utilized an ensemble of three models, each based on the byt5-base model[6], trained on different seeds, and employing a majority voting. The training of these models incorporated both the train and dev datasets, as well as a small dataset automatically generated from abbreviations sourced from various dictionaries such as Morfeusz [9], sjp.pl, and Wiktionary [7].

Jakub Karbowski (2nd place submission) trained a sequence-to-sequence model based on the plt5-base model[8]. The input to the model consisted of a context with a masked abbreviation, a target base form, and inflected forms of the expanded abbreviation. The initial training was performed on a synthetic dataset generated from the Polish Wikipedia. The dataset was created by randomly selecting contexts of varying lengths and shortening consecutive words using one of several strategies, such as using the first few letters, the first and last letters, or the first, middle, and last letters. The base form was generated using Spacy[9]. Then, the model was fine-tuned on the PolEval dataset.

IV. Task 3: Passage retrieval

A. Problem statement

Passage Retrieval is a crucial part of modern open-domain question-answering systems that rely on precise and efficient

[6] https://huggingface.co/google/byt5-base

[7] https://www.wiktionary.org

[8] https://huggingface.co/allegro/plt5-base

[9] https://spacy.io

Abbr	Expanded form	Base form	Context
s.	sobota	sobota	Karpaty Siepraw (n. 16). IV liga, grupa wschodnia: Olimpia Wojnicz - Grybovia (16), Orkan Szczyrzyc - Wolania Wola Rzędzińska (s. 16), Sandecja II Nowy Sącz -
d.	dawniej	dawniej	poinformowała w piątek na swej stronie internetowej rosyjska korporacja państwowa Rostech (Rostechnologii). Nie podano daty przechwycenia amerykańskiego drona. „Dron MQ-5B,"
n.	niedziela	niedziela	11) Gościbia - Piast (s. 16) Wiślanka - Sęp (s. 16); Skawinka - pauza Sęp - Gościbia (16.30) Piast - Hejnał (n. 17) Orzeł - Jordan (n. 17) Czarni - Nadwiślanka (s.
pkt. proc.	punktu procentowego	punkt procentowy	proc. Kolejne 0,12 pkt. proc. wynika ze spadku popytu na polski eksport, a 0,08 z zaburzeń na rynku wewnętrznym" - oszacowali.
rp.pl.	rp.pl.	rp.pl.	Jutro rozpocznie się proces posła ruchu Palikota - dowiedziała się Biedroń został oskarżony o naruszenie nietykalności cielesnej funkcjonariusza policji

Fig. 2. Examples from the training dataset

retrieval components to find passages containing correct answers. Traditionally, lexical methods, such as TF-IDF or BM25 [18], have been used to power retrieval systems. They are fast, interpretable, and don't require training (and therefore a training set). However, they can only return a document if it contains a keyword present in a query. In addition, their understanding of text is limited because they ignore word order.

Recently, neural retrieval systems (e.g. Dense Passage Retrieval [8]) have surpassed these traditional methods by fine-tuning pre-trained language models on a large number of (query, document) pairs. They solve the aforementioned problems of lexical methods but at the cost of the need to label training sets and poor generalisation to other domains. As a result, in a zero-shot setup (i.e. no training set), lexical methods are still competitive or even better than neural models.

B. Task description

The aim of the *passage retrieval* task was to develop a system for cross-domain question-answering retrieval. For each test question, the system should retrieve an ordered list of the ten most relevant passages (i.e. containing the answer) from the given corpus. The system is evaluated on the basis of its performance on test examples from three different domains, namely trivia, law, and customer support.

C. Dataset

1) Training set: The training set consisted of 5,000 trivia questions from the PolQA dataset [19]. Each question was accompanied by up to five passages from Polish Wikipedia containing the answer to the question. In total, the training set consisted of 16,389 question-passage pairs. In addition, we provided a Wikipedia corpus of 7,097,322 passages. The raw Wikipedia dump was parsed with WIKIEXTRACTOR[10] and split into passages at the end of paragraphs or if the passage was longer than 500 characters.

2) Test sets: The systems were evaluated on three test sets with questions from different domains. The first dataset consisted of 1,291 trivia questions similar to those in the training set.

The second dataset consisted of 900 questions and 921 passages related to the large Polish e-commerce platform - Allegro[11]. The dataset was created based on help articles and lists of frequently asked questions available on the Allegro website. Each question-passage pair was manually checked and edited where necessary.

The third dataset contained over 700 legal questions. It was created by randomly selecting the passage and manually writing a question. We also provided a corpus of approximately 26,000 passages extracted from over a thousand acts of laws published between 1993 and 2004.

D. Evaluation

The submitted systems were evaluated using Normalised Discounted Cumulative Gain for the top 10 most relevant passages [7, NDCG@10], where the score of each relevant passage depends on its position in descending order:

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{5}$$

$$\mathrm{IDCG}_p = \sum_{i=1}^{|REL_p|} \frac{rel_i}{\log_2(i+1)} \tag{6}$$

$$\mathrm{NDCG}_p = \frac{\mathrm{DCG}_p}{\mathrm{IDCG}_p} \tag{7}$$

where rel_i is the relevance of the i-th passage and REL_p is the list of relevant passages ordered by their relevance.

E. Results

Seven teams submitted a final solution to the task (see Table IX). All systems followed a similar architecture. First, the retriever was used to find the top N most relevant passages, and then the ranker scored these passages in order of importance to select the final 10 most relevant passages. Below are brief descriptions of the submitted systems, starting with the highest scoring ones.

Jakub Pokrywka implemented a retriever using the BM25 algorithm with text stemming using Polimorf.[12] To improve

[10] https://github.com/attardi/wikiextractor

[11] https://allegro.pl/

[12] https://github.com/dzieciou/pystempel

TABLE IX
COMPARISON OF PASSAGE RETRIEVAL TASK SUBMISSIONS

Submission	Retriever	Ranker	External Datasets	Model per domain	NDCG@10
Jakub Pokrywka	BM25	mt5-3B, mt5-13B, custom	No	Yes	69.36
Marek Kozlowski	Hybrid	mt5-13B	Yes	No	68.19
Konrad Wojtasik	Hybrid	mt5-13B, custom	Yes	No	67.44
Norbert Ropiak	Hybrid	MiniLM-L12, mDeBERTa	No	Yes	63.27
Anna Pacanowska	BM25	MiniLM-L6, custom	No	No	54.23
Maciej Kazuła	BM25	MiniLM-L6	Yes	No	51.78
Daniel Karaś	Hybrid	mBERT	No	No	51.71

the ranking of answers, separate rankers were used for each domain. For the Allegro and legal domains, an ensemble of mt5-3B[13] and mt5-13B[14] models was used, considering a pool of 1,500 candidates. Conversely, for trivia domain, Jakub Pokrywka also used the mt5-3B model but it was supplemented by a custom-trained cross-encoder models, mDeBERTa[15], and mmarco-mMiniLMv2-L12-H384-v1[16]. For trivia domain, the system included 3,000 candidate passages for effective ranking.

Marek Kozlowski used a system consisting of three retrievers: a lexical retriever (BM25) and two neural retrievers based on roberta-base[17] and roberta-large[18] [3]. The BM25 retriever used the ElasticSearch engine with the Morfologik analyser[19] for lemmatisation. For the neural encoders, fine-tuning the Roberta models involved using the MultipleNegativeRankingLoss loss function, large batch sizes and training data consisting of a mixture of Poleval training and translated MSMARCO [13] data sets. After retrieval, a re-ranking step was performed, with the mt5-13B model yielding the best results.

Konrad Wojtasik used an ensemble of several retrieval algorithms, starting with the BM25 algorithm, followed by various multilingual retrievers such as mContriever [6], mDPR [1] and LaBSE [4]. To further reduce the number of passages for reranking, he trained the plT5-large model [2] on the translated MSMARCO dataset. The final ranking was performed with mT5-13B on about 350 candidate passages from different sources.

Norbert Ropiak used both lexical (BM25) and neural retrievers (mContriever) and combined the results of both for further processing. He used ms-marco-MiniLM-L-12-v2[20] and mDeBERTa cross-encoders for ranking.

Anna Pacanowska's solution was a combination of several models. First, BM25 was used on lemmatised text to retrieve 1,000 candidate passages. Various statistics were calculated on these candidates, such as BM25 on unlemmatised data or on bigrams. The retrieved passages were then translated into English using OPUS-MT[21], which allowed the English MiniLM-L6 cross-encoder[22] to be used to calculate various scores, including those on raw question/passage pairs and on pairs with answers generated using GPT-3. Finally, logistic regression was used to combine all the results into a final score.

Maciej Kazuła used the BM25 passage retrieval algorithm together with the word inflection dictionary[23] to normalise the text. He fine-tuned the MiniLM-L6 cross-encoder for the ranking process. The cross-encoder was trained on the translated MSMARCO Polish dataset. A new tokeniser was created on the Poleval dataset, as well as on the translated MSMARCO data, in order to better represent Polish words in terms of word forms.

Daniel Karaś used two retrievers, a lexical search using BM25 and neural search using a slightly fine-tuned MiniLM-v6[24] model. Both retrievers were used to find approximately 1,000 candidates per question, except for Allegro where all passages were selected. In a second step, all candidate passages were fed into the mBERT[25], which was used without any additional training.

F. Summary

All submitted systems used the BM25 algorithm as a retriever, but differed in the way they normalised the text. Many lemmatised the passages, while others favoured stemming or using a dictionary of different word forms. In addition, some teams also used the neural retrievers and combined the candidates from these two approaches.

Given a pool of retrieved candidate passages, the systems used different methods to sort them and select the most relevant ones. The most popular were the cross-encoders, either trained on the multilingual data or fine-tuned by the contestants on the Polish examples. Most teams ensembled several models to achieve better performance.

[13] https://hf.co/unicamp-dl/mt5-3B-mmarco-en-pt

[14] https://hf.co/unicamp-dl/mt5-13b-mmarco-100k

[15] https://hf.co/cross-encoder/mmarco-mdeberta-v3-base-5negs-v1

[16] https://hf.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1

[17] https://huggingface.co/sdadas/polish-roberta-base-v2

[18] https://huggingface.co/sdadas/polish-roberta-large-v2

[19] https://github.com/allegro/elasticsearch-analysis-morfologik

[20] https://hf.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[21] https://huggingface.co/Helsinki-NLP/opus-mt-pl-en

[22] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

[23] https://sjp.pl

[24] sentence-transformer/multi-qa-MiniLM-L6-cos-v1

[25] amberoad/bert-multilingual-passage-reranking-msmarcoreranker

Three teams used external datasets to train their models. In all cases, they automatically translated the MSMARCO dataset into Polish.

Although the goal of the task was to create a system for cross-domain passage retrieval, it was allowed to submit different systems for different domains. Three participants chose this approach, including the winning system.

Regarding the results, it is observed that the performance of the systems was very much dependent on the ranker. The first three systems that achieved the results in the range of 67-69 NDCG points used a very large mt5-13B model as the reranker. The fourth model which achieved 63 points, used MiniLM-L12 and mDeBERTa. The last three models scoring 51-54 points used only MiniLM-L6 or multilingual BERT (with the exception of Anna Pacanowska's system, which also utilized a custom model). It seems that the retriever did not play an important role in the task, since the best system used only BM25 model. It is also interesting to observe that none of the systems used a learning-to-rank approach. One of the deficiencies of the evaluation is the lack of consideration for the computational heaviness of the approaches, which might be considered in the future incarnations of this task.

V. Conclusions and Future Plans

As each year we observe a growing interest in the PolEval challenge (the number of submissions and participating teams is growing), we plan to continue our efforts to identify new tasks, which are current and interesting in the research area of NLP and Polish language. The next editions will be specifically interesting, considering the current developments in the area of generative AI and language models.

We also plan to organize the datasets created for all the editions of the challenge in a repository to facilitate their distribution and encourage other researchers to use them for their work.

Acknowledgments

This work was supported by the European Regional Development Fund as a part of 2014–2020 Smart Growth Operational Programme, CLARIN — Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19, investment in the CLARIN-PL research infrastructure funded by the Polish Ministry of Science and Higher Education. We gratefully acknowledge Poland's high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016304.

References

[1] Asai, A., Yu, X., Kasai, J., Hajishirzi, H.: One question answering model for many languages with cross-lingual dense passage retrieval. In: NeurIPS (2021)

[2] Chrabrowa, A., Dragan, Ł., Grzegorczyk, K., Kajtoch, D., Koszowski, M., Mroczkowski, R., Rybak, P.: Evaluation of transfer learning for Polish with a text-to-text model. In: Proceedings of the Thirteenth Language Resources and Evaluation Conference. pp. 4374–4394. European Language Resources Association, Marseille, France (Jun 2022), https://aclanthology.org/2022.lrec-1.466

[3] Dadas, S., Perełkiewicz, M., Poświata, R.: Pre-training polish transformer-based language models at scale. In: Artificial Intelligence and Soft Computing. pp. 301–314. Springer International Publishing (2020)

[4] Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 878–891. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.62, https://aclanthology.org/2022.acl-long.62

[5] Hlubík, P., Španěl, M., Boháč, M., Weingartová, L.: Inserting Punctuation to ASR Output in a Real-Time Production Environment. In: Sojka, P., Kopeček, I., Pala, K., Horák, A. (eds.) Text, Speech, and Dialogue. pp. 418–425. Springer International Publishing, Cham (2020)

[6] Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., Grave, E.: Unsupervised dense information retrieval with contrastive learning (2021). https://doi.org/10.48550/ARXIV.2112.09118, https://arxiv.org/abs/2112.09118

[7] Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**, 422–446 (10 2002). https://doi.org/10.1145/582415.582418

[8] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.t.: Dense passage retrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 6769–6781. Association for Computational Linguistics, Online (Nov 2020). https://doi.org/10.18653/v1/2020.emnlp-main.550, https://aclanthology.org/2020.emnlp-main.550

[9] Kieraś, W., Woliński, M.: Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego. Język Polski **XCVII**(1), 75–83 (2017)

[10] Mikołajczyk, A., Wawrzyński, A., Pęzik, P., Adamczyk, M., Kaczmarek, A., Janowski, W.: PolEval 2021 Task 1: Punctuation Restoration from Read Text. In: Ogrodniczuk, M., Kobyliński, Ł. (eds.) Proceedings of the PolEval 2021 Workshop. pp. 21–31. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2021)

[11] Mroczkowski, R., Rybak, P., Wróblewska, A., Gawlik, I.: HerBERT: Efficiently pretrained transformer-based language model for Polish. In: Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing. pp. 1–10. Association for Computational Linguistics, Kiyv, Ukraine (Apr 2021), https://www.aclweb.org/anthology/2021.bsnlp-1.1

[12] Nguyen, T.B., Nguyen, Q.M., Nguyen, T.T.H., Do, Q.T., Luong, C.M.: Improving Vietnamese Named Entity Recognition from Speech Using Word Capitalization and Punctuation Recovery Models. In: Proceedings of Interspeech 2020. pp. 4263–4267 (2020). https://doi.org/10.21437/Interspeech.2020-1896, http://dx.doi.org/10.21437/Interspeech.2020-1896

[13] Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: MS MARCO: A Human Generated MAchine Reading COmprehension Dataset (November 2016), https://www.microsoft.com/en-us/research/publication/ms-marco-human-generated-machine-reading-comprehension-dataset/

[14] Ogrodniczuk, M., Kobyliński, Ł. (eds.): Proceedings of the PolEval 2021 Workshop. Institute of Computer Science, Polish Academy of Sciences, Warsaw (2021)

[15] Pappagari, R., Żelasko, P., Mikołajczyk, A., Pęzik, P., Dehak, N.: Joint Prediction of Truecasing and Punctuation for Conversational Speech in Low-Resource Scenarios. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). pp. 1185–1191 (2021). https://doi.org/10.1109/ASRU51503.2021.9687976

[16] Pęzik, P.: Spokes – a Search and Exploration Service for Conversational Corpus Data. In: Linköping Electronic Conference Proceedings. Selected Papers from CLARIN 2014. pp. 99–109. Linköping University Electronic Press (2015)

[17] Pęzik, P., Krawentek, G., Karasińska, S., Wilk, P., Rybińska, P., Cichosz, A., Peljak-Łapińska, A., Deckert, M., Adamczyk, M.: DiaBiz – an Annotated Corpus of Polish Call Center Dialogs. In: Proceedings of the Language Resources and Evaluation Conference. pp. 723–726. European Language Resources Association, Marseille, France (Jun 2022), https://aclanthology.org/2022.lrec-1.76

[18] Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Found. Trends Inf. Retr. **3**, 333–389 (2009)

[19] Rybak, P., Przybyła, P., Ogrodniczuk, M.: Improving question answering performance through manual annotation: Costs, benefits and strate-

gies (2022). https://doi.org/10.48550/ARXIV.2212.08897, https://arxiv.org/abs/2212.08897

[20] S., K., Cichosz, A., M., A., P., P.: Evaluating Punctuation Prediction in Conversational Language (2023), Forthcoming

[21] Sirts, K., Peekman, K.: Evaluating Sentence Segmentation and Word Tokenization Systems on Estonian Web Texts. In: Utka, A., Vaičenonienė, J., Kovalevskaitė, J., Kalinauskaitė, D. (eds.) Human Language Technologies – The Baltic Perspective. Frontiers in Artificial Intelligence and Applications, vol. 328, pp. 174–181 (2020). https://doi.org/10.3233/FAIA200620

[22] Sunkara, M., Ronanki, S., Bekal, D., Bodapati, S., Kirchhoff, K.: Multimodal Semi-supervised Learning Framework for Punctuation Prediction in Conversational Speech. In: Proceedings of Interspeech 2020. pp. 4911–4915 (2020). https://doi.org/10.21437/Interspeech.2020-3074, http://dx.doi.org/10.21437/Interspeech.2020-3074

[23] Wang, X.: Analysis of Sentence Boundary of the Host's Spoken Language Based on Semantic Orientation Pointwise Mutual Information Algorithm. In: Proceedings of the 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). pp. 501–506 (2020). https://doi.org/10.1109/ICMTMA50254.2020.00114

[24] Yi, J., Tao, J., Bai, Y., Tian, Z., Fan, C.: Adversarial Transfer Learning for Punctuation Restoration (2020)

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1251–1254

DOI: 10.15439/2023F1633
ISSN 2300-5963 ACSIS, Vol. 35

Punctuation Prediction for Polish Texts using Transformers

Jakub Pokrywka
Adam Mickiewicz University
Faculty of Mathematics and Computer Science,
Email: jakub.pokrywka@amu.edu.pl

***Abstract*—Speech recognition systems typically output text lacking punctuation. However, punctuation is crucial for written text comprehension. To tackle this problem, Punctuation Prediction models are developed. This paper describes a solution for Poleval 2022 Task 1: Punctuation Prediction for Polish Texts, which scores 71.44 Weighted F1. The method utilizes a single HerBERT model finetuned to the competition data and an external dataset.**

I. Introduction

AUTOMATIC Speech Recognition (ASR) systems produce speech transcripts, which typically do not contain punctuation. This may negatively impact the overall clarity of the transcribed text. For several reasons, punctuation is important:

- Punctuation reduces ambiguity in communication. The Sentences "Let's eat, children" and "Let's eat children" have completely different meanings, but they only vary in a comma.
- Punctuation helps in clarifying the intended meaning of a text. It provides cues to understand the structure of the text. Punctuation marks like commas, periods, question marks, and exclamation marks indicate pauses, sentence endings, and changes in tone or intent.
- Punctuation conveys tone and emotion behind the text. E.g., an exclamation mark may indicate excitement and a question mark may denote uncertainty.
- Punctuation enhances the readability of the written words. Breaking down complex sentences into smaller parts with the use of commas, colons, and semicolons creates pauses, which aids in understanding the text

Many post-processing steps may be taken to circumvent this problem and the lack of capitalization problem. Such tasks are:

- Punctuation Restoration (PR)
- Punctuation Prediction (PP)
- Capitalization Restoration (CR)

The task of Punctuation Restoration is defined as the act of reinstating the original punctuation found in read speech transcripts.

This work describes the solution to Poleval 2022 Task 1: Punctuation Prediction from conversational language. The solution is based on the HerBERT model [1] fine-tuned to the competition data and an external dataset.

II. Related Work

In the previous PolEval edition, a task similar to Punctuation Prediction was assigned, precisely PolEval 2021 Task: Punctuation restoration from read text [2]. The challenge unveiled WikiPunct, a fresh collection of text and audio corpus comprising 39 hours of audio and approximately 38,000 text transcripts. Four submissions [3], [4], [5], [6] applied transformer-based methods for token classification, from which two authors utilized ensembles. Additionally, one author explored the integration of a bi-LSTM layer at the top of the transformer, along with vectors acquired from a wave2vec model.

When it comes down to other languages, authors of [7] developed a method on Support Vector Machines with Conditional Random Field (CRF) classifiers, using part-of-speech (POS) and morphological data for Arabic texts. Authors of [8] used Deep Neural Networks and Convolutional Neural Networks for English texts and authors of [9] used transformers for English medical texts.

Recently, The Sentence End and Punctuation Prediction for many languages shared task was launched [10]. All of the teams explored neural network models, particularly transformers. The winning team described their solution in [11].

III. Competition Description

The three datasets are provided for in the competition: train, dev, and test. For each dataset, input audio WAV files with text transcribed by an ASR system are delivered. The input text is segmented, where a single space separates each word. Each word is prepended by a word start timestamp and word end timestamp in milliseconds.

The missing punctuation symbols are as in table I.

TABLE I
Punctuation symbols in the challenge.

symbol description	symbol character
Fullstop	.
Comma	,
Question Mark	?
Exclamation Mark	!
Hyphen	-
Ellipsis	. . .

The competition dataset is based on three resources summarized in Table II.

IEEE Catalog Number: CFP2385N-ART ©2023, PTI

Thematic track: Challenges for Natural Language Processing

TABLE II
THE FULL COMPETITION DATASET (TRAIN, DEV, TEST) STATISTICS.

Subset	Corpus	Files	Words	Audio [s]	Speakers	License
CBIZ [12]	DiaBiz	69	36 250	16 916	14	CC-BY-SA-NC-ND
VC	Video conversations	8	44 656	17 123	20	CC-BY-NC
Spokes [13]	Casual conversations	13	42 730	20 583	19	CC-BY-NC

The dataset is split into three subsets as described in Table III.

TABLE III
COMPETITION DATASET STATISTICS SPLIT INTO TRAIN, DEV, TEST.

Dataset	Files	Words	Audio [s]	License
Train	69	98 095	44 030	CC-BY-SA-NC-ND
Dev	11	12 563	4 718	CC-BY-NC
Test	10	12 978	5 874	CC-BY-NC

The annotation scheme is not publicly available during the competition and will described in [14].

There is one sample data from the training dataset in the subsection below.

A. Sample data

Input wav file : audio/AU1_P1_w_drodze_do_sklepu.wav

Input text : I:5880-5880 teraz:5940-6180 mamy:6330-6450 drugi:6480-6900 dzień:6960-7080 takiej:7170-7410 ładnej:7440-7650 pogody:7830-8400 Ała:8430-8430 Nie:8760-8820 bij:8850-8970 mnie:9120-9330 kijem:9450-9870 To:10020-10080 boli:10170-10260

Golden truth : I teraz mamy drugi dzień takiej ładnej pogody... Ała! Nie bij mnie kijem! To boli!

B. Utilized Data

In our final solution, we did not use any audio data. Additionally, we decided not to include start and stop timestamps as we did not observe any significant improvement in their score after conducting multiple experiments. Throughout the training process, we experimented with four different sources.

- Poleval 2022 Task 1: Punctuation Prediction from Conversational Language (this competition training dataset)
- Poleval 2021 Task 1: Punctuation Restoration from Read Text [2] (training dataset)
- Poleval 2021 Task 1: Punctuation Restoration from Read Text (test dataset)
- europarl-v7.pl-en.pl [15]

Regrettably, the europarl-v7.pl-en.pl dataset did not lead to a score improvement. Therefore, it was not utilized in our final solution.

We have carried out normalization procedures. Firstly, we transformed the text format from being split with timestamps to raw text format with timestamps included. Secondly, we replaced all three consecutive full stop characters "." (Unicode code: 81) with a single ellipsis character "…" (Unicode code: 8230). This modification was essential for utilizing the punctuation prediction library explained in Section IV.

Table IV presents the statistics for the training datasets used and competition final test data: test-B. Some punctuation marks are more popular than others, which is consequent in all the datasets. There are some differences between training and testing datasets, but they are insignificant. E.g., the Fullstop character is more common in the test-B dataset than in the train dataset (104.022 vs. 78.338). The same stays true for Comma (133.303 vs. 112.923). The PolEval 2022 dataset exhibits much more significant differences than the PolEval 2021 dataset. This is particularly evident in the Mean Words per Sample metric, as well as in most punctuation characters. While some characters like Fullstop, Comma, and Ellipsis are more prevalent in the PolEval 2022 dataset, Hyphen is less frequent, and the Exclamation mark remains relatively unchanged.

Below are samples of golden truths from each dataset, with the last two examples shortened.

1) Sample Poleval 2022 Task 1 test-B sentence: No dzień dobry pani. Tu mi się jakaś opłata za kartę pobrała.

2) Sample Poleval 2022 Task 1 train sentence: I teraz mamy drugi dzień takiej ładnej pogody… Ała! Nie bij mnie kijem! To boli!

3) Sample Poleval 2021 Task1 train sentence: w wywiadzie dla "polski" jarosław kaczyński podkreślił, że informacje dotyczące radosława sikorskiego zagrażają interesowi państwa. "to naprawdę wszystko, co mogę na ten temat powiedzieć "-odpowiedział, gdy dziennikarz pytał o bardziej szczegółowe informacje. premier kaczyński sugeruje, że dobry kandydat po na szefa dyplomacji to np. jacek saryusz- wolski wymieniony polityk zyskał uznanie braci kaczyńskich za dotychczasową działalność w charakterze dyplomaty i dużą wiedzę."

4) Sample Poleval 2021 Task1 test sentence: 801 co znaczy, że beginki "padły ofiarą reformacji"? grzesie2k wpis na słabym poziomie bzdurna informacja o 50 spalonych waldensach; po co w bibliografii pseudonaukowa książka magdaleny ogórek? fragment recenzji z księgarni gandalf: "magdalena ogórek do inkwizycji oraz kościoła ma stosunek jednoznaczny, pisząc o inkwizycyjnej pożodze oraz występkach heretyków spreparowanych przez inkwizytorów, którzy siali spustoszenie oraz o tym jak to w połowie xiii w? duchowni skupiali się na obsadzaniu stanowisk kościelnych, budowaniu zamętu przez interdykty, schizmy i walki, lekceważyli obowiązki duszpasterskie. nie ukrywa też, że jej celem jest próba rehabilitacji heretyków. takie jednoznacznie ideologiczne ustawienie problematyki nie ma wiele wspólnego z prawdą o epoce, obiektywizmem historycznym.

TABLE IV
Datatasets Statistics. The number of punctuation symbols is normalized per 1000 words.

Dataset	Samples	Mean Words per Sample	Fullstop	Comma	Question Mark	Exclamation Mark	Hyphen	Ellipsis
Poleval 2022 Task1 test-B	1642	7.90	104.022	133.303	18.493	0.848	0.154	33.981
Poleval 2022 Task1 train	10601	8.87	78.338	112.923	16.718	2.574	1.67	47.039
Poleval 2021 Task1 train	800	206.39	63.405	61.364	4.827	0.715	14.826	0.018
Poleval 2021 Task1 test	200	204.21	62.999	61.163	3.648	0.563	15.205	0.0
europarl-v7.pl-en.pl	632565	20.26	50.086	76.627	1.383	3.354	7.32	0.097

TABLE V
final testing dataset test-B scores.

model	Weighted-F1	Fullstop-F1	Comma-F1	Question Mark-F1	Exclamation Mark-F1	Hyphen-F1	Ellipsis-F1
allegro-herbert-large-cased-pl	71.44	78.67	72.25	74.96	16.67	100.00	43.72
polish-roberta-pl	66.23	74.56	68.31	72.77	28.57	100.00	29.86

TABLE VI
preliminary testing dataset test-A scores.

model	Weighted-F1	Fullstop-F1	Comma-F1	Question Mark-F1	Exclamation Mark-F1	Hyphen-F1	Ellipsis-F1
allegro-herbert-large-cased-pl	67.30	77.32	70.31	76.23	6.2	100.00	38.20
polish-roberta-pl	62.17	71.6	66.88	69.15	22.86	100.00	28.92

C. Metric

The challenge metric is the Weighted F1 score. The evaluation script is implemented in the GEval evaluation tool [16]. The challenge was hosted on the gonito platform [17]. The final evaluation is done on the test-B dataset on all the domains. The metric definition is meticulously described in Poleval 2021 Task1 summary paper [2].

IV. Method

Our method was based on FullStop: Multilingual Deep Models for Punctuation Prediction [11] library. We slightly modified the library to work on a different set of punctuation marks than it was intended to. The final solution model was based on a single HerBERT [1], a neural model of transformer architecture [18] trained on a corpus of Polish texts. The model was finetuned to the data described in Section III-B with the aforementioned text preprocessing steps. We used scripts available at https://github.com/oliverguhr/fullstop-deep-punctuation-prediction/blob/main/other_languages/readme.md. The Polish RoBERTa [19] model was evaluated as well, but not used for the final solution due to worse results. Both evaluations are available in Tables V and VI. We also conducted experiments with XLM-RoBERTa [20], but unfortunately, we did not achieve better results again.

V. Results

The final model using achieved a third-place score of 71.44 in the competition's Weighted F1 category. While it falls behind the first-place score of 83.30 and the second-place score, it still surpasses the baseline score of 35.30. Frequent punctuation symbols like full stops and commas (occurring above ten times per 1000 words) consistently scored between 70 and 80 in F1. However, the F1 scores varied greatly for less frequent symbols, with scores of 16.67, 100.00, and 43.72.

The subsections below illustrate some correct and incorrect predictions from the test-B dataset.

A. Correct predictions

Predicted: Nie rozumiem powodu, dla którego komuś za ciężko jest rozbić jajko.

Predicted: A ty dasz radę zabrać to wszystko?

B. Incorrect predictions

Expected:Ona nie będzie już**,**

Predicted:Ona nie będzie już**...**

Expected:Stary d**-** delegacyjny sprzęt z czasów PRLu, ale może być przydatny.

Predicted:Stary d**,** delegacyjny sprzęt z czasów PRLu, ale może być przydatny.

Expected:Zamknęli nam łazienkę**...** dranie**...**

Predicted:Zamknęli nam łazienkę**,** dranie

VI. Conclusions

In this paper, we proposed our solution to Poleval 2022 Task 1: Punctuation Prediction for Polish Texts. The method uses a single HerBERT model fine-tuned to the competition training data and other external datasets. The achieved score is 71.44, which falls behind the two best solutions but is significantly better than a baseline.

References

[1] R. Mroczkowski, P. Rybak, A. Wróblewska, and I. Gawlik, "HerBERT: Efficiently pretrained transformer-based language model for Polish," in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, (Kiyv, Ukraine), pp. 1–10, Association for Computational Linguistics, Apr. 2021.

[2] A. Mikołajczyk, A. Wawrzynski, P. Pezik, M. Adamczyk, A. Kaczmarek, and W. Janowski, "Poleval 2021 task 1: Punctuation restoration from read text," *Proceedings ofthePolEval2021Workshop*, p. 21.

[3] K. Wróbel, "Punctuation restoration with transformers," *Proceedings ofthePolEval2021Workshop*, pp. 33–37.

[4] N. Ropiak, M. Pogoda, J. Radom, K. Gawron, M. Swędrowski, and B. Bojanowski, "Comparison of translation and classification approaches for punctuation recovery," *Proceedings ofthePolEval2021Workshop*, pp. 39–46.

[5] M. Marcińczuk, "Punctuation restoration with ensemble of neural network classifier and pre-trained transformers," *Proceedings ofthePolEval2021Workshop*, pp. 47–53.

[6] T. Ziętkiewicz, "Punctuation restoration from read text with transformer-based tagger," *Proceedings ofthePolEval2021Workshop*, pp. 55–60.

[7] M. Attia, M. Al-Badrashiny, and M. Diab, "Gwu-hasp: Hybrid arabic spelling and punctuation corrector," in *Proceedings of the EMNLP 2014 workshop on Arabic natural language processing (ANLP)*, pp. 148–154, 2014.

[8] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 654–658, 2016.

[9] M. Sunkara, S. Ronanki, K. Dixit, S. Bodapati, and K. Kirchhoff, "Robust prediction of punctuation and truecasing for medical asr," *arXiv preprint arXiv:2007.02025*, 2020.

[10] D. Tuggener and A. Aghaebrahimian, "The sentence end and punctuation prediction in nlg text (sepp-nlg) shared task 2021," in *Swiss Text Analytics Conference–SwissText 2021, Online, 14-16 June 2021*, CEUR Workshop Proceedings, 2021.

[11] O. Guhr, A.-K. Schumann, F. Bahrmann, and H. J. Böhme, "Fullstop: Multilingual deep models for punctuation prediction," June 2021.

[12] P. Pęzik, G. Krawentek, S. Karasińska, P. Wilk, P. Rybińska, A. Cichosz, A. Peljak-Łapińska, M. Deckert, and M. Adamczyk, "DiaBiz," 2022. CLARIN-PL digital repository.

[13] P. Pęzik, "Spokes- a search and exploration service for conversational corpus data," pp. 99–109, Selected papers from the CLARIN 2014 Conference, 2014.

[14] S. Karasińska, S. Cichosz, and P. Pęzik, "Evaluating punctuation prediction in conversational language," *Forthcoming*.

[15] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of Machine Translation Summit X: Papers*, (Phuket, Thailand), pp. 79–86, Sept. 13-15 2005.

[16] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki, "GEval: Tool for debugging NLP datasets and models," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Florence, Italy), pp. 254–262, Association for Computational Linguistics, Aug. 2019.

[17] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń, "Gonito.net – open platform for research competition, cooperation and reproducibility," in *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language* (A. Branco, N. Calzolari, and K. Choukri, eds.), pp. 13–20, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[19] S. Dadas, M. Perełkiewicz, and R. Poświata, "Pre-training polish transformer-based language models at scale," in *Artificial Intelligence and Soft Computing*, pp. 301–314, Springer International Publishing, 2020.

[20] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *CoRR*, vol. abs/1911.02116, 2019.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1255–1264
DOI: 10.15439/2023F839
ISSN 2300-5963 ACSIS, Vol. 35

Abbreviation Disambiguation in Polish Press News Using Encoder-Decoder Models

Krzysztof Wróbel
0000-0002-3485-7825
Jagiellonian University, Enelpol
Krakow, Poland
Email: krzysztof@wrobel.pro

Jakub Karbowski
0009-0009-3051-7354
AGH University
of Science and Technology
Krakow, Poland
Email: carbon225@proton.me

Paweł Lewkowicz
0009-0000-5752-7610
AGH University
of Science and Technology,
Krakow, Poland
Email: pawlew@agh.edu.pl

Abstract—**The disambiguation of abbreviations and acronyms is a longstanding problem in Natural Language Processing (NLP) that has garnered significant attention from researchers. Previous approaches have employed statistical methods, semantic similarity metrics, and machine learning algorithms. Various languages and document types have been explored, with English being the most commonly studied language. Recent advances have been driven by the application of pre-trained transformer models. Standardization and addressing the challenges of multilingual and multi-document type disambiguation remain ongoing goals in the field of NLP. This paper presents an in-depth exploration of abbreviation disambiguation using state-of-the-art neural Encoder-Decoder models, specifically the ByT5 and plT5 architectures. Advanced synthetic data generation techniques are introduced and their effect on model performance is analysed. The methods are evaluated in the context of the PolEval abbreviation disambiguation competition, where the authors achieve top ranking.**

I. Introduction

THE problem of disambiguation of both acronyms and abbreviations has been the subject of interest for many researchers in the field of Natural Language Processing (NLP) for many years. Even before the era of widely used machine learning algorithms and text recognition using Deep Neural Networks (DNN), methods based solely on statistics were used. An example of such work is the paper [1] from 2004, which used a semantic similarity metric. The author determines the adequacy of abbreviation expansion candidates based on the similarity between the context of the target abbreviation and that of its expansion candidate.

The motivation for recognizing abbreviations often stemmed from the need to understand passages in documents such as provisions in law or medical notes. However, due to the richness, diversity, and uniqueness of languages, it is difficult to generalize the solution for expanding abbreviations or acronyms. Articles [2], [3] examine Jewish Law documents written in Hebrew, while [4], [5] present research on clinical papers using methods such as Support Vector Machines (SVM). Scientific papers are usually dedicated to only one language for which the datasets were prepared and the most widely used language in datasets is English. But another example of a different language is the research analysis [6] in Chinese, where the authors present their unconventional method based on Integer Linear Programming (ILP) and decode abbreviations from the generated candidates. Meanwhile, [7] analyze the Russian language by comparing methods such as SVM, Random Forest (RF), and Gradient Boosting (GB). Research has also been conducted in the Polish language, for example, in the paper [8], which utilized the bidirectional long short-term memory (LSTM) neural network architecture and compared two methods: automatically selecting all words in a text and using clustering of abbreviation occurrences.

Another aspect worth noting is the diversity of approaches to solving the disambiguation problem. As it turns out, supervised methods such as Convolutional Neural Networks (CNN), which are primarily used for image analysis, can also be used for this purpose in NLP, as evidenced by the article [9]. A diffrent example is the utilization and combination of pre-trained models such as RoBERTa and SciBERT, based on the transformer architecture, to create their own model named hdBERT, as presented in the research [10]. In this study, the authors also compared many state-of-the-art non-deep and deep learning methods up to 2017.

In 2020, the article [11] presented Google's T5 model as the Unified Text-to-Text Transformer. Since then, models with this architecture have found applications not only in text translation but also in expanding abbreviations, as shown in the publication [12]. In another article [13], expansions of acronyms were presented using pre-trained language models such as BERT and T5 for datasets consisting of four categories: Legal English, Scientific English, French, and Spanish.

The authors of the article [14] built their end-to-end acronym expander system named AcX and compared various existing methods such as Cosine Similarity (Cossim), RF, Logistic Regression (LR), and SVM. Based on this, they also prepared a benchmark on various types of datasets, including those from biomedical document, scientific papers and Wikipedia.

Based on the above considerations, research on abbreviation disambiguation over the years can be divided into three main categories:

- the type of documents from which the dataset was created, e.g. medicine, law, articles, or news
- the language in which the datasets were prepared

- the method used, which is almost always related to various machine learning algorithms

Former articles mainly focused on one type of document, one language, and one or two methods. However, today we increasingly encounter works related to multilingual models such as mT5 (multilingual T5), trained over 101 languages. However, these models undoubtedly require more memory (T5-base model - 220M parameters, mT5-base model - 580M parameters). The area of recognizing abbreviations is moving towards standardization and dealing with multiple languages and document types at once, but achieving satisfactory results in this area still poses a challenge for the field of NLP. Currently, promising models for this task appear to be encoder-decoder models like T5, pre-trained on a multi-task mixture of unsupervised and supervised tasks.

This article presents an attempt to standardize and formalize different aspects of abbreviation disambiguation, methods, challenges and limitations. State-of-the-art methods are evaluated on the PolEval 2022/23 competition, specifically in Task 2: Abbreviation disambiguation [1]. Additional dataset augmentation techniques are described, such as dictionary-lookup and algorithmic generation of arbitrary abbreviations. A unified training framework for abbreviation disambiguation is provided in a public online code repository. The additional created datasets are also provided. By combining the above methods, the authors achieve number one ranking on the PolEval contest.

II. Data

A. Training, validation and test datasets

The training, validation, and test datasets have been provided by the organizers of PolEval. As part of the PolEval competition, a training set called `train` and a validation set called `dev-0` with expected output were created, along with two test sets: `test-A` and `test-B` with implicit output. The collection and preparation of the datasets were carried out by:

- Michał Marcińczuk (Wrocław University of Science and Technology)
- Łukasz Kobyliński (Institute of Computer Science, Polish Academy of Sciences / Sages)

1) Assumptions: During the preparation, the authors of the reference corpus based their work on three assumptions:

- focus on abbreviations of common words or phrases ending with a dot (excluding initials, acronyms, and proper names)
- the context and common knowledge should be sufficient to expand the abbreviation (excluding incomplete or confusing examples)
- the base forms should follow the guidelines of phrase lemmatization from PolEval 2019 Task 2 [2] with some exceptions, such as abbreviations joined with other abbreviations or phrases

TABLE I
Examples of abbreviation disambiguation

Abbr	**Context**	**Inflected form**	**Base form**
t.	a na Dolnym Śląsku - 530-540 zł/**<mask>**	tonę	tona
gat.	czystą miedź (gat. M1) i mosiądz niklowy (**<mask>** M55N, CuZn35Ni6Mn4Si0,2)	gatunek	gatunek
j. ukr.	Kier. Natalia Szelest (nagroda burmistrza Węgorzewa) Wertep punktu naucz. **<mask>** w Baniach Mazurskich	języka ukraińskiego	język ukraiński
ład.	Żyjemy w okresie przejściowym, międzyepoce poprzedzającej nowy **<mask>**	ład.	ład.

Table I presents several examples of abbreviations disambiguation found in these datasets.

2) Input and expected output: In the system, the input data in the `in.tsv` file consists of two columns as we can see in Table II: the first column contains a phrase to be analyzed, and the second contains the context in which the phrase appears. If a masked token in an input is not an abbreviation then both columns in expected file are the same as masked token.

TABLE II
Input and expected output format

in.tsv		**expected.tsv**	
l.	ciągników serii 1523 i 1221 otrzymały sześciocylindrowe silniki o pojemności 7,2 **<mask>** Także te modele nie były wysilone mocowo i osiągały 158 KM w przypadku Belarusa	litra	litr

The occurrence of the phrase is marked by the keyword `<mask>`. The output file `expected.tsv` is also composed of two columns: the first column contains the inflected form, and the second contains the base form.

3) Data processing: The corpus was created based on the following four steps:

1. The datasets were built based on a collection of press news.
2. Regular expressions were used to collect potential abbreviations.
3. Each candidate was represented as a matched phrase and the context with several words before and after the match.
4. Candidates were selected and manually annotated.

4) Dataset cleanup: During the review, the authors removed any examples where the text was somehow corrupted, making it difficult to analyze.

1. Chi lij ski gi gant mie dzio wy (16,1 mld do **<mask>** war to ści) roz pro szył do brym zy skiem oba wy inwesto rów o je go wzrost mi mo

[1] http://poleval.pl/tasks/task2
[2] http://2019.poleval.pl/index.php/tasks/task2

2. l.,ulica,ulica,A) 9. U**l. Kościuszki** na odcinku od Lelewela do al. Krasińskiego 10. U****<mask>** Zwierzyniecka** od al. Krasińskiego do ul. Retoryka 11. Ul**. Bieżanowska** na

Above two examples of corrupted texts: the first with words divided into syllables, and the second with "**" characters within the abbreviation.

5) Challenges: The task poses several challenges that need to be addressed in order to solve it.

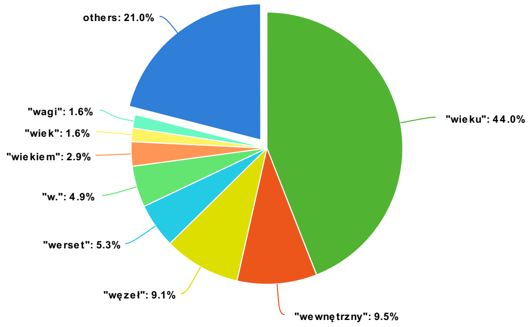
a) Challenge – ambiguous forms: When looking at one-word abbreviations, there is a large pool of words to which these abbreviations can be expanded.

TABLE III
AMBIGUOUS ONE-WORD ABBREVIATION FORMS EXPANDED TO THE LARGEST VARIETY OF WORDS (IN THE TRAIN SET)

Abbreviation	Base forms	Inflected forms
p.	38	57
w.	21	41
s.	20	36
m.	14	30

It can be observed in Table III that practically every word can be shortened to a single letter, and based on the context, one can infer the intended word. However, the challenge lies in correctly identifying this word.

Fig. 1. Distribution of abbreviation "w." extensions



The Figure 1 shows that there are multiple potential forms for each abbreviation, and the distribution is not such that one form constitutes 90% of expansions. What makes this task interesting is that there are many expansions for each abbreviation. In each case, there is a dominant word, for example, for the abbreviation `w.`, the most frequent expansion is `wieku` which accounts for 44.0% of expansions, while the remaining cases below 1.6% each sum up to 21.0%.

b) Challenge – unbounded space of abbreviations: The second element that affects the complexity of this task is the practically unbounded set of phrases that are subject to analysis. This task needs to be approached creatively because many cases not present in the training set may appear in the test set and should be handled correctly.

Based on Table IV, it can be assumed that about half of the phrases are non-abbreviated elements, while the other half are actual abbreviations that require expansion.

TABLE IV
THE NUMBER OF DISTINCT PHRASES AND ABBREVIATIONS TO BE EXPANDED

Set	Distinct phrases	Distinct abbreviations
train	∼ 1013	∼ 500
full	>1600	>900

c) Challenge – mixed abbreviations and non-abbreviations: Another challenge is distinguishing whether a given phrase is an abbreviation or not. There are also cases where a phrase can be both an abbreviation and a regular word that does not require expansion, which introduces an additional level of complexity to the task.

TABLE V
OCCURRENCES OF DIFFERENT ABBREVIATION FORMS

Set	Train set
Distinct phrases	∼ 1013
Non-abbreviations	∼ 480
Abbreviations	∼ 540
Both	26

Furthermore, certain phrases are a combination of abbreviations and non-abbreviations, marked as `others` in Table V, for example, replacing `mln. ton.` with `miliona ton.`

6) Special cases: There are several specific cases that have appeared to some extent in the corpus and go beyond the previous assumptions.

a) Special case – ambiguous forms: There are instances where certain abbreviations can be expanded into multiple words, as in the example presented in Table VI, where `m.` can be expanded into both `miejscowości` and `miasta`.

TABLE VI
AN EXAMPLE OF A CASE WITH A POSSIBLE DUAL OUTPUT

in.tsv	expected.tsv	
m. które przyszło na świat, gdy już był w więzieniu. Sam miał lat 40 i pochodził z **<mask>** Luboml pow. Kowel. Był urzędnikiem w składnicy Monop. Spirytusowego. (...) Nadmieniam,	miejscowości; miasta	miejscowość; miasto

Wherever this ambiguity can be resolved through context, such as certain signals indicating that it refers to one specific form, we expect only one form to appear in the output. However, in cases where there is no ambiguity, we assume that both forms should appear in the output, and they will be compared in this way.

b) Special case – non-abbreviation: In situations exposed in Table VII where a given phrase is not an abbreviation, we expect the output to repeat the phrase, which will be recognized as a non-abbreviation. Therefore, in this case the second part of answer is not a base form.

Both the base form and any inflected forms should be inflected in the same way as in the input, including the dot.

TABLE VII
AN EXAMPLE OF A CASE WITH A NON-ABBREVIATION

in.tsv		**expected.tsv**	
Goi.	Fascynowały ją obrazy **<mask>** Czytywała klasyków. Dowodzą tego jej, właśnie odnalezione, zapiski - pisze Bartosz	Goi.	Goi.

c) Special case – abbr and non-abbr: When there is a combination of an abbreviation and a non-abbreviated element, the abbreviated fragment should be expanded, while the non-abbreviated one should be preserved in the same form as in the input phrase.

TABLE VIII
AN EXAMPLE OF A CASE WITH A MIXED-ABBREVIATION

in.tsv		**expected.tsv**	
ws. T.	też: "Będziemy sprawdzać. czy nie fałszowano dowodów". Prokuratura bada śledztwo **<mask>** Komendy	w sprawie T.	w sprawie T.

In the example shown in Table VIII, these are initials, which we also do not want to expand.

d) Special case – lemmatization of joined abbreviations: In yet another case, there may be a combination of abbreviations that affects lemmatization. In the example below, two abbreviations appear consecutively, and each of these abbreviations should be expanded separately.

TABLE IX
AN EXAMPLE OF A CASE WITH A JOINED ABBREVIATIONS

in.tsv		**expected.tsv**	
mm. Temp. maks.	opady deszczu lub burze. Na zachodzie i południu prognozowana wysokość opadu do 25 **<mask>** od 19 do 23 st., nad morzem od 13 do 18 st. Wiatr północno-wschodni, slaby, na wybrzeżu	milimetrów Temperatura maksymalna	milimetrów temperatura maksymalna

In Table IX, the one-element abbreviation `mm.` is expanded to `millimeters` (it is an annotation error, it should be in singular: `millimeter`), and the two-element abbreviation `Temp. maks` is expanded to `Maximum temperature` which are then joined together.

B. Dictionary-based additional data

From dictionaries we extracted abbreviations with expanded forms, e.g. `bdb. -> bardzo dobry`. Then from Polish corpus CC100 [15] we extracted text fragments with inflected expanded forms of abbreviations and replaced them with abbreviations. An example is in Table X.

Only samples with unique inflected forms were taken into consideration giving 1982 new data points. The process creates also incorrect samples, e.g. `najlepszym` is abbreviated to `db..` This dataset lacks non-abbreviation examples.

TABLE X
DICTIONARY-BASED ADDITIONAL EXAMPLE

in.tsv		**expected.tsv**	
bdb.	pełno dyspozycyjny, zaangażowany, pracowity, komunikatywny, bardzo dobrze znam budowe komputera jak i **<mask>** obsługa komputera jak i programów biurowych Microsoft Office, Open Office itd. Prosze o kontakt	bardzo dobra	bardzo dobry

1) Morfeusz: Morfeusz [16] is a morphological analysis tool for the Polish language. With its help, all the abbreviations with their expanded form were filtered out from the dictionary. The pairs were selected on the basis of the morphosyntactic tags `brev:pun` or `brev:npun`. The `brev` feature indicates the base form of an abbreviation expansion, while `pun` and `npun` denote the presence or absence of a dot after the abbreviation.

TABLE XI
NUMBER OF ABBREVIATIONS FOUND IN THE MORFEUSZ DICTIONARY ACCORDING TO THE MORPHOSYNTACTIC TAGS

tag		
brev:pun	**brev:npun**	total
279	154	433

Table XI shows that there are more abbreviations without a dot at the end. Due to the problem posed in the task, `pun` abbreviations may be more useful since they can occur anywhere in a sentence, whereas `npun` abbreviations only appear at the end of a sentence, to be followed by a dot.

2) Wiktionary: Based on the free, multilingual dictionary Wiktionary [3], 554 different abbreviations were extracted with their meaning or meanings serving as an extension of the abbreviation. An example of multiple meanings in this dataset is, for example, `wyd.`, which can mean: `wydanie`, `wydawca`, `wydawnictwo`, `wydawniczy`.

TABLE XII
DISTRIBUTION OF THE NUMBER OF MEANINGS FOR ABBREVIATIONS IN THE DATASET WIKTIONARY

Number of meanings for the abbreviation	**Number of abbreviations with a given number of meanings**	
1	419	
2	83	
3	29	
4	13	
5	5	
6	1	
7	1	
8	1	
14	1	
19	1	
Number of meanings	**Number of abbreviations**	
803	554	**total**

[3] https://www.wiktionary.org

As we can see in Table XII, most of the found abbreviations have only one meaning, while in some cases, there are as many as 14 meanings for `n.` or 19 for `a.`.

In addition to the meanings, examples were also extracted from the Wiktionary, serving as context; however, this information was not utilized in solving the task.

3) SJP: More abbreviations than in previous dictionaries, a total of 1199, were found in the Polish language dictionary SJP. Just like in Wiktionary, some abbreviations had multiple meanings, for example, `woj.` could stand for `województwo`, `wojewoda`, `wojewódzki`, `wojenny`, `wojskowy`. The abbreviations were selected by reviewing all the words or phrases in the dictionary and then filtering out those that had periods at the end.

TABLE XIII
DISTRIBUTION OF THE NUMBER OF MEANINGS FOR ABBREVIATIONS IN THE DATASET SJP

Number of meanings for the abbreviation	Number of abbreviations with a given number of meanings	
1	1051	
2	97	
3	30	
4	10	
5	11	
Number of meanings	**Number of abbreviations**	
1430	1199	**total**

Table XIII shows that there are no abbreviations with as many meanings as sometimes found in Wiktionary, but the number 5 can be considered the maximum number of meanings that almost always appeared in SJP and Wiktionary.

C. Synthetic additional data

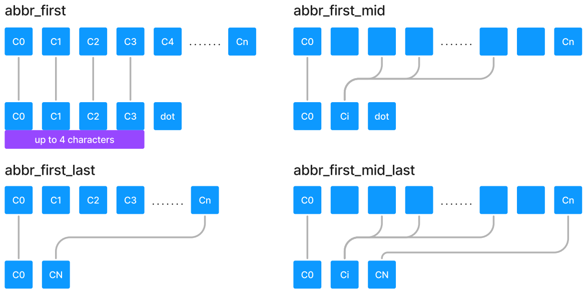
Collecting a sufficiently large and diverse dataset with accurately annotated abbreviations can be a challenging and time-consuming task. In this section, we describe the methodology used to generate synthetic data.

1) Data collection and preprocessing: The source corpus was the Polish Wikipedia. A sliding window is used to randomly select a context of 140 to 200 characters. This context length is representative of the PolEval abbreviation disambiguation dataset. Within each such context, a continuous span of words is randomly selected. These words are then processed with algorithmic abbreviation.

2) Algorithmic abbreviation: The custom abbreviation algorithm operates with four different strategies, as seen in Figure 2:

1. `abbr_first`: `profesor` → `prof.`
   Choose 1 to 4 of the first characters.
2. `abbr_first_last`: `profesor` → `pr`
   Choose the first and last characters.
3. `abbr_first_mid`: `profesor` → `pf.`
   Choose the first and one middle character.
4. `abbr_first_mid_last`: `profesor` → `pfr`
   Choose the first, one middle and last characters.

Fig. 2. Algorithmic Abbreviation Schemes



The algorithm applies a random strategy to each word in the span. It must be noted that the generated abbreviations are not guaranteed to be grammatically correct.

3) Base form prediction: The base forms of all words from the span before abbreviation are generated with the `spaCy` `pl_core_news_lg` model [17].

TABLE XIV
SYNTHETIC WIKIPEDIA EXAMPLE

in.tsv	**expected.tsv**	
bs. jest zgodny ze światem, w którym istnieje problem zła i cierpienie, a **<mask>** miłość jest ukryta przed wieloma osobami. Podobną argumentację	boska	boski

Each such context containing an abbreviated span is used as a dataset sample. Table XIV shows one generated example. The process repeats until reaching the end of the Wikipedia corpus.

4) Considerations: This synthetic dataset applies to a broader task of corrupted text restoration, where abbreviation disambiguation can be seen as a sub-task. In the context of abbreviation disambiguation, it is a low-quality dataset. This disadvantage is countered by its vast size of 14 million examples, which is 3375 times larger than the PolEval training dataset.

The exact version of the created dataset cannot be deterministically reproduced because of multi-process random number generation used during processing. A snapshot of the dataset is provided online[4].

III. EVALUATION

The process of abbreviation disambiguation i.e. replacing abbreviations with their appropriate expansions in text can be divided into two stages.

The first stage is to find the abbreviation in a dictionary and replace it with an appropriate word or phrase that is its base forms.

The second stage is to transform the result from the first stage into its correctly inflected grammatical form based on the context in which the abbreviation appears in the text.

[4] https://huggingface.co/datasets/carbon225/poleval-abbreviation-disambiguation-wiki

It is worth noting that sometimes there is more than one meaning for a given abbreviation, so in the first stage it is also important to take the context into account in order to better predict which meaning is intended in a specified example.

Two metrics were used to objectively evaluate the replacement of abbreviation with their appropriate expansion in text:

- Af - the accuracy of provided expanded forms of abbreviations
- Ab - the accuracy of provided base forms of abbreviations

The matching check for both metrics was case-insensitive.

Based on the above metrics, the ultimate formula was defined to determine the final score:

$$Acc = 0.25 \cdot Af + 0.75 \cdot Ab \qquad (1)$$

Therefore, the task of finding the appropriate base form is three times more important than the task of finding its appropriate expansion.

IV. Methods

The solutions are based on a sequence to sequence model using the T5 [18] architecture. Krzysztof Wróbel's submission used the ByT5 [19] model, while Jakub Karbowski used the plT5 [20] model.

Both submissions used a similar workflow. The input to the transformer encoder is the context with the abbreviation. The transformer decoder generates both the base and inflected forms. Multiple methods of encoding the input and output of the model were used. They are described in detail in their corresponding sections.

In order to improve the results, majority voting with multiple models has been applied. The final decision is determined by the majority vote, where each model's prediction contributes one vote, and the outcome with the most votes is selected as the final prediction. This approach leverages the collective knowledge and expertise of multiple models to improve the accuracy and robustness of predictions in scientific studies. For this task, majority voting has been applied separately for inflected and base form.

V. Experiments

A. PolEval submissions

Initial experiments were carried out with limited time because of the competition deadlines. They were the base for further post-competition research. First, the exact methods used to produce the competition submissions are described.

1) Krzysztof Wróbel submissions: Proper validation is very important in every competition. The original validation (dev-0) dataset has only 300 samples which is insufficient for tracking scores with a precision of 0.1 percentage points. Therefore, 1000 samples from the training data were moved to the validation set.

The input data was prepared as follows: an abbreviation in the sentence was surrounded by <abbrev> and </abbrev>, e.g. Komunistyczny deputowany, <abbrev>b.</abbrev> śledczy Prokuratury Generalnej. The output data is structured as follows:

- for abbreviations: inflected form, separator <sep>, and base form, e.g. były <sep> być
- for non-abbreviations: the form, e.g. b.

The tokens <abbrev>, </abbrev>, and <sep> were added to the model vocabulary.

Initial experiments using Adafactor as an optimizer showed that the plT5 models performed slightly worse than the ByT5 models.

The training dataset was augmented by extracting abbreviations from dictionaries and applying them into sentences sourced from a corpus.

The final submission was created using majority voting on 3 models:

- trained on the training data and dictionary-based additional data using the development data for selecting the best model
- trained on the training data, development data, and dictionary-based additional data with two different seeds

The training parameters were as follows:

- model: byt5-base
- max input length: 250
- max output length: 100
- batch size: 16
- gradient accumulation: 16
- epochs: 24
- learning rate: 0.001
- scheduler: linear with warmup 0.1
- optimizer: Adafactor

TABLE XV
Krzysztof Wróbel's submissions to PolEval. The name is the same as in official leaderboard.

Description	Name	test-A	test-B
train	3	90.78	
train + dict	5	91.32	
train + dev + dict, seed 1	8	92.18	91.69
train + dev + dict, seed 2	9	92.14	91.65
voting (final)	11	**92.76**	**92.01**

Table XV presents the results of Krzysztof Wróbel's submissions to the PolEval competition. The table includes different models and their corresponding scores on the test-A and test-B datasets.

The second model, named as 5 was trained on the training data along with additional dictionary-based data. This model performed better by 0.5 percentage points than model trained only on the training data.

The next two models, named as 8 and 9 were trained on the training data, development data, and dictionary-based additional data, using different random seeds for each. Model 8 achieved a score of 92.18 on the test-A dataset and 91.69 on the test-B dataset, while model 9 achieved a score of 92.14 on the test-A dataset and 91.65 on the test-B dataset.

The final model, named as `11` is the result of majority voting on three models: `5`, `8`, and `9`. This model achieved the highest scores among all the submissions, with a score of 92.76 on the test-A dataset and 92.01 on the `test-B` dataset.

2) Jakub Karbowski submissions: Input data was encoded in a similar way to Krzysztof Wróbel's submission. The only difference is that `<abbrev>` and `</abbrev>` are not added as special tokens and are tokenized as raw text by the model's tokenizer. Instead of `<abbrev>` they are called `<mask>`.

The output format was the same as in Krzysztof Wróbel's submission, except the output of the model does not differ between abbreviations and non-abbreviations. The output format is: inflected form; base form, e.g. `były; być`.

Although ByT5 was considered because of its high performance on noisy data, plT5 was chosen because of limited training hardware available to the author of the submission. Training was performed on single GTX 1080 GPU with 8 GB of VRAM within a single day. Training ByT5 on this hardware would not be feasible.

First, pre-training was carried out on the Wikipedia dataset with synthetic abbreviations.

Pre-training parameters:

- model: plt5-base
- batch size: 4
- gradient accumulation: 64
- training steps: 3300
- learning rate: 0.0000928
- scheduler: linear with warmup 2000 steps
- optimizer: AdamW
- weight decay: 0.001

The training lasted 6 hours and was terminated after just 6% of the dataset. The pre-trained score achieved on the PolEval `dev-0` dataset was 29.18%.

The pre-trained model was then fine-tuned on the PolEval `train` dataset.

Training parameters:

- model: plt5-base (wiki pre-trained)
- batch size: 8
- gradient accumulation: 32
- epochs: 223
- learning rate: 0.000015
- scheduler: linear with warmup 10%
- optimizer: AdamW
- weight decay: 0.0001

The per-device batch size could be increased because of a decrease in sequence length compared to the pre-training dataset. The score of the final submission with pre-training was 91.75% on `test-A` and 91.27% on `test-B`.

B. Post-competition experiments

After the announcement of the competition results, the top two contestants combined their work to evaluate the performance of their methods, with respect to:

- model architectures
- used datasets and their combinations
- a broad range of hyperparameters
- original optimizations and solutions

1) Setup: A unified codebase for training was created[5]. It combines all of the methods and datasets used:

- ByT5 and plT5 models
- PolEval, dictionary-based and synthetic Wikipedia datasets
- majority voting

It also contains the hyperparameters and sweep configurations used during experimentation.

2) Configurations: Eight different configurations were chosen for final assessment. All combinations of the following options were used:

- Base model:
  - ByT5
  - plT5
- Pre-training dataset:
  - None
  - Wikipedia
- Fine-tuning dataset:
  - PolEval `train`
  - PolEval `train` with additional dictionary-based data

3) Pre-training: As pre-training on the large synthetic Wikipedia dataset was computationally expensive, sweeps on this dataset were not conducted. Instead, results from fine-tuning runs and manual experimentation provided the hyperparameters for pre-training.

4) Fine-tuning: For each configuration, a hyperparameter sweep was conducted. The sweeps considered: learning rate, weight decay, epochs, optimizer (AdamW or Adafactor). To provide a fair comparison between the two model architectures, each sweep was given 24h of computational time on an A100 GPU. The four sweeps with ByT5 managed to perform 16 training runs each, while plT5 sweeps performed 80 runs each.

5) Voting: Experiments involving majority voting were conducted to evaluate the performance of the best plT5 and ByT5 models. For each model, a set of 10 models was trained using identical parameters but different random seeds.

VI. RESULTS

Table XVI shows scores for experiments using plT5 and ByT5 models trained on different datasets. The highest scores are obtained using pre-training on synthetic data and then fine-tuned on `train` data with `dictionary-based` data. The models are shared at Hugging Face[6]. ByT5 consistently achieves higher scores than plT5. The results on the `dev` dataset do not correlate with the test data due to the small size of the `dev` dataset.

Using the synthetic Wikipedia dataset for pre-training improves the performance of both models. For plT5, the `test-B`

[5] https://github.com/Carbon225/poleval-2022-abbr

[6] https://huggingface.co/carbon225/plt5-abbreviations-pl, https://huggingface.co/carbon225/byt5-abbreviations-pl

score improves by around 1% when considering both the pure PolEval `train` dataset and the additional dictionary data. For ByT5, the improvement is under 1%. Using additional dictionary data improves the scores by around 0.5%.

TABLE XVI
RESULTS FOR PLT5 AND BYT5 MODELS ON DIFFERENT TRAINING DATASETS

		plT5			ByT5	
	dev	test-A	test-B	dev	test-A	test-B
train	91.80	90.76	90.06	94.10	92.10	91.73
wiki-train	91.39	**91.76**	91.32	93.61	92.46	92.53
train-dict	91.31	91.21	90.44	94.34	92.30	92.20
wiki-train-dict	91.31	91.64	**91.37**	93.44	**92.71**	**92.92**

TABLE XVII
SCORES OBTAINED USING MAJORITY VOTING AMONG 1 TO N MODELS. MODELS ARE SORTED BY TEST-A SCORE AND USED IN THAT ORDER.

	plT5		ByT5	
Models	test-A	test-B	test-A	test-B
1	91.72	90.96	92.71	92.92
1-2	91.74	91.22	92.65	92.80
1-3	91.91	91.42	93.00	93.06
1-4	91.88	91.45	**93.33**	93.15
1-5	92.03	91.57	93.25	**93.27**
1-6	92.00	91.59	93.20	93.19
1-7	91.99	91.56	93.16	93.14
1-8	92.05	91.57	93.12	93.11
1-9	92.10	**91.62**	93.12	93.18
1-10	**92.12**	91.59	93.13	93.17

Table XVII provides the test-A and test-B scores for different combinations of plT5 and ByT5 models. The row labeled `1` represents the score obtained when only the first model is used. Subsequent rows, labeled `1-2`, `1-3`, and so on, indicate the scores obtained when additional models are included in the majority voting process. The maximum improvement observed through this process is approximately 0.5 percentage points.

TABLE XVIII
POLEVAL BEST RESULTS AND SCORES BY DIFFERENT SUBMISSIONS.

	test-A	test-B
Krzysztof Wróbel	**92.76**	**92.01**
Jakub Karbowski	91.75	91.27
Marek Kozlowski	89.00	88.73
Jakub Pokrywka	65.48	66.25
Rafał Prońko		19.09

Table XVIII presents the best results and scores achieved by different submissions in the PolEval competition. The table includes two test metrics: `test-A` and `test-B`.

Krzysztof Wróbel emerged as the highest scorer, surpassing Jakub Karbowski by 0.74 percentage points in the test-B metric. Krzysztof Wróbel's success can be attributed to the implementation of the ByT5 model, majority voting, and the utilization of a larger validation dataset. Incorporating the pretraining step and utilizing the AdamW optimizer, as introduced in Jakub Karbowski's solution, has the potential to yield scores higher by more than 1 percentage point.

A. Error analysis

The error analysis of 50 randomly selected errors made by the ByT5 model in the `wiki-train-dict` variant revealed that the model correctly predicted the answers for half of them. The dataset annotation needs to be improved.

More technical issues apply to about 1.5% of the examples. Approximately 0.76% of the examples in the dataset are annotated with multiple possible answers separated by a semicolon, such as `przeciw; przeciwko`. These cases were not properly taken into account during evaluation, About 0.72% of the examples in the dataset consist of multiword abbreviations with tokens separated by more than one space.

VII. CONCLUSIONS

In this paper, we addressed the problem of abbreviation disambiguation in Polish press news using encoder-decoder models. The task involved replacing abbreviations with their appropriate expansions in text, taking into account the context.

Our experiments included submissions to the PolEval competition and post-competition research. In the PolEval competition, we achieved first and second place rankings. In the post-competition experiments, we conducted evaluations using different configurations, including pre-training on synthetic Wikipedia data and fine-tuning on additional data, which achieved a new state-of-the-art on the PolEval competition.

In conclusion, our study contributes valuable insights into the abbreviation disambiguation task in Polish press news. We emphasize the importance of proper validation, the trade-off between optimizer choice and memory usage, importance of pre-training, and the effectiveness of majority voting as a simple technique for improving results. Further research can build upon these findings to explore more advanced architectures, optimizations, and techniques for even better performance in Polish abbreviation disambiguation tasks.

Our approach can be easily applied to other languages and various types of texts.

VIII. APPENDIX

Table XIX presents errors of the ByT5 model in the `wiki-train-dict` variant.

IX. ACKNOWLEDGMENT

The competition submissions made by the team of Krzysztof Wróbel and Paweł Lewkowicz were completely independent of Jakub Karbowski's submissions. However, after the competition, the authors decided to collaborate and write a joint article due to the similarities in the methods used.

We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2023/016304

The research has been supported by a grant from the Faculty of Management and Social Communication under the Strategic Programme Excellence Initiative at Jagiellonian University.

References

[1] A. Terada, T. Tokunaga, and H. Tanaka, "Automatic expansion of abbreviations by using context and character information," *Information Processing & Management*, vol. 40, no. 1, pp. 31–45, 2004. doi: https://doi.org/10.1016/S0306-4573(02)00080-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457302000808

[2] Y. HaCohen-Kerner, A. Kass, and A. Peretz, "Abbreviation disambiguation: Experiments with various variants of the one sense per discourse hypothesis," in *Natural Language and Information Systems, 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008, London, UK, June 24-27, 2008, Proceedings*, ser. Lecture Notes in Computer Science, E. Kapetanios, V. Sugumaran, and M. Spiliopoulou, Eds., vol. 5039. Springer, 2008. doi: 10.1007/978-3-540-69858-6_5 pp. 27–39. [Online]. Available: https://doi.org/10.1007/978-3-540-69858-6_5

[3] Y. HaCohen-Kerner, A. Kass, and A. Peretz, "Combined one sense disambiguation of abbreviations," in *Proceedings of ACL-08: HLT, Short Papers*, 2008, pp. 61–64.

[4] Y. Wu, J. Xu, Y. Zhang, and H. Xu, "Clinical abbreviation disambiguation using neural word embeddings," in *Proceedings of BioNLP 15*, 2015, pp. 171–176.

[5] A. M. M. Jaber and P. Martínez Fernández, "Disambiguating clinical abbreviations using pre-trained word embeddings," 2021.

[6] L. Zhang, L. Li, H. Wang, and X. Sun, "Predicting Chinese abbreviations with minimum semantic unit and global constraints," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014. doi: 10.3115/v1/D14-1147 pp. 1405–1414. [Online]. Available: https://aclanthology.org/D14-1147

[7] A. Berdichevskaia, "Atypical lexical abbreviations identification in russian medical texts," *2022 12th International Conference on Pattern Recognition Systems (ICPRS)*, pp. 1–5, 2022.

[8] A. Mykowiecka and M. Marciniak, "Experiments with ad hoc ambiguous abbreviation expansion," in *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*. Hong Kong: Association for Computational Linguistics, Nov. 2019. doi: 10.18653/v1/D19-6207 pp. 44–53. [Online]. Available: https://aclanthology.org/D19-6207

[9] R. Kai and W. Shi-Wen, "Applying convolutional neural network model and auto-expanded corpus to biomedical abbreviation disambiguation." *Journal of Engineering Science & Technology Review*, vol. 9, no. 6, 2016.

[10] Q. Zhong, G. Zeng, D. Zhu, Y. Zhang, W. Lin, B. Chen, and J. Tang, "Leveraging domain agnostic and specific knowledge for acronym disambiguation," *CoRR*, vol. abs/2107.00316, 2021. [Online]. Available: https://arxiv.org/abs/2107.00316

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[12] A. Rajkomar, E. Loreaux, Y. Liu, J. Kemp, B. Li, M.-J. Chen, Y. Zhang, A. Mohiuddin, and J. Gottweis, "Deciphering clinical abbreviations with a privacy protecting machine learning system," *Nature Communications*, vol. 13, no. 1, p. 7456, 2022.

[13] G. Song, H. Lee, and K. Shim, "T5 encoder based acronym disambiguation with weak supervision," *SDU@ AAAI-22*, 2022.

[14] J. L. Pereira, J. Casanova, H. Galhardas, and D. Shasha, "Acx: system, techniques, and experiments for acronym expansion," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2530–2544, 2022.

[15] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, "CCNet: Extracting high quality monolingual datasets from web crawl data," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020. ISBN 979-10-95546-34-4 pp. 4003–4012. [Online]. Available: https://aclanthology.org/2020.lrec-1.494

[16] W. Kieraś and M. Woliński, "Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego," *Język Polski*, vol. XCVII, no. 1, pp. 75–83, 2017.

[17] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020. doi: 10.5281/zenodo.1212303

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[19] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "Byt5: Towards a token-free future with pre-trained byte-to-byte models," *CoRR*, vol. abs/2105.13626, 2021. [Online]. Available: https://arxiv.org/abs/2105.13626

[20] A. Chrabrowa, Ł. Dragan, K. Grzegorczyk, D. Kajtoch, M. Koszowski, R. Mroczkowski, and P. Rybak, "Evaluation of transfer learning for Polish with a text-to-text model," in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, Jun. 2022, pp. 4374–4394. [Online]. Available: https://aclanthology.org/2022.lrec-1.466

TABLE XIX
EXAMPLES OF ERRORS BY THE BEST MODEL

input	expected		predicted	
m. możliwy grad. Przewidywana wysokość opadów w burzach od 10 mm do 15 mm, w górach do 20 **<mask>** Temperatura maksymalna od 15 st.C w rejonie Zatoki Gdańskiej, 21 st.C na Suwalszczyźnie	metrów	metr	milimetrów	milimetr
p. procentowych (do 100 proc.) przy dopłatach dla 1 osoby i o 10 punktów procentowych (do 40 **<mask>** proc.) dla każdej kolejnej osoby w gospodarstwie domowym najemcy. Ważne zmiany	pikseli	piksel	punktów	punkt
róż. Pazdanowi żona, Dominika, z wyraźnym onieśmieleniem przyjęła bukiet biało-czerwonych **<mask>** Piłkarz zrewanżował się własną reprezentacyjną koszulką. Potem nastąpiła seria	róż	róż	róż.	róż.
o. (są) do indywidualnego uzgodnienia z władzami' uczelni', czyli i tak daje ludziom **<mask>** Rydzyka zupełną dowolność. Wyższa Szkoła Kultury Społecznej i Medialnej w Toruniu	ojciec	ojciec	ojca	ojciec
cm. 38-letniego Granta. Amerykański pięściarz mierzy 201 cm, natomiast 33-letni Adamek - 187 **<mask>** Polak znacznie przegrywa z Grantem również pod względem zasięgu ramion. Dla Adamka,	centrymetrów	centrymetr	centymetrów	centymetr
p. C-331/94, Komisja p. Grecji, ECLI:EU:C:1996:211, pkt 10; C-111/05, Aktiebolaget NN **<mask>** Skatteverket, ECLI:EU:C:2007:195, pkt 55-58. [10] Z. Knypl, Polskie obszary morskie,	przeciwko	przeciwko	przeciw; przeciwko	przeciw; przeciwko
p. również osoby bez obywatelstwa, którzy publicznie znieważają osoby, wymienione w **<mask>** 1 Ustawy, przeszkadzają w realizacji praw osób walczących o niezależność Ukrainy	punkcie; paragrafie	punkt; paragraf	paragrafie	paragraf
r. zawodników urodzonych w 2001 roku i młodszych. Wcześniej we Włocławku walczyli piłkarze **<mask>** 1997. Tym razem nie będzie to turniej międzynarodowy, ponieważ cztery zaproszone	rocznik	rocznik	rocznika	rocznik
p. pogwałcenia praw osoby w świetle takich oświadczeń (uchwał polskich samorządów – **<mask>** A.J), ale na razie KE pilnie analizuje sytuację. Nie otrzymała, ale przecież mogła	przypis	przypis	pani	pani
d. nadzwyczajnej, którą odbyli w dniu wczorajszym, są deputowani Iwano-Frankowska (**<mask>** Stanisławów). „Krwawe stłumienie w sercu Europy pokojowych zgromadzeń przez uzbrojonych	dawny	dawny	dawniej	dawniej
m. Przemysłowym. Tramwaje linii 3, 23, 33>pl. Jana Pawła II skierowano objazdem przez **<mask>** Sikorskiego, Dubois, Nowy Świat. Tramwaje linii 10 i 20>pl. Jana Pawła II skierowano	most	most	mosty	most
p. konkursu. Oferty należy składać do 5 grudnia (nie decyduje data stempla pocztowego) w **<mask>** 223 w Starostwie Powiatowym. Obecnie placówkę prowadzi Zgromadzenie Sióstr św.	pokoju	pokój	pokój	pokój
s. William P. Young, Chata, tłum. A. Reszka, Wydawnictwo Nowa Proza, Warszawa 2009, **<mask>** 281. Ponad 6.000.000 sprzedanych egzemplarzy robi wrażenie na każdym, kto ma styczność	stron	strona	strona	strona
m. małopolskiego (i na 389. miejscu w Polsce) oraz II LO im. Tytusa Chałubińskiego na 70 **<mask>** (poza pierwszą 500 najlepszych liceów w Polsce). W ubiegłym roku „Kościuszko" był	miejscu	miejsce	metrach	metr
f. powierzchniową. Jak piszą Allaud L.A. i Martin M. bracia Schlumberger'owie przekonali **<mask>** Royal Dutch Shell, po powtórzeniu pomiarów i sprawdzeniu ich wiarygodności, że ta	firmę	firma	firmie	firma
m. dąbrowski) - rzeka Wisła o 69 cm m. Szczucin (pow. dąbrowski) - rzeka Szreniawa o 8 cm **<mask>** Biskupice (powiat miechowski) - rzeka Wisła o 145 cm Pustynia (powiat oświęcimski) -	miejscowość	miejscowość	miasto	miasto
ub.roku. krajowymi. Kupili ich w pierwszym kwartale o ponad 9 mld zł więcej niż na koniec **<mask>** – To pokazuje, że zagranica nie ucieka od naszego długu. Równoległy spadek nierezydentów	ubiegłego roku.	ubiegły roku.	ubiegłego roku	ubiegły rok
zew. wreszcie zbliża się ten dzień / wielki dreszcz emocji w nas / i wolności poczuj **<mask>** / Euro w barwach szczęścia jest / więc ramiona w górę wznieś / a dopóki piłka w grze	zew.	zew.	zewnętrznie	zewnętrznie
zm. pierwszym francuskim zwycięzcą Ligi Mistrzów. Zbigniew Pacelt (ur. 26 sierpnia 1951, **<mask>** 4 października 2021) - pływak i pięcioboista, dwukrotny olimpijczyk (1968 i 1972).	zmarły	zmarły	zmarł	zmarł
l. Parku Wodnego dla 3 l. koni półkrwi, Godz. 13.40 – gonitwa czwarta Puchar D&D dla 3 **<mask>** koni półkrwi, Godz. 14.25 – gonitwa piąta dla 3 l. ogierów i wałachów półkrwi, Godz.	letnich	rok	litrów	litr

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1265–1269

DOI: 10.15439/2023F9253
ISSN 2300-5963 ACSIS, Vol. 35

Passage Retrieval of Polish Texts Using OKAPI BM25 and an Ensemble of Cross Encoders

Jakub Pokrywka
Adam Mickiewicz University
Faculty of Mathematics and Computer Science,
Email: jakub.pokrywka@amu.edu.pl

***Abstract*—Passage Retrieval has traditionally relied on lexical methods like TF-IDF and BM25. Recently, some neural network models have surpassed these methods in performance. However, these models face challenges, such as the need for large annotated datasets and adapting to new domains. This paper presents a winning solution to the Poleval 2023 Task 3: Passage Retrieval challenge, which involves retrieving passages of Polish texts in three domains: trivia, legal, and customer support. However, only the trivia domain was used for training and development data. The method used the OKAPI BM25 algorithm to retrieve documents and an ensemble of publicly available multilingual Cross Encoders for Reranking. Fine-tuning the reranker models slightly improved performance but only in the training domain, while it worsened in other domains.**

I. Introduction

PASSAGE retrieval involves the task of retrieving a set of relevant text passages from a large collection of documents based on a given query. Typically, these passages are presented in descending order of relevance. The most commonly used method for passage retrieval is through lexical approaches like OKAPI BM25. Though, lexical models cannot capture semantic relationships between words, phrases, and sentences. To address this, neural language models can be employed. These models are often pretrained on extensive text corpora and then fine-tuned specifically for passage retrieval. There are two common setups for utilizing neural models in this task: complete passage retrieval using a neural model or combining another retrieval engine to retrieve a subset of passages, followed by using the neural model to select the most relevant ones. The latter approach is employed when the reranking model is too slow to process an entire document collection.

The Poleval 2023 Task 3: Passage Retrieval challenge aims to identify the best method for passage retrieval in Polish texts. The competition's test dataset comprises three domains: wiki-trivia, legal-questions, and allegro-faq. However, only the wiki-trivia domain is provided as the training and development dataset.

In this paper, we discuss the two-stage approach that achieved a score of 69.36 NDCG@10 on the final test competition dataset. Our method involves two phases. Firstly, we use the OKAPI BM25 algorithm to retrieve relevant passages. Then, an ensemble of Cross Encoder models is employed to rerank these passages. These models are publicly available multilingual models that have been trained on various languages (including Polish) and finetuned on multilingual corpora for passage reranking, as outlined in [1]. We used these models with no further finetuning on the challenge dataset for two domains: legal-questions and allegro-faq. For the wiki-trivia domain, one model was fine-tuned and used in combination with models that had no further finetuning.

TABLE I
Dataset statistics split into given domains. rel. passages stand for relevant passages.

-	wiki-trivia	legal-questions	allegro-faq
train questions	4401	0	0
dev questions	599	0	0
test-A questions	400	400	400
mean test-A rel. passages	3.46	1.97	1.09
test-B questions	891	318	500
mean test-B rel. passages	3.39	2.03	1.05
passages	7097322	26287	921
mean word per passage	44.6	155.1	50.0

II. Related Work

A. Reranker models and modern neural Information Retrieval

MS MARCO [2] is a large publicly available reranking dataset retrieved by Bing. The dataset includes queries, retrieved documents by search engine, and a label on whether a user clicked a document. The corpus is in the English language. Recently, authors of mMARCO [1] translated this corpus into many languages (but not into Polish though) and trained Cross Encoder reranker models on it. The base models were multilingual. The performance was effective not only for translated languages but also for not translated languages, only visible by models in the semisupervised pretraining phase.

BEIR [3] is an Information Retrieval benchmark for Zero-shot Evaluation between different domains. The authors provided many comparisons between different retrieval architectures. Very recently, the benchmark for Polish Information Retrieval was released in BEIR-PL paper [4].

B. Language models working on Polish texts

There are a few transformer language models trained for the Polish language: HerBERT [5], plt5 [6], Polish RoBERTa [7]. There are also many multilingual language models working on Polish languages, such as XLM-RoBERTa [8], multilingual DeBERTa [9], and mT5 [10].

III. PolEval 2023 Task 3: Passage Retrieval Challenge

A. Data

The task is to retrieve the relevant passages given a query. The queries and passages are in the Polish language. There are separate domains: wiki-trivia, legal-questions, allegro-faq. In the below subsection, each domain is presented. There are the following datasets: training (train), development (dev), test-A (preliminary test set), and test-B (final test set). For the training and development dataset, golden truth data was released during the competition, but the golden truth dataset was not. After competitions, the test set golden truth was released to https://github.com/poleval/2022-passage-retrieval-secret. Training and development datasets consist of only wiki-trivia, but the test dataset consists of all three domains. Below all domains are described. Some dataset statistics are given in Table I. Domains vary greatly in the number of passages and mean relevant passages per query.

1) wiki-trivia: Questions are general-knowledge typical for TV quiz shows, such as *Fifteen to One* or Polish equivalent *Jeden z dziesięciu*. For each question, there were manually selected up to five relevant passages (the mean number for the training dataset is 3.28 with a standard deviation of 1.45). The passages corpus consists of 7097322 elements. This domain was selected for train, dev, and test datasets. There are 4041 questions in the train dataset, 599 in the dev dataset, 400 in the test-A dataset, and 891 in the test-B dataset. Below, one example question with all correct passages is presented.

Example Question: "Czy w państwach starożytnych powoływani byli posłowie i poselstwa?"
Correct passage: "Poselstwo do Chin. Chińska "Księga Późniejszych Hanów" ("Houhanshu") zanotowała informację, iż w roku 166 drogą morską przez Ocean Indyjski do kraju Jinan (Annam) przybyło poselstwo "króla Andun z Daqin" (Rzymu), oferując kość słoniową, rogi nosorożca i szylkret."
Correct passage: "Z okresu starożytnych Indii pochodzą pierwsze zachowane instrukcje na temat zadań dyplomaty (rozpoznawać i informować) oraz cech dyplomaty (wykształcony, zręczny, zjednujący sobie ludzi). Dyplomację stosowali już starożytni Grecy, od których wywodzi się termin „diplóos" oznaczający złożony we dwoje dokument – rodzaj listów uwierzytelniających w które wyposażany był poseł."
Correct passage: "Ze starożytnej Grecji pochodzi również przywilej nietykalności posła i poselstwa (immunitet), które już wtedy były uznawane za zasadę będącą elementem prawa narodów. Pierwotną formą quasi-dyplomacji była wymiana posłów przez społeczności plemienne w celu przekazania i wyjaśnienia przekazu mocodawcy, lub wynegocjowania jakiegoś porozumienia (np. o zakończeniu wojny)."

2) legal-questions: A portion of the legal questions were generated by randomly selecting provisions and formulating questions based on their content. The task is similar to SQuAD and requires only identifying relevant passages rather than answering the question. The questions were supplemented with 26287 provisions derived from over one thousand laws published between 1993 and 2004. There are 400 questions in the test-A dataset and 318 in the test-B dataset. Below, the example questions from the test-A dataset are provided.

Example Question: "Ile trwa kadencja szefa służby cywilnej?"
Correct passage: "Ustawa z dnia 18 grudnia 1998 r. o służbie cywilnej Rozdział 1 Przepisy ogólne","text":"Art. 9. 1. Szefa Służby Cywilnej powołuje, po zasięgnięciu opinii Rady Służby Cywilnej, Prezes Rady Ministrów spośró d urzędników służby cywilnej. 2. Kadencja Szefa Służby Cywilnej trwa 5 lat, licząc od dnia powołania; Szef Służby Cywilnej pełni obowiązki do dnia powołania jego następcy. 3. Kadencja Szefa Służby Cywilnej wygasa w razie jego śmierci lub odwołania. 4. Prezes Rady Ministrów odwołuje Szefa Służby Cywilnej w razie: 1) rezygnacji ze stanowiska, 2) utraty zdolności do pełnienia powierzonych obowiązków na skutek długotrwałej choroby, trwającej co najmniej 6 miesięcy. 5. Prezes Rady Ministrów odwołuje Szefa Służby Cywilnej także w przypadku, gdy przestał on odpowiadać jednemu z warunków określonych w art. 4. Odwołanie w przypadkach, o których mowa w art. 4 pkt 4 i 5, następuje za zgodą co najmniej 23 składu Rady Służby Cywilnej."

3) allegro-faq: Questions regards the large e-commerce platform- Allegro.pl were created using help articles and lists of frequently asked questions. There are 400 questions in the test-A dataset and 500 questions in the test-B dataset. In total, there are 921 passages. Here is an example question from the test-A dataset:

Example Question: "Otrzymałem rekompensatę z POK, a później zwrot od Sprzedającego. Co mam zrobić?"
Correct passage: "Jeśli wypłaciliśmy Ci rekompensatę w ramach Programu Ochrony Kupujących a Ty otrzymasz zwrot pieniędzy od Sprzedającego, masz obowiązek zwrócić nam rekompensatę. Zgodnie z punktem 6 Część IV Załącznika nr 9 do Regulaminu Allegro, na zwrot rekompensaty masz 7 dni od naprawienia szkody przez Sprzedającego."

B. Evaluation Metric

The GEval evaluation tool [11] uses Normalized Discounted Cumulative Gain for the top ten passages (NDCG@10) as the challenge metric. The challenge was hosted on the Gonito platform [12], and the final evaluation was conducted on the test-B dataset across all domains. It should be noted that the sample split between domains is not equal, which means that some domains have a greater impact on the final score.

IV. Method

The solution involves two stages: Retrieval and Reranking. Retrieval is carried out using the lexical method OKAPI BM25, which is quick but not as effective as a neural ranking model. Additionally, it does not require training. The best performing method for Reranking is through Cross Encoders, but it is slow as it requires processing every query-passage pair. Due to its time-consuming nature, it can only operate on a limited set of passages, except for the allegro-faq domain, which consists of only 921 passages.

A. Retrieval phase

For retrieval model We used OKAPI BM25 algorithm using parameters k_1=1.2 , b=0.75, ε=0.25. The utilized library may be accessed via https://github.com/zhusleep/fastbm25. The preprocessing included tokenization using the nltk library, specifically nltk.tokenize.word_tokenize, lowercase normalization, stemming using pystempel (accessed via https://github.com/dzieciou/pystempel) with the Polimorf [13] stemmer, and removal of Polish stopwords.

B. Reranking phase

The reranking phase was performed using an ensemble of multilingual reranker models based on Cross-Encoder architecture. We used different sets of the ensemble for wiki-trivia domain and legal-questions with allegro-faq questions. Both are described in the following section. The ensembles were created by summing up all the individual models' probability scores. Finetuning, if performed, was loosely based on a script from Sentence-Transformer library [14], namely https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/ms_marco/train_cross-encoder_scratch.py. The process of finetuning and inference was completed on A100 GPU card. We used one 100 negative query-passage pair for each positive passage selected from the training dataset. The negative passage selection was from the top 2000 passages returned by the described OKAPI BM25 algorithm. The used Loss was BCEWithLogitsLoss with a constant learning rate scheduler of 1e-6 and 2000 warmup steps. The best-performing model was selected for inference from training for ten epochs.

1) wiki-trivia: Reranking was based on the top 3000 results from the OKAPI BM25 algorithm. Because wiki-trivia passages are relatively short, they only require a little time, although, during experiments, we observed that reranking with above 1000 passages, there is not much gain in the metric score.

The ensemble consisted of three models:

- Publicly available reranker based on multilingual T5 (mT5) model [10] (also trained on Polish corpora) and fine-tuned to automatically translated reranking corpus MS MARCO [2] into Portuguese. The model unicamp-dl/mt5-13b-mmarco-100k via https://huggingface.co/unicamp-dl/mt5-13b-mmarco-100k was used as it is without further fine-tuning to the competition training dataset. Therefore model works in a zero-shot manner as described in [1].
- Reranker cross-encoder/mmarco-mMiniLMv2-L12-H384-v1 (accesed via https://huggingface.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1.
  The multilingual base model MiniLMv2 [15] is fine-tuned on mMARCO dataset (MS MARCO translated into multiple languages). Please note MMARCO dataset does not contain the Polish language, but MiniLMv2 was trained on Polish. However, it performed well on the dev dataset. We then fine-tuned it further on the competition train dataset, slightly improving it.
- Reranker cross-encoder/mmarco-mdeberta-v3-base-5negs-v1 (https://huggingface.co/cross-encoder/mmarco-mdeberta-v3-base-5negs-v1) based on multilingual DeBERTaV3 [9] finetuned on MMARCO dataset. During the competition, the model was publicly available but was removed before the time of writing this article. We further fine-tuned the model to the competition training dataset.

2) legal-questions and allegro-faq: Reranking was performed on top 1500 passages for legal-questions. The limit was lower than for wiki-trivia due to the length of passages collection and longer computation time. For the allegro-faq domain, reranking was performed on all the passages since the whole collection consists of only 921 passages. For both domains, the same ensemble was used. The following models were used without further finetuning to the competition dataset. We conducted experiments using models fine-tuned to wiki-trivia, but their performance dropped drastically. Finally, we used the following models:

- Model unicamp-dl/mt5-13b-mmarco-100k via https://huggingface.co/unicamp-dl/mt5-13b-mmarco-100k described in the previous section.
- Model unicamp-dl/mt5-3B-mmarco-en-pt via https://huggingface.co/unicamp-dl/mt5-3B-mmarco-en-pt, which is the same as above but in the 3B parameters version.

V. Results

The presented method scores 75.40 NDCG@10 on preliminary test-A and on 69.36 NDCG@10 on final test-B data. The experiments code is available at https://github.com/kubapok/poleval22. The analysis of single models on different reranking size limits is presented in Table III for test-A and in Table II for test-B. The results vary between domains, probably because of text nature, as well as different passage collection sizes and different size mean relevant passages per one query. All the presented reranking models score better than the OKAPI BM25 baseline. With the reranking size limit, the performance is better. However, the gain isn't great beyond the reranking limit of 500. Finetuning models increase their performance only on the wiki-trivia domain and worsen on other domains. Unfortunately, these results are not included in the presented tables as we didn't save them.

VI. Other Experiments

We have tried other approaches as well. These experiments were very preliminary and may yield better results if we spend more time on them. However, we decided to include them in this paper anyway.

TABLE II

NDCG@10 results for the whole final testing dataset test-B and split into domains. ft stands for the model fine-tuning to the competition data, whereas no-ft stands for no fine-tuning. The number at the right of the model name stands for the reranking size from the OKAPI BM25 algorithm. Some experiments were not conducted or saved. In this case, the score is labeled as "-"

model	test-B	wiki-trivia	legal-questions	allegro-faq
final ensemble	69.36	55.13	86.39	83.88
OKAPI BM25	42.55	23.48	81.31	51.87
mmarco-mMiniLMv2-L12-H384-v1 no-ft 10	48.85	28.45	83.00	63.47
mt5-3B-mmarco no-ft 10	50.31	29.47	84.35	65.81
mt5-13B-mmarco no-ft 10	50.36	29.63	83.59	66.15
mmarco-mMiniLMv2-L12-H384-v1 no-ft 50	56.18	35.88	85.26	73.84
mt5-3B-mmarco no-ft 50	59.04	38.06	86.75	78.80
mt5-13B-mmarco no-ft 50	59.79	39.30	85.30	80.08
mmarco-mMiniLMv2-L12-H384-v1 no-ft 100	57.76	38.22	85.54	74.91
mt5-3B-mmarco no-ft 100	61.42	41.24	87.06	81.09
mt5-13B-mmarco no-ft 100	62.65	43.17	85.63	82.75
mmarco-mMiniLMv2-L12-H384-v1 no-ft 500	58.52	39.86	85.61	74.56
mt5-3B-mmarco no-ft 500	63.48	44.41	86.67	82.70
mt5-13B-mmarco no-ft 500	65.04	47.21	85.42	83.86
mmarco-mMiniLMv2-L12-H384-v1 no-ft 1000	58.91	40.49	85.66	74.72
mt5-3B-mmarco no-ft 1000	64.12	45.48	86.64	83.01
mt5-13B-mmarco no-ft 1000	65.59	48.13	85.22	84.21
mmarco-mMiniLMv2-L12-H384-v1 no-ft 1500	58.99	40.70	85.51	74.72
mmarco-mMiniLMv2-L12-H384-v1 ft 1500	-	47.64	-	-
mmarco-mdeberta-v3-base-5negs-v1 no-ft 1500	-	45.30	-	-
mmarco-mdeberta-v3-base-5negs-v1 ft 1500	-	51.73	-	-
mt5-3B-mmarco no-ft 1500	64.46	46.17	86.55	83.01
mt5-13B-mmarco no-ft 1500	65.99	48.96	85.04	84.21

TABLE III

NDCG@10 results for the whole preliminary testing dataset test-A and split into domains. ft stands for the model fine-tuning to the competition data, whereas no-ft stands for no fine-tuning. The number at the right of the model name stands for the reranking size from the OKAPI BM25 algorithm. Some experiments were not conducted or saved. In this case, the score is labeled as "-"

model	test-A	wiki-trivia	legal-questions	allegro-faq
final model	75.40	52.25	86.48	87.48
OKAPI BM25	52.67	22.26	81.78	53.96
mmarco-mMiniLMv2-L12-H384-v1 no-ft 10	58.81	26.03	84.95	65.46
mmarco-mdeberta-v3-base-5negs-v1 no-ft 10	59.52	26.69	84.79	67.09
mt5-base-mmarco-v2 no-ft 10	58.60	25.91	83.96	65.95
mt5-3B-mmarco no-ft 10	60.14	27.09	84.74	68.60
mt5-13B-mmarco no-ft 10	60.09	27.30	84.00	68.98
mmarco-mMiniLMv2-L12-H384-v1 no-ft 50	65.13	32.81	85.60	76.98
mmarco-mdeberta-v3-base-5negs-v1 no-ft 50	66.96	35.17	85.92	79.80
mt5-base-mmarco-v2 no-ft 50	64.97	33.14	84.31	77.45
mt5-3B-mmarco no-ft 50	68.24	35.81	85.57	83.33
mt5-13B-mmarco no-ft 50	68.78	36.70	84.90	84.75
mmarco-mMiniLMv2-L12-H384-v1 no-ft 100	66.31	35.32	85.96	77.66
mmarco-mdeberta-v3-base-5negs-v1 no-ft 100	68.39	38.27	86.28	80.63
mt5-base-mmarco-v2 no-ft 100	65.70	35.04	84.35	77.70
mt5-3B-mmarco no-ft 100	69.97	38.82	86.10	84.99
mt5-13B-mmarco no-ft 100	70.83	40.43	85.42	86.63
mmarco-mMiniLMv2-L12-H384-v1 no-ft 500	67.11	37.46	85.80	78.05
mmarco-mdeberta-v3-base-5negs-v1 no-ft 500	69.31	41.02	85.92	80.99
mt5-base-mmarco-v2 no-ft 500	65.85	36.50	83.81	77.25
mt5-3B-mmarco no-ft 500	71.40	42.13	86.14	85.93
mt5-13B-mmarco no-ft 500	72.45	43.94	85.50	87.91
mmarco-mMiniLMv2-L12-H384-v1 no-ft 1000	67.37	38.29	85.69	78.11
mmarco-mdeberta-v3-base-5negs-v1 no-ft 1000	69.73	42.30	85.85	81.04
mt5-base-mmarco-v2 no-ft 1000	65.98	36.96	83.66	77.32
mt5-3B-mmarco no-ft 1000	71.84	43.29	86.20	86.03
mt5-13B-mmarco no-ft 1000	73.06	45.54	85.66	88.00
mmarco-mMiniLMv2-L12-H384-v1 no-ft 1500	67.35	38.45	85.50	78.11
mmarco-mMiniLMv2-L12-H384-v1 ft 1500	-	45.84	-	-
mmarco-mdeberta-v3-base-5negs-v1 no-ft 1500	69.82	42.58	85.82	81.04
mmarco-mdeberta-v3-base-5negs-v1 ft 1500	-	48.99	-	-
mt5-base-mmarco-v2 no-ft 1500	65.99	37.11	83.54	77.32
mt5-3B-mmarco no-ft 1500	72.01	43.78	86.22	86.03
mt5-13B-mmarco no-ft 1500	73.28	46.26	85.57	88.00

A. Translating Polish texts into English.

We translated Polish passages and queries into English using a machine translation model accessed by https://huggingface.co/gsarti/opus-mt-tc-en-pl [16]. English Cross Encoder reranking models did not perform on the translated texts better than multilingual reranking models on Polish texts tough.

B. Bi Encoder models

We experimented with various publicly available Bi Encoder models, using them as one-stage retrieval models. Unfortunately, their performance was significantly inferior to that of the OKAPI BM25 algorithm operating alone. However, combining the OKAPI BM25 and Bi Encoder models as retrieval models for further reranking with the Cross Encoder model may lead to improved results and is a promising area for research. Our highest Bi Encoder score for untranslated documents was 9.26 NDCG@10, achieved using the sentence-transformers/distiluse-base-multilingual-cased-v1 model. For translated texts into English, our highest score was 21.00, obtained using the sentence-transformers/all-mpnet-base-v2 model.

C. Translating MS MARCO into Polish

MMARCO does not include translations for Polish texts. We've attempted translating MS MARCO into English using model gsarti/opus-mt-tc-en-pl and training several reranking models on this data. The approach is similar to [4]. Nevertheless, this work was published after the competition. In our case, this approach didn't yield better results than large multilingual models.

VII. Conclusions

This paper summarizes our solution to Poleval 2023 Task 3: Passage Retrieval. The system operates in two stages, utilizing OKAPI BM25 for retrieval and a multilingual ensemble of Cross Encoders for reranking. However, the system's performance varies between domains due to the limited availability of training data for only one domain. While fine-tuning the neural model can enhance results for this domain, it may have a negative impact on other domains.

References

[1] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira, "mmarco: A multilingual version of the ms marco passage ranking dataset," 2021.

[2] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated machine reading comprehension dataset.," *CoRR*, vol. abs/1611.09268, 2016.

[3] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[4] K. Wojtasik, V. Shishkin, K. Wołowiec, A. Janz, and M. Piasecki, "Beir-pl: Zero shot information retrieval benchmark for the polish language," *arXiv preprint arXiv:2305.19840*, 2023.

[5] R. Mroczkowski, P. Rybak, A. Wróblewska, and I. Gawlik, "HerBERT: Efficiently pretrained transformer-based language model for Polish," in *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, (Kiyv, Ukraine), pp. 1–10, Association for Computational Linguistics, Apr. 2021.

[6] A. Chrabrowa, Ł. Dragan, K. Grzegorczyk, D. Kajtoch, M. Koszowski, R. Mroczkowski, and P. Rybak, "Evaluation of transfer learning for polish with a text-to-text model," *arXiv preprint arXiv:2205.08808*, 2022.

[7] S. Dadas, M. Perełkiewicz, and R. Poświata, "Pre-training polish transformer-based language models at scale," in *Artificial Intelligence and Soft Computing*, pp. 301–314, Springer International Publishing, 2020.

[8] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *CoRR*, vol. abs/1911.02116, 2019.

[9] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," in *International Conference on Learning Representations*, 2021.

[10] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (Online), pp. 483–498, Association for Computational Linguistics, June 2021.

[11] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki, "GEval: Tool for debugging NLP datasets and models," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, (Florence, Italy), pp. 254–262, Association for Computational Linguistics, Aug. 2019.

[12] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń, "Gonito.net – open platform for research competition, cooperation and reproducibility," in *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language* (A. Branco, N. Calzolari, and K. Choukri, eds.), pp. 13–20, 2016.

[13] W. Kieraś and M. Woliński, "Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego," *Język Polski*, vol. XCVII, no. 1, pp. 75–83, 2017.

[14] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 11 2020.

[15] W. Wang, H. Bao, S. Huang, L. Dong, and F. Wei, "MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, (Online), pp. 2140–2151, Association for Computational Linguistics, Aug. 2021.

[16] J. Tiedemann and S. Thottingal, "OPUS-MT — Building open translation services for the World," in *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, (Lisbon, Portugal), 2020.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1271–1276
DOI: 10.15439/2023F8119
ISSN 2300-5963 ACSIS, Vol. 35

Hybrid retrievers with generative re-rankers

Marek Kozlowski
National Information Processing Institute
Warsaw, Poland
Email: marek.kozlowski@opi.org.pl

Abstract—**The passage retrieval task was announced during PolEval 2022 (SemEval-inspired evaluation campaign for natural language processing tools for Polish). Passage retrieval is a crucial part of modern open-domain question answering systems that rely on precise and efficient retrieval components to identify passages that contain correct answers. Our solution to this task is a multi-stage neural information retrieval system. The first stage consists of a candidate passage retrieval step in which passages are retrieved using federated search over sparse (BM25) and dense indexes (two FAISS indexes built using bi-encoder type retrievers based on Polish RoBERTa models). The second stage consists of a re-ranking step of the previously selected passages with a neural model, mt5-13b-mmarco. The model scores each passage by its relevance to a given query. The highest-scoring passages are then retained as the final result. Our system achieved second place in the competition.**

I. Introduction

PASSAGE retrieval is a crucial part of modern information retrieval systems that rely on highly efficient retrieval components to identify passages (mostly represented as paragraph(s)) that contain correct answers.

Information retrieval is a popular research domain that focuses on obtaining relevant information from a collection of diverse data resources (mainly textual ones). When working with information retrieval tasks, one can rely on using bag-of-words (BOW) systems (such as the BM25) or different approaches supported by deep learning models (such as dense retrievers or re-ranking modules).

Recently, neural information retrieval has surpassed the lexical methods based on BOW (such as TF-IDF + cosine similarity or BM25) by fine-tuning pre-trained language models, including generative ones such as BART, T5, and representation ones such as BERT, RoBERTa. Although they beat the classical methods in terms of quality efficiency, they are not free of drawbacks, such as the need for a relevant training set. They solve many problems of lexical methods, including poor semantic capabilities, but at the cost of an expensive training process that uses a relevant number of labeled examples. There is also evidence that neural information retrieval systems are characterized by poor generalizability to other domains. This means that in a zero-shot or few-shot setup (i.e. no or little training data), lexical methods remain competitive with, or even better than, neural models.

The lack of Polish-language datasets, relevant evaluations, and benchmarks encouraged the Polish AI community to establish the PolEval initiative, a SemEval-inspired evaluation campaign for natural language processing tools for Polish. Submitted tools compete against each other using available data in tasks selected by the organizers, and are evaluated according to pre-established procedures. In 2022, the following tasks were announced: punctuation prediction from conversational language, abbreviation disambiguation, and passage retrieval.

The goal of the passage retrieval task was to develop a system for cross-domain question answering retrieval in the Polish language.

The participants were given a training set that comprised question–passage pairs from the trivia domain—the type of general-knowledge questions that are typical on popular television quiz shows. For each test question, the systems were tasked with retrieving ordered lists of the ten most relevant passages (i.e. those that contain the answer) from the provided corpus. The systems were scored based on their performance on all three test sets.

Using the PolEval data and our own (the translated MS MARCO dataset), we evaluated various approaches toward passage retrieval, where the sentence-level queries are given and the corpus is counted in millions of passages. We present the best of our submitted approaches.

This article is structured as follows: In Section 2, we discuss related work. Section 3 is devoted to presenting the datasets. In Section 4, we present our approach and the results of our evaluations. Section 6 outlines our conclusions.

II. Related Work

We commence this section from the standard formulation of the passage retrieval task. From a finite, but arbitrarily large collection of passages $P = \{p_1, p_2...\}$, the system's task, given a query q, is to return a top-N ranking of the passages that maximizes a metric of quality, such as normalized discounted cumulative gain (NDCG) or average precision. NDCG is a measure of ranking quality. Highly relevant documents are more useful than moderately relevant documents, which are, in turn, more useful than irrelevant documents. In the subsections below, we explain sparse and dense retrieval, as well as dense re-ranking.

A. Sparse retrieval

Traditionally, retrieval has been dominated by lexical approaches like TF-IDF + cosine similarity, and BM25.

BM25 is a BOW retrieval function that ranks sets of documents based on the query terms that appear in them, regardless

This work corresponds to the Poleval 2022—Passage Retrieval competition.

of their proximity [1]. BM25 is a retrieval model based on the probabilistic retrieval framework. BM25 often achieves better performance compared to TF-IDF, which rewards term frequency and penalizes document frequency. BM25 goes beyond this to consider document length and term frequency saturation.

The results [2] demonstrate that BM25 is a robust baseline. However, these approaches suffer from a lexical gap and are able to retrieve only documents that contain keywords present within the query. In addition, lexical approaches treat queries and documents as BOW by not considering word order.

To overcome this lexical gap, techniques to improve lexical retrieval systems using neural networks have been proposed. Sparse methods, such as docT5query [3], identified document expansion methods that use sequence-to-sequence models that generate possible queries for which a given document would be relevant. At base, it involves training a model that predicts questions for which the input document might contain answers. The generated questions are then appended to the original documents, which are indexed. The docT5query model takes its name from the generative model T5. The primary advantage of this approach is that expensive neural inference is pushed to indexing time.

B. Dense retrieval

Recently, dense retrieval approaches have also been proposed. They are capable of capturing semantic matches, and attempt to overcome the (potential) lexical gap. Dense retrievers map queries and documents in a single, common dense vector space. A bi-encoder architecture based on BERT-type models demonstrated strong performance for various open-domain question answering tasks.

An important recent innovation for passage retrieval is the introduction of dense retrieval models that take advantage of a bi-encoder design. Bi-encoders produce two corresponding embeddings for a given two-sentence pair (e.g. a query and a passage), which can then be compared efficiently using cosine similarity.

Bi-encoders are used whenever a sentence embedding is needed in a vector space for efficient comparison in applications such as information retrieval, semantic search, or clustering. Cross-encoders would be the wrong choice for these applications, because a cross-encoder does not produce a sentence embedding; it processes both sentences simultaneously through the Transformer network, which is very computationally expensive with such a large scale of data.

With sufficient labeled data, we can learn encoders (typically, Transformer-based models) that project queries and documents into a dense (semantic) representation space (e.g. 768 dimensions) where the relevance ranking can be recast as a nearest neighbor search over representation vectors [4].

Bi-encoders can be used also as re-rankers, working not on all documents in corpora, but only on subsets of them. There are settings in which the first-stage retriever returns a limited number of documents and passes them to the re-ranker. Re-ranking can be also performed as a second stage retrieve on a limited collection of documents from stage 1 without any top-k constraints.

The two most popular bi-encoders are DPR and ANCE. DPR [5] is a two-tower bi-encoder trained with the hard negatives and single-batch negatives of a single BM25. The Multi-DPR model is a BERT-base-uncased model trained on four QA datasets: NQ, TriviaQA, WebQuestions, and CuratedTREC. ANCE [6] is a bi-encoder that uses approximate nearest neighbor negative contrastive learning, which selects hard training negatives globally from the entire corpus.

C. Dense re-ranking

Modern search engines are developed as multi-stage retrieve & ranking architectures in which a first-stage retriever generates candidate documents that are then re-ranked by deep learning models. Re-ranking requires feeding the model both the query and the candidate text. Neural re-ranking approaches use the output of a first-stage retrieval system to create a better order of the retrieved documents.

Significant improvements in the performance of re-ranking has been achieved using the cross-encoder mechanism based on BERT-type models. However, it entails the disadvantage of high computational overhead, because cross-encoders do not scale well for large datasets.

In the generative model era, the T5 model [7] was applied in an identical manner to classical cross-encoders, and yielded SOTA results in zero-shot scenarios [8]. MonoT5 is an adaptation of the T5 model [7] proposed by Nogueira [9]. The model uses query–document pairs as input and generates probability scores that quantify the relevance between them. The model is asked to generate either a “true” or a “false” token for a source prompt that contains a query and a document, from which we can extract the probability of relevance used to sort the candidates. In [8], large re-rankers such as monoT5-3B outperform distilled ones and dense models of equivalent size on zero-shot tasks. This approach outperformed other fine-tuned re-ranking models significantly in data scarcity scenarios. The average results of monoT5-3B [8] demonstrate that strong zero-shot effectiveness in new text domains can be achieved by increasing the number of model parameters and without fine-tuning in-domain data. In summary, the monoT5 model, fine-tuned on the MS MARCO passage dataset, achieves state-of-the-art results on the TREC Deep Learning Track, as well as impressive zero-shot effectiveness on BEIR and many other datasets [4].

III. Dataset

Quality, representativeness, and quantity are crucial aspects of any dataset. Neural language models must usually be pre-trained and fine-tuned on high-quality labeled examples, such as documents, queries, or passages. For many languages, the available training and test datasets are limited or biased. In the sub-sections below, we present the dataset provided by the organizers, then our Polish translation of MS MARCO—a collection of datasets that focuses on deep learning in search.

A. The PolEval dataset

The PolEval organizers prepared datasets that consist of question–passage pairs from domains as diverse as general knowledge, legal matters, and the FAQs section of Polish e-commerce website Allegro[1].

Training dataset. The training set consists of 5000 trivia questions: the type of general-knowledge questions that are typical on popular television quiz shows. Each question in the training set has up to five passages from the Polish-language version of Wikipedia manually assigned. These contain the answer to the question. The training set consists of 16 389 question–passage pairs. Additionally, the PolEval organizers released a Wikipedia corpus of 7 million passages. The raw Wikipedia dump was parsed using WikiExtractor and split into passages at the ends of the paragraphs, or if the passage was longer than 500 characters. The training set comprises only data from the general knowledge domain[2].

Test dataset. There were three partial test sets with questions from different domains. The first consists of 1291 general knowledge questions that are similar to those from the training set. The second consists of 900 questions and 921 passages regarding the Polish e-commerce platform Allegro. The dataset was created based on FAQs available on the Allegro portal. Each question–passage pair was verified manually. The third dataset contains over 700 questions from the legal domain. The dataset was built in reverse because some part of the questions were created by random selection of the provisions and asking questions based on their content. The legal-domain-oriented passages count approximately 26 000 provisions extracted from more than 1000 laws published between 1993 and 2004.

Using those test sets, the organizers created the validation, test-A, and test-B datasets. They contain 599, 1200, and 1709 examples, respectively.

B. The Polish translation of MS MARCO

Microsoft Machine Reading Comprehension (MS MARCO)[3] [10] is a collection of datasets that focuses on the evaluation of modern machine learning methods in different search challenges. The first dataset was a question-answering set that features 100 000 real Bing questions and human-generated answers. Over time, the collection has expanded to at least one-million-questions, a natural language generation dataset, a passage ranking dataset, a keyphrase extraction dataset, a crawling dataset, and a conversational search dataset.

MS MARCO has one drawback: it is a large-scale dataset that focuses chiefly on the English language. A translation is available of these English corpora. MMARCO[4] [11] is a multilingual version of the MS MARCO passage ranking dataset that comprises 13 languages. It was created using machine translation. This dataset demonstrates good transfer learning capabilities, as well as being a popular choice for the evaluation of deep learning models. Using the machine translation approach to create new datasets minimizes the high costs of extensive manual data annotation processes. However, MMARCO does not contain data for the Polish language.

To address this at the National Information Processing Institute (with the assistance of Dr. Sławomir Dadas), we prepared the Polish training set using a translation of MS MARCO into Polish (approximately 39 million triplet translations) with two type of models: a) mbart-large-50-one-to-many-mmt (a fine-tuned checkpoint for multilingual machine translation of the mBART-large-50 model) [12] and b) our in-house English–Polish convolutional neural machine translation models trained using the Fairseq sequence modeling toolkit[5]. We used different neural machine translation models because the quality of Polish translations varies across the MS MARCO dataset. Finally, we mixed them heuristically. The process of machine translation lasted a few days and consumed 8xV100 GPUs.

IV. Approach

Our approach is inspired by [13], in which the authors describe NeuralSearchX, a metasearch engine based on a multi-purpose large re-ranking model that merges results and highlights sentences.

Our solution is a multi-stage neural information retrieval system. The first stage involves a candidate passage retrieval step in which passages are retrieved using federated search over sparse (BM25) and dense (two FAISS indices built using dedicated RoBERTa-based encoders, the result of the bi-encoders' training) indices. The second stage involves a re-ranking step of the previously selected passages with a fine-tuned neural model, mT5. The model scores each document by its relevance to a given query. The top-scoring documents are then retained as a final result.

A. Candidate passage retrieval

In the first stage, we used both the sparse and dense retrieval methods provided by the BEIR library [2]. BEIR is a heterogeneous benchmark that contains diverse information retrieval tasks. It also provides a common and easy framework for evaluation of various natural-language-processing-based retrieval models within the benchmark. We opted to use BEIR to take advantage of its support for lexical and dense retrievers.

In our solution, we use Elasticsearch[6]—which applies the BM25 method for scoring documents against queries—as a lexical retriever. BM25 is Elasticsearch's default similarity ranking algorithm. Elasticsearch is a distributed search and analytics engine built on Apache Lucene. Since its release, Elasticsearch has quickly become the most popular search engine in developed systems and is commonly used for full-text search when huge masses of textual data are involved. To support Polish language, we installed the morfologik plugin[7] in our Elasticsearch instance. This plugin is crucial for

[1] https://beta.poleval.pl/challenge/2022-passage-retrieval

[2] http://poleval.pl/tasks/task3

[3] https://microsoft.github.io/msmarco/

[4] https://github.com/unicamp-dl/mMARCO

[5] https://github.com/sdadas/polish-nlp-resources#convolutional-models-for-fairseq

[6] https://www.elastic.co

[7] https://github.com/allegro/elasticsearch-analysis-morfologik

normalization purposes—specifically, we used it to perform lemmatization of words to improve the search efficiency.

The dense retrievers in the proposed approach are Sentence-BERT-type encoders. We trained them in the Bi-encoder architecture, and used them with FAISS index[8] support. FAISS is a library for the efficient similarity search and clustering of dense vectors. Bi-encoder architectures are used because sentence embedding in a vector space is needed for efficient comparison during semantic search or clustering. The queries and passages are passed independently to the sentence transformer to produce fixed-size embeddings. These can then be compared using cosine similarity to identify matching passages for a given query. The training of the dense retrievers is performed by fine-tuning encoders—specifically, we fine-tuned the RoBERTa model used in our bi-encoder architecture. During training, we used a loss function called MultipleNegativesRankingLoss, $number\_of\_epochs = 1-10$ and $batch\_size = 32$. We pass triplets in the format: $(query, positive\_passage, negative\_passage)$, where negative passage is a hard negative example (not positive one, but lexically similar to the positive one) that is retrieved by lexical search in the whole passage corpora. We used Elasticsearch to obtain ($max = 10$) hard negative examples for given positive passages. We trained two types of dense retrievers: a) using RoBERTa-base-v2[9] as a transformer model, and training one epoch on 500 000 triplets (135 000 Poleval ones and 370 000 Polish MS Marco ones randomly selected); and b) using RoBERTa-large-v2[10] as a transformer model, and training 10 epochs on a few million triplets (several million Polish MS Marco ones randomly selected). We then fine-tuned for one epoch on all 135 000 PolEval triplets.

More information on how to train bi-encoders effectively can be found in the sentence-transformers github[11].

Since all of the retrievers are independent, we could run them in parallel—therefore not creating a significant overhead in the process and maintaining an adequate latency.

In the last phase, we collected the top-K results from three retrievers (one sparse and two dense), where the limit K is set for each retriever, respectively.

B. Re-ranking

After the candidate passage retrieval, the next step involved merging all of the candidate passages into a single list, and then ranking the documents so that the most relevant ones were at the top of the results list. To re-rank, we used generative models of type T5 [7]—specifically, the multilingual version of mT5: the mt5-13b-mmarco-100k model[12]. Re-ranking is performed in the Polish language, however, we used the multilingual language model already fine-tuned as a re-ranker, using the following prompt:

Algorithm 1 Batch re-ranking relevance predictions using the T5 class model, function: get list of query–passage pairs to verify and return list of probabilities of their relevance.

```
def predict(self,
        query_passage_pairs: List[Tuple[str, str]],
        batch_size: int = 16) -> List[float]:

    probability_scores = []

    # create batches
    batches = []
    for i in range(0, len(query_passage_pairs),
                       batch_size):
        batches.append(
        query_passage_pairs[i: i + batch_size])

    for batch in tqdm(batches):
        #"Query: {q_txt} Passage: {p_txt} Relevant:"
        prompts = [f"Query: {q_p_pair[0]} " \
                f"Document: {q_p_pair[1]} Relevant:"
                for q_p_pair in batch]

        res = self.modelT5.predict_in_batch(prompts)

        for label, prob in zip(res[0], res[1]):
            final_prob = prob
            if label != 'true':
                final_prob = 1 - final_prob

            probability_scores.append(final_prob)
    return probability_scores
```

$$Query : \{query\_text\}\ Document : \{pass\_text\}\ Relevant :$$

The model was asked to generate for a given prompt either a "true" or a "false" token, from which we could extract the probability of relevance used to sort the candidates. Some pseudo-code of the batch prediction is contained in Listing 1.

The mT5 model used in our solution contains 13 billion parameters; in other words, it is an XXL model. It is based on the T5 model [7], and its adaptation to re-ranker was proposed by Nogueira [9]. It has recently been demonstrated that this model yields state-of-the-art results in zero-shot scenarios [8]. We used a variant based on the multilingual version of T5 called mT5, which was pre-trained on the multilingual mC4 dataset. The mt5-13b-mmarco-100k has been already fine-tuned for re-ranking, using the mMARCO dataset, a multilingual version of the MS MARCO passage ranking dataset that comprises 13 languages and was created using machine translation.

The mT5-based re-ranker computes the relevance (as a probability) of each passage for a given query. After all passages are scored, the list of results is re-ordered according to those scores.

V. Results

Using the validation, test, and train PolEval datasets and our own (the translated MS MARCO dataset), we present the various approaches toward a specific information retrieval challenge: the problem of passage retrieval where the sentence-

[8] https://github.com/facebookresearch/faiss

[9] https://huggingface.co/sdadas/polish-roberta-base-v2

[10] https://huggingface.co/sdadas/polish-roberta-large-v2

[11] https://github.com/UKPLab/sentence-transformers/blob/master/examples/training/ms_marco/train_bi-encoder_mnrl.py

[12] https://huggingface.co/unicamp-dl/mt5-13b-mmarco-100k

TABLE I
EVALUATION OF DIFFERENT RE-RANKERS THAT HAVE THE SAME CLASSICAL, LEXICAL RETRIEVER: BM25, AND TOP-K=100 RESULTS RETRIEVED FROM BM25. NDCG METRICS WERE CALCULATED ON THE POLEVAL VALIDATION DATASET.

Method name	Retriever	Re-ranker	NDCG@10(%)
baseline	BM25 (default)	None	21.05
baseline@plugged	BM25 (morfologik)	None	27.67
bi-enc@base	BM25 (morfologik)	Bi-encoder (RoBERTa-base)	38.03
gpt3@curie	BM25 (morfologik)	GPT3 (curie)	40.06
bi-enc@large	BM25 (morfologik)	Bi-encoder (RoBERTa-large)	41.19
mT5@base	BM25 (morfologik)	mT5-base-mmarco	42.87
mT5@xxl	**BM25 (morfologik)**	**mT5-13b-mmarco**	**45.88**

TABLE II
EVALUATION OF DIFFERENT RETRIEVERS' SETTINGS (SPARSE—BM25, AND TWO DENSE RETRIEVERS BASED ON BI-ENCODER ARCHITECTURE, AND TWO TYPES OF ROBERTA MODELS) OF OUR SOLUTION. THE NDCG METRICS WERE CALCULATED ON THE POLEVAL TEST-A DATASET, THE FIRST RELEASED DATASET THAT WAS USED FOR PUBLIC LEADERBOARD PURPOSES.

Top@K sparse retriever – BM25	Top@K dense retriever – bi-encoder (RoBERTa-base)	Top@K dense retriever – bi-encoder (RoBERTa-large)	NDCG@10(%)
1	28	28	73.71
7	50	0	74.61
7	0	50	74.71
7	**25**	**25**	**75.32**
7	45	45	74.83

level queries are given and the corpus is counted in millions of passages.

First we evaluated various re-ranking methods on the PolEval validation dataset. In table I, we present some of the results, where we have one fixed retriever (BM25) plugged with the morfologik analyzer to introduce Polish lemmatization[13]. We tested the following re-rankers:

1) a bi-encoder based on RoBERTa-base, fine-tuned for one epoch on approximately 500 000 triplets, combining 135 000 PolEval training triplets and randomly selected triplets from the Polish MS Marco dataset;
2) a bi-encoder based on RoBERTa-large, initially fine-tuned for 10 epochs on a few million triplets (several million Polish MS Marco triplets randomly selected), and next fine-tuned for one epoch on 500 000 triplets, combining 135 000 PolEval training triplets and not-yet-used triplets from the Polish MS Marco dataset;
3) a generative model: GPT3 (curie), fine-tuned for one epoch on approximately 200 000 triplets, combining 135 000 PolEval training triplets and randomly selected triplets from the Polish MS Marco dataset;
4) a generative model: mT5-base-mmarco, fine-tuned for one epoch on 135 000 PolEval training triplets (Polish MS Marco was omitted because it is covered semantically by the mMARCO multi-language dataset);
5) a generative model: mT5-13b-mmarco, no fine-tuning.

As presented in Table I, the best results were achieved by mT5-13b-mmarco, an mT5-XXL model, fine-tuned on mMARCO (the multi-language version of MS Marco). Impressive results were achieved in the NeuCLIR track of TREC 2022 by the Unicamp team [14]. Despite the mT5 model being fine-tuned only on query–document pairs of the same language, it proved to be viable for cross-lingual information retrieval tasks, where query–document pairs are in different languages. The results in [14] demonstrate outstanding performance across all tasks and languages. For that reason, we used the same model for the Polish language—even though the mMARCO dataset does not contain Polish. We attempted to fine-tune the mT5-13b-mmarco model using the PolEval training dataset, but this process demanded very expensive infrastructure: at least four GPUs with high RAM capacity (such

[13] https://github.com/allegro/elasticsearch-analysis-morfologik

as the A100 80GB), and $ddp\_sharded$ strategy during fine-tuning to shard the entire model weights across all available GPUs. This approach enables the model's size to be scaled while using efficient communication to reduce overhead. Our initial experiments toward distributed fine-tuning failed to demonstrate any statistically significant improvement. This means that we could have trained it better. Ultimately, our solution used the original mT5-13b-mmarco, without fine-tuning on the PolEval datasets.

After selecting the best re-ranker, we analyzed how to improve the retrieval phase. After a number of experiments, we realized that bi-encoders as dense retrievers complement the BM25 results. Table II presents the evaluation of different retrievers' settings (sparse—BM25, and two dense retrievers based on bi-encoder architecture, as well as two types of RoBERTa models) in our solution. The results suggest that all three retrievers are needed. The first three rows of the table present different distributions of all 57 results sent for re-ranking: a) in the first row, we almost eliminated the BM25 results; b) in the second and third rows, we used only one dense retriever; c) we used seven results from BM25, and 25 from each of the dense retrievers; and d) we attempted to verify whether more dense retrievers results would help the results. Results a) and b) prove that all of of the retrievers are neeeded; result d) proves that increasing the number of dense retrievers by too many fails to improve the final NDCG metric.

VI. Conclusion

The goal of the PolEval 2022 Passage Retrieval task was to develop a system for cross-domain question answering retrieval in the Polish language.

The participants were given a training set that comprised question–passage pairs from the general knowledge domain, as well as three separate test sets with unpaired questions and passages from different domains: general knowledge, legal matters, and customer support. For each test question, participants were tasked with retrieving ordered lists of the ten most relevant passages from the provided corpora. The systems were scored using the NDCG metric.

Our solution was a multi-stage neural information retrieval system. The first stage involved a candidate passage retrieval step in which passages were retrieved using federated search over sparse (BM25) and dense indices (two FAISS indices built using dense retrievers based on bi-encoder architecture and Polish RoBERTa models). The second stage involved a re-ranking step of the previously-selected passages with a neural model, mt5-13b-mmarco. The model scored each passage by its relevance for a given query. The top-scoring passages were then retained as the final result.

Our system achieved second place in the competition. The results between top three entries did not differ significantly, and the quality of the solutions was high.

Acknowledgment

We thank Dr. Slawomir Dadas for his support—particularly in the bi-encoder training, and the Polish translation of the MS MARCO dataset.

References

[1] S. Robertson, H. Zaragoza *et al.*, "The probabilistic relevance framework: Bm25 and beyond," *Foundations and Trends® in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.

[2] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [Online]. Available: https://openreview.net/forum?id=wCu6T5xFjeJ

[3] R. Nogueira, J. Lin, and A. Epistemic, "From doc2query to docttttt-query," *Online preprint*, vol. 6, 2019.

[4] J. Lin, D. Alfonso-Hermelo, V. Jeronymo, E. Kamalloo, C. Lassance, R. Nogueira, O. Ogundepo, M. Rezagholizadeh, N. Thakur, J.-H. Yang *et al.*, "Simple yet effective neural ranking and reranking baselines for cross-lingual information retrieval," *arXiv preprint arXiv:2304.01019*, 2023.

[5] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[6] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. Bennett, J. Ahmed, and A. Overwijk, "Approximate nearest neighbor negative contrastive learning for dense text retrieval," *arXiv preprint arXiv:2007.00808*, 2020.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[8] G. M. Rosa, L. Bonifacio, V. Jeronymo, H. Abonizio, M. Fadaee, R. Lotufo, and R. Nogueira, "No parameter left behind: How distillation and model size affect zero-shot retrieval," *arXiv preprint arXiv:2206.02873*, 2022.

[9] R. Nogueira, Z. Jiang, R. Pradeep, and J. Lin, "Document ranking with a pretrained sequence-to-sequence model," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020. doi: 10.18653/v1/2020.findings-emnlp.63 pp. 708–718. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.63

[10] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated machine reading comprehension dataset," *choice*, vol. 2640, p. 660, 2016.

[11] L. Bonifacio, V. Jeronymo, H. Q. Abonizio, I. Campiotti, M. Fadaee, R. Lotufo, and R. Nogueira, "mmarco: A multilingual version of the ms marco passage ranking dataset," *arXiv preprint arXiv:2108.13897*, 2021.

[12] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv preprint arXiv:2008.00401*, 2020.

[13] T. S. Almeida, T. Laitz, J. Seródio, L. H. Bonifacio, R. Lotufo, and R. Nogueira, "Neuralsearchx: Serving a multi-billion-parameter reranker for multilingual metasearch at a low cost," *arXiv preprint arXiv:2210.14837*, 2022.

[14] V. Jeronymo, R. Lotufo, and R. Nogueira, "Neuralmind-unicamp at 2022 trec neuclir: Large boring rerankers for cross-lingual retrieval," *arXiv preprint arXiv:2303.16145*, 2023.

DOI: 10.15439/2023F3900
ISSN 2300-5963 ACSIS, Vol. 35

Multi-index Retrieve and Rerank with Sequence-to-Sequence Model

Konrad Wojtasik
Wrocław University of Science and Technology
Email: konrad.wojtasik@pwr.edu.pl

***Abstract*—This paper presents the solution to PolEval 2022 Task 3: Passage Retrieval. The main goal of the task, was to retrieve relevant text passages for the query. There were three different domains of passages: wikipedia passages, allegro faq and legal documents. The proposed solution incorporated both dense and lexical indexes, as well as, reranking model and reached 67.44 NDCG@10 score in official evaluation.**

I. Introduction

INFORMATION retrieval is a task that aims at finding relevant information from a collection of documents. It involves searching through the whole collection based on a query provided by a user. The query is usually a question, and the system has to provide the documents or text fragments that contain the answer to the question.

The landscape of existing information retrieval datasets is predominantly populated by English language resources. This presents a challenge for the development and evaluation of models designed to handle a variety of languages. In response to this, the BEIR benchmark[11] was developed, a unique benchmark that focuses on the evaluation of information retrieval models in a zero-shot setting. The primary training dataset utilized in this benchmark is MS MARCO[8], a large-scale dataset in English, while a diverse array of other datasets are employed for zero-shot evaluation.

Recognizing the need for multilingual resources, a multilingual version of MS MARCO[1] was created. This version was translated into various languages using state-of-the-art automated machine translation techniques, expanding the reach of the dataset beyond English. However, it was observed that the Polish language was conspicuously absent from this collection.

In an effort to fill this linguistic gap and foster the development of Polish language information retrieval models, a Polish version of the dataset was introduced in the BEIR-PL benchmark[13], as well as Massive Automatically-created Polish Question Answering Dataset[10], which is a large collection of question and passage pairs.

The main goal of PolEval 2022 Information Retrieval task was to propose a cross-domain question-answering retrieval system in Polish language. The task encompassed three distinct domains of queries and documents. The training set was exclusively composed of data related to the trivia domain. The other domains, namely legal and customer support, were approached from a zero-shot perspective.

II. Related Work

The most common approach to the information retrieval systems is incorporating the two-step retrieval process with reranking. With this two-step process, the information retrieval system can provide more accurate and relevant results, enhancing the overall effectiveness.

A. Retrieval

In the retrieval phase of information retrieval, the system matches a user's query with the indexed collection of documents to identify the initial set of relevant items. It should be fast and efficient, as the collections may contain millions of documents.

One way to perform retrieval is lexical matching. Usually, it utilizes the Best Matching 25 (BM25), which compares the frequency of terms in the query and the document. BM25 is a ranking function used by search engines to estimate the relevance of a document to a given search query based on the terms it contains.

Elasticsearch[1], an open-source search engine, is a popular implementation of this approach. It uses BM25 as its default scoring function. This method has become a standard baseline approach for most retrieval benchmarks due to its good performance, effectiveness and lack of training.

Recent trends indicate that neural retrievers are capable of surpassing the performance of lexical term matching[6]. The neural network, based on pre-trained transformer model, encode both the query and the document into a low-dimensional space. The encodings are compared using inner product or cosine similarity. By pre-encoding the corpus into the index, retrieval can become very efficient and run online with millisecond level latency with libraries that support similarity search of dense vectors, such as FAISS[2]. Neural retrievers are trained as bi-encoders with contrastive loss, which makes the representations of passages and queries with the same information similar.

B. Reranking

The reranking phase is a subsequent process that follows the initial retrieval. It involves reordering the retrieved documents based on more complex models or additional features to

[1] https://www.elastic.co/

[2] https://github.com/facebookresearch/faiss

improve the ranking and elevate the most relevant documents to the top positions in the ranked list. The reranking problem can be formulated as a classification problem, where the query and the document are passed jointly to the model and the result is a binary classification, if the document is relevant for the query or not. Pretrained transformer models, like BERT model, can be effectively utilized for this task. The BERT model[9], trained using cross-entropy loss, is capable of performing the classification based on the representation of the [CLS] token. Additionally, a single-layer neural network is employed to compute the probability of a passage's relevance to a given query. This approach leverages the power of transformer architectures to capture complex semantic relationships in the data, thereby enhancing the accuracy of the classification task. Encoder-decoder models, such as T5, have also been employed for the reranking task. In the context of classification tasks, the tokenizer is augmented with two unique tokens. The first token, representing a 'true' value, is generated when the text passage demonstrates relevance to the query in question. On the other hand, the second token, denoting a 'false' value, is generated in instances where the passage does not exhibit relevance to the query.

III. Dataset

Data domains in PolEval 2022 Information Retrieval task:

- trivia domain - knowledge based general questions from a popular quizzes and passages from Wikipedia pages. The corpus contains 7M passages.
- customer support domain - FAQ based customer questions from the allegro.pl platform. The corpus includes 921 passages.
- legal domain - dataset was constructed based on legal documents, with questions formulated around the content of these documents. The corpus comprises a total of 26287 passages.

The training set contained only data related to the trivia domain, other domains were treated as zero-shot approach.

IV. Solution

The final solution incorporates three dense indexes, where documents are embedded with different neural encoders and one lexical BM25 index. The reranking is performed by the multilingual T5 model.

A. Experiment Setup

Most of the experiments were performed on NVIDIA RTX 3090 with 24GB GPU memory, except the final submission with mT5-13B model, which was run on A100 with 80GB GPU memory, due to the model size and computational complexity.

B. Experiments

The system was constructed from a combination of various dense indexes, each created with different models, as well as a lexical index created using the BM25 algorithm. From each index, the top documents were retrieved, and the collective set of all these documents was then forwarded to the reranking model for further refinement.

TABLE I
Retrievers results on TEST A without reranking. Combined* represents a score of all taken set of all returned passages from all retrievers at top k.

Retriever	NDCG@10	Recall@10	Recall@100	Recall@1000
BM25	50.77	37.55	45.29	51.65
mContriever	**58.43**	**56.03**	**70.05**	**77.93**
LaBSE	29.84	32.01	50.59	62.87
mDPR	31.42	33.95	51.32	66.09
Combined*	-	*63.84*	*75.15*	*81.34*

Experimental results shown in the table I demonstrated that, the best performance achieve mContriever dense retriever. Other dense retrievers got lower NDCG@10 and Recall@10 metrics than BM25, but they achieve better recall score at higher number of retrieved passages. Combined* results show recall when all top passages are combined from all four retrievers. In practice, due to duplicates, there are on average 3 times of the k passages. So for Recall@10, there were on average 30 passages taken into account for each query.

In the final solution, the following three dense retriever models were employed:

- mContriever-base-msmarco [3] - mBERT[7] based retriever trained in unsupervised manner with contrastive loss on multilingual data and afterward fine-tuned on English MS MARCO dataset[5].
- LaBSE [4] - is a language-agnostic BERT model for sentence embedding[3].
- mDPR-question-nq [5] and mDPR-passage-nq [6] - mDPR is a multilingual Dense Passage Retriever[15], a bi-encoder network where query and passage are encoded with different encoders trained in contrastive manner.

For reranking stage, different rerankers were taken into account. Initial testing were performed with following rerankers fine-tunned on MS MARCO dataset to reranking task:

- mMiniLMv2 [7] - multilingual MiniLMv2 model[12].
- mDeBERTa - improved multilingual DeBERTa model[4].
- mBERT[8] - multilingual BERT model[7].
- plT5[9] - T5-based language model train on Polish corpora[2]. Fine-tunned on Polish MS MARCO from BEIR-PL.
- mT5[10] - multilingual text-to-text transformer[14].

As shown in the table II, the bigger model, the better result. That is why, in the final solution, the mT5 model[11] with 13 billion parameters was employed. This model is very

[3] https://huggingface.co/nthakur/mcontriever-base-msmarco

[4] https://huggingface.co/sentence-transformers/LaBSE

[5] https://huggingface.co/castorini/mdpr-question-nq

[6] https://huggingface.co/castorini/mdpr-passage-nq

[7] https://huggingface.co/cross-encoder/mmarco-mMiniLMv2-L12-H384-v1

[8] https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco

[9] https://huggingface.co/clarin-knext/plt5-base-msmarco

[10] https://huggingface.co/unicamp-dl/mt5-3B-mmarco-en-pt

[11] https://huggingface.co/unicamp-dl/mt5-13b-mmarco-100k

TABLE II
RERANKER RESULTS ON TEST A RERANKING TOP 1000 BM25 RESULTS.

Reranker	NDCG@10 TEST A
plt5-base	66.02
plt5-large	69.09
mMini-lm	67.87
mDeberta	68.67
mBert	59.78
mT5-3B	**70.28**

TABLE III
FINAL RESULT ON TEST A AND TEST B.

	NDCG@10 TEST A	NDCG@10 TEST B
Final solution	**74.28**	**67.44**

computationally expensive to run, that is why there are no additional experiments performed on this model.

The final solution was achieved by retrieving the top 100 passages for each query from each dense retriever, and the top 100 reranked passages using the plT5-large model from the top 1000 retrieved from the BM25 index. The use of plT5 was dictated by the computational complexity of the mT5-13B model. As shown in table I, combined retrievers achieve already very high 75.15 recall at top 100 passages from each retriever.

Subsequently, the set of all top 100 passages from all sources was reranked using the mT5-13B model. The total number of passages to rerank was approximately 310 instead of 400, as some passages were duplicated. The final results are shown in the table III.

V. CONCLUSION

The optimal strategy to enhance the results of information retrieval involves the utilization of various dense retrievers, in conjunction with lexical BM25 matching. This approach amplifies the overall recall of the system, ensuring that passages not deemed relevant by one model may be identified as such by another. Another key insight is the importance of employing an effective reranker. The performance of the reranker is often correlated with the size of the model, with larger models typically achieving superior scores compared to their smaller counterparts. However, this advantage is accompanied by an increase in computational cost, which must be taken into account. Furthermore, reranking a larger number of top retrieved passages enhances the likelihood that a relevant passage is included in the set of reranked passages. This strategy, while potentially more computationally intensive, can significantly improve the precision of the retrieval system at the top ranks, which is often a critical requirement in many information retrieval applications.

ACKNOWLEDGMENT

The work was financed by the European Regional Development Fund as a part of the 2014-2020 Smart Growth Operational Programme, CLARIN - Common Language Resources and Technology Infrastructure, project no. POIR.04.02.00-00C002/19.

REFERENCES

[1] Luiz Henrique Bonifacio, Israel Campiotti, Roberto de Alencar Lotufo, and Rodrigo Nogueira. mmarco: A multilingual version of MS MARCO passage ranking dataset. *CoRR*, abs/2108.13897, 2021.

[2] Aleksandra Chrabrowa, Łukasz Dragan, Karol Grzegorczyk, Dariusz Kajtoch, Mikołaj Koszowski, Robert Mroczkowski, and Piotr Rybak. Evaluation of transfer learning for polish with a text-to-text model. *arXiv preprint arXiv:2205.08808*, 2022.

[3] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[4] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543, 2021.

[5] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021.

[6] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, November 2020. Association for Computational Linguistics.

[7] Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. How language-neutral is multilingual bert?, 2019.

[8] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. November 2016.

[9] Rodrigo Frassetto Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[10] Piotr Rybak. MAUPQA: Massive automatically-created Polish question answering dataset. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 11–16, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[11] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *CoRR*, abs/2104.08663, 2021.

[12] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *CoRR*, abs/2012.15828, 2020.

[13] Konrad Wojtasik, Vadim Shishkin, Kacper Wołowiec, Arkadiusz Janz, and Maciej Piasecki. Beir-pl: Zero shot information retrieval benchmark for the polish language, 2023.

[14] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020.

[15] Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. Towards best practices for training multilingual dense retrieval models, 2022.

DOI: 10.15439/2023F586
ISSN 2300-5963 ACSIS, Vol. 35

Passage Retrieval in question answering systems in Polish language

Anna Pacanowska

Abstract—**This paper describes the submissions to Task 3 of PolEval 2022. Passage Retrieval is a problem of retrieving a passage relevant to the given query. It is an important problem with many practical use cases, especially in question answering. It is very beneficial if a model is generalizable, that is effective in various domains, even the ones it was not trained on. This is a challenge for many state-of-the-art models. In this paper I describe and test many different methods of approaching this problem – from standard techniques, such as BM25 and lemmatization to recently developed methods based on deep learning and transformers.**

I. Introduction

The aim of Task 3 of PolEval 2022 was to find a tool that retrieves the passages that contain the answer to the given question. The tool should work well not only on the domain it was trained on, but also other domains. The passages and queries are in Polish. I tested and compared different techniques and their combinations in order to find out which one will be the most effective – from the standard statistical approaches to advanced neural network based solutions. My goal was to find a solution that is not only effective, but also does not require great amount of resources to use – it should be memory and time efficient.

The best solution consisted of two stages. In the first I retrieved 1000 candidate passages using BM25 from Elasticsearch[1] on a corpus and queries lemmatized using Morfeusz2 [3]. In the second stage I calculated different scores and joined them using logistic regression. The scores were BM25 on lemmatized texts (with Elasticsearch), on unlemmatized texts and on bigrams. The rest of the scores were calculated using miniLM [13] on the texts translated with OPUS-MT [12]. They were computed on pairs (question, passage), (generated_answer, extracted_answer), (generated_answer question, extracted_answer passage). The answers were generated with GPT3 [1] and extracted with DistilBERT [10]. The code is available on GitHub[2].

II. Task

A. Description

The model's task was to retrieve 10 passages that were most relevant to the given question. To encourage generalizability the data was split into three distinct domains: Wikipedia passages, legal articles and Allegro FAQ. All training and development data came from the first domain and the other domains were present only in the test dataset.

TABLE I
Dataset sizes and number of questions from each domain

domain	questions source	passages source	avg length	corpus size
wiki-trivia	Jeden z dziesięciu	Wikipedia	44	7097322
legal-questions	generated from passages	legal acts	153	26287
allegro-faq	FAQ	help articles	48	921

B. Domains

Each domain contains a separate corpus of passages and different kinds of questions. The domains are very diverse – the samples in each of them have unique characteristics that influence the solution. They vary in passage length, corpus size, question types, number of matches and other important details (Table I).

1) wiki-trivia: The questions come from "Jeden z dziesięciu" and are classical trivia questions. They have short, factoid answers and require only common knowledge. The passages are fragments of articles from Polish Wikipedia extracted using WikiExtractor. The passages are quite short, but the corpus is very large: it contains over 7 million passages. The questions were created first, and the passages were independently matched later. That means that the answer is usually worded in a different way. Sometimes it requires good understanding of the text to notice the passage does in fact contain the answer. Usually there are multiple relevant passages to each question.

2) legal-questions: The passages are Polish legal acts. The questions were artificially generated from passages by people – a volunteer first looked at the provision, and then generated a question for it. This means that, unlike in wiki-trivia, the answers are usually similarly worded and the passage contains a direct answer. The language is quite heavy and contains specialist vocabulary. Answering the questions requires in-depth knowledge of Polish law, not only common knowledge. The questions sometimes are ambiguous and make sense only in the context of the passage they were generated from, for example "Kto sprawuje nadzór nad Akademią?" (which Academy?). The passages are often very long, but the corpus is significantly smaller with about 26 thousand passages.

3) allegro-faq: The questions and passages are fragments of FAQ and help articles from Allegro. There are a lot of 'How to' questions. These questions are often open-ended, there are multiple possible responses that can be worded in many different ways. The answers are usually specific to Allegro. For most questions there is only one matching passage. This is the smallest dataset with only 921 passages.

[1] https://www.elastic.co/
[2] https://github.com/aniapacanowska/passage-retrieval

C. Evaluation

The submissions were evaluated using NDCG@10 with binary relevance scores. The overall score was a mean of the NDCG@10 score for each question in the test set. For the test-A dataset only the total score was visible – the correct answers were not public. BM25 was the baseline solution provided by the task's authors.

III. Methods

In this chapter I describe all the methods I used in the experiments.

A. BM25

In the first experiments I tried to improve the baseline solution by using lemmatization. Lemmatization can be used to improve the performance of BM25 – similar technique was used in the best solution to the Question Answering task in 2021 PolEval [7]. There are many different models capable of lemmatizating texts. I tried four different options – Morfeusz2 [3], spaCy [2], a hybrid of these two and modified Morfeusz2.

Morfeusz2 is a dictionary-based morphological analyzer for Polish language. SpaCy is an open-source library for natural language processing. It features much more complex techniques, such as state-of-the-art neural networks.

One of major distinctions between these models is how they handle ambiguity (words with multiple possible lemmas). Morfeusz2 is not capable of choosing the correct lemma based on the context. In the ambiguous cases it simply provides all possibilities. At first I tried to take all of the provided lemmas. This may be beneficial for two reasons. There is no risk I will lose a potential match because of choosing the wrong lemma. The other benefit is that words in the same inflected form always have identical set of lemmas, so they will often result in multiple matches. That means they will be more valuable than words in different grammatical form that have only one matching lemma. This way the words in different forms will still count as a match, but less than ones with exactly the same inflection. On the other hand, it will also result in many false matches of lemmas that are incorrect in the given context.

SpaCy on the other hand always tries to pick the correct lemma. However, it is sometimes wrong and picks the wrong one. It can also return words that are not valid lemmas or even do not exist in the Polish language. Morfeusz2 never makes up invalid words – it has a very large dictionary, but in case the model encounters an unknown word it is returned unchanged.

The hybrid approach tries to get the advantages of both models. It uses spaCy to pick the correct lemma from the ones provided by Morfeusz2. If the spaCy result was not returned by Morfeusz2, it takes the most popular lemma in the corpus. The popularity was measured on the corpus lemmatized using only Morfeusz2. Lastly, I wanted to check if spaCy is really useful in the hybrid model. Perhaps simply always choosing the most frequent lemma could work just as well.

B. Bigrams

I tested also a variation of the BM25 scoring function in which the terms are bigrams instead of words. This might be useful, because a bigram match is a stronger indicator that the passage is relevant than a single word match. This is not a sufficient method on its own, but can be useful in addition to other scores.

C. Deep learning

The next step was to use more advanced and recent models. Each of the following experiments consisted of two stages. In the first stage I retrieved top 100 passages with standard BM25 from Elasticsearch on texts lemmatized with Morfeusz2 lemmatizer (the basic approach). This was necessary, because many of these models require a lot of computing power, so it would not be feasible to run them on each possible pair. The second stage was re-ranking the retrieved passages using the selected method.

D. BERT

The first approach was to utilize BERT in a way that was outlined in [6]. A passage and a query separated with SEP token are encoded by a pre-trained model. The CLS vector is passed to a simple classification layer to predict whether the passage is relevant to the query. The model is fine-tuned on this task. I wanted to find out whether this approach would work well for the wiki domain and if it would be capable of generalizing to other domains. I used HerBERT [4] as the pre-trained Polish model.

E. Translation

Most state-of-the-art architectures for passage retrieval are very large and training them requires a lot of time and computing power. For this reason I decided it would be beneficial to use a pre-trained model. However, I could not find a good model for passage retrieval in Polish. There is much greater choice of models for passage retrieval in English. I used the OPUS-MT model [12] from Huggingface[3] in order to translate all of my data into English. I tested if translating the data and using more powerful models would improve the results.

F. miniLM

MiniLM [13] is a small model trained using knowledge distillation with BERT-base. One of its variants was fine-tuned on MSMARCO passage ranking dataset [5] with the ensemble of BERT-base, BERT-large and ALBERT-large as teacher models. This version achieved the best results for most IR tasks according to the BEIR paper [11], which is why miniLM was my first choice. An additional benefit is that small number of parameters makes the inference really efficient. MiniLM calculates a relevance score for a pair of query and passage.

[3] https://huggingface.co/Helsinki-NLP/opus-mt-pl-en

TABLE II
LIST OF ALL SCORES USED (IN DIFFERENT COMBINATIONS) WITH LOGISTIC REGRESSION

Logistic regression features
BM25 (elasticsearch) for lemmatized texts
BM25 for unlemmatized texts
BM25 for bigrams (unlemmatized)
miniLM for (question, passage)
miniLM for (GPT3_answer question, DistillBERT_answer passage)
miniLM for (GPT3_answer, DistillBERT_answer)
miniLM for (chatGPT_answer question, DistillBERT_answerpassage)
miniLM for (chatGPT_answer, DistillBERT_answer)

G. Answer generation

There are two types of question answering models: extractive, which extract the answer from the given passage and generative, which generate the answer based only on the model's knowledge. Similarity between the answer generated by a generative model and the answer extracted from given passage could help miniLM decide whether the passage is relevant. If these answers are the same or similar it can be a good indicator that the passage is relevant. It can be misleading as well – the answer from either model may be incorrect or the same only by accident. I tested two variants. In the first I concatenated the generated and extracted answers to the beginning of the question and the passage, respectively. In the second I calculated the relevance score only between the answers.

As the extractive model I used DistilBERT [10], which is a distilled version of BERT. I tested two different generative models, both based on GPT: GPT-3 and chatGPT (based on GPT-3.5). These models are not open source – it was necessary to use the API provided by OpenAI to connect with them. Even if they were available I would not be able to run them locally as they are too large.

H. Ensamble

The last idea to improve the results was to gather the scores from multiple methods instead of relying on a single model. The test dataset is very different from the train set, so the function combining the scores had to be simple. I decided to use logistic regression.

IV. Experiments

A. BM25 with lemmatization

These experiments were conducted with the use of Elasticsearch. I created and indexed a lemmatized corpus of passages with each lemmatizer (Morfeusz2, spaCy, hybrid and Morfeusz2-freq). Next I lemmatized the questions and retrieved the most relevant passages using Elasticsearch. The results on the dev dataset are shown in the Table III.

Lemmatizing the texts significantly improved the results. The impact considerably depended on the chosen model. The score with Morfeusz2 and spaCy was similar, with spaCy slightly higher. Morfeusz2 is even 20 times faster (on CPU) than spaCy, which makes it much easier to use. This is because Morfeusz2 is dictionary-based and spaCy is a complex neural network.

TABLE III
DIFFERENT LEMMATIZATION METHODS (DEV DATASET)

method	NDCG@10
no lemmatization	18.62
spaCy	21.41
Morfeusz2	21.15
hybrid	24.47
Morfeusz2-freq	**25.24**

The hybrid approach clearly outperformed both models. Morfeusz2 almost always provides the correct lemma, but usually together with a few incorrect ones. SpaCy on the other hand sometimes returns incorrect or even invalid lemmas. The hybrid approach eliminates both these issues – Morfeusz2 is used to check if the spaCy lemma is valid. If it is not, the most popular lemma gets chosen.

Surprisingly, the last technique turned out to be the best. The lemma was picked solely based on its frequency in the corpus lemmatized with Morfeusz2. That means spaCy was not necessary at all in the hybrid approach – the alternative method of choosing the correct lemma worked even better.

All methods have problems with proper names (such as surnames or places). There are too many of them for any model to know, so they are not properly lemmatized. For example, the lemmas from texts 'Bilbo Baggins' and 'Bilba Bagginsa' will be different.

B. Deep learning

In the following experiments I tested approaches based on deep learning and transformers. The passages to be re-ranked were fetched using Elasticsearch on a corpus lemmatized with Morfeusz2 (basic approach). I worked with the Huggingface library [14]. The submissions are collected in the Table IV. I did not calculate the results of all models on the dev dataset, because it would unnecessarily take a lot of time and resources. Test-B data appeared in the last two weeks, so I tried only the models that did well on test-A.

The submissions were evaluated on the PolEval website. The other scores in this section were calculated with my script. There are slight differences (around 0.5) that are a result of different behavior on questions where the correct passage is repeated.

In the following sections I describe in detail all of the methods I used.

C. BERT re-ranking

The first method to re-rank the passages was to fine-tune a BERT model (3 in Table IV). The input consisted of a question (sentence A) and a passage (sentence B). On top of BERT there was a simple classification layer (BertForSequenceClassification head), which was trained to predict whether the input pair is relevant based on the CLS vector. I fine-tuned the HerBERT-base model on the train dataset for one epoch. The model did better than the previous methods on the dev

TABLE IV
SUBMISSIONS SENT TO POLEVAL. SIZE: NUMBER OF PASSAGES TO RE-RANK, COMBINATION: METHOD OF JOINING THE SCORES FROM DIFFERENT MODELS, LR-DEV: LOGISTIC REGRESSION TRAINED ON THE DEV DATASET, LR-DEV: LOGISTIC REGRESSION TRAINED ON THE TRAIN DATASET

Methods				NDCG@10		
id	models	combination	size	dev	test-A	test-B
1	Baseline (from PolEval)	-	-	-	50.76	-
2	BM25 (k=1.0, b=0.5)	-	-	19.59	48.82	-
3	BERT	-	100	24.5	17.03	-
4	miniLM	-	100	31.36	58.19	-
5	miniLM	lr-train	100	31.97	59.76	-
6	miniLM	lr-dev	100	-	60.17	51.55
7	miniLM, GPT3	lr-dev	100	-	60.97	52.15
8	miniLM, GPT3, chatGPT	lr-dev	100	-	56.18	48.69
9	miniLM, GPT3	neural network	100	-	53.64	-
10	miniLM, GPT3	-	100	-	58.45	-
11	miniLM, GPT3	lr-dev	1000	-	**62.51**	**54.23**
12	miniLM, GPT3 (selected)	lr-dev	1000	-	-	54.15
13	miniLM, GPT3, chatGPT	lr-dev	1000	-	-	51.82
14	miniLM	lr-dev	1000	-	-	53.20

dataset, but the results on the test dataset dropped three times compared to baseline. The model learned well for the wiki domain, but was completely unprepared for the legal and allegro datasets. It would suggest that this technique of BERT re-ranking generalizes very poorly to other domains.

D. Translation

For the next experiments I had to translate the data to be able to use the models trained in English. The maximum length of the input for OPUS-MT model is 512 tokens. There are many longer passages in the datasets (especially in the legal corpus), so they had to be split. I tried to avoid splitting the sentences – passing only a fragment of a sentence might confuse the translator. However, I noticed that OPUS-MT does not work well for long texts – even if they fit in the limit. It often translates only some fragment of the input and leaves out the rest. This was not problematic for Wikipedia passages, which are usually short, but started to be noticeable for longer texts. So I split the passages into even smaller pieces (at most 0.3*512) and translated each separately. Then I joined all translated fragments.

E. miniLM

The first model I tested on the translated data was miniLM[4] with 6 layers fine-tuned on MSMARCO. Again, there were some problems with passages that were too long to fit into the model (the input can be at most 512 tokens). I split the passages into overlapping smaller pieces. Each fragment starts in the middle of the previous one – that way I want to avoid a situation where part of the answer is in one fragment and the rest is in the other. The fragments are at most 0.7*512 words to take into account that some words translate into multiple tokens. When ranking the passage, its score is the maximum of its fragments' scores. Re-ranking based on the scores computed by miniLM made a significant improvement over the baseline (4 in Table IV).

F. Logistic regression

I tried to join the results from the models I tested so far. This way I could aggregate the information stored in different scores. The features I considered were:

- BM25 (elasticsearch) for lemmatized texts
- BM25 for unlemmatized texts
- BM25 for bigrams (unlemmatized)
- miniLM score

I used the logistic regression model from sklearn library [8] and trained it on the train dataset (5). This resulted in a small, but noticeable improvement. Surprisingly, even larger improvement (on the test dataset) was achieved by training on the dev dataset instead (6). It might be because the train dataset is over 7 times larger and logistic regression started overfitting to the wiki-trivia domain.

In the next experiments I continued to use logistic regression and simply added new scores as additional features.

G. Answer generation

The next step was to use the question answering models. I generated an answer for each question using a generative model and extracted the answer from each (question, passage) pair using an extractive model. I did not want to exceed the free quota of the generative models, so the answers were generated only for the dev and test datasets. I was able to fit all my requests into the free trial. I calculated the miniLM score for each pair ('generated_answer question', 'extracted_answer passage') as well as just (generated answer, extracted answer). These results were then added as additional features to the logistic regression (trained on the dev dataset). As the extractive model I used DistilBERT[5] fine-tuned on the SQuAD dataset [9].

1) GPT3: The first generative model I tried was GPT3 – specifically gpt3-davinci, which is described in the documentation as the 'most capable'. I used the API provided by OpenAI. Each prompt was based on the question translated into English.

Normally, the answers given by the model are quite elaborate and if it does not have the requested information, the answer is 'unknown'. I wanted the answers to be concise and informative – even if the model had no information, I wanted it to guess the answer. I decided the short answers are better, because they do not contain unnecessary descriptions which could confuse miniLM. This also significantly reduced the cost of each query, because the responses are priced by the number of tokens. The guessed answers, even if incorrect, still might contain some useful information, such as the type of the answer (a number, a name etc.). I modified the prompts in order to get the desired results. To each question I added "Shortest answer. NOT unknown.". Temperature was set to a low value of 0.1. This way the model would usually get the true answer if it had enough knowledge, but was able to guess if necessary.

[4] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2

[5] https://huggingface.co/distilbert-base-cased-distilled-squad

Adding these answers to the dataset and calculating the scores as described above resulted in another small improvement (7 in Table IV). By inspecting the answers I noticed that they are usually correct for wiki-trivia questions, but often wrong for questions from other domains. This makes sense, because the wiki questions are based on common knowledge. Other domains require specific knowledge of Polish law or the policies of Allegro. The questions can be open-ended, which also makes it more difficult to guess the correct answer.

2) ChatGPT: I tried also the new chatGPT model. This time the questions could be in Polish as well as in English because this model is multilingual. I decided to use the Polish ones because chatGPT is likely better in translation than OPUS-MT that I used until now. However, it can have a downside – the translations might be inconsistent. The same expression in Polish could be translated differently in the answer by chatGPT than in the passage by OPUS-MT.

As before, I wanted to receive a concise answer in English. Interesting thing is that unlike GPT3, chatGPT almost never answers ’unknown’. It is guessing even without the suggestion in the prompt I had to add before. For the system prompt I left the default "You are a helpful assistant". To the questions in the legal domain I added "W Polsce" and in allegro domain "W allegro" in hopes it would help the model answer correctly. I finished each user prompt with "Shortest answer in english". Surprisingly, it turned out that this addition not only did not improve the results, but in fact made them much worse (8).

H. Combining scores

I compared a few different methods of combining the scores I gathered so far. The method should be quite simple to avoid overfitting to only wiki domain. I tried training a simple two layer neural network for classification. This turned out to be much worse than logistic regression (9). Ranking only by miniLM score with GPT3 answers also gave considerably worse results (10).

I. Bigger data

Sometimes the relevant passages are not included in the top 100 retrieved using BM25. The number of relevant passages found in the different positions in the ranking created using BM25 scores can be found in the Table V. The results were calculated using the training dataset and different lemmatizers.

The passages that fall outside of the top 100 will of course never be found by the re-ranker. On the other hand, adding a lot of new possible passages can result in false positives. I repeated some of the previous experiments on the top 1000 passages (retrieved by Elasticsearch with Morfeusz2, as before). In every case the outcome was considerably better. Re-ranking top 1000 passages with miniLM and GPT3 combined with logistic regression resulted in the best score I was able to achieve (11).

The main downside of this solution is a much larger computational cost. The most expensive task is translation, which was a problem especially for wiki domain that has the largest corpus. Another problem was that Elasticsearch sometimes retrieves less passages than requested when the corpus is small compared to the requested size. It is especially visible for allegro-faq, where there are only 921 passages, but Elasticsearch often retrieves even less. This happened for the top 100 as well, but rarely.

TABLE V
NUMBER OF RELEVANT PASSAGES IN DIFFERENT RANKING FRAGMENTS

lemmatization	:10	11:100	101:1000	1001:10000	10001:
none	2918	2755	2648	2078	4049
spaCy	3447	3476	2930	2023	2572
Morfeusz2	3385	3341	3044	2181	2497
hybrid	3952	3841	3064	1897	1694
Morfeusz2-freq	4073	3863	3091	1839	1582

I tried to simplify the re-ranking method and pass only the features that seemed most important: unlemmatized BM25 score, bigrams BM25 score, miniLM score and miniLM score on pairs concatenated with answers. This turned out to be only minimally worse than providing all scores (12).

J. Incorrect answers

In this section I analyze the incorrect answers given by the best model. The data is far too large to fully analyze it manually, so these are only some observations. The model always provides exactly 10 passages even though the number of correct ones is smaller. This means that retrieving a passage without the answer is not an error as long as it is lower in the ranking than the relevant passages. The incorrect passages are often relevant to the topic, but do not contain the answer to the given question. Sometimes the passage describes something similar, but not the same – for example the results for the question about the host of the show ’Zrób to sam’ talk about the hosts of shows such as ’Sam tego nie rób’.

There are also some errors in the annotations. Some passages that were correctly retrieved by the model were not marked as relevant. There were also cases in which the annotated passage did not contain the answer. Other times it did have the answer, but without the necessary context. For example for the question "In which book Adam Mickiewicz describes Jankiel’s concert?", one of the annotated passages says only that Jankiel is a character in "Pan Tadeusz" movie – does not mention the concert or the book’s author. This does not happen only in the allegro domain because there the passages were verified manually. However, I think that despite these problems the datasets are still useful in measuring the performance of the models.

K. PolEval results

The final results are in the Table VI. My models clearly outperformed the baseline. The score of my best model is around the middle between the basic BM25 and the best solution of PolEval 2022.

After the contest ended, the answers for the test datasets were published. I compared the performance of my best model and the baseline on different domains (Table VII). It turned out that wiki domain was clearly the most difficult. The first reason

TABLE VI
FINAL RESULTS

model	test-A	test-B
baseline BM25 (Elasticsearch)	50.38	38.84
my best model	62.51	54.23
PolEval best solution	75.40	69.36

TABLE VII
SCORES ON DIFFERENT DOMAINS

	test-A			test-B		
model	wiki	legal	allegro	wiki	legal	allegro
baseline BM25 (Elasticsearch)	19.76	81.10	49.16	18.45	80.32	48.05
my best model	38.27	77.70	69.96	37.11	79.00	67.92

for that is a very large corpus (over 7 million passages). The other can be that the passages and the questions were created independently and matched later, so the answers are often indirect or differently worded. The score on allegro domain was much better, probably because of a very small number of possible passages (only 921). On both of these domains my model was much better than BM25. The legal domain was definitely the easiest. I believe it is because the questions were generated based on the passages, so the wording was usually very similar. Additionally, the questions often contained some unique words that pointed to a certain passage. It is the only domain where the new solution was slightly worse. The differences between the domains explain the lower score on test-B, where the majority of passages came from the wiki domain.

V. CONCLUSIONS

In this article I have explored various methods of passage retrieval, both traditional statistical approaches as well as recent models based on transformers.

The standard statistical methods, such as BM25 are a good starting point. They can be improved by means such as lemmatization. The choice of the lemmatizer largely impacts the performance. These methods don't have a problem with generalizability, because they are independent of the domain. Deep learning approaches are definitely more capable, but it comes at the cost of a much higher computational complexity. The most optimal way to use them is together with less accurate, but faster statistical methods. A good approach is to fetch a subset of passages with BM25 and then re-rank it with an advanced model, such as miniLM. Retrieving more passages to re-rank improves the results, but significantly increases the necessary resources, so it is important to find a balance. Knowledge distillation is an incredibly useful technique to reduce the computational cost of using a model. Translation is a good method when there are no models pre-trained in the correct language. It can be a great alternative to training a model from scratch, especially with limited resources.

Question answering is another good way to boost the results. Generative models are powerful, but expensive to use. However, they need to be used only once per question, unlike other methods that need to calculate a score for each pair of passage and query. It is important to choose the right model – GPT3 answers helped the re-ranking, but chatGPT did the opposite. Crafting a good prompt matters as well.

The best result was achieved by joining different techniques using logistic regression. Even a model that is less accurate on its own can still be useful to increase the score of a better model.

VI. NOTE

This paper is taken from my master's thesis written under the direction of dr Paweł Rychlikowski at the University of Wrocław.

REFERENCES

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[2] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020.

[3] Witold Kieraś and Marcin Woliński. Morfeusz 2 – analizator i generator fleksyjny dla języka polskiego. *Język Polski*, XCVII(1):75–83, 2017.

[4] Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. HerBERT: Efficiently pretrained transformer-based language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10, Kiyv, Ukraine, April 2021. Association for Computational Linguistics.

[5] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human-generated MAchine reading COmprehension dataset, 2017.

[6] Rodrigo Frassetto Nogueira and Kyunghyun Cho. Passage re-ranking with BERT. *CoRR*, abs/1901.04085, 2019.

[7] Maciej Ogrodniczuk and Łukasz Kobyliński, editors. *Proceedings of the PolEval 2021 Workshop*, Warsaw, Poland, 2021. Institute of Computer Science, Polish Academy of Sciences. pg. 123-140.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[11] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[12] Jörg Tiedemann and Santhosh Thottingal. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal, November 2020. European Association for Machine Translation. https://aclanthology.org/2020.eamt-1.61.

[13] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

[14] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing, 10 2020.

FedCSIS 2023 Challenge: Cybersecurity Threat Detection in the Behavior of IoT Devices

Andrzej Janusz

Marcin Michalak

CYBERSECURITY Threat Detection in the Behavior of IoT Devices was the 9th competition organized in association with the FedCSIS conference series at KnowledgePit.ai. The goal was to detect attacks on IoT devices on the basis of data provided by Efigo company, describing the changing behavior of devices, and known moments of cyberattack attempts. The data set was generated as a part of a SPINET project aiming at improving the cybersecurity of IoT device networks.

The increasing significance of many IoT device applications motivates scientists to develop techniques for cyber safety improvement in many ways. In our case, based on the changing device behavior, it was expected that the profile of processes running on the device should change during the attack attempts.

The competition data was collected in a simulated environment - IoT devices were emulated in a separated network, where attacking servers were also plugged in. The scenario of attacks was known, so it was possible to tag the behavioral data as “normal” and “unusual”. Based on that information participants tried to develop classification models to predict whether the device is being attacked or not.

The top four competitor groups were invited to submit a paper describing their solutions to our special event at the FedCSIS 2023 conference. These papers are included in this chapter of the conference proceedings and are preceded by a paper describing in detail the competition, authored by the organizers. The most of presented approaches were based on gradient-boosting algorithms. That is not surprising - such models play an essential role in different fields of application. However, they are not so easy to interpret which may cause difficulties in better understanding the nature of IoT devices’ behavior change during cyberattacks.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1289–1293

DOI: 10.15439/2023F3089
ISSN 2300-5963 ACSIS, Vol. 35

Cybersecurity Threat Detection in the Behavior of IoT Devices: Analysis of Data Mining Competition Results

Michał Czerwiński[†‡], Marcin Michalak[§‖], Piotr Biczyk[†**], Błażej Adamczyk[¶‖], Daniel Iwanicki[†], Iwona Kostorz[§],
Maksym Brzęczek[¶], Andrzej Janusz[†‡], Marek Hermansa[§], Łukasz Wawrowski[§], Artur Kozłowski[§]

[†]QED Software, Mazowiecka 11/49, 00-052 Warsaw, Poland
Email: {michal.czerwinski, piotr.biczyk, daniel.iwanicki, andrzej.janusz}@qed.pl
[§]Łukasiewicz Research Network - Institute of Innovative Technologies EMAG, Leopolda 31, 40-189 Katowice, Poland
Email: {marcin.michalak,iwona.kostorz,marek.hermansa,lukasz.wawrowski,artur.kozlowski}@emag.lukasiewicz.gov.pl
[¶]EFIGO sp. z o.o., M.Kopernika 8/6 , 40-064 Katowice, Poland
Email: {blazej.adamczyk, maksym.brzeczek}@efigo.pl
[‡]Faculty of Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
Email: {m.czerwinski4}@uw.edu.pl, {a.janusz}@mimuw.edu.pl
{[‖]Department of Computer Networks and Systems, ** Faculty of Automatic Control, Electronics and Computer Science}
Silesian<file://Silesian> University of Technology, Akademicka 16, 44-100 Gliwice, Poland
Email: {marcin.michalak, blazej.adamczyk}@polsl.pl, pbiczyk@gmail.com<mailto:pbiczyk@gmail.com>

***Abstract*—The paper discusses a data science competition centered around the development of an anomaly detection system for IoT devices. The competition utilized a unique environment that allowed for the operation and monitoring of real IoT devices, including scheduling of attacks on these devices. The environment was used to collect the data, which included both normal and attack-induced behavior of IoT devices. The paper presents the background of the competition, the top models submitted, and the competition results. The paper also includes a discussion about restrictions related to the use of synthetic attack data as input for constructing anomaly detection systems.**

***Index Terms*—data science competitions; KnowledgePit.ai platform; cybersecurity; ML applications in log analysis; ML data quality**

I. Introduction

THE INCREASING number of Internet of Things (IoT) devices being used in present times implies the need to pay attention to ensure a proper level of their safety. The market review indicates that there is a lack of products that can increase the security of IoT devices while reducing the risk of successful attacks. This conclusion led to the idea of an IoT-dedicated system for detecting anomalies, which could be the result of an attack.

The consortium of EMAG, QED, and EFIGO runs a project focused on IoT devices cybersecurity – SPINET. Within that project, an environment that allows running and monitoring real IoT devices as well as collecting data describing their behavior has been developed. The environment also offers the possibility to schedule and perform attacks on monitored devices. The collected data, which describes both normal and attack-induced behavior of IoT devices, became the basis of the FedCSIS 2023 challenge.

This paper briefly presents the background of the competition, showcases the best models submitted to the challenge, and discusses the competition results and their potential impact on further system development.

II. Related Literature

Anomaly detection is a well–known approach for data analysis in many specific domains. As the IoT issues are becoming more and more interesting it is intuitive that any new or improved models should be tested on some data with anomalies to evaluate their capabilities. During the last decades, dozens of datasets related to network traffic security, operating systems or IoT monitoring were published. A brief summary is presented in Table I.

To reflect the nowadays trends in the data, we limited our search to datasets not older than 6 years and closely related to the IoT domain. Their short descriptions are presented below.

The environment for Bot–IoT [4] data capturing consisted of three components: network platforms, simulated IoT services, and extracting features. The network platforms included normal and attacking virtual machines. The IoT services simulating various IoT sensors were connected to the public IoT hub. The network environment that the dataset was collected from contained a combination of normal and botnet traffic. The dataset provides original packet capture (PCAP) files, generated Argus files and CSV files. The files separation is based on attack categories and subcategories. The dataset

TABLE I
DATASETS RELATED TO NETWORK TRAFFIC, OPERATING SYSTEM OR IOT SECURITY MONITORING (N- NETWORK, OS - OPERATING SYSTEM, IOT - INTERNET OF THINGS/INDUSTRIAL INTERNET OF THINGS DEVICES)

Dataset name	Owner	Monitoring	Reference
ADFA-LD	University of New South Wales	N, OS	[1]
Aposemat IoT–23	Stratosphere Laboratory	N, OS, IoT	[2]
CAIDA	Center of Applied Internet Data Analysis	N	[3]
Bot–IoT	University of New South Wales	N, IoT	[4]
CDX	United State Military Academy	N	[5]
DARPA 98-99	MIT Lincoln Laboratory	N, OS	[6]
KDD Cup 1999	University of California	N, OS	[7]
IoT Botnet	Ontario Tech University	N, OS, IoT	[8]
ISCX2012	University of New Brunswick	N	[9]
Kyoto	Kyoto University	N	[10]
Malware on IoT	Stratosphere Laboratory	N, OS, IoT	[2]
NSL-KDD	Canadian Institute for Cyersecurity	N, OS	[11]
RegSOC	Łukasiewicz–EMAG	N	[12], [13]
TON IoT	University of New South Wales	N, OS, IoT	[14]
Twente	University of Twente	N	[15]
UNSW-NB15	University of New South Wales	N	[16]
UMASS	University of Massachusetts	N	[17]
WUSTL-IIOT-2021	Washington University in St. Louis	N, IoT	[18]
Edge-IIoTset	Guelma Univ., De Montfort Univ., Annaba Univ., Edith Cowan Univ.	N, IoT	[19]

files include denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks, operating system and service scanning, keyloggers and data exfiltration.

The TON_IoT datasets [14] are IoT and Industrial IoT (IIoT) datasets which files contain heterogeneous data sources collected from IoT and IIoT sensor telemetry data sets, Windows 7 and 10 operating systems datasets, as well as Ubuntu 14 and 18 Transport Layer Security (TLS) and network traffic datasets. The data was collected in a realistic and large-scale network. A testbed network was created for the Industry 4.0 network, which includes IoT and IIoT networks. The test platform was deployed using multiple virtual machines and hosts of Windows, Linux, and Kali operating systems to manage connections between the three tiers of IoT, Cloud, and Edge / Fog. Various attack techniques such as DoS, DDoS, and ransomware targeting web applications, IoT gateways, and computer systems on the IoT/IIoT network were conducted. The datasets were collected in parallel processing to collect several normal and cyber-attack events from network traffic, Windows audit trail, Linux audit trail, and IoT telemetry data.

The IoT-23 [20] is a dataset of network traffic from Internet of Things devices. The dataset consists of 23 captured different IoT network traffic scenarios. These scenarios were divided into twenty network captures from infected IoT devices that the malware samples were performed in each scenario and three network captures of the actual network traffic of the IoT devices. In each malicious scenario a specific malware sample was run on a Raspberry Pi. Scenarios included the following malware samples used to infect the device (Mirai, Torii, Trojan, Gagfyt, Kenjiro, Okiru, Hakai, IRCBot, Linux Mirai, Linux Hajime, Muhstik, Hide and Seek).

Malware on IoT [2] is a dataset of the monitoring of real IoT devices infected by malware. The dataset consists of labeled network traffic files stored during the long-lived real IoT malware traffic. It is divided into five subsets containing results of network traffic capturing during the Mirai malware attack and two subsets of honeypot network traffic capturing logs including protocols (HTTP, SSL, TCP, UDP) and connections statistics. The honeypot was a network camera.

The WUSTL-IIOT-2021 dataset [18] contains network data from industrial Internet of Things (IIoT) monitoring. The dataset was developed on test bench involving the industrial control systems (ICS) model including supervisory control and data acquisition (SCADA) systems. The testbed was dedicated to controlling a water storage tank, which is a part of the process of water treatment and distribution. The dataset was preprocessed and cleaned (rows with missing or corrupted values and extreme outliers removed. Artificially generated Command Injection, reconnaissance and DoS attacks were recorded in the set. It accounted for about 8% of network traffic.

The Edge-IIoTset is a dataset [19] containing monitoring data from IoT devices and IIoT applications. The IoT data was generated from more than 10 types of IoT devices such as low-cost digital temperature and humidity sensors, ultrasonic sensors, water sensors, level detection sensors, pH meters, soil moisture sensors, heart rate sensors, flame sensors, etc. Fourteen attacks related to IoT and IIoT communication protocols were identified and analyzed (divided into five threats) including DoS/DDoS attacks, information gathering, man-in-the-middle, injection attacks, and malware attacks.

The number of available datasets from IoT and IIoT devices monitoring is still relatively small in comparison to the rapidly growing number of such devices in the world (it is estimated that this year, their number will reach approx. 17 billion devices). Most of the available datasets contain data from network monitoring during normal operations and attacks. The data sets described in detail contain data from the audit of real and simulated IoT devices and their network environment. The available datasets providing kernel event monitoring data are from Solaris (DARPA 98-99).

It is also worth noting that this data science competition is the second cybersecurity-related challenge organized on the KnowledgePit.ai platform [21]. The first one, IEEE BigData 2019 Cup: Suspicious Network Event Recognition, was organized in 2019, jointly with Security-on-Demand company [22]. The platform also hosted a competition related to the monitoring of network devices [23]. Furthermore, KnowledgePit.ai has been a host to a number of data science competitions related to monitoring hazardous environments using networks of sensors [24].

III. FedCSIS 2023 Challenge

The challenge data were generated within the simulation environment which was an extension of the software framework described in detail in [25]. This real components simulation environment consisted of an IoT device (Raspberry Pi) and additional devices responsible for HTTP traffic generating and performing the attacks. The whole environment was plugged into a separated network to assure no other influences on the monitoring device.

It was necessary to model a normal way of device operation as well as to simulate some attacks to assure that the collected dataset contains both, safe and unsafe states of device usage.

In order to obtain a data sample describing both the normal operations of devices and the moments when attacks were carried out, it was necessary to ensure typical network traffic and triggering processes characteristic for it, as well as to prepare a scenario of external attacks on the device.

Typical operation conditions were generated continuously in several independent ways:

- SSH sessions: with the interval from 10 seconds up to 12 hours an administrator logs into the device and runs from 3 up to 10 commands (the time between each command varies from 0.5 to 11.5 seconds), later the administrator logs off;
- HTTP WAN traffic: the device has a built-in HTTP server, so it was possible to simulate cyclic queries; queries based on the real (other) WAN-connected device and their intervals were also taken from the historic data;
- file transfer: the file transfer service was run on the device (the endpoint) to simulate a periodic software update: a binary file of a size varying from 512 to 1,024 bytes was sent with the random interval from 1 to 12 hours;
- specialized HTTP queries: the device contained a dedicated endpoint for outer status checking/device clock synchronizing so it was possible to send the query that implied the “date -date now” command run (such a query was released with 9.5–11.5 second interval).

The environment provided the ability to perform two kinds of attacks: remote code execution and path traversal. In the case of the first one, the attack is carried out through a vulnerable endpoint “clock.php” and a query that uses a command injection vulnerability is invoked. Then, a reverse connection (with the attacking host) is established and an interactive session of the console “sh” program is run. Afterward, random commands are invoked with an interval of up to 20 minutes.

A path traversal vulnerability is used to upload the file into an unusual location (path) on the device. Files were saved into one of the following locations:

- /dev/shm/
- /var/tmp/
- /tmp/

The name of the file was random, as well as its size (from 20 to 5,024 bytes). Also, the number of files was varying (from 1 to 10) and the time between uploads was between 0.5 and 10 seconds.

A. Data preparation

System logs of each device were extracted, saved, and preprocessed resulting in a tabular dataset consisting of statistical characteristics of each feature aggregated over a rolling window of a fixed size.

The data created in such a way had certain characteristics typical of simulated data:

- The generated dataset contained a huge amount of information. Within this data, only a small fraction was collected during attacks on IoT devices. This resulted in big files which were hard to operate on containing only a small amount of data that could serve as valuable training data.
- Because the number of continuous attacks was small (not exceeding 20) it is reasonable to assume that the dataset makes it impossible to train a general-purpose IoT-attack-detection model. The methods chosen for generating attacks represent only a small fraction of the attack classes identified in the wild [26].
- Most of the created data was highly repeatable, resulting in a dataset of low diversity. This is normal behavior for IoT devices that operate in a repetitive manner.
- The training and testing data were created from a single source. This made it possible to achieve a near 100% accuracy on the testing data by identifying the process id (PID) values of processes that were targeted during an attack and using this knowledge to identify malicious activities in system logs. This is a highly improbable scenario in reality since restarting a process (or restarting the whole system) results in a new PID being assigned to the processes.

B. Evaluation procedure

The task in this challenge was to design an accurate method to predict whether system logs from an IoT system indicate the occurrence of cyberattacks or not. The quality of submissions was evaluated using the ROC AUC measure. The solutions were evaluated online and the preliminary results were published on the public leaderboard. The preliminary score was calculated on a small subset of test records, which was the same for all participants. The final evaluation was conducted after the completion of the competition using the remaining portion of the test records.

C. A baseline solution

Two baseline scores were established. The first one assumes a realistic scenario, while the second one was tailored to the dataset used in this competition, leveraging the knowledge that classification of PIDs enabled achieving a near 100% accuracy score on a portion of the test dataset.

1) Baseline score 1: The first baseline was calculated using an XGBoost model. Since the problem is a highly unbalanced classification task, the XGBoost's prior score was set according to the proportion of the system logs containing attacks (≈0.97).

The XGBoost model in this scenario was selected after comparing its results to Random Forest models (with and without class weights according to the proportion of the system logs containing attacks) and an XGBoost model without a prior score.

The baseline score achieved this way was 0.691 (ROC AUC). Examining the feature importances revealed that over 65% of the result was dependent on features created using the 'SYSCALL_pid' column which led to investigating the PID-based dependencies in the data and creating another classifier which gave the second baseline score.

2) Baseline score 2: Since the data used in this experiment was generated from a single artificial source, the PIDs corresponding to attacks were constant over time. For this reason, it is possible to list the PIDs of processes present during attacks and classify them in the test dataset as attacks.

This technique can also be altered by not strictly looking for all malicious PIDs in the test dataset but looking for PIDs that frequently occurred during attacks. Such an approach makes it possible to introduce a margin of error and thus filter out PIDs that could be falsely labeled in the training dataset as being part of an attack.

Performing a search-based classification as described above without any ML model resulted in a ≈0.93 ROC AUC score. Since a frequency-based method of classifying malicious PIDs was used; this score can be easily improved by further examining the PID values distribution in the training dataset.

IV. Challenge Outcomes

The competition was quite successful, with 78 participating teams and nearly 600 correctly formatted submissions. The majority of submitted solutions follow a general pattern of processing/cleaning the data → performing feature engineering → feature selection → model construction. However, there were some differences in the approach due to the hierarchical/complex form/format of the dataset. The internal data structure, i.e., a single observation is given as a separate file with a variable number of entries imposed the need for some form of aggregation. Among the submissions, we could observe different approaches in this regard. Some of the teams aggregated each of the input data files resulting in a representation where each observation (each file) was represented as a single row, while other teams concatenated all input files performing the aggregation only at the very end on the basis of predictions of classifiers working the level of file entries. The feature extraction/engineering stage was also approached differently by different teams. The total number of constructed features ranged from as few as several to several hundred thousand (including binary-encoded features). The most popular machine learning models used among the contestants fall into the category of gradient boosting machines - with particular implementations provided by commonly used open-source libraries like XGBoost, LightGBM, and CatBoost. However, also several other models could be encountered, including classical ones like decision trees, random forest, kNN, and logistic regression, as well as, custom methods, e.g., using micro-predictors build on top of features constructed using target guided binning, which achieved one of the best final scores.

TABLE II
Final results of the competition. The scores of the top 10 teams and their number of submitted solutions are shown.

Rank	Team name	Preliminary	Final score	#subs
1	MathLogic	1.0000	0.9999	76
2	dymitr	1.0000	0.9997	59
3	The Fellowship of the Cybersecurity	0.9997	0.9995	5
4	DML	0.9999	0.9993	176
5	Y-Team	1.0000	0.9986	8
6	Cyan	0.9940	0.9966	69
7	PisaTeam	1.0000	0.9957	10
8	hieuvq	0.9772	0.9718	101
9	Stan	0.9190	0.9293	14
10	baseline	0.9633	0.9257	-
...	...	...	...	...

Most of the competition participants decided to use the PID analysis-based approach to solve the task due to its high effectiveness. In such a case, the most significant differentiating factor between solutions from different teams was their approach to feature engineering. The final results of the top 10 teams are shown in Table II.

V. Conclusions and Future Work

The method of constructing the simulation environment and creating the competition dataset was sufficient to train and compare various machine learning models, but only to determine potential directions for future research and development specifically for cybersecurity data from IoT devices. It should be noted that such data is not suitable for training general-purpose machine learning models.

Based on the results of the competition and techniques employed, we plan future work to focus on the aspect of training data sets quality. Specifically - on tuning techniques for creating synthetic data sets that reflect various characteristics of real-life data. We plan to explore a hybrid approach, in which such synthetic data is used to augment data gathered from production IoT systems, in such a way as to create training sets that are optimal for training of a production system for analysis of anomalies in IoT behavior.

The end goal is to create a system that will be usable in various fields of application - most notably one that works on data gathered from utility providers (electricity, gas, water), manufacturers of video surveillance devices, and smart city infrastructure (interactive road signs, passenger information systems, control systems). Those companies will receive a toolkit that can be implemented in their own products. Additionally, the solutions can be utilized for protecting home devices such as smart lighting, electrical installations, alarm systems, and others.

VI. Acknowledgements

This work is a part of the "Development of a tool for protecting users, systems and devices of the Internet of Things, based on machine learning and behavioral analysis." project, which was co-financed by the Polish National Centre for Research and Development as part of the fourth CyberSecIdent–Cybersecurity and e–Identity competition (agreement number: CYBERSECIDENT/489240/IV/NCBR/2021).

References

[1] Creech G. and Hu J., "Generation of a new IDS test dataset: Time to retire the KDD collection," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2013, pp. 4487–4492, ISSN: 1558-2612.

[2] Stratosphere, "Stratosphere laboratory datasets," 2020, Retrieved March 15, 2021, from https://www.stratosphereips.org/datasets-overview.

[3] CAIDA, "Center of applied internet data analysis," 1998-2013, Retrieved March 16, 2021, from https://www.caida.org/catalog/datasets/completed-datasets/.

[4] Koroniotis N., Moustafa N., Sitnikova E., and Turnbull B., "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," 2018.

[5] Sangster B., O'Connor T. J., Cook T., Fanelli R., Dean E., Adams W. J., Morrell C., and Conti G., "Toward instrumenting network warfare competitions to generate labeled datasets," in *Proceedings of the 2nd Conference on Cyber Security Experimentation and Test*, USA, 2009, CSET'09, p. 9, USENIX Association.

[6] MIT Lincoln Laboratory MIT, "Mit lincoln laboratory - darpa datasets," 1998-1999, Retrieved March 16, 2021, from https://www.ll.mit.edu/r-d/datasets.

[7] Stolfo S.J., Fan W., Lee W., Prodromidis A., and Chan P.K., "Cost-based modeling for fraud and intrusion detection: results from the jam project," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, 2000, vol. 2, pp. 130–144 vol.2.

[8] Ullah I. and Mahmoud Q. H., "A technique for generating a botnet dataset for anomalous activity detection in iot networks," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 134–140.

[9] Shiravi A., Shiravi H., Tavallaee M., and Ghorbani A. A., "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, no. 3, pp. 357–374, 2012.

[10] Kyoto University, "Traffic data from kyoto university's honeypots," 2015, Retrieved March 17, 2021, from https://www.takakura.com/Kyoto_data/.

[11] Tavallaee M., Bagheri E., Lu W., and Ghorbani A. A., "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul 2009, p. 1–6.

[12] Wawrowski Ł., Michalak M., Białas A., Kurianowicz R., Sikora M., Uchroński, and Kajzer A., "Detecting anomalies and attacks in network traffic monitoring with classification methods and xai-based explainability," *Procedia Comput. Sci.*, vol. 192, no. C, pp. 2259–2268, jan 2021.

[13] Wawrowski Ł, Białas A., Kajzer A., Kozłowski A., Kurianowicz R., Sikora M., Szymańska-Kwiecień A, Uchroński M., M. Białczak, Olejnik M., and Michalak M., "Anomaly detection module for network traffic monitoring in public institutions," *Sensors*, vol. 23, no. 6, 2023.

[14] Moustafa N., Ahmed M., and Ahmed S., "Data analytics-enabled intrusion detection: Evaluations of ton_iot linux datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2020, pp. 727–735.

[15] Sperotto A., Sadre R., van Vliet F., and Pras A., "A labeled data set for flow-based intrusion detection," in *IP Operations and Management*, Giorgio Nunzi et al., Eds., Netherlands, Oct. 2009, Lecture Notes in Computer Science, pp. 39–50, Springer, 9th IEEE International Workshop on IP Operations and Management, IPOM 2009, IPOM ; Conference date: 29-10-2009 Through 30-10-2009.

[16] Moustafa N. and Slay J., "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[17] Yang S., Kurose J., and Levine B., "Disambiguation of residential wired and wireless access in a forensic setting," in *2013 Proceedings IEEE INFOCOM*, 04 2013, pp. 360–364.

[18] Maede Zolanvari, "Wustl-iiot-2021 dataset for iiot cybersecurity research," https://www.cse.wustl.edu/~jain/iiot2/index.html, 2021.

[19] Ferrag M.A., Friha O., Hamouda D., Maglaras L., and Janicke H., "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications: Centralized and federated learning," https://dx.doi.org/10.21227/mbc1-1h68, 2022.

[20] Garcia S., Parmisano A., and Erquiaga M. J., "IoT-23: A labeled dataset with malicious and benign IoT network traffic," Jan. 2020.

[21] Janusz A. and Ślęzak D., "KnowledgePit Meets BrightBox: A Step Toward Insightful Investigation of the Results of Data Science Competitions," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022*, 2022, vol. 30 of *Annals of Computer Science and Information Systems*, pp. 393–398.

[22] Andrzej Janusz, Daniel Kałuża, Agnieszka Chądzyńska-Krasowska, Bartek Konarski, Joel Holland, and Dominik Ślęzak, "IEEE BigData 2019 Cup: Suspicious Network Event Recognition," in *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*. 2019, pp. 5881–5887, IEEE.

[23] Andrzej Janusz, Mateusz Przyborowski, Piotr Biczyk, and Dominik Ślęzak, "Network Device Workload Prediction: A Data Mining Challenge at Knowledge Pit," in *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, FedCSIS 2020, Sofia, Bulgaria, September 6-9, 2020*, 2020, vol. 21 of *Annals of Computer Science and Information Systems*, pp. 77–80.

[24] Andrzej Janusz, Marek Sikora, Łukasz Wróbel, Sebastian Stawicki, Marek Grzegorowski, Piotr Wojtas, and Dominik Ślęzak, "Mining data from coal mines: Ijcrs'15 data challenge," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*. Springer, 2015, pp. 429–438.

[25] Adamczyk B., Brzęczek M., Michalak M., Kostorz I., WawrowskiŁ., Hermansa M., Czerwiński M., and Jamiołkowski A., "Dataset generation framework for evaluation of iot linux host–based intrusion detection systems," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 6179–6187.

[26] "MITRE ATT&CK® Adversarial Tactics, Techniques, and Common Knowledge," https://attack.mitre.org/versions/v13/, 2023.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1295–1299

DOI: 10.15439/2023F2385
ISSN 2300-5963 ACSIS, Vol. 35

Tackling Variable-length Sequences with High-cardinality Features in Cyber-attack Detection

Chang Lin
State Key Laboratory of Information Photonics and Optical Communications
Beijing University of Posts and Telecommunications
Beijing, China
bupt.ipoc@yandex.com

***Abstract*—Internet of Things (IoT) based systems are vulnerable to various cyber-attacks and need advanced and smart techniques in order to achieve their security. In the FedCSIS 2023 big-data competition, participants are asked to construct scoring models to detect whether anomalous operating systems were under attack by using logs from IoT devices. These log files are variable-length sequences with high cardinality features. Through in-depth and detailed analysis, we find out concise and efficient methods to handle these huge volumes, variety, and veracity of data. On the basis of this, we create detection rules using the fundamental knowledge of mathematical statistics and train gradient boosting machine (GBM) based classifier for attack detection. Experimental and competition results prove the effectiveness of our proposed methods. Our final AUC score is 0.9999 on the private leaderboard.**

***Index Terms*—Internet of Things; Cyber security; Machine Learning; Variable-length Sequences; High-cardinality Features**

I. Introduction

INTERNET of Things (IoT) plays an essential role in remote monitoring and control operations. IoT based systems are widely used in the fields of environment, home automation, healthcare, smart grid, transportation, agriculture, military, surveillance, etc. In 2023, the number of devices connected to networks is expected to be 3 times higher than the global population [1]. With the IoT, sensors collect, communicate, analyze, and act on information. This offers new ways for technology, media and telecommunications businesses to create value. But it also creates new opportunities for that information to be compromised. The IoT connect systems, applications, data storage, and services become a new gateway for cyber-attacks as they continuously offer services but lack of adequate security protection. In 2020, nearly 1.5 billion cyber-attacks on IoT devices were reported [1]. These attacks may steal important and sensitive information that causes economic and societal damages. To address critical challenges related to the authentication and secure communication of IoT, many people (such as Jarosz et al.[2]) have developed various authentication and key exchange protocols for IoT devices. But software piracy and malware attacks remain high risks to compromise the security of IoT. This brings with it a particular challenge: securing IoT based systems against cyber-attacks.

In the FedCSIS 2023 challenge: Cybersecurity Threat Detection in the Behavior of IoT Devices [3], participants are asked to construct scoring models to detect whether anomalous operating systems were under attack by using logs from IoT devices. This competition has important theoretical and practical value for increasing IoT cyber security. It provides rich and detailed data for participants to analyze cyber-attacks from various perspectives and to train and test their models. Thereby we can understand attacker's intent, learn their behavior, and track the tactics, techniques, and procedures that they utilize to achieve their goals. We believe that all predictive models thoughtfully and elaborately constructed by each participant will definitely help to detect attacks as early as possible, determine the scope of the compromise rapidly and predict how they will progress, and eventually empower organizations to better respond to attacks.

In the past decade, traditional machine learning techniques (such as Support Vector Machine, Decision Tree, K-Nearest Neighbor, Random Forests, Naive Bayes, etc.) have been widely used by the cyber security community to automatically identify IoT attacks. Many papers (such as [4]) have provided various reference implementation on state-of-the-art machine learning methods for data preprocessing, feature engineering, model fitting, and ensemble blending. And paper [5] discusses in detail the existing machine learning and deep learning solutions for addressing different security problems in IoT networks. However, with the continuous expansion and evolution of IoT applications, attacks on these IoT applications continue to grow rapidly.

The complexity and quantity of attacks push for more efficient detection methods. In the recent years, deep learning techniques have been used in an attempt to build more reliable systems. For example, Martin Kodys et al. proposed a novel solution which deployed two CNN architectures (ResNet-50 and EfficientNet-B0) on the same data to observe how their performance differs to detect the intrusion attacks against IoT devices [6]. Kumar Saurabh et al. developed Network Intrusion Detection System (NIDS) models based on variants of LSTMs (namely, stacked LSTM and bidirectional LSTM and validated their performance) [7]. Compared with traditional machine learning, the deep learning brings an end-to-end approach combining feature selection and classification which can speed up the defense response against the fast-evolving cyber-attacks. However, some authors declare that deep learning methods have proved far better than the traditional machine learning models in terms of accuracy, precision with the ability to handle large amounts of data, and the inability to scale the data poses a large limitation to the extensive use of any conventional machine learning model [7]. This is not always the case.

In this competition, we apply basic data processing approaches, and leverage the feature selection and model building methods mentioned in our ICME2023 paper [8], combined with fundamental knowledge of mathematical statistics for cyber security threat detection. Our methods are fast and accurate, and achieve near-perfect prediction results. Our work provides examples for processing large scale data and extracting effective features to get better detection accuracy with less computational cost.

The paper is structured as follows: Section II introduces data analysis and processing methods. Section III applies basic knowledge of mathematical statistics to construct rules for attack detection. Section IV discusses how to perform feature selection and build binary classification models for threat prediction. Section V explains the experiment design and presents the results of the experiments. Section VI discusses the pros and cons of our proposed approaches and suggests future research directions.

II. Data Analysis and Processing

The available training data and test data in this competition contain 15027 and 5017 log files respectively. Each log file includes 40 fields and it contains 1-minute logs of all related system calls. There are a total of 28,339,158 and 10,060,209 lines of records in the training and test sets, respectively. The size of the data set is over 21.4 gigabytes. Therefore, one of the main tasks of this competition is to analyze and process these data efficiently and thereby construct effective features for attack detection.

In the training set, 522 files were identified as being under attack. Therefore, the chance of cyber-attack is 3.47375%. After the end of the competition, the organizer published the labels of the test set for the participants to do further research. There are 176 files which are under attack in the test set. It seems that the data set is divided by a "stratified K-Fold" manner to let the test set has the same proportion of target variable as the entire data set.

Among the 40 fields, 17 fields only have unique values (for example, 'SYSCALL_arch' always equals to 'aarch64'). These fields are useless. Besides, (1) 'SYSCALL_timestamp' indicates the number of milliseconds relating to the datetime 2023-04-12-00:00:00. It should not be used for prediction, otherwise it will cause over-fitting. (2) Column 'SYSCALL_exit' and 'SYSCALL_exit_hint' take almost similar values, so column 'SYSCALL_exit_hint' can be ignored.

For the remaining columns, we split their values into the smallest units. We call these smallest units "basic items". It is obvious that every unique combination of basic items represents a new kind of attribute, which can dramatically increase the amount of data. These will cause the high cardinality problem which means that there can be many possible values for a single column. To solve the high cardinality problem, a simple and straightforward approach is to subdivide the content of each column, and left the problem of digging the correlations between these basic items to the subsequent algorithms. Take the following line of record as an example:

```
53824,aarch64,openat,yes,6.0,ps,/usr/bin/ps,>systemd>/usr/sbin/
cron>/usr/sbin/cron>/bin/sh>/usr/bin/bash,"['/proc/uptime',
'/proc/meminfo', '/proc', '/proc/647524/status']", "['/lib/aarch64-
linux-gnu/libprocps.so.8', '/lib/aarch64-linux-gnu/libdl.so.2',
'/lib/aarch64-linux-gnu/libc.so.6',
'/lib/aarch64-linux-gnu/libsystemd.so.0',
'/lib/aarch64-linux-gnu/librt.so.1',
'/lib/aarch64-linux-gnu/liblzma.so.5',
'/lib/aarch64-linux-gnu/libzstd.so.1',
'/lib/aarch64-linux-gnu/liblz4.so.1',
'/lib/aarch64-linux-gnu/libgcrypt.so.20',
'/lib/aarch64-linux-gnu/libpthread.so.0',
'/lib/aarch64-linux-gnu/libgpg-error.so.0',
'/lib/aarch64-linux-gnu/libnss_files.so.2']",root,root,6,649165,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,[],,,,,/usr/bin/ps,,
```

We split the content of field 'PROCESS_PATH' (column 7) into the following small items:

systemd
/usr/sbin/cron
/bin/sh
/usr/bin/bash

and 'CUSTOM_openFiles' (column 8) into these items:

/proc/uptime
/proc/meminfo
/proc
/proc/647524/status

and 'CUSTOM_libs' (column 9) into these items:

/lib/aarch64-linux-gnu/libprocps.so.8
/lib/aarch64-linux-gnu/libdl.so.2
/lib/aarch64-linux-gnu/libc.so.6
/lib/aarch64-linux-gnu/libsystemd.so.0
/lib/aarch64-linux-gnu/librt.so.1
/lib/aarch64-linux-gnu/liblzma.so.5
/lib/aarch64-linux-gnu/libzstd.so.1
/lib/aarch64-linux-gnu/liblz4.so.1
/lib/aarch64-linux-gnu/libgcrypt.so.20
/lib/aarch64-linux-gnu/libpthread.so.0
/lib/aarch64-linux-gnu/libgpg-error.so.0
/lib/aarch64-linux-gnu/libnss_files.so.2

As for the columns 'SYSCALL_exit' (column 4) and 'SYSCALL_pid' (column 13), since 'exit code' and 'pid' take a very large range of values, we use the simple method of data binning to combine every 100 values into one group. After this, we count the occurrence of each basic item in the training log files and in files which are under attack, respectively. Examples of the obtained statistics are shown in Table 1.

Table 1. Example of statistical analysis results of column 2 ('SYSCALL_syscall')

Under attack times	Total occurrence	Items
491	14199	write
132	3521	exit
266	8123	bind
517	14942	exit_group
514	14701	socket
515	14939	execve
0	19	kill
520	14977	openat
501	14406	connect
522	15026	close
517	14956	clone
517	14962	mmap
0	51	listen
519	14963	munmap
0	49	bpf

These results can be further applied to construct detection rules and create features for classification.

III. Building Rules for Attack Detection

List all the basic items with attacked chance equal to 100% and number of occurrences >= 5 and are not included in other items, we can get the following list:

Table 2. Rules used for attack detection

Colum n index	Occurrence in training set	Occurrence in test set	Items
8	169	18	/proc/647524/stat
8	145	50	/proc/573203/stat
8	106	45	/proc/671015/stat
8	59	54	/proc/600849/stat
13	6	1	[576000000-576099999]
13	5	1	[574500000-574599999]

From the list we can find that basic item "/proc/647524/stat" appears 169 times in different log files, and all these files are identified as being under attack. From this we can infer that if a log file contains "/proc/647524/stat", it definitely indicates a cyber-attack has occurred.

Suppose "/proc/647524/stat" is an ordinary event, then the probability that "/proc/647524/stat" consecutively occurs 169 times in and only in the attacked files is 0.0347^169 = 0. According to the impossibility principle of small probability events, a small probability event is practically impossible to happen in a single trial. And once it does happen, we can reasonably reject the null hypothesis. In fact, it only needs five consecutive occurrences, then we can reasonably infer that an event has close relationship with cyber-attack.

By applying the above 6 rules, we are able to accurately detect 169 compromised files from the test set.

Furthermore, using the same method, we can confirm that 4003 files are secure (i.e., there are no attack events in these log files).

Applying these simple rules for threat prediction yields an AUC = 0.9985 on the test set.

IV. Feature Selection and Model Building

The aforementioned rule-based intrusion detection methods use only a small fraction of the data and cannot take advantage of the complex nonlinear relationships between various features. In this section we apply the sequential floating forward and backward (SFFB) feature selection method [8] for feature selection, and train a binary classification model based on GBM for attack prediction.

When creating features, we use the target encoding method to replace the categorical values with the mean of the target variable, and introduce a smoothing parameter to regularize towards the unconditional mean. We found this to be helpful in improving the predictive performance of the subsequent algorithms. We also find that the "K-fold target encoding" preferred by many people cannot mitigate over fitting risks. In fact, for high cardinality features "K-fold target encoding" method will lead to serious data leakage. This can be easily verified.

After feature encoding, we calculate the maximum, minimum and average chance of being attacked of each field. We also count their number of the contained basic items. Subsequently, these features are concatenated to form a feature set of equal length. We then use SFFB method to select features. The optimal subsets selected by the SFFB method are somewhat random. In most cases, the selected subset will only contain 10 features, such as:

PROCESS_comm_count, PROCESS_exe_count,
PROCESS_PATH_mean, CUSTOM_openFiles_max,
CUSTOM_openFiles_min, SYSCALL_pid_min,
SYSCALL_pid_mean, SYSCALL_pid_count,
PROCESS_name_mean, PROCESS_name_count.

*_max, *_min and *_mean means the maximum, minimum and average attacked chance of the fields. *_count means the number of basic items of the fields.

Training a GBM model with these 10-dimensional features leads to a classification result of AUC=0.9997 on the test set. Figure. 1 shows the gain contribution of these features.

V. Experiment Design and Experiment Results

Cybersecurity threat detection always is a majority-minority classification problem. Class imbalance in the dataset can dramatically skew the performance of classifiers. Therefore a reliable cross-validation method is essential to train a good classifier.

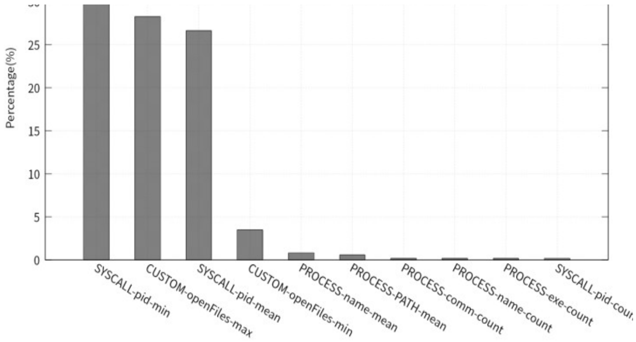


Fig. 1. Gain contribution of the 10 selected features

In our experiments, we estimate the performance of the classifier by using 3-fold cross-validation. At each fold, we completely hide the validation set when processing data and performing feature engineering. The average AUC score of 3-fold cross-validation is 0.9997 in local test. However, the classifiers trained in this way cannot achieve optimal scores on the public leaderboard. In fact, when the score of local CV is greater than 0.998, the changing trends of the local CV score is not consistent with the trends of the public leader-board. To address this problem, we randomly select 2/3 of the data from the training set at a time to train several classifiers, and then weighted averaging the prediction result of each classifier. In this way, we try to eliminate the effects of class imbalance and sample bias.

Finally, we ensemble the results obtained from the rules prediction with those predicted by the GBM model, and achieve an AUC score 0.9999 on the private leaderboard. After the organizer published the labels of the test set, we found that by correctly ensembling the prediction results from sections 3 and 4, we could obtain an AUC score 0.99995 on the test set. This is equivalent to the total accuracy can up to 99.88%. The ensemble method is:

1. If rule-based prediction results are equal to 1, then:

*ensemble results = 0.85 + 0.15*GBM prediction results.*

2. If rule-based prediction results are equal to 0, then:

*ensemble results = 0.15*GBM prediction results.*

3. Otherwise,

ensemble results = GBM prediction results

The total time (includes data processing, feature construction, feature selection, classifier training, and target prediction) required to obtain this result on our i7-10700 desktop is less than 30 minutes.

VI. Conclusion and Future Work

In this cyber security threat detection challenge, we only apply the fundamental methods of machine learning, but achieve near-perfect detection results. Many big-data competition participants like to apply ready-to-use GBM or deep learning frameworks. They prefer the end-to-end approaches that automates data processing, feature selection and classification, and expect to get good answers just by tuning the parameters. But our experiments show that each algorithm has a different application scenario.

In this competition, we conduct in-depth, detailed analysis of the massive-volumes data, and propose concise and efficient methods to process these data. (A significant portion of our work is C++ programming. To master the methodologies and techniques of contemporary C++ in the age of new technologies and challenges, one can start by reading paper [9].) Our proposed approaches are useful for solving variable-length, high-dimensional and high-cardinality problems.

However, our detection method still has obvious limitations: it is good at detecting known attacks but may fail at detecting attacks which have not been seen before. As more and more IoT devices are added, the potential for new and unknown threats grows exponentially. For this reason, an intelligent security framework for IoT networks must be developed that can identify such threats (e.g., detect any anomaly which rises from any deviation from normal behavior of the IoT network, or monitor network traffic to identify potential threats). In these research directions, conventional machine learning methods will still play an important role.

Acknowledgements

Many thanks to Łukasiewicz Research Network, EFIGO, and QED Software for their great efforts in organizing this challenging competition. Thanks to all the participants. They all did an amazing job.

References

[1] IoT Cybersecurity in 2023: Importance & Tips To Deal With Attacks. https://research.aimultiple.com/iot-cybersecurity/

[2] Michał Jarosz, Konrad Wrona, Zbigniew Zieliński. Formal verification of security properties of the Lightweight Authentication and Key Exchange Protocol for Federated IoT devices. Proceedings of the 17th Conference on Computer Science and Intelligence Systems, ACSIS, Vol. 30, pages 617-625 (2022). DOI: http://dx.doi.org/10.15439/2022F169.

[3] FedCSIS 2023 Challenge: Cybersecurity Threat Detection in the Behavior of IoT Devices. https://knowledgepit.ai/fedcsis-2023-challenge/

[4] Eyad Kannout, Michał Grodzki, Marek Grzegorowski. Considering various aspects of models' quality in the ML pipeline - application in the logistics sector. Proceedings of the 17th Conference on Computer Science and Intelligence Systems. ACSIS, Vol. 30, pages 403-412 (2022). DOI: http://dx.doi.org/10.15439/2022F296.

[5] F. Hussain, R. Hussain, S. A. Hassan and E. Hossain. Machine Learning in IoT Security: Current Solutions and Future Challenges. in IEEE Communications Surveys & Tutorials, vol.22, no.3, pp.1686-1721, 2020. DOI: https://doi.org/10.1109/COMST.2020.2986444.

[6] Martin Kodys, Zhi Lu, Kar Wai Fok, et al. Intrusion Detection in Internet of Things using Convolutional Neural Network. https://arxiv.org/pdf/2211.10062.pdf. DOI: https://doi.org/10.1109/PST52912.2021.9647828.

[7] Kumar Saurabh, Saksham Sood, P. Aditya Kumar, et al. LBDMIDS: LSTM Based Deep Learning Model for Intrusion Detection Systems for IoT Networks. https://arxiv.org/pdf/2207.00424.pdf. DOI: https://doi.org/10.48550/arXiv.2207.00424

[8] Chang Lin. Predicting Frags in Tactic Games using Machine Learning Techniques and Intuitive Knowledge (in press). In Proceedings of the 2023 IEEE International Conference on Multimedia and Expo. IEEE, 2023.

[9] Bogusław Cyganek. Modern C++ in the era of new technologies and challenges - why and how to teach modern C++?. Proceedings of the 17th Conference on Computer Science and Intelligence Systems. ACSIS, Vol. 30, pages 35-40 (2022). DOI: http://dx.doi.org/10.15439/2022F308.

DOI: 10.15439/2023F7166
ISSN 2300-5963 ACSIS, Vol. 35

Beating Gradient Boosting: Target-Guided Binning for Massively Scalable Classification in Real-Time

Dymitr Ruta, Ming Liu, Ling Cen
EBTIC, Khalifa University, UAE
{dymitr.ruta,liu.ming,cen.ling}@ku.ac.ae

***Abstract*—Gradient Boosting (GB) consistently outperforms other ML predictors especially in the context of binary classification based on multi-modal data of different forms and types. Its newest efficient implementations, including XGBoost, LGBM and CATBoost, push GB even further ahead with fast GPU-accelerated compute engine and optimized handling of categorical features. In an attempt to beat GB in both the performance and processing speed we propose a new simple yet fast and robust classification model based on predictive binning. At first all features undergo massively parallelized binning into a unified ordinally compressed risk representation, independently optimized to maximize the AUC score against the target. The resultant array of summarized micro-predictors, resembling 0-depth decision trees, directly expressing oridnally represented target risk, are then passed through the greedy feature selection to compose a robust wide-margin voting classifier, whose performance can beat GB while the extreme build and execution speed along with highly compressed representation welcomes extreme data sizes and real-time applicability. The model has been applied to detect cyber-security attacks on IoT devices within FedCSIS'2023 Challenge and scored 2^{nd} place with the $AUC \approx 1$, leaving behind all the latest GB variants in performance and speed.**

I. Introduction

THE RISE of AI to prominence in access and control of just about everything is upon us and so is the expanding infrastructure network of fixed and mobile sensing and computing devices, capable of sending and receiving data, known as the Internet of Things (IoT). In such digitized environment security and integrity of every individual node as well as the whole network is critical and hence the cyber-security against the internal and external threats of different nature and scale is of crucial importance. A survey presented in [1] offers a thorough overview of different security threats IoT devices are exposed to, reviews the current security mechanisms, trying to address them, and identifies a continuous challenge in this very fast evolving ecosystem, in which the evidence based designed security solutions are always playing a catch-up game and leave a lagged gap, in which new threats or attacks may inflict a lot of damage before they are detected, analysed and neutralized. Machine Learning (ML) has been growing in parallel to these revolutionary changes and since the outset offered methods for automated detection of security threats based on data, both by learning from historically labelled attack examples and by discovering and flagging anomalies from normal operation of the IoT devices. Several reviews of ML deployment in the IoT cyber-security environment have been presented recently like in [2] assessing various ML models while focusing further on the SVM application to smart city traffic flows prediction, or in [3] where similar classification, utility and suitability analysis of the most common ML methods, applied in various aspects of IoT cyber-security, is carried out with perhaps a deeper focus on deep learning.

To our surprise, however, gradient boosting methods developed around the start of our century by the pioneering work in [4], [5], that have ever since consistently been winning big data and ML competitions ([6]-[11]), have been rather scarcely covered in the literature dedicated to cyber-security. We can argue that in more practical realistic ML applications to cyber-security like detecting threads based on complex log-extracted data, the old favourite models like SVM are simply not scalable enough [2], [11], while high-performing deep learning networks cannot easily encode the multi-modal unstructured data coming in a variety of forms and types, i.e. in quite different form than regular images or time-series, deep learning are performing well for. For such data GB models appear much more suitable and easy to be applied.

In this paper, however, we attempt to go a step further than a standard application of the optimized gradient boosting models. Inspired by the essence of what makes GB work well we propose the target guided binning (TGB) process that transforms all input features into an array of independent AUC-optimized and robust micro-predictors of the binary target with which a simple voting can outperform the latest GB variants both in terms of performance, transparency, data handling overhead and the processing speed. While binned features resemble somewhat 0-depth decision trees, they leave TGB process already AUC-pre-optimized to maximally suppress unnecessary complexities within the original feature domains while maximally exposing and summarizing its predictive power against the known binary target.

Neither optimising with respect to AUC [16], [17] nor binning [17], [18] is new in the context of classification. Since the advantages of AUC-optimized classifier design have been widely exposed [16], there was a fast-paced evolution towards AUC-optimized classification that eventually culminates with the advent of CATBoost [17], and the ingenious way it handles high cardinality categorical data utilising target statistic as a robust form of feature re-engineering. We build up on this direction by more explicit target guidance in a form of predictive binning uniformly applied to both categorical and numerical features, yet to guard against performance damaging *target-leakage*, extensively investigated in [18], we simply build the cross-validation process into the binning process.

We intend to comparatively demonstrate the advantages of the proposed TGB over GB in an objectively evaluated FedCSIS'2023 challenge dedicated to predicting cyber attacks on IoT devices based on logs data files that the latest GB variants should perform very well for. We specifically focus on and optimise TGB for scalability utilizing massive parallelization of the processing pipeline that can be deployed along multiple dimensions to approach real-time readiness even for such large scale problems as the cyber-security attacks prediction. For that reason we deliberately leave, typically critical, feature engineering aspects unexplored to great depths, instead focusing on TGB algorithmics and comparative experimentation applied to only one family of features, that nevertheless produced excellent results that scored the 2^{nd} place in the competition after leading throughout the whole preliminary phase.

The remainder of the paper is organized as follows. The FedCSIS'2023 Challenge is briefly described in Section II. The critical element of the proposed target guided bining process along with its fast, massively parallelized implementation are discussed in Section III. The composition of wide-margin voting classifier from TGB-binned features that includes important aspects of feature selection and the evaluation criteria they use, are covered in Section IV, followed with the description, presentation and discussion of the experimental results in Section V and the conclusions drawn in Section VI.

II. FedCSIS'2023 Challenge

The FedCSIS'2023[1] competition focused on detecting cyber-security threats based on the behavior of IoT devices captured in their detailed logs data. Over 20k of log files have been provided to the competitors, each of which representing a single data sample of timestamped variable-length sequence of events capturing specific IoT device interaction/operation over a fixed period of time. System calls, system and user processes' details, lists of open files and libraries, counts of various events, errors, integrity checks are just some of 24 raw features included in the log files both in numerical and categorical (text) format. Out of the total 20044 samples the correct binary target label (*ATTACK*) was provided for 15027 training samples, leaving the remaining unlabeled 5017 examples for AUC testing on the KnowledgePit.ml platform, hosting the competition. Before the final evaluation of the submitted full solutions, i.e. throughout the competition, KnowledgePit.ml operated a leader-board of competitors' solutions evaluated based on the preliminary set of unknown 10% of the full testing set. FedCSIS'2023 Challenge is sponsored by the Łukasiewicz Research Network - Institute of Innovative Technologies, EMAG and EFIGO sp. z o.o. companies.

III. Target-guided binning (TGB)

Given the data examples are provided in a composite form a variable-length table of time-ordered event features it was imminent that any kind of feature engineering strategy would involve some form of aggregation over the whole log table of typically thousands of records. Moreover given relatively large number of unique values observed for several categorical features it is expected that potential number of possible derived features could be large. In an attempt to extract possibly fullest predictive value form such evidence we decided to reduce feature engineering to measuring the per-log-frequency of all observed unique feature values and simultaneously transform these frequencies into summarized ordinal target-risk levels monotonically increasing with the target likelihood conditioned on the intervals or subsets contained within each risk level. Such predictive TGB transforms all feature space irrespective of their form or data type into unified, numerically stable and additive micro-predictors of the target.

Target-guided binning focuses on a single, very simple goal: how to exploit the guidance of the binary target to bin the input feature in a way that maximally improves its generalized predictive power over the target. An objective, scale- and threshold- invariant measurement of feature predictive power in binary classification is the area under the receiver-operator curve (AUC). Denoting by x and $y(x) = y_x$ our input variable and the binary target, respectively, and by $AUC(y_x, x)$ the empirical AUC between y_x and x, the target-guided binning process can be formally defined by the transformation function T that maps all values of x into $x_T \in \{1, 2, .., k\}$ such that the $AUC(y_x, x_T)$ is maximized:

$$T: \quad x_T = T(x, y_x), \quad x_T = \arg\max_{x_T} AUC(y_x, x_T) \quad (1)$$

At the first glance, this task seems trivial given the relation of AUC and the Wilcoxon-Mann-Whitney statistic [14], [15], that gives an AUC a simple interpretation of the probability of a random *positive* $x_+ = \{x_i : y(x_i) = 1\}$ being larger than the random *negative* $x_- = \{x_j : y(x_j) = 0\}$:

$$AUC(y, x) = P(x_+ > x_-) \quad (2)$$

It is trivial to show that to maximize such probability it is sufficient to simply transform x to the ranking of target rates along all unique values of x. The recipe for (y) target-guided binning of x that maximizes $AUC(y, x)$ seems, therefore, to be just finding unique values of x: x_u, computing target rates y_{xu} for all x_u, and replacing x with the positions they appear in x_u sorted by y_{xu}. Assuming Matlab coding syntax, finding x_T becomes straightforward:

```
[∼,j]=sort(y_xu); [∼,x_T]=ismember(x,x_u(j));
```

Although such logic is in principle correct it ignores a fundamental property of a good predictor: the generalization ability and would likely fail on two accounts. First, the binning for numerical variable has to provide the mapping for the entire domain, not just unique values observed in the training set, otherwise the binning is unable to allocate previously unseen values of x into any bin. Second, a feature binned as described above essentially over-fits the observed data with the degree dependent on the the number of unique inputs x_u. In the extreme case of (almost) all unique values, that is typical for continuous floating point features, the target rates will be

[1] https://knowledgepit.ml/fedcsis-2023-challenge

extremely unreliable as computed on the basis of just a single or a few examples and would wildly differ in the unseen testing set, on which the binning should be designed to perform well.

To address these two cases our target guided binning would operate on the intervals (subsets) that at all times span the entire domain (universal set) of numerical (categorical) feature and attempt to efficiently and optimally define their numbers, edges (set members) and label-permutation that maximises AUC, but not over the training data (x_u, y_{xu}), on which the bins are built and merged, but on the previously unseen validation set (v_u, y_{vu}). The bottom-up approach is proposed for our TGB algorithm, which starts from singleton intervals (subsets) exclusively covering all unique values x_u and then greedily merge to maximise $AUC(y_x, x_T)$ but only until monitored validation set performance $AUC(y_v, v_T)$ starts falling.

A. Massively parallelized TGB implementation

Target guided binning has been developed with extreme efficiency and scalability in mind. Each feature is binned independently and in parallel using the same binary target as a guidance in such a way that the AUC score of its binned representation against the target is maximized in a generalization sense, i.e. AUC is measured via cross-validation on different data partitions than those on which the bin definitions were built.

TGB starts from building singleton intervals (or subsets for categorical features) containing all unique values and reordering them along the rising likelihood of positive target. Then the intervals/subsets proceed to greedy merger process which continues until no further gain in validation AUC can be achieved through further mergers. Figure 1 illustrates a sample TGB process for numerical feature which starts from individual singleton intervals and then proceeds through the greedy neigboring interval merger until no more validation AUC improvement is possible, which in the depicted sample scenario converges to 6 intervals. The AUC-optimized intervals are then mapped to ordinal bin labels calibrated or scaled according to user preferences yet always monotonic with the conditional positive target rate. Feature 1 also illustrates readiness for massively parallelized implementation of the TGB process which could be efforlessly applied along the feature level, cross-validation partitions or testing for optimal merger along running pairs of neighboring intervals.

The missing data (NaNs) are mapped to a bin that has the closest posterior target probability as the one observed for missing data. Similarly previously unseen data are provisioned to receive bin that has the target posterior probability the closest to the target prior. Greedy interval merger follows very fast vectorized test of the impact of the decomposed AUC score implemented on matrix formulation that allows to compute all simulated pairs merger in a single step per round resulting in tabular formatted bin definitions mapping all original feature domain into optimised incrementally summarized intervals/subsets labelled with target-monotonic ordinal risk levels. Given the final bin definitions, transforming any new data into bins is equally lightning fast and, importantly, represented by *uint8* data type reducing the binned data complexity to just 1 byte per value. Compared to the original data typically coming in double precision or text format, TGB typically reduces the size of the memory required to hold the data around 100-fold, while obfuscating the original values behind ordinal risk mask.

IV. Wide-margin voting classifier composition

Constructing a robust classifier based on TGB-transformed (binned) data is straightforward and in its simplest form can be executed by a simple voting i.e. by adding up all feature bin values (risk votes). Further AUC-measured performance gains can be achieved by more or less sophisticated feature selection strategies, which for the voting classifier simply translates into finding a sum of binned features that maximises AUC against the binary target. Two highly scalable heuristic optimisation methods have been developed to execute such robust additive selection of binned features and both can deal with hundreds of thousands of features in seconds if supported by multi-core parallel processing and/or dedicated capable GPU.

A. Greedy forward selection (GFS)

Greedy forward selection of binned features starts from the strongest feature and keeps adding features that maximally improve the appended sum's AUC against the target in each round until this is no longer possible. The process of testing for optimal addition is fast since the current best subset is constantly retained in the form of collapsed running sum and stored indices of selected members, while testing the AUC improvement when adding another binned feature is vectorized and massively parallelized with additional speedups possible when executed on the GPU. In practical applications, when faced with tens to hundreds of thousands of features, each round of finding a binned feature that maximally improves the pool's AUC usually takes around 1s. In the latest implementation this greedy search was additionally improved by reducing the data type of vectors holding the sums to *uint16* and allowing each feature to be added multiple times - thereby equipping the method with a fast feature weighting capability.

B. Fast probability based incremental learning (FPBIL)

Probability based incremental learning (PBIL) is a simple population based heuristic optimization that is perfectly suited for simple evaluation functions based on binary encoded feature selection. Beyond that fit, PBIL has been chosen to help with feature selection also for two other reasons. Its critical operation is constant sampling from the probability vector that involves generation of random number matrices of enormous sizes that can be massively accelerated on the GPU. Moreover, evaluation of the population of solutions at each generation involves preparation of the intermediate voting sums corresponding to binarized selection vectors sampled from the evolving probability vector, all of which has been very efficiently vectorized and passed on to equally optimized and parallelized evaluation of the AUC. Operating such PBIL on the GPU with the Philox based random number generator

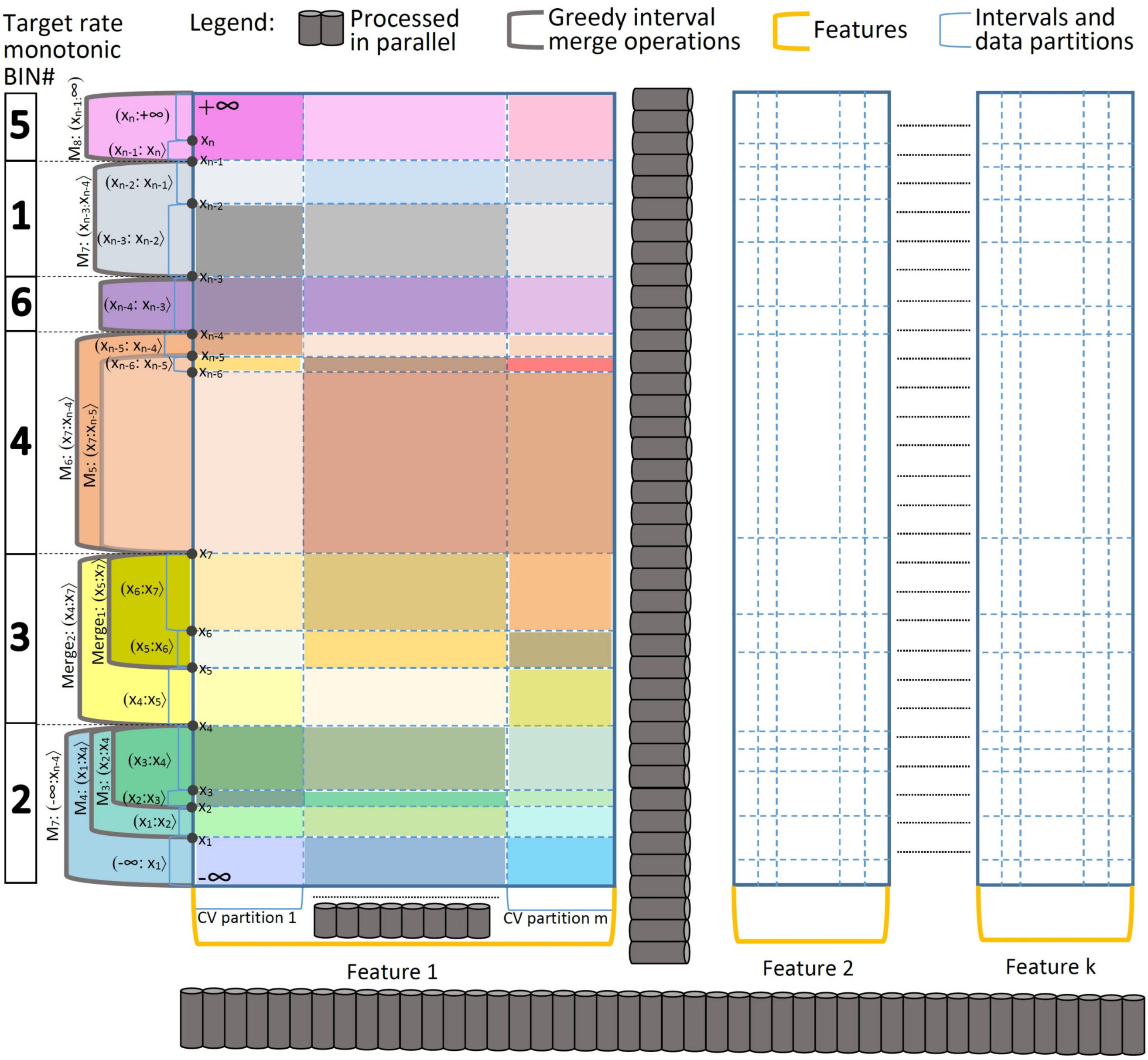


Fig. 1. An annotated example of the target guided binning (TGB) for numerical feature x. In the orginal domain $(-\inf, +\inf)$ first all unique values: $\{x_1 : x_n\}$ are found and wrapped within singleton intervals: $(-\inf : x_1], (x_1, x_2], .., (x_n : +\inf)$. Then neigboring intervals are greedily merged along multiple cross-validation partitions until the validation sets AUC against the target no longer improves leaving the final optimised intervals ordinally labelled to represent target-monotonic risk levels. The process is ready for massive parallelization along multiple dimensions: independent features, cross-validation partitions and the neigboring pairs of intervals examination for optimal merger. Distinct colors are representing the conditional target rate heatmap and the effect of its aggregation after mergers.

on a population of 1000 100k-elements solutions with a learning rate of 0.5-1 typically converges after a couple of hundreds of generations at the pace of multiple generations per second. Compared to the greedy forward selection which normally converges with up to 200 out of 100k features, accelerated PBIL-based selection converges with thousands of features and typically better AUC-score

C. Criteria for feature selection evaluation

Ideal evaluation criterion for feature selection is the actual classifier performance for the selected features. The only reason why much simpler proxy measures are normally used is that evaluating the classifier with different set of features is expensive and normally requires a rebuilt of the whole model from scratch to extract new classifier output. For our case, however, the voting classifier only needs to add the newly selected feature values to the cumulative sum from features already in the pool to update the classifier output, hence it follows an online update process and is therefore very fast. For this reason TGB with voting enables us to use directly the powerful threshold independent classifier performance measures like AUC as a feature selection criterion which supports classifier robustness while keeping it simple and fast.

1) Area under the ROC curve (AUC): AUC measure has already been discussed above as the prime threshold-less indicator of the overall predictive power of a feature x against the binary target y. In the context of target guided binning for the reason of being AUC-optimized and also due to the fact that computing AUC for binned features likely involves fewer unique bin-label ordinal values, we have introduced a dedicated summarized representation of the relationship between x and y for the purpose of AUC computation called feature predictive structure $P = [u, c, v]$ that contains a sorted list u of unique values of x, and the corresponding lists of their counts c as well as the counts of the positive targets $v = \sum(y|x)$. For simplicity we will replace the original x and y with their summarized representation in P such that $P = [x, c, y]$. Note that such summarized representation significantly reduces the sizes of both the feature x with respect to the target y down to the essential statistics sufficient to evaluate its full predictive power. Given P, the cumulative true and false positive vectors can be readily computed for multiple features or targets by this vectorized Matlab code:

```
tp=[0;cumsum(flipud(y))];
fp=bsxfun(@minus,[0;cumsum(flipud(c))],tp);
tp=bsxfun(@rdivide,tp,sum(y));
fp=bsxfun(@rdivide,fp,sum(y));
```

such that the AUC can be accurately and rapidly computed for multiple features or targets using a 1-liner:

```
auc=sum(diff(fp).*(tp(1:end-1,:)+tp(2:end,:))/2);
```

2) Kolmogorov-Smirnov Distance (KSD): Kolmogorov-Smirnov distance, test or statistic in our context expresses simply the maximum absolute difference between the cumulative rate of positive targets and the cumulative rate of negative examples along the sorted unique inputs x. Given our compact predictive structure $P = [x, c, y]$ KSD can be rapidly computed for multiple inputs/targets using:

```
y=cumsum(y); s=y(end); c=cumsum(c)-y; n=c(end)-s;
ksd=bsxfun(@rdivide,y,s)-bsxfun(@rdivide,c,n);
ksd=max(abs(ksd));
```

3) Classification Impurity Score (CIS): This new measure utilizes the specificity of working with binned feature votes and is designed to stimulate stable wide margin classifier especially for very high performance close to AUC=1. The measure works on the sorted predictor outputs (sums of bin votes) and focuses on the interval, within which samples are not classified 100% correctly. For every sample falling within this interval it then simply adds up distances between the prediction (sum) for these samples and the interval boundary that if reached would eliminate the misclassification for any threshold. Since our voting classifier simply holds the sum of selected binned features, the logic of this measure is to evaluate how many votes need to be added (for false negative) to or subtracted (for false positive) from the current sum of votes such that the sample would be correctly classified irrespective of the applied threshold. Formally, assuming sorted classifier outputs x_i, $i = 1, .., n$, the corresponding binary targets y_i and the interval of indices $j = k, .., l$ such that $1 \leq k < l \leq n$ and

$$\begin{aligned} &\underset{s:x_s \leq x_k}{\forall} y_s = 0 \\ &\underset{s:x_s \geq x_l}{\forall} y_s = 1 \end{aligned} \tag{3}$$

then the CIS can be defined by:

$$CSI = \sum_j (x_l - x_j) y_j + \sum_j (x_j - x_k) \bar{y_j} \tag{4}$$

Using Matlab code the above definition can be readily captured as follows:

```
i=find(x<x(find(y,1,'first')),1,'last');
if isempty(i) i=1; end
j=find(x>x(find(~y,1,'last')),1,'first');
if isempty(j) j=numel(x); end
l=i:j;
cis=sum(x(j)-x(l(y(l)))) + sum(x(l(~y(l)))-x(i));
```

Note that in case of 100% accurate classification the impurity measure could be adjusted to receive negative values proportional to the gaps or margins in votes that needed to be bridged to observe classification impurity, hence such measure can be very effective for very high performance wide-margin classification with AUC scores very close to 1, which happens to be the case of the FedCSIS' 2023 Challenge.

V. Experimental results

All features have been generated in the exactly same form capturing the frequency (counts) of unique values observed within the log files. For the datetime and other continuous numerical variables the domain has been split into 100 equipercentile intervals and the derived features measured de-facto frequencies of observed percentile values. For features listing all open filenames and libraries with paths the two variants of unique elements were applied: the whole unique paths separated by commas and, in the second variant, all the unique path sub-strings separated by \ character. Such feature engineering process resulted in over 300k 1-hot-encoded style features that after reduction by eliminating duplicates and constant features shrunk to a set of about 40k of unique raw features. These features have then been passed on to the TGB process allowing up to 20 (and later 100) unique bins applied only on the training set of 15027 labelled samples and resulted with bin definitions re-applied on both the training and testing sets to achieve the final transformed training and testing sets taking ordinal values from 1 to 20 (100).

Feature selection process followed on the binned training set in all combinations of the presented feature selection methods and evaluation metrics, however, we only show the results for AUC and CIS since KSD produced the results similar to AUC.

The feature subset sums obtained as a result of all the selection-evaluation combinations along with outputs from many other variants of restricted feature subsets and gradient boosting models applied for comparison have been normalized within (0,1) interval and submitted for evaluation on the preliminary testing set containing only 10% of all testing samples. The feedback received was in line with the results received from the validation sets with the top scored model variants, notably achieving leaderboard's top AUC=1.0 submitted as final solutions for the evaluation on the full testing set.

Comparative AUC performance results of our target guided binning (TGB) with gradient boosting (GB) variants classifiers and various combinations of feature-selection and evaluation criteria are presented in Table I. For both the training and validation sets we have observed GFS performing slightly better with AUC than CIS criterion however the opposite was observed for FPBIL selection method. The FPBIL applied with CIS metric typically returned solutions of a couple of thousands of features with the final converged impurities of just about 50-500, while starting from impurities in the order of millions. On the other hand the GFS typically converged with about only 100-200 features, for which the added score produced the AUC reaching extremely close to 1. What produced the best results, however, was sequentially applied GFS interchangeably with AUC and CIS criteria, until no further improvement in the validation AUC could be achieved. Although results for 100-bin TGB appear to show slightly better validation results than for TGB with up to 20 bins, final testing revealed later that 20-bin TGB could have performed better, i.e. 100-bin TGB appeared to be slightly over-fitted with too fine granularity and the best results could be expected somewhere in between for example 50-bin TGB. Although in our validation TGB on its own i.e. with simple voting outperforms all GB model variants, final testing revealed that CATBoost could climb to similar performance levels if applied on top of the 20-binned rather than raw features and could most likely have improved our final combined testing score thanks to a significant diversity with TGB-generated results.

TABLE I
COMPARATIVE PERFORMANCE OF GB/TGB VARIANTS COMBINED WITH DIFFERENT FEATURE SELECTION/EVALUATION CRITERIA.

Classifier	FSelection	Criterion	AUC Score	
			VAL	TST
XGBoost	ALL-LogLoss		0.9992	0.9980
LGBM	ALL-LogLoss		0.9967	0.9954
CATBoost	ALL-LogLoss		0.9994	0.9983
TGB20-SUM	GFS	AUC	0.9996	0.9985
	GFS	CIS	0.9994	0.9982
	FPBIL	AUC	0.9992	0.9981
	FPBIL	CIS	0.9994	0.9983
	GFS	AUC-CIS	0.9997	0.9991
TGB100-SUM	GFS	AUC-CIS	0.9997	0.9989
	FPBIL	CIS	0.9998	0.9979
XGB-BIN20	ALL-LogLoss		0.9992	0.9985
CATBoost-BIN20	ALL-LogLoss		0.9994	0.9993
MEAN(TOP5(TGB))	NA		1	0.9997

VI. CONCLUSION

Presented target guided binning rapidly transforms any input evidence into a robust array of 1-feature micro-predictors of the binary target and offers readily available, high quality classification by voting with ordinal-risk represented binned feature outputs in near-real time. Further performance gains are available through fast parallelized gready feature selection and gpu-optimized FPBIL features selection methods utilizing both AUC and newly introduced CIS as evaluation criterion to achieve stable high margin perfomance. In the competitive setup of detecting cyber-security attacks on the IoT devices based on log files data the presented methodology appears to consistently beat gradient boosting models in all aspects: the speed of building the model, the classification performance, simplicity, transparency and added security layer, topping the preliminary evaluation on the leader-board of the FedCSIS'2023 Challenge with the score of AUC=1 and eventually scoring the 2^{nd} place with AUC=0.9997 in the final testing.

REFERENCES

[1] F. Alaba, M. Othman, I. Hashem, F. Alotaibi. Internet of Things security: A survey, *Journal of Network and Computer Applications* 88:10-28, 2017.

[2] M. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A. Sheth, Machine learning for internet of things data analysis: a survey, *Digital Communications and Networks*, 2018.

[3] M. Garadi, A. Mohamed, A. Ali, X. Du, I. Ali and M. Guizani. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security, *IEEE Communications Surveys & Tutorials*, 2020.

[4] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent In S.A. Solla, T.K. Leen and K. Müller (eds): Advances in Neural Information Processing Systems 12: 512–518, MIT Press, 1999.

[5] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5): 1189-1232, 2001.

[6] D. Ruta, M. Liu, L. Cen. Feature Engineering for Prediction of Frags in Tactical Games. *Proc. Int. Conf. 2023 IEEE International Conference on Multimedia and Expo*, 2023.

[7] D. Ruta, M. Liu, L. Cen and Q. Hieu Vu. Diversified gradient boosting ensembles for prediction of the cost of forwarding contracts. *Proc. 17th Int. Conf. on Computer Science and Intelligence Sys.*, pp 431-436, 2022.

[8] D. Ruta, L. Cen, M. Liu and Q. Hieu Vu. Automated feature engineering for prediction of victories in online computer games. *Proc. Int. Conf on Big Data*, pp 5672-5678, 2021.

[9] Q. Hieu Vu, D. Ruta, L. Cen and M. Liu. A combination of general and specific models to predict victories in video games. *Proc. Int. Conf. on Big Data*, pp 5683-5690, 2021.

[10] D. Ruta, L. Cen and Q. Hieu Vu. Deep Bi-Directional LSTM Networks for Device Workload Forecasting. *Proc. 15th Int. Conf. Comp. Science and Inf. Sys.*, pp 115-118, 2020.

[11] D. Ruta, L. Cen, Q. Hieu Vu. Greedy Incremental Support Vector Regression. *Proc. Fed. Conf. on Comp. Sci. and Inf. Sys.*, pp 7-9, 2019.

[12] Q. Hieu Vu, D. Ruta and L. Cen. Gradient boosting decision trees for cyber-security threats detection based on network events logs. *Proc. IEEE Int. Conf. Big Data*, pp 5921-5928, 2019.

[13] Q. Hieu Vu, D. Ruta, A. Ruta and L. Cen. Predicting Win-rates of Hearthstone Decks: Models and Features that Won AAIA'2018 Data Mining Challenge. *Int. Symp. Advances in Artificial Intelligence and Apps (AAIA)*, pp 197-200, 2018.

[14] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics* 1:80–83, 1945.

[15] H.B. Mann, D.R. Whitney. On a test whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics* 18: 50–60, 1947.

[16] C.X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In Int. Joint Conf. on Artificial Intelligence, pp 519–526, 2003.

[17] Z. Yang, Q. Xu, S. Bao, Y. He, X. Cao, Q. Huang. Optimizing Two-way Partial AUC with an End-to-end Framework. IEEE Trans. on Pattern Analysis and Machine Intelligence 45:10228-10246, 2023

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A.V. Dorogush and A. Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio at al (eds.): Advances in Neural Information Processing Systems 31:6638–6648, Curran Associates, Inc, 2018.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1307–1310
DOI: 10.15439/2023F7136
ISSN 2300-5963 ACSIS, Vol. 35

Spotting Cyber Breaches in IoT Devices

Sławomir Pioroński
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Uniwersytetu Poznańskiego 4 Street
61-614 Poznań, Poland
Email: slawomir.pioronski@amu.edu.pl

Tomasz Górecki
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Uniwersytetu Poznańskiego 4 Street
61-614 Poznań, Poland
Email: tomasz.gorecki@amu.edu.pl

***Abstract*—In the ever-growing realm of the Internet of Things (IoT), ensuring the security of interconnected devices is of paramount importance. This paper discusses the process of spotting cyber breaches in IoT devices, a significant concern that needs urgent attention due to the susceptibility of these devices to hacking and other cyber threats. With billions of IoT devices worldwide, the detection and prevention of cybersecurity breaches are critical for maintaining the integrity and functionality of networks and systems.**

In this paper, we showcase the outcomes achieved by employing the LightGBM technique for a cyberattack prediction challenge, which was a part of the FedCSIS 2023 conference.

***Index Terms*—cybersecurity, data mining competition, LightGBM**

I. Introduction

As we step further into the digital era, the Internet of Things (IoT) continues to reshape the landscape of our daily lives, driving advancements in various sectors such as healthcare, transportation, smart homes, and industrial automation. Despite the remarkable benefits, the rapid proliferation of IoT devices has significantly heightened the stakes in the domain of cybersecurity. The interconnected nature of these devices poses unique vulnerabilities, making them attractive targets for cyberattacks. An essential part of combating this growing threat involves the ability to effectively identify and predict cybersecurity breaches in IoT systems.

Numerous machine learning methodologies can be deployed for the prediction of cyberattacks [1]. However, we opted for a gradient boosting algorithm, specifically LightGBM [2], due to its impressive combination of speed and precision. In this paper, we aim to highlight the effectiveness of our strategy. Our discussion will serve to underscore the integral role of data science in augmenting cybersecurity measures in an increasingly interconnected world. By delving into this topic, we hope to provide valuable insights for future research endeavors and practical applications aimed at advancing the field of cybersecurity for IoT.

The organization of this paper is as follows: after this introduction, we review relevant literature and provide a brief overview of the FedCSIS 2023 challenge. In Section IV, we delve into the processes involved in data handling and preparation. We detail the model deployed in our experiment in Section V, followed by a comprehensive presentation of our findings in the succeeding section. We conclude in Section VII by summarizing our observations and contemplating potential avenues for future exploration.

II. Related work

The practice of automatically detecting cyberattacks has a well-established history in the field. A diverse range of methods have been employed to accomplish this task. It has been suggested through numerous studies that machine learning techniques could be potentially beneficial, with many researchers opting to use unsupervised algorithms to navigate identification challenges [3], [4]. However, there is a notable drawback to using unsupervised machine learning methods for recognizing anomalies in a network, distinguishing between standard cyberattacks, and detecting outliers. The sparse occurrence of these outliers can have an asymmetric impact on both the success rate and the identification of abnormalities.

To achieve more dependable results, supervised machine learning methods are often employed. These algorithms are trained using metadata with labels indicating whether the given instances have previously been classified as cyberattacks. Examples of such supervised learning algorithms include Support Vector Machines and Artificial Neural Networks [5], Random Forests [6], the k-Nearest Neighbor (k-NN) technique [7], the Naive Bayes algorithm [8], and LightGBM [9].

In our solution, we decided to use LightGBM (Light Gradient Boosting Machine) due to several reasons [2], [10], [11]:

- **Efficiency**. LightGBM uses a novel technique of Gradient-based One-Side Sampling (GOSS) to filter out the data instances for finding a split value, which can result in a more efficient learning process. This is particularly useful when dealing with large volumes of data generated by IoT devices.
- **High Performance**. LightGBM can handle large data sets while maintaining high efficiency. It uses the leaf-wise tree growth algorithm, unlike the traditional level-wise tree growth algorithm, which can result in a better performance in terms of speed and accuracy.
- **Handling Categorical Features**. LightGBM can naturally handle categorical features, which can be very beneficial when dealing with IoT data, as IoT devices can produce a variety of data types.

- **Scalability**. LightGBM is highly scalable and can work well with large datasets that often characterize IoT networks.
- **Accuracy**. LightGBM can achieve lower prediction errors by employing complex tree architectures, boosting its accuracy, which is crucial for detecting subtle signs of cyberattacks in IoT networks.

It is also worth noting that gradient-boosting models have been used in previous data mining competitions. In the IEEE BigData 2019 Cup: Suspicious Network Event Recognition, the best solutions used tree-based boosting models [12]. In particular, first place went to an ensemble of two models [13], LightGBM and XGBoost [14]. In another competition, the FedCSIS 2020 Challenge: Network Device Workload Prediction [15], the situation was similar. The 2nd and 3rd place solutions used XGBoost models. Of course, it is important to remember that proper preprocessing is required to use these models. Furthermore, we have used a similar approach (LightGBM + appropriate preprocessing) for other competitions with outstanding results [16].

III. Challenge description

A. Data

The data provided consists of CSV table log files, each with a randomized uuid4 name. All original timestamps have been standardized to a specific timestamp, which is 2023-04-12-00:00:00. A separate TXT file was provided for the training set, containing the names of log files associated with cyber attacks. After the competition concluded, similar information regarding the test set was also made available. The sizes of the datasets are as follows:

- training data: 15 027 files (522 indicates cyberattack),
- test data: 5 017 files (176 indicates cyberattack).

As we can see, a small number of files indicated a cyberattack (3.48% for the training dataset and 3.50% for the test dataset).

B. Task

Our goal is to develop an accurate method that can detect cyberattacks on an IoT system based on its logs.

C. Evaluation

In this competition, participants submitted their solutions to the online evaluation system as text files that included predictions for the test instances. Each test instance in the solution file was accompanied by a single number within the $[0, 1]$ range, representing the probability of a cyberattack. These predictions were arranged according to the lexicographic ordering of the log files from the test set.

The effectiveness of the submitted entries was assessed using the ROC AUC (Receiver Operating Characteristic Area Under Curve) metric, a widely used evaluation metric for binary classification problems [17]. The ROC curve is a plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the true positive rate (TPR), against the false positive rate (FPR), at various threshold settings. Precisely

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}},$$

where TP is the number of True Positives, FN is the number of False Negatives, FP is the number of False Positives and TN is the number of True Negatives.

Calculation of the AUC (Area Under the ROC Curve) is slightly more complex as it involves integration over all possible classification thresholds. But practically, it's usually calculated using the trapezoidal rule [18]. An AUC of 1 indicates a perfect classifier, while an AUC of 0.5 signifies a classifier that performs no better than random chance [19].

Initial scores were evaluated via the KnowledgePit online platform [20] and published on a challenge leaderboard calculated on a small subset of the test set fixed for all participants. The final score was published after the challenge using the remainder of the test data set.

IV. Data preprocessing

Due to the format of the data (we have a separate file for each observation, and therefore a separate table with data), we had to process it in an appropriate way. We focused on the approach to have one row of data for a single observation. Each file contains 40 columns: 21 numeric, 17 string, and 2 with only null values (based on training data). We skip these two columns with nulls and now proceed to preprocess the data by type.

A. Numerical data

We focused on the numerical data first. For each file, we took the smallest, largest, and average values (omitting features with the same values, we obtained 17 features). With this simple approach, we will get very good predictions. So, we now move on to data of string type.

B. String data

The main idea was to focus on finding significant differences in this data type without considering numerical data. Of particular note is the "Custom_openFiles" feature. To begin with, we selected unique values for this feature separately for the files that represented the logs with and without the attack. Then, from the unique values from files with attacks, we removed all the values that were present in files without attacks. Finally, for each file, an indicator was created indicating whether any value from the "Custom_openFiles" column belonged to that set. Using the same set this was repeated for the test data. Passing this indicator as the probability of a cyberattack, we obtained a score of 96.77% (by ROC AUC measure) on the public part of the test set.

C. *Addtional preprocessing*

In the test set, the shortest log file contains 68 items. So we took only those files from the training set that are not shorter than it. Thus, we get rid of 65 (0.43%) files from the training set. In addition, we replaced one of the string-type features with a numeric one. Namely, in the feature "SYSCALL_exit _hint" we have numeric and string type values. So in place of the string, for example, "ENOENT(No such file or directory)", we inserted nulls. Then we calculated the average, minimum, and maximum as in Section IV-A.

V. Model

We used the gradient boosting model for testing, and the choice was LightGBM [2]. We used Microsoft's FLAML library [21] to optimize the hyperparameters.

With the above preprocessing, the models achieve high predictive quality very quickly. We can see a comparison of the performance of the models for different subsets of features with hyperparameter optimization taking 3 minutes in Table I.

TABLE I
STRATIFIED 5-FOLD CROSS-VALIDATION RESULTS FOR DIFFERENT FEATURE SETS (3 MIN OF HYPERPARAMETER OPTIMIZATION, AUC MEASURE) AND THE RESULT ON THE TEST SET.

Feature set	Cross-validation	Test set
IV-A	0.99932	**0.99954**
IV-A + IV-B	0.99993	0.99951
IV-A + IV-C	0.99813	0.98224
IV-A + IV-B + IV-C	**1.00000**	0.99603

In Table II, we have a list of optimized hyperparameters and values for the best model from Table I.

TABLE II
FINAL MODEL HYPERPARAMETERS FOR FEATURE SET IV-A AT 3 MINUTES OF OPTIMIZATION (TO FIVE DECIMAL PLACES).

Hyperparameter	Value
n_estimators	1098
num_leaves	120
min_child_samples	5
learning_rate	0.19275
max_bin	1023
colsample_bytree	0.73337
reg_alpha	0.00098
reg_lambda	0.24821

VI. Experimental results

We see that the first two cases in Table I gave the best results on the test set. In both cases, we have another feature that is most relevant according to gain importance [14]. The most significant feature for the 1st model is "SYSCALL_pid_max" (which is the maximum of the "SYSCALL_pid" feature from each file) as we see in Figure 1.

On the other hand, the graph for the second model appears quite similar, except that the newly added feature (indicator based on "Custom_openFiles", described in IV-B) is now positioned at the beginning.

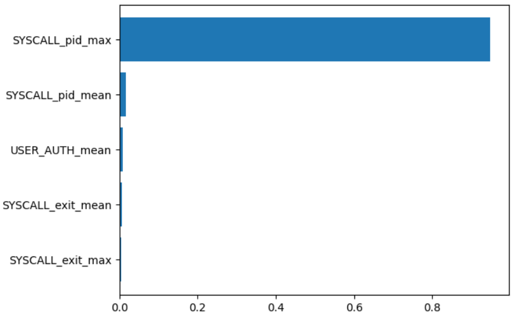


Fig. 1. Top 5 features by gain for the first model. The values were divided by the sum of all gains.

We now set the search times for hyperparameters to 30 minutes. We can see the results in Table III.

TABLE III
STRATIFIED 5-FOLD CROSS-VALIDATION RESULTS FOR DIFFERENT FEATURE SETS (30 MIN OF HYPERPARAMETER OPTIMIZATION, AUC MEASURE) AND THE RESULT ON THE TEST SET.

Feature set	Cross-validation	Test set
IV-A	0.99946	**0.99959**
IV-A + IV-B	0.99996	0.99901
IV-A + IV-C	0.0.99888	0.0.98733
IV-A + IV-B + IV-C	**1.00000**	0.99660

The outcomes show minimal variation from the 3-minute version, as demonstrated more accurately in the learning curve of one feature set presented in Figure 2.

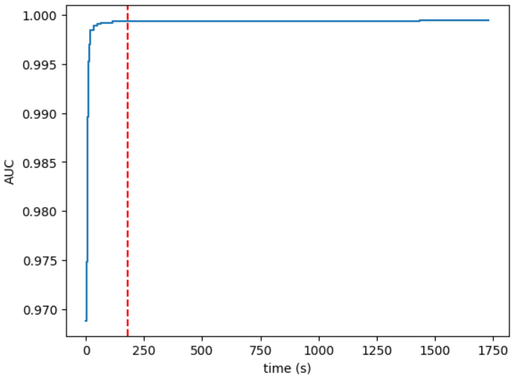


Fig. 2. Learning curve for model trained on IV-A feature set. The red line marks 3 minutes.

We have 4.5 times more false positives than false negatives (in the case of the first model), which is good behavior since it is better to verify claims with no attacks than to omit those with attacks. We can see this in the confusion matrix in Figure 3.

VII. Conclusions

To detect cyberattacks, we utilized the renowned LightGBM model along with some data preprocessing. Our approach

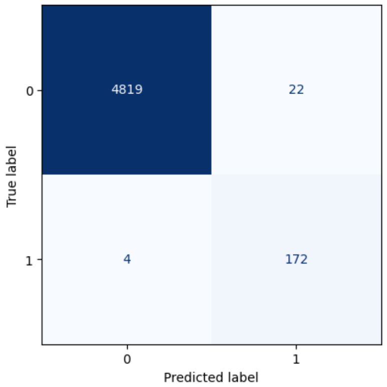


Fig. 3. Confusion matrix for model trained on IV-A feature set.

was successfully trained within a mere 3 minutes, securing a commendable 3rd place in the competition. The top 4 results were closely matched, with only a marginal difference of 0.0002 between the following positions.

The achieved result was already commendable, making it difficult to anticipate a substantial enhancement in performance. Nonetheless, for future endeavors, it is crucial to concentrate on extracting valuable insights from string-type attributes. Furthermore, it is possible to expect that more advanced data preprocessing techniques may contribute to marginal enhancements in performance.

References

[1] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*, ser. Springer Series in Statistics. Springer, 2009.

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017, pp. 3146–3154.

[3] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in *Mobile Networks and Management*, J. Hu, I. Khalil, Z. Tari, and S. Wen, Eds. Cham: Springer International Publishing, 2018, pp. 30–44.

[4] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Network and Distributed Systems Security (NDSS) Symposium*, 2018.

[5] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 2016, pp. 1–6.

[6] Y. Meidan, M. Bohadana, A. Shabtai, M. Ochoa, N. O. Tippenhauer, J. D. Guarnizo, and Y. Elovici, "Detection of unauthorized IoT devices using machine learning techniques," 2017.

[7] M. Aljabri, A. A. Alahmadi, R. M. A. Mohammad, F. Alhaidari, M. Aboulnour, D. M. Alomari, and S. Mirza, "Machine learning-based detection for unauthorized access to IoT devices," *Journal of Sensor and Actuator Networks*, vol. 12, no. 2, 2023.

[8] C. Malathi and I. N. Padmaja, "Identification of cyber attacks using machine learning in smart iot networks," *Materials Today: Proceedings*, vol. 80, pp. 2518–2523, 2023.

[9] M. Al-kasassbeh, M. A. Abbadi, and A. M. Al-Bustanji, "LightGBM algorithm for malware detection," in *Intelligent Computing*, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham: Springer International Publishing, 2020, pp. 391–403.

[10] A. Anghel, N. Papandreou, T. Parnell, A. D. Palma, and H. Pozidis, "Benchmarking and optimization of gradient boosting decision tree algorithms," 2019.

[11] C. Bentéjac, A. Csörgo, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2020.

[12] A. Janusz, D. Kałuza, A. Chądzyńska-Krasowska, B. Konarski, J. Holland, and D. Ślęzak, "Ieee bigdata 2019 cup: Suspicious network event recognition," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019. doi: 10.1109/BigData47090.2019.9005668 pp. 5881–5887.

[13] Q. H. Vu, D. Ruta, and L. Cen, "Gradient boosting decision trees for cyber security threats detection based on network events logs," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019. doi: 10.1109/BigData47090.2019.9006061 pp. 5921–5928.

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2939672.2939785 pp. 785–794.

[15] A. Janusz, M. Przyborowski, P. Biczyk, and D. Ślęzak, "Network device workload prediction: A data mining challenge at knowledge pit," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*, 2020. doi: 10.15439/2020F159 pp. 77–80.

[16] S. Pioroński and T. Górecki, "Using gradient boosting trees to predict the costs of forwarding contracts," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022. doi: 10.15439/2022F299 pp. 421–424.

[17] K. A. Spackman, "Signal detection theory: Valuable tools for evaluating inductive learning," in *Proceedings of the Sixth International Workshop on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 160–163.

[18] R. L. Burden, *Numerical analysis*, 8th ed. Thomson Brooks/Cole, 2005.

[19] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[20] "FedCSIS 2023 challenge: Cybersecurity threat detection in the behavior of IoT devices," https://knowledgepit.ml/fedcsis-2023-challenge/, accessed: 2023-07-05.

[21] C. Wang, Q. Wu, M. Weimer, and E. Zhu, "FLAML: A fast and lightweight AutoML library," in *MLSys*, 2021.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1311–1315

DOI: 10.15439/2023F4457
ISSN 2300-5963 ACSIS, Vol. 35

Gradient boosting models for cybersecurity threat detection with aggregated time series features

Ming Liu, Ling Cen, Dymitr Ruta
EBTIC, Khalifa University, UAE
{liu.ming,cen.ling,dymitr.ruta}@ku.ac.ae

***Abstract*—The rapid proliferation of Internet of Things (IoT) devices has revolutionized the way we interact with and manage our surroundings. However, this widespread adoption has also brought forth significant cybersecurity challenges. IoT devices, with their interconnectedness and varying functionalities, present a unique threat landscape that requires tailored detection techniques. Traditional approaches to cybersecurity, primarily focused on network monitoring and anomaly detection, often fall short in effectively identifying threats originating from IoT devices due to their dynamic and complex behaviors. This paper addresses our solution for FedCSIS 2023 Challenge: Cybersecurity Threat Detection in the behavior of IoT Devices. First, we aggregated time series features, and then at the feature selection stage, we filtered and combined different categorical and numerical features to generate four different feature sets. The Gradient boosting models, i.e. lightgbm, catboost and xgboost, are applied and trained individually with hyper-parameter tuning. The final three submissions are two best individual lightgbm models with the AUC scores of 0.9999 and 0.9998, respectively on the different feature sets, which secured the 4th place with a final score of 0.9993, and one ensemble result with a AUC score of 0.9998 from combination of xgboost, catboost and lightgbm, which has the final score of 0.9997 while unluckily was missing in the final three evaluation entries.**

***Index Terms*—Cybersecurity threat detection, Gradient Boosting Trees, CatBoost, XGBoost, LightGBM, Stacking, Ensemble Learning.**

I. Introduction

WITH the exponential growth of Internet of Things (IoT) devices, ensuring the security and integrity of these interconnected systems has become a paramount concern. The dynamic and heterogeneous nature of IoT devices presents a unique challenge for traditional cybersecurity approaches.

The survey paper in [1] comprehensively discusses the security issues faced in IoT environments and presents an overview of the existing security mechanisms and solutions. It covers various aspects of IoT security, including authentication, access control, privacy preservation, secure communication, and intrusion detection. The paper also highlights the unique security challenges posed by IoT and provides insights into ongoing research efforts to address those challenges.

Machine learning (ML) techniques have emerged as promising solutions for addressing IoT device security challenges. ML algorithms can analyze vast amounts of data collected from IoT devices to identify patterns, detect anomalies, and make predictions. By leveraging ML, IoT security can be improved through proactive threat detection, effective intrusion detection, and robust anomaly detection. In [2], various machine learning methods applied into IoT have been explored and a Support Vector Machine (SVM) application use case is presented. In [3], this survey paper offers insights into the application of various ML/DL techniques, such as deep learning, support vector machines, and decision trees, in enhancing IoT device security. It discusses the challenges faced in securing IoT devices and highlights the potential of ML methods in addressing those challenges. The paper also provides an overview of different use cases, datasets, and evaluation metrics used in the context of IoT device security.

In this paper, a model utilizing gradient boosting decision trees in conjunction with effective feature engineering and optimized model hyper-parameters, forming an ensemble learning approach has been developed for predicting the cybersecurity breaches to address the task given in the FedCSIS 2023 Challenge [4][1]. The objective of the challenge, which is sponsored by Łukasiewicz Research Network - Institute of Innovative Technologies EMAG and EFIGO sp. z o.o. companies, is to detect the cybersecurity breaches in log data from IoT devices.

The remainder of the paper is organized as follows. The FedCSIS 2023 Challenge is briefly described in Section II. Time series data aggregation and feature engineering is presented in Section III, followed with the description of the gradient boosting models and ensemble learning in Sections IV and V, respectively. The experimental results in Section VI. Concluding remarks are provided in Section VII.

II. FedCSIS 2023 challenge

The FedCSIS 2023 data mining competition focused on the detection of cybersecurity breaches in log data from IoT devices. The data sets contain 1-minute logs of all related system calls. The task for the competition participants is to develop a model that assesses the chances that a cyber attack was ongoing during the monitored period. Such a model could play a vital role in improving the safety of IoT systems. The knowledgepit.ai platform, on which the competition was hosted operated a leaderboard, which provided the feedback to the competitive model prediction submissions in a form of the preliminary AUC score [2] computed over the small subset of the testing set, while the final AUC score for the complete testing set - constituting the final results, were provided after the submissions' closure.

[1] https://knowledgepit.ml/fedcsis-2023-challenge

[2] https://en.wikipedia.org/wiki/Receiver_operating_characteristic

III. Data aggregation and feature engineering

The available training data provided by the competition organizers contain 15027 log files each given in a .csv table format with a uuid4 random name, in which the 1-minute logs of all related system calls are listed together with the timestamps given in the datetime formmat of yyyy-mm-dd-hh:mm:ss, eg. 2023-04-12-00:00:00. A small fraction in the training data is indicated to be hit by a cyberattack. The test data contains 5017 log files having the same format and naming scheme as the training files, while the cyberattack indication is not given.

To consolidate the feature sets, we wrote a python code to aggregate the time series log files, a generic aggregation filter fitting into all columns of the dataset as described below:

1) For numerical columns eleven self-explanative aggregators were applied, including:
   - *minimum*,
   - *maximum*,
   - *mean*,
   - *median*,
   - *sum*,
   - and *standard deviation*

   across all timestamps.
2) For categorical columns the aggregation treatment was made dependent on the most common frequency number of unique values.

In this way, each of the log files has been converted to a vector of aggregation features, representing a sample in both training and testing datasets. A binary label is given to each training sample, which represents whether or not a cyberattack is experienced.

IV. Gradient boosting models

Gradient boosting decision trees (GBDT) algorithms have emerged as a powerful and widely used technique in machine learning and data mining. GBDT combines the strengths of decision trees and boosting, resulting in a highly accurate and robust predictive model. This approach has been successfully applied in various domains, including finance, healthcare, and online advertising [5]. In [6], it provides a comprehensive overview of GBDT algorithms, explaining their theoretical foundations, practical implementation details, and empirical results. The fundamental idea behind GBDT is to iteratively train weak decision trees and sequentially add them to an ensemble, where each subsequent tree aims to correct the mistakes made by the previous ones. This process is guided by a loss function that measures the discrepancy between the actual and predicted values. By minimizing the loss function, GBDT optimizes the model's ability to make accurate predictions. One of the key advantages of GBDT is its ability to handle both numerical and categorical features effectively. Through a process called feature engineering, GBDT algorithms transform raw data into meaningful and informative representations, enhancing the model's predictive capabilities. Efficient feature engineering techniques play a crucial role in improving the accuracy and interpretability of the GBDT model. We applied three popular GBDT algorithms XGBoost, CatBoost, and lightGBM to construct the ensemble learning model for predicting the cybersecurity breaches for this challenge. Moreover, for years our team has been participating in the data science competitions series organized by the KnowledgePit platform[3] using GBDT related algorithms for classification, regression and other related tasks [7] - [22] with outstanding results. Gradient boosting decision trees algorithms, with their ability to handle diverse data types, efficient feature engineering, and model hyper-parameter optimization, have proven to be a powerful tool for predictive modeling in various domains.

Initial tests conducted on the primary dataset provided evident findings, demonstrating that gradient boosting models outperformed other methods in terms of predictive accuracy, while also exhibiting favorable computational efficiency. Notably, when compared to simple linear regression and deep networks, the performance of gradient boosting models was significantly superior. Within the category of gradient boosting models, specifically XGBoost, LightGBM, and CatBoost, were employed and subsequently fine-tuned during the competition. Different variations of these models, trained with diverse parameters, were utilized in second-level ensembles, employing both simple aggregation and stacked retraining techniques.

Modern Machine Learning models have reached a level of sophistication where they offer extensive customization and adaptability to cater to diverse options, versions, and parametric configurations during the model construction process. Gradient boosting models serve as prime examples of such models, as they provide numerous algorithmic, representational, modeling, and statistical parameters that can be fine-tuned to effectively capture and represent the data. The ultimate goal is to learn a reliable regression function that accurately predicts continuous output based on the input variables, ensuring robust generalization on unseen data.

In order to handle the challenge of tuning a large number of parameters for each distinct model, we opted to utilize a fast and efficient rotational grid search approach built on the general grid search hyperparameter tuning [23] for the gradient boosting models: XGBoost, CatBoost, and LightGBM. This method involves assigning up to a set of unique values to each optimizable parameter, covering a comprehensive range within the parameter's search space, regardless of whether it is numerical or categorical.

Unlike an exhaustive parametric grid search, which would be computationally infeasible given the number of parameters involved, our approach focuses on incrementally finding local optima for a specific parameter while keeping the remaining parameters fixed. This process continues in a rotational manner, moving on to the next parameter only after no further improvement can be achieved from any local changes. By adopting this strategy, we can efficiently explore the parameter space without exhaustive evaluation.

[3] https://knowledgepit.ai/

To enhance the reliability of the best parameter configurations discovered, we applied Repeated Stratified 10-Fold cross validator. This technique helps eliminate the possibility of accidentally selecting configurations with unusually high performance. However, to mitigate the additional computational cost associated with cross-validation, we simplified the process of searching for local optima for each parameter. Specifically, we only performed a pair of neighboring checks for each turn, evaluating the performance above and below the current parameter value. The optimal value was then adjusted to the value that exhibited the maximum performance improvement.

By employing this rotational grid search hyperparameter tuning approach and integrating 10-fold cross-validation, we aimed to efficiently and effectively determine the most suitable parameter configurations for the gradient boosting models, ensuring reliable and high-performing models for our purposes.

Optimizing the hyper-parameters of the GBDT model is essential to achieve superior performance. Determining the appropriate values for hyper-parameters such as the learning rate, tree depth, and regularization parameters can significantly impact the model's predictive accuracy and generalization ability. Therefore, model hyper-parameter optimization is a crucial step in harnessing the full potential of GBDT algorithms.

This parameters optimization process is terminated when no improvement in cross-validated AUC performance was found from any local changes of parameters.

V. Ensemble model

For the final ensemble construction, we employed three fundamental gradient boosting models: XGBoost (XGB), LightGBM (LGBM), and CatBoost (CatB). To enhance the generalization performance of these models, we applied filters. The purpose of filter techniques is to expand the classifier into multiple versions that differ from each other, train them on either the entire training set or subsets of it, and then apply them to the testing set. The outputs of these model versions are subsequently aggregated together.

To further enhance diversity and seek improved predictive performance, we trained all baseline regression models on different feature subsets generated by our feature engineering engine. The primary distinction between these feature subsets was that the second set included a greater number of sparse columns obtained from an extensive application of one-hot-encoding to categorical features. This approach aimed to introduce more varied and complementary information for prediction.

By employing the combination of baseline gradient boosting models, diversification filters, and diverse feature subsets, we aimed to construct a final ensemble that not only exhibited enhanced diversity but also delivered superior predictive performance.

Furthermore, in order to explore additional avenues for performance improvement, we introduced an additional stacked layer consisting of simple linear regression. This layer was trained using the outputs generated by the baseline models. To facilitate the stacking layer's integration, we divided the training data into two distinct parts. The first part was utilized to construct the baseline models, while the second part was reserved specifically for learning the parameters of the linear regression model within the stacking layer.

In the end, we merged the outputs of each individual model and the outputs from the linear regression-based stacking by taking their average. The architecture, represented as a flow chart, showcasing the structure of the final ensemble, can be observed in Figure 1.

VI. Experimental results

Throughout the competition, we used sklearn packages, xgboost, lightgbm and catboost under Python3 Jupyter Notebook[4] in a Windows Server Virtual Machine with 128G RAM memory and Intel(R) Xeon(R) Gold 6230R CPU@2.10GHz, 2 Processors to run simulations. We did intensive feature aggregation and different feature combination by removing or filtering some columns as mentioned below in Table I.

The different features sets, along with their corresponding impact on the performance of individual models on the limited and sparse training and testing datasets, is summarized in Table II.

While numerous parametric variants demonstrated strong performance throughout the competition, we obtained our best individual model scores by utilizing specific model parameters, as indicated in Table III.

It can be easily seen from Table II, the performance of each model for different feature datasets has only slightly difference, even for 40 features only we can also achieve AUC 0.9999, it is most probably due to the very limited and unbalanced training and test dataset, and as AUC score is near 1, so honestly speaking it is very difficult to improve the model performance consider the trade off potential or already model overfitting problem which is very much challenging in terms of considering model stability rather than accuracy performance.

To be safe and ensure our model is more robust enough, we only consider the full aggregated features of 149 features and the compact version of 40 features, and also build our ensemble model on top of the three individual model LGBM, XGB and CatB to make it more robust. Our chosen datasets, (AUC) results, along with the optimal model parameters determined for each model are described below:

1) Feature set 1 with 149 features:
   - CatB (learning rate 0.02, depth 3, iterations 1000): 0.9983
   - LGBM (learning rate 0.02, depth 3, iterations 1000): 0.9999
   - XGB (learning rate 0.01, depth 3, iterations 3000): 0.9976
2) Feature set 2 with 40 features:
   - CatB (learning rate 0.02, depth 3, iterations 1000): 0.9986
   - LGBM (learning rate 0.02, depth 3, iterations 1000): 0.9998

[4] https://jupyter.org/

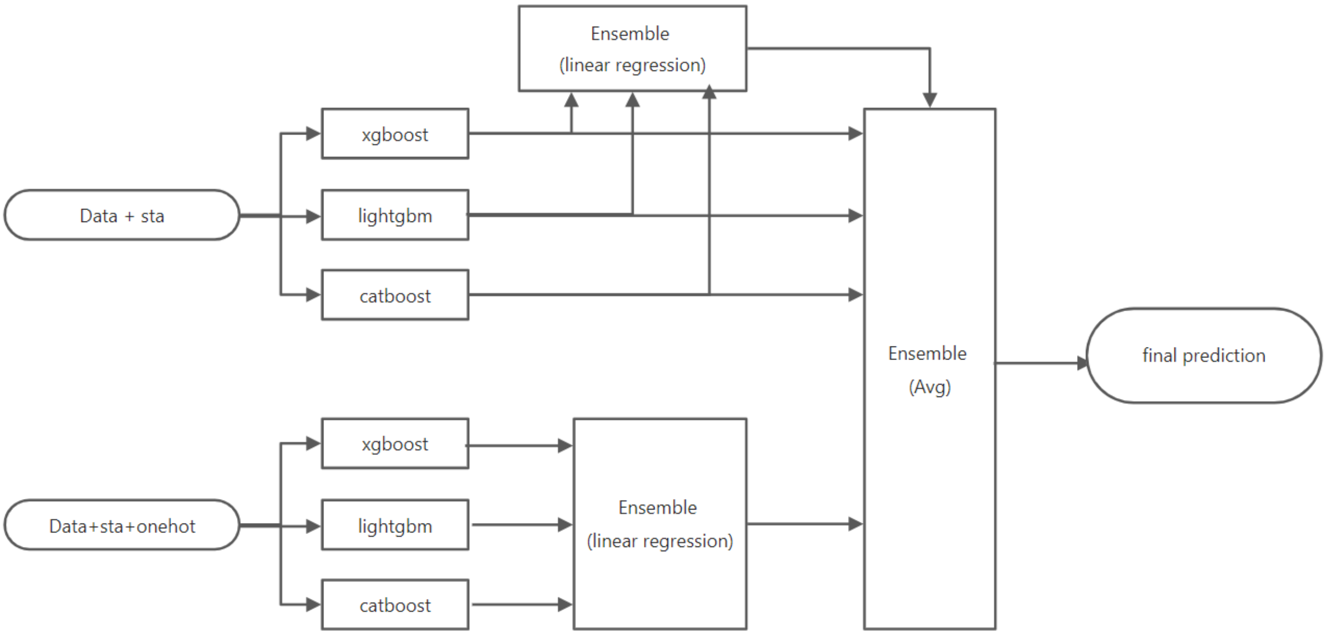


Figure 1. Flowchart of the final ensemble model.

Table I
VERSION OF FEATURES COMBINATION DATASETS

Version	Number of Features	Ignored Columns Numbers	Ignored columns names
V1	149	0	Full aggregated columns
V2	40	19 columns ignored on top of V1	"USER_AUTH","USER_MGMT_COUNT","CRED_COUNT", "USER_ERR_COUNT","USYS_CONFIG_COUNT","CHID_COUNT", "SELINUX_ERR_COUNT","SYSTEM_COUNT","SERVICE_COUNT", "DAEMON_COUNT","NETFILTER_COUNT","SECCOMP_COUNT", "AVC_COUNT","ANOM_COUNT","INTEGRITY_COUNT", "KERNEL_COUNT","RESP_COUNT","SELINUX_MGMT_COUNT", "CUSTOM_openSockets"
V3	28	2 columns ignored on top of V2	"KILL_process","KILL_uid"
V4	23	5 columns ignored on top of V3	"SYSCALL_exit_hint_common", "USER_ACTION_op_common", "USER_ACTION_src_common", "USER_ACTION_res_common", "USER_ACTION_addr_common"

Table II
FEATURES AND AUC-MEASURED INDIVIDUAL MODEL PERFORMANCE

Version	Number of Features	LGBM	XGB	CatB
V1	149	0.9999	0.9983	0.9976
V2	40	0.9998	0.9988	0.9986
V3	28	0.9993	0.9975	0.9982
V4	23	0.9990	0.9976	0.9981

Table III
OPTIMIZED INDIVIDUAL MODEL PARAMETERS

Model	Encoder	Iterations	Learning Rate	Tree Depth
LGBM	onehot	1000	0.02	3
XGB	ordinal	3000	0.02	3
CAT	onehot	1000	0.01	3

- XGB (learning rate 0.01, depth 3, iterations 3000): 0.9988

And, stacking ensemble models based on linear regression were trained using the outputs from the diversified individual models. These stacking models exhibited preliminary AUC performance as follows:

- Stacking model with 149 features: 0.9998
- Stacking model with 40 features: 0.9996

The final predictions are obtained by averaging the outcomes of both stacking models and diversified individual baseline models using an ensemble approach. This ensemble averaging results in an AUC of approximately 0.9998 in the preliminary score. However due to some mistakes, this ensemble model submission was missing in the three final entries, so we only submitted two entries of LGBM which scored the 0.9993 as the 4th place, while this ensemble model can achieve 0.9997 AUC score evaluated with the final released test labels which means the ensembled model is more robust than the single model.

VII. Conclusions

We endeavored to enhance the predictive performance of the already robust regression models within the gradient boosting family, namely XGBoost, LGBM, and CatBoost. To accomplish this challenge, we applied a range of GBDT methods, combined with different ensemble combination techniques and observed improved performance achieved through aggregating the expanded set of diverse model versions. Additionally, we employed linear regression-based stacking and selected the most effective ensemble candidates based on the trade-off between performance and diversity.

We applied this proposed ensemble approach to the challenging task of advance prediction of cybersecurity breaches in IoT device log data, which involved various types and forms of data. Our solution was implemented and evaluated within the competitive framework of the FedCSIS 2023 data mining challenge. In the preliminary leader-board of the challenge, our proposed solution achieved the fifth position with an AUC of 0.9999, while in the final ranking we are 4th place with AUC score 0.9993 even though unluckily our ensemble model entry was missing in the final three entries and this ensemble model can achieve the AUC score 0.9997 with most stable and robust compared the AUC score 0.9998 in the preliminary leader-board. Our solution holds potential for enabling network service providers to better anticipate hacker threats and bolster their cybersecurity measures.

References

[1] F. Alaba, M. Othman, I. Hashem, F. Alotaibi, Internet of Things Security: A Survey, *Journal of Network and Computer Applications*, vol. 88, pp. 10-28, 2017.

[2] M. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A. Sheth, Machine learning for internet of things data analysis: a survey, *Digital Communications and Networks*, 2018.

[3] M. Garadi, A. Mohamed, A. Ali, X. Du, I. Ali and M. Guizani, A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security, *IEEE Communications Surveys & Tutorials*, 2020.

[4] A. Janusz, A. Kozłowski, B. Adamczyk, D. Iwanicki, M. Brzęczek, M. Michalak, M. Tynda, M. Czerwiński, P. Biczyk, Predicting the Cybersecurity Threat Detection in the Behavior of IoT Devices: Analysis of Data Mining Competition Results, *Proceedings of the 18th Conference on Computer Science and Intelligent Systems (FedCSIS)*, 2023.

[5] L. Mason, J. Baxter, P.L. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent In S.A. Solla and T.K. Leen and K. Müller. Advances in Neural Information Processing Systems 12: 512–518, MIT Press, 1999.

[6] J.H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29(5): 1189-1232, 2001.

[7] D. Ruta, M. Liu, L. Cen. FEATURE ENGINEERING FOR PREDICTING FRAGS IN TACTICAL GAMES. *Proc. Int. Conf. 2023 IEEE International Conference on Multimedia and Expo*, 2023.
FEATURE ENGINEERING FOR PREDICTING FRAGS IN TACTICAL GAMES

[8] D. Ruta, M. Liu, L. Cen and Q. Hieu Vu. Diversified gradient boosting ensembles for prediction of the cost of forwarding contracts. *Proc. Int. Conf. 2022 17th Conference on Computer Science and Intelligence Systems*, 2022.

[9] Q. Hieu Vu, L. Cen, D. Ruta and M. Liu. Key Factors to Consider when Predicting the Costs of Forwarding Contracts. *Proc. Int. Conf. 2022 17th Conf. on Computer Science and Intelligence Systems*, 2022.

[10] D. Ruta, L. Cen, M. Liu and Q. Hieu Vu. Automated feature engineering for prediction of victories in online computer games. *Proc. Int. Conf on Big Data*, 2021.

[11] Q. Hieu Vu, D. Ruta, L. Cen and M. Liu. A combination of general and specific models to predict victories in video games. *Proc. Int. Conf. on Big Data*, 2021.

[12] D. Ruta, L. Cen and Q. Hieu Vu. Deep Bi-Directional LSTM Networks for Device Workload Forecasting. *Proc. 15th Int. Conf. Comp. Science and Inf. Sys.*, 2020.

[13] L. Cen, D. Ruta and Q. Hieu Vu. Efficient Support Vector Regression with Reduced Training Data. *Proc. Fed. Conf. on Comp. Science and Inf. Sys.*, 2019.

[14] D. Ruta, L. Cen and Q. Hieu Vu. Greedy Incremental Support Vector Regression. *Proc. Fed. Conf. on Computer Science and Inf. Sys.*, 2019.

[15] Q. Hieu Vu, D. Ruta and L. Cen. Gradient boosting decision trees for cyber security threats detection based on network events logs. *Proc. IEEE Int. Conf. Big Data*, 2019.

[16] L. Cen, A. Ruta, D. Ruta and Q. Hieu Vu. Regression networks for robust win-rates predictions of AI gaming bots. *Int. Symp. Advances in AI and Apps (AAIA)*, 2018.

[17] Q. Hieu Vu, D. Ruta, A. Ruta and L. Cen. Predicting Win-rates of Hearthstone Decks: Models and Features that Won AAIA'2018 Data Mining Challenge. *Int. Symp. Advances in Artificial Intelligence and Apps (AAIA)*, 2018.

[18] L. Cen, D. Ruta and A. Ruta. Using Recommendations for Trade Returns Prediction with Machine Learning. *Int. Symp. on Methodologies for Intelligent Sys. (ISMIS)*, 2017.

[19] A. Ruta, D. Ruta and L. Cen. Algorithmic Daily Trading Based on Experts' Recommendations. *Int. Symp. on Methodologies for Intelligent Systems (ISMIS)*, 2017.

[20] Q. Hieu Vu, D. Ruta and L. Cen. An ensemble model with hierarchical decomposition and aggregation for highly scalable and robust classification. *12th Int. Symposium Advances in AI and Applications (AAIA)*, 2017.

[21] L. Cen and D. Ruta. A Map based Gender Prediction Model for Big E-Commerce Data. *The 3rd IEEE Int. Conf. on Smart Data*, 2017.

[22] D. Ruta and L. Cen. Self-Organized Predictor of Methane Concentration Warnings in Coal Mines. *Proc. Int. Joint Conf. Rough Sets, LNCS*, Springer, 2015.

[23] https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/.

Center for Artificial Intelligence Challenge on Conversational AI Correctness

CENTER for Artificial Intelligence Challenge on Conversational AI Correctness was organized as part of the 1st Symposium on Challenges for Natural Language Processing. The goal of this competition was to develop Natural Language Understanding models that are robust against speech recognition errors.

Regardless of near-human accuracy of Automatic Speech Recognition in general-purpose transcription tasks, speech recognition errors can significantly deteriorate the performance of a Natural Language Understanding model that follows the speech-to-text module in a virtual assistant. The problem is even more apparent when an ASR system from an external vendor is used as an integral part of a conversational system without any further adaptation. The contestants were expected to develop Natural Language Understanding models that maintain satisfactory performance despite the presence of ASR errors in the input.

The data for the competition consist of natural language utterances along with semantic frames that represent the commands targeted at a virtual assistant. The approach used to prepare the data for the challenge was meant to promote models robust to various types of errors in the input, making it impossible to solve the task by simply learning a shallow mapping from incorrectly recognized words to the correct ones. It reflects real-world scenarios where the NLU system is presented with inputs that exhibit various disturbances due to changes in the ASR model, acoustic conditions, speaker variation, and other causes.

This chapter includes the paper discussing the objectives, evaluation rules and results of the competition, authored by the organizers followed by the detailed description of the leading solution contributed by the winners of the challenge.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1319–1324
DOI: 10.15439/2023F6058
ISSN 2300-5963 ACSIS, Vol. 35

Center for Artificial Intelligence Challenge on Conversational AI Correctness

Marek Kubis, Paweł Skórzewski
0000-0002-2016-2598
0000-0002-5056-2808
Adam Mickiewicz University
Faculty of Mathematics and Computer Science
ul. Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland
Email: {marek.kubis, pawel.skorzewski}@amu.edu.pl

Marcin Sowański, Tomasz Ziętkiewicz
0000-0002-9360-1395
0000-0002-2594-4660
Samsung Research Poland
Plac Europejski 1, 00-844 Warsaw, Poland
Email: {m.sowanski, t.zietkiewicz}@samsung.com

***Abstract*—This paper describes a challenge on Conversational AI correctness with the goal to develop Natural Language Understanding models that are robust against speech recognition errors. The data for the competition consist of natural language utterances along with semantic frames that represent the commands targeted at a virtual assistant. The specification of the task is given along with the data preparation procedure and the evaluation rules. The baseline models for the task are discussed and the results of the competition are reported.**

I. Introduction

REGARDLESS of the near-human accuracy of Automatic Speech Recognition in general-purpose transcription tasks, speech recognition errors can significantly deteriorate the performance of a Natural Language Understanding model that follows the speech-to-text module in a virtual assistant. The problem is even more apparent when an ASR system from an external vendor is used as an integral part of a conversational system without any further adaptation. The goal of this competition is to develop Natural Language Understanding models that are robust to speech recognition errors.

The approach used to prepare data for the challenge is meant to promote models robust to various types of errors in the input, making it impossible to solve the task by simply learning a shallow mapping from incorrectly recognized words to the correct ones. It reflects real-world scenarios where the NLU system is presented with inputs that exhibit various disturbances due to changes in the ASR model, acoustic conditions, speaker variation, and other causes.

II. Related Work

The robustness of Natural Language Understanding models to various types of errors is a subject of several publications. Some authors proposed to use word confusion networks to improve models' robustness to ASR errors [1], [2], [3], [4]. Reference [5] developed a learning criterion that prefers NLU models that are robust to ASR errors by adding a loss term that measures the distance between the prediction distribution from transcriptions and ASR hypotheses. Reference [6] studied the performance of intent classification and slot labeling models with respect to several kinds of perturbations, such as substituting abbreviations and synonyms, changing casing and punctuation, paraphrasing, and introducing misspellings and morphological variants. Speech characteristics are among three aspects of robustness investigated by [7] in the assessment of task-oriented dialog systems. Reference [8] investigated data-efficient techniques that apply to a wide range of natural language understanding models used in large-scale production environments to make them robust against speech recognition errors, using domain classification as an example. The authors compared the effectiveness of several such techniques in terms of time-varying usage patterns and distribution of ASR errors.

Several benchmarks exist to evaluate NLU models regarding their robustness to ASR errors. RADDLE [9], a benchmark for evaluating the performance of dialog models, prefers models robust to language variations, speech errors, unseen entities, and out-of-domain utterances. ASR-GLUE [10] is a benchmark consisting of 6 different NLU tasks, for which the input data were recorded by six different speakers and at three different noise levels.

Mitigating the impact of ASR errors on downstream tasks was the subject of several contests. In [11], the authors proposed a challenge for improving the recognition rate of an ASR system on the basis of incorrect ASR hypotheses paired with reference texts. Post-edition of ASR output was also the objective of the shared task held by [12]. Speech-aware dialogue state tracking was the topic of a recent competition conducted by [13].

The data preparation procedure outlined in Section III involves combining a TTS model and an ASR system. Augmentation of speech corpora with the use of synthesized speech was investigated by [14] and [15]. Reference [13] uses synthesized inputs along with spoken utterances in their challenge.

Holding competitions as a method for finding promising solutions to scientific problems has a long history in computer science, particularly in natural language processing [16], [17].

This research was partially funded by the "CAIMAC: Conversational AI Multilingual Augmentation and Compression" project, a cooperation between Adam Mickiewicz University and Samsung Electronics.

This contest is organized under the 1st Symposium on Challenges for Natural Language Processing (CNLPS), a part of the 18th Conference on Computer Science and Intelligence Systems (FedCSIS 2023). FedCSIS Conference Series hosted a wide range of data mining competitions through the years that covered topics such as identifying key risk factors for the Polish State Fire Service [18], network device workload prediction [19], and predicting the costs of forwarding contracts [20]. In the process of running our CNLPS challenge, we followed the best practices set out by the organizers of FedCSIS data mining competitions.

III. Data

The data for the task are derived from the Leyzer dataset [21]. The samples consist of user utterances and the semantic representation of the commands targeted at a virtual assistant (VA). A fraction of the utterances in the training set is contaminated with speech recognition errors; however, we left most of the utterances intact to make the task more challenging. The erroneous samples were obtained from user utterances using a TTS model followed by an ASR system.

A. Preparation of Base Text Corpus

We used the second version of the Leyzer corpus, which consists of more utterance variations when compared to the version described in the original paper. The second version of the corpus introduced two additional sub-intent differentiation levels called *naturalness level* (or simply *level*) and *verb pattern*. Although we have not implicitly used this information in this contest, it allowed us to create more variant corpus for the task. Leyzer consists of 20 domains across three languages: English, Spanish, and Polish, with 186 intents and a wide range of samples per intent. Domains can be grouped into several topics that can be found in the most popular VAs:

- **Communication** with Email, Facebook, Phone, Slack, and Twitter domains in that group, which all relate to communication and the transfer of ideas,
- **Internet** with Web Search and Wikipedia that groups domains related to the search for information on the web; therefore, these domains will have a lot of open-title queries,
- **Media and Entertainment** with Spotify, YouTube, and Instagram domains in that group, which relate to multimedia content with named entities connected to artists or titles,
- **Devices** with Air Conditioner and Speaker domains, which represent simple physical devices that can be controlled by voice,
- **Self-management** with Calendar and Contacts, which consist of actions that involve time planning and people,
- **Other**, uncategorized domains (Fitbit, Google Drive, News, Translate, Weather, Yelp) represent functions and language not shared by other categories. In this sense, the remaining domains can be understood as intentionally not matching the other domains.

Using scripts provided in the Leyzer repository, we generated the text corpus from JSGF grammars. The corpus was divided into `train`, `valid`, `test-A`, and `test-B` parts using the splitting script provided in the Leyzer repository. First, we differentiate `test-B` from the rest of the corpus. For `test-B`, a minimum of 1 test case and up to 20% of the total available sentences for each intent, level, and verb pattern were selected, and the remaining test cases were left in the development corpus. From the development part of the corpus, we further differentiate `test-A` using the same procedure as for `test-B`, which extracted a minimum of 1 and up to 20% of test cases for each intent, level, and verb pattern triplet. The remaining corpus was divided into `train` and `valid` subsets. The `valid` subset is 20% of randomly selected test cases without assuring that it contains at least 1 test case for each intent, level, and verb pattern triplet.

B. Augmenting Corpus with Back-transcription

Back-transcription is a technique that can be used to produce speech transcripts from text-only data. Textual data are fed to a TTS engine to produce a speech signal, which in turn is fed to an ASR system, producing an augmented text. Depending on the performance of both models and differences in text normalization performed on the input text, as well as inside these models, the resulting text can be identical to the input or may contain differences introduced in either processing stage. The technique has been used to develop post-processing [22] and error correction [23] models for ASR systems.

We use back-transcription to simulate a virtual assistant user's behavior. The user speaks to the system, and their speech is converted into text by an ASR model, which is subsequently processed by an NLU model (see Fig. 1). NLU text prompts from the Leyzer corpus are synthesized using a TTS engine. The resulting sound signal is used as input to an ASR model producing back a text with an augmented NLU prompt. The procedure is illustrated in Fig. 2. To perform Text-To-Speech synthesis, we used the FastSpeech 2[1] model [24] for English, the VITS model [25] for Polish, and Tacotron 2 [26] for Spanish, both from the Coqui TTS library [27]. Speech recognition was performed using the Whisper[2] model [28] for all three languages.

C. CAICCAIC Dataset

The training data are located in the `train` directory of the contest's repository[3]. The `train` directory contains two files:

- `in.tsv` with four columns:
  1) sample identifier: *306*,
  2) language code: *en-US*,
  3) data split type: *train*,
  4) utterance: *adjust the temperature to 82 degrees fahrenheit on my reception room thermostat.*
- `expected.tsv` with three columns representing:

[1] https://huggingface.co/facebook/fastspeech2-en-ljspeech

[2] https://huggingface.co/openai/whisper-large

[3] https://github.com/kubapok/cnlps-caiccaic

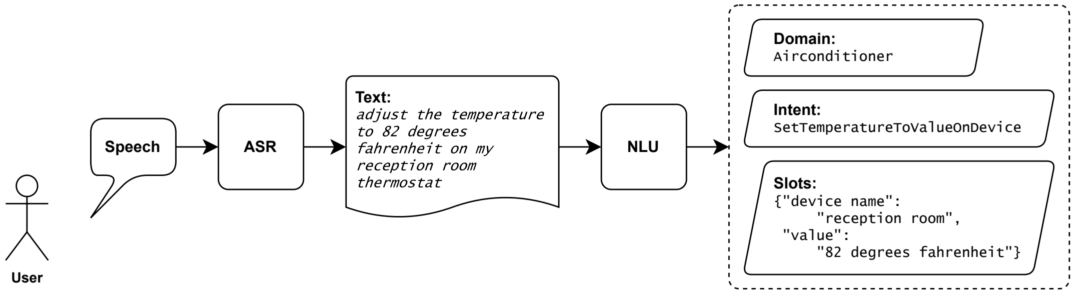


Fig. 1. Spoken Language Understanding.

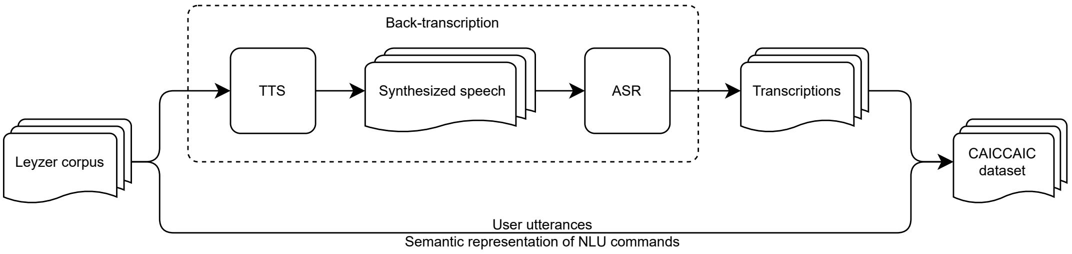


Fig. 2. Dataset preparation pipeline.

1) domain label: *Airconditioner*,
2) intent label: *SetTemperatureToValueOnDevice*,
3) slot values:
`{"device_name": "reception room",`
`"value": "82 degrees fahrenheit"}`

For experimentation, we provide the validation dataset in the `dev-A` directory of the contest's repository. It was created using the same pipeline as the `train` dataset. The test data are located in `test-A` and `test-B` directories and contain only input values, while expected values hidden for contestants are used by the evaluation platform to score submissions.

IV. Baseline Models

We use XLM-RoBERTa Base [29] as a baseline model for intent detection and slot-filling. The XLM-RoBERTa model, also known as XLM-R, is a transformer-based multilingual masked language model that employs a multilingual masked language model (MLM) objective using only monolingual data. During training, streams of text from each language are sampled, and the model is trained to predict the masked tokens in the input. Subword tokenization is applied directly to raw text data using SentencePiece [30] with a unigram language model. The model does not use language embeddings, which allows it to handle code-switching better. It uses a large vocabulary size of 250K with a full softmax.

XLM-R was pre-trained on 2.5 TB of filtered CommonCrawl data containing 100 languages. This large-scale training led to significant performance gains for various cross-lingual transfer tasks. The model significantly outperforms multilingual BERT (mBERT) on various cross-lingual benchmarks.

Our baseline models were trained independently on the entire training set and optimized on the evaluation set. All baseline models have 12 layers, 768 hidden units, and 12 attention heads, totaling 270M parameters, and a size of 1.1 GB.

We use the `leyzer-fedcsis`[4] dataset from the Hugging Face Model Hub in the baseline training process. Each language-specific portion is processed individually, retaining only the `utterance` and `intent` columns. The processed datasets are then merged and split into training, validation, and testing sets. The model is defined for a sequence classification task using the `AutoModelForSequenceClassification` class, with the number of labels corresponding to the unique intents in the training dataset. Training hyperparameters were set to a learning rate of 2×10^{-5}, a training batch size of 16, a weight decay of 0.01, and 10 training epochs. Evaluations are performed after each epoch.

Finally, performance metrics such as accuracy and F_1 score are computed to assess the model's effectiveness in its classification task. The final epoch checkpoint evaluation results on the test set are presented in Table II in the "official baseline" row. All baseline intents models achieved results above 90% accuracy, with Spanish, Polish, and all-language models achieving above 95%. We analyzed misclassification errors and found that most of them could be resolved if a model resisted token distortion and could separate syntactically similar classes.

The error analysis of the intent recognition models for English, Spanish, and Polish languages reveals similarities and differences across the models. The *Spotify* domain tends to be the most problematic for all three languages, suggesting that these models may struggle with understanding and predicting

[4] https://huggingface.co/datasets/cartesinus/leyzer-fedcsis

TABLE I
UTTERANCE LENGTH DISTRIBUTION IN THE DATASET.

Locale	Split	Utterances	Mean Length	Length StdDev	Min	50%	Max
en-US	test	3344	9.951	4.322	1	9	33
	train	13022	9.345	3.718	1	9	33
	valid	3633	9.281	3.799	1	9	30
es-ES	test	3520	13.214	6.110	1	12	36
	train	15043	13.369	6.022	1	12	39
	valid	3546	13.152	5.948	1	12	39
pl-PL	test	3494	8.927	3.059	1	9	22
	train	12753	8.972	3.028	1	9	26
	valid	3498	9.018	3.053	1	9	23

TABLE II
EVALUATION RESULTS.

#	submission	description	pl-PL EMA	es-ES EMA	en-US EMA	Slot WRR	Intent accuracy	Domain accuracy	EMA
1	8850	mbart-large-50	**0.799**	**0.884**	0.569	**0.872**	0.916	0.963	**0.754**
2	8774	flan-t5-large	0.649	0.787	0.628	0.805	0.922	0.969	0.689
3	8347	*official baseline*	0.767	0.595	**0.686**	0.752	**0.945**	**0.980**	0.682
4	8812	flan-t5-large+context	0.648	0.794	0.548	0.770	0.898	0.955	0.665
5	8687	flan-t5-large	0.550	0.716	0.435	0.738	0.822	0.931	0.569
6	8846	flan-t5	0.495	0.503	0.479	0.692	0.898	0.958	0.493
7	8853	transformer t5	0.516	0.389	0.481	0.626	0.866	0.949	0.461
8	8869	dfd	0.469	0.457	0.411	0.627	0.675	0.959	0.446
8	8856	flan-t5-base	0.463	0.475	0.389	0.624	0.849	0.945	0.443
10	8847	all done	0.344	0.368	0.278	0.451	0.582	0.926	0.331

intents related to music streaming or the specific language used in this domain. *Slack* and *Console* domains also prove problematic for the English and Polish models, while for the Spanish model, the recognition of the *Airconditioner* and *Email* domains was the most challenging. Regarding specific intents, the English model has the most trouble with *ConsoleEdit* and *AddAlbumToPlaylist*, the Spanish model struggles with *PlayAlbumOfTypeByArtist* and *TurnOn*, and the Polish model with *SetPurposeOnChannel* and *PlayAlbumOfTypeByArtist*. These intents may be harder to recognize due to their semantic complexity, similarity to other intents, or underrepresentation in training data.

All models are available on the Hugging Face platform with details of how each model was trained and how to execute them:

- intent: en-US[5], es-ES[6], pl-PL[7], and all[8] that was trained and evaluated on all three languages together
- slot: en-US[9], es-ES[10], pl-PL[11]

V. Evaluation

The solutions for the task were submitted via the Gonito platform [31] challenge available at https://gonito.csi.wmi.amu.edu.pl/challenge/cnlps-caiccaic. For `in.tsv` file located in `test-A` directory, the participants were expected to provide `out.tsv` file in the same directory containing the predictions. The format of `out.tsv` was the same as the format of `train/expected.tsv`. Participants were allowed to use any publicly available data and models. Manual labeling was forbidden. A maximum of five submissions per day were allowed.

The submissions were scored using *Exact Match Accuracy* (EMA), i.e., the percentage of utterance-level predictions in which domain, intent, and all the slots are correct. Besides EMA scores, we also report the following auxiliary metrics:

- *domain accuracy*, i.e., the percentage of utterances with correct domain prediction;
- *intent accuracy*, i.e., the percentage of utterances with the correct intent prediction;
- *slot word recognition rate*, i.e., word recognition rate (WER) calculated on slot annotations, which is the percentage of correctly annotated slot values.

All scores were calculated using the GEval [32] library, which was also made available to participants for offline use.

VI. Results

We received 28 submissions from 9 teams. Table II presents the final ranking with cumulative metrics for all languages[12]. Notably, most submissions are based on pre-trained Transformer models [33] adapted to the task, with the Flan-T5 model [34] being the preferred choice. However, the winning

[5] https://huggingface.co/cartesinus/fedcsis-intent_baseline-xlm_r-en
[6] https://huggingface.co/cartesinus/fedcsis-intent_baseline-xlm_r-es
[7] https://huggingface.co/cartesinus/fedcsis-intent_baseline-xlm_r-pl
[8] https://huggingface.co/cartesinus/fedcsis-intent_baseline-xlm_r-all
[9] https://huggingface.co/cartesinus/fedcsis-slot_baseline-xlm_r-en
[10] https://huggingface.co/cartesinus/fedcsis-slot_baseline-xlm_r-es
[11] https://huggingface.co/cartesinus/fedcsis-slot_baseline-xlm_r-pl

[12] Detailed results can be found at https://gonito.csi.wmi.amu.edu.pl/challenge/cnlps-caiccaic/allentries.

TABLE III
MOST PROBLEMATIC FEATURES FOR THE WINNING MODEL COMPARED WITH THE BASELINE MODEL.

metric	feature	count	metric δ
Intent acc.	in<4>:hundred	357	-0.423
Intent acc.	in<4>:eight	224	-0.442
Intent acc.	in<4>:six	206	-0.408
Intent acc.	in<4>:images	417	-0.290
Intent acc.	out:FindImages[..][13]	31	-1.000
Intent acc.	in<2>:en-US	3344	-0.090
Intent acc.	in<4>:being	55	-0.582
Intent acc.	in<4>:small	48	-0.604
Intent acc.	out:	2013	-0.103

TABLE IV
MOST PROBLEMATIC FEATURES FOR THE BASELINE MODEL COMPARED WITH THE WINNING MODEL.

metric	feature	count	metric δ
EMA	out:subject	707	-0.808
EMA	exp:subject	711	-0.802
EMA	exp:SendEmail[..][14]	766	-0.756
EMA	out:SendEmail[..][15]	771	-0.752
EMA	out:message	835	-0.677
EMA	exp:message	837	-0.671
EMA	in<4>:un	982	-0.609
EMA	exp:to	1020	-0.593
EMA	in<4>:email	748	-0.686
EMA	out:to	885	-0.567

solution [35] used the mBART model [36] as its basis to train a joint, text-to-text model of domain, intent, and slots. This model achieved an Exact Match Accuracy of 0.754 across all the samples, with top results attained for Polish and Spanish NLU commands (0.799 and 0.884 EMA, respectively). It demonstrated outstanding performance in slot recognition with a slot WRR of 0.872 (0.067 better than the second-best solution). Although the winning solution performed well overall, it was within the accuracy of XLM-RoBERTa baseline models regarding domain and intent accuracy. This observation is intriguing and could be a valuable starting point for future research on developing joint models for domains, intents, and slots.

To gain more insight into the differences between the winning model and the baseline, we performed the analysis using the Geval tool [32]. Geval's "most worsening feature" function was used to analyze cases for which one of the models is problematic while the other behaves correctly. The function calculates the difference in a chosen metric between two models being compared, on cases containing a specific feature. The results are reported for cases for which the difference is statistically significant. Table III shows the features that had the most negative impact on the winning results compared to the baseline submission. It appears that numbers in their written form in English input are problematic for the mBART model. Also, it is not surprising to see that English inputs, in general, are easier for the baseline solution compared to the winning one, considering the overall results presented in Table II. Additionally, the mBART model has problems with one of the image-finding intents, which is consistent with the problematic word "images" in input sentences. Conversely, Table IV presents features that were problematic for the winning submission while being easier for the baseline model. The most problematic features are connected with the *Email* domain. It looks as baseline model has problems with identifying all kinds of slots of commands used for sending emails. These observations should prompt the authors of the winning submission and anyone else who wants to improve on these results to take a closer look into the specific causes of these particular types of errors and work towards addressing them.

REFERENCES

[1] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond ASR 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006. doi: https://doi.org/10.1016/j.csl.2005.07.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230805000495

[2] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "LATTICE RNN: Recurrent neural networks over lattices," in *Interspeech 2016*, 2016. [Online]. Available: https://www.amazon.science/publications/lattice-rnn-recurrent-neural-networks-over-lattices

[3] G. Tur, A. Deoras, and D. Hakkani-Tür, "Semantic parsing using word confusion networks with conditional random fields," in *Annual Conference of the International Speech Communication Association (Interspeech)*, Sep. 2013.

[4] X. Yang and J. Liu, "Using word confusion networks for slot filling in spoken language understanding," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[5] W. Ruan, Y. Nechaev, L. Chen, C. Su, and I. Kiss, "Towards an ASR error robust spoken language understanding system," in *Interspeech 2020*, 2020. [Online]. Available: https://www.amazon.science/publications/towards-an-asr-error-robust-spoken-language-understanding-system

[6] S. Sengupta, J. Krone, and S. Mansour, "On the robustness of intent classification and slot labeling in goal-oriented dialog systems to real-world noise," in *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*. Online: Association for Computational Linguistics, Nov. 2021. doi: 10.18653/v1/2021.nlp4convai-1.7 pp. 68–79. [Online]. Available: https://aclanthology.org/2021.nlp4convai-1.7

[7] J. Liu, R. Takanobu, J. Wen, D. Wan, H. Li, W. Nie, C. Li, W. Peng, and M. Huang, "Robustness testing of language understanding in task-oriented dialog," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021. doi: 10.18653/v1/2021.acl-long.192 pp. 2467–2480. [Online]. Available: https://aclanthology.org/2021.acl-long.192

[8] Y. Nechaev, W. Ruan, and I. Kiss, "Towards NLU model robustness to ASR errors at scale," in *KDD 2021 Workshop on Data-Efficient Machine Learning*, 2021. [Online]. Available: https://www.amazon.science/publications/towards-nlu-model-robustness-to-asr-errors-at-scale

[9] B. Peng, C. Li, Z. Zhang, C. Zhu, J. Li, and J. Gao, "RADDLE: An evaluation benchmark and analysis platform for robust task-oriented dialog systems," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021. doi: 10.18653/v1/2021.acl-long.341 pp. 4418–4429. [Online]. Available: https://aclanthology.org/2021.acl-long.341

[10] L. Feng, J. Yu, D. Cai, S. Liu, H. Zheng, and Y. Wang, "ASR-GLUE: A new multi-task benchmark for ASR-robust natural language understanding," *CoRR*, vol. abs/2108.13048, 2021. doi: 10.48550/arXiv.2108.13048. [Online]. Available: https://arxiv.org/abs/2108.13048

[11] M. Kubis, Z. Vetulani, M. Wypych, and T. Ziętkiewicz, "Open challenge for correcting errors of speech recognition systems," in *Human Language Technology. Challenges for Computer Science and Linguistics*, Z. Vetulani, P. Paroubek, and M. Kubis, Eds. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-031-05328-3_21. ISBN 978-3-031-05328-3 pp. 322–337.

[12] D. Koržinek, “Results of the PolEval 2020 Shared Task 1: Post-editing and Rescoring of Automatic Speech Recognition Results,” in *Proceedings of the PolEval 2020 Workshop*, 2020, pp. 9–14.

[13] H. Soltau, I. Shafran, M. Wang, A. Rastogi, J. Zhao, Y. Jia, W. Han, Y. Cao, and A. Miranda, “Speech Aware Dialog System Technology Challenge (DSTC11),” 2022. [Online]. Available: https://arxiv.org/abs/2212.08704

[14] T. Hayashi, S. Watanabe, Y. Zhang, T. Toda, T. Hori, R. Astudillo, and K. Takeda, “Back-translation-style data augmentation for end-to-end ASR,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018. doi: 10.1109/SLT.2018.8639619 pp. 426–433.

[15] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, “You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation,” in *2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2020. doi: 10.1109/CISP-BMEI51763.2020.9263564 pp. 439–444.

[16] R. Grishman and B. Sundheim, “Message Understanding Conference-6: A brief history,” in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. [Online]. Available: https://aclanthology.org/C96-1079

[17] A. Kilgarriff and M. Palmer, “Introduction to the special issue on SENSEVAL,” *Comput. Humanit.*, vol. 34, no. 1-2, pp. 1–13, 2000. doi: 10.1023/A:1002619001915. [Online]. Available: https://doi.org/10.1023/A:1002619001915

[18] A. Janusz, A. Krasuski, S. Stawicki, M. Rosiak, D. Ślęzak, and H. S. Nguyen, “Key risk factors for Polish state fire service: a data mining competition at knowledge pit,” in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 2. IEEE, 2014. doi: 10.15439/2014F507 pp. 345–354. [Online]. Available: http://dx.doi.org/10.15439/2014F507

[19] A. Janusz, M. Przyborowski, P. Biczyk, and D. Ślęzak, “Network device workload prediction: A data mining challenge at knowledge pit,” in *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, S. Agarwal, D. N. Barrell, and V. K. Solanki, Eds., vol. 21. IEEE, 2020. doi: 10.15439/2020KM159 pp. 77–80. [Online]. Available: http://dx.doi.org/10.15439/2020KM159

[20] A. Janusz, A. Jamiołkowski, and M. Okulewicz, “Predicting the costs of forwarding contracts: Analysis of data mining competition results,” in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 30. IEEE, 2022. doi: 10.15439/2022F303 p. 399–402. [Online]. Available: http://dx.doi.org/10.15439/2022F303

[21] M. Sowański and A. Janicki, “Leyzer: A dataset for multilingual virtual assistants,” in *Text, Speech, and Dialogue*, P. Sojka, I. Kopeček, K. Pala, and A. Horák, Eds. Cham: Springer International Publishing, 2020. doi: 10.1007/978-3-030-58323-1_51. ISBN 978-3-030-58323-1 pp. 477–486.

[22] C. Park, J. Seo, S. Lee, C. Lee, H. Moon, S. Eo, and H. Lim, “BTS: Back TranScription for speech-to-text post-processor using text-to-speech-to-text,” in *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*. Online: Association for Computational Linguistics, Aug. 2021. doi: 10.18653/v1/2021.wat-1.10 pp. 106–116. [Online]. Available: https://aclanthology.org/2021.wat-1.10

[23] J. Guo, T. N. Sainath, and R. J. Weiss, “A spelling correction model for end-to-end speech recognition,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. doi: 10.1109/ICASSP.2019.8683745 pp. 5651–5655.

[24] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, “FastSpeech 2: Fast and high-quality end-to-end text to speech,” in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[25] J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul. 2021, pp. 5530–5540. [Online]. Available: https://proceedings.mlr.press/v139/kim21f.html

[26] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, “Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. doi: 10.1109/ICASSP.2018.8461368 pp. 4779–4783.

[27] G. Eren and The Coqui TTS Team, “Coqui TTS,” Jan. 2021. [Online]. Available: https://github.com/coqui-ai/TTS

[28] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 28 492–28 518. [Online]. Available: https://proceedings.mlr.press/v202/radford23a.html

[29] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020. doi: 10.18653/v1/2020.acl-main.747 pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[30] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018. doi: 10.18653/v1/D18-2012 pp. 66–71. [Online]. Available: https://aclanthology.org/D18-2012

[31] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń, “Gonito.net - Open Platform for Research Competition, Cooperation and Reproducibility,” in *Branco, António and Nicoletta Calzolari and Khalid Choukri (eds.), Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, 2016, pp. 13–20. [Online]. Available: http://4real.di.fc.ul.pt/wp-content/uploads/2016/04/4REALWorkshopProceedings.pdf

[32] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki, “GEval: Tool for debugging NLP datasets and models,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019. doi: 10.18653/v1/W19-4826 pp. 254–262. [Online]. Available: https://www.aclweb.org/anthology/W19-4826

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS’17. Red Hook, NY, USA: Curran Associates Inc., 2017. ISBN 9781510860964 p. 6000–6010.

[34] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, “Scaling instruction-finetuned language models,” 2022.

[35] S. Jadczak and R. Jaworski, “Boosting conversational AI correctness by accounting for ASR errors using a sequence to sequence model,” in *Proceedings of the 18th Conference on Computer Science and Intelligence Systems*, 2023.

[36] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, “Multilingual denoising pre-training for neural machine translation,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020. doi: 10.1162/tacl_a_00343. [Online]. Available: https://aclanthology.org/2020.tacl-1.47

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1325–1328

DOI: 10.15439/2023F9627
ISSN 2300-5963 ACSIS, Vol. 35

Boosting conversational AI correctness by accounting for ASR errors using a sequence to sequence model

Szymon Jadczak
Adam Mickiewicz University in Poznań
Faculty of Mathematics and Computer Science
Email: szyjad@st.amu.edu.pl

Rafał Jaworski
0000-0001-8827-3318
Adam Mickiewicz University in Poznań
Faculty of Mathematics and Computer Science
Email: rjawor@amu.edu.pl

***Abstract*—This paper describes the winning submission to the challenge CAICCAIC: Center for Artificial Intelligence Challenge on Conversational AI Correctness. The aim of the challenge was to design a mechanism of natural language understanding capable of interpreting user prompts. The prompts were the output of an automatic speech recognition system and therefore contained errors. In this scenario, it was necessary to apply techniques of accounting for these errors. As per the results of the challenge, the most effective technique proved to be an original use of a sequence to sequence model. The key idea was the concatenation of labels before passing them to the model for training and prediction.**

I. Problem formulation

AUTOMATIC Speech Recognition (ASR) systems are immensely popular in today's world. They find use in assistant applications for phones, cars or at home. The ASR techniques have been perfected over many years to achieve maximum available output quality. However, it is not always possible to recognize speech with perfect accuracy due to numerous factors, such as:

- external noise,
- individual voice features (prosody),
- ambiguity of spoken language

and many others. Problems with the accuracy of ASR may also arise from using imperfect models which produce sub-optimal output quality.

In such scenarios it is well justified to apply an error proof natural language understanding (NLU) module on the results of ASR. The goal of that module is the conversion of the text output of ASR into semantically meaningful objects. Typically, NLU modules operate on ASR output which is assumed to be correct. If ASR makes an error, NLU is not necessarily expected to interpret the output of ASR correctly. In this challenge, however, the text input to NLU is noisy.

This simulates the real-life data which is typically presented to ASR systems in the form of user commands. Being able to counter the challenges of processing this data allows for the creation of more robust and usable voice command interpreters. This research has potentially very high impact on the experience of the users of those systems who are often left frustrated by the ASR module not functioning properly. This frustration not only lowers the user's satisfaction but often causes the user to refrain from using the system completely. Proper handling of ASR errors can enable the usage of voice commands more frequently and in more different scenarios.

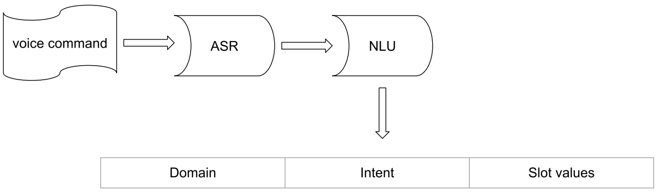


Fig. 1. The usage of the ASR system

The CAICCAIC: Centre for Artificial Intelligence Challenge on Conversational AI Correctness [1] was organized to spark the research on ASR error correction. The challenge was published on the Gonito.net platform [2]. The data set consisted of utterances of user commands annotated with the following data:

- Domain
- Intent label
- Slot values

The usage of the ASR system is presented in Figure 1 (source of the figure: [1]).

Domain is a general indication of the environment that the user is interacting with. For instance, this may be the name of the voice operated appliance, such as *Airconditioner*.

Intent is the specific action the user would like to see accomplished by using a voice command. The intent in the data set is given as a string label representing a specific function of the appliance. In the domain of an air conditioner, the example intent label is *SetTemperatureToValueOnDevice*.

Slot values are pieces of specific information being passed along with the voice command. Exemplary slot values are presented below:

```
{'device_name': 'reception room',
'value': '82 degrees fahrenheit'}
```

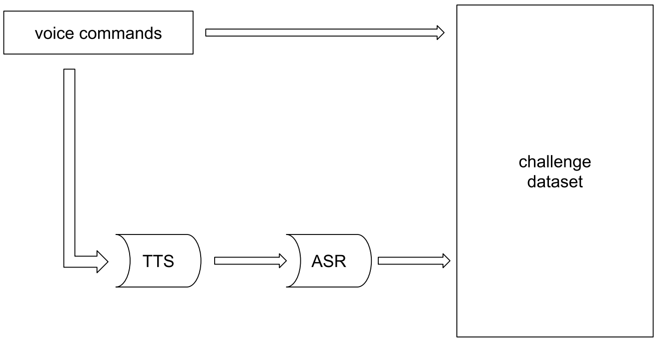


Fig. 2. Challenge data preparation

TABLE I
STATISTICS OF THE DATA

Language	Set	Utterances	Mean length
English	test	3344	9.95
English	train	13022	9.34
English	valid	3633	9.28
Spanish	test	3520	13.21
Spanish	train	15043	13.37
Spanish	valid	3546	13.15
Polish	test	3494	8.93
Polish	train	12753	8.97
Polish	valid	3498	9.02

Some of the annotated user commands in the data set of the challenge were intentionally distorted by the task organizers. However, in order not to let the participants of the challenge train NLU modules on those distortions, majority of user commands were left intact.

Distortion was achieved by first feeding the original utterances into a Text-To-Speech (TTS) system in order to obtain their audio versions. These, in turn, were converted back to text with the means of an ASR system. Since both TTS and ASR are prone to errors, the final text output was distorted. The utterances were prepared according to the scheme presented in Figure 2 (source of the figure: [1]).

The data was prepared in English, Spanish and Polish. Annotated utterances were split into train, test and validation sets. The statistics of the data are presented in Table I.

II. Related research

The problem of identifying the domain and intent can be seen as classification of distorted data. Such problems were typically approached with the use of statistical methods. One such solution is described in [3]. This work was dealing with the problem of thematic classification of texts. The texts were articles from a collection of digital libraries and were output of an OCR mechanism. Noisy data was not corrected but instead classified as is with the use of Latent Dirichlet Allocation.

The same text data was also used in another research - automatic prediction of the year of publication of the article. This was organized as the RetroC challenge [4] on the aforementioned Gonito.net platform. Winning submissions to this challenge also did not venture to correct OCR errors but instead performed the classification on the raw text data.

The problem of filling the slots, on the other hand, requires not only classification but deeper understanding of spoken language. The article [5] describes the challenges of this task and lists current solutions. Among the main challenges is the nature of ASR errors. These errors are significantly different than those observed in text (typing, spelling or grammar mistakes). In ASR whole words and phrases are substituted with fragments sounding similarly but carrying completely different meaning.

According to the authors [5], majority of researchers approaching the problem of interpreting noisy ASR output focus on correcting the errors first with the use of text correction tools, such as in [6]. The corrected ASR output is then interpreted using a NLU module. However, in recent years a new trend is observed. Experiments with direct understanding of spoken language (SLU - Spoken Language Understanding) have yielded impressive results (see for instance [7]).

The approach assumed in the CAICCAIC challenge, however, is based only on text processing. This has the following advantages over SLU:

- independence of the ASR module,
- not requiring specialistic speech-to-meaning data sets,
- ability to take full advantage of recent advances in text modelling and generation.

Among natural language processing techniques known to operate well on text containing errors we can mention character-based neural networks. The paper [10] presents research on error correction using character-based attention architecture. Thanks to operating on the character level, the solution is able to deal with out-of-vocabulary words.

III. Solution

This section describes the author's solution to the problem formulated by the CAICCAIC challenge which was evaluated as the winning submission.

The classic solution to this problem would involve training three separate classifiers – for domain, intent and slot values. The problem of identifying the domain is relatively the easiest. It can be viewed as a classification problem with few classes (there were not many distinct domain labels in the data set). Such problem could have been solved with classic text feature extraction methods (TF-IDF) and statistical classification mechanisms, such as SVM. Since the domain is heavily dependent on some specific keywords in the user prompt (e.g. temperature - air conditioner, event - calendar etc.)

Similarly, the intent classification could probably also be approached this way. Here, however, the spectrum of possible values of the intent label is wider. Moreover, the intent is not necessarily well correlated with specific words in the user prompt. This is due to the fact that a single intent can be expressed in many ways by the user. Consider the following example: the intent of checking the current temperature on an air conditioner can be expressed in the following ways (examples taken from the training set of the challenge):

- how many fahrenheits degrees are on my cooling system
- show me the temperature on the playroom thermostat
- give me temperature on my air conditioning

This makes the problem of classifying the intent label more difficult than the classification of the domain.

Moreover, the problem of filling the slot values is even more challenging. In this case it is not sufficient to guess a value from a narrow set. This problem consists in interpreting the user prompt and extracting the most important piece of information. This problem should rather be approached using text generation techniques.

This the exact idea behind the author's solution to the whole problem. In the rise of sequence to sequence language models the problem was approached with this exact technique.

Instead of training three separate models, only one is trained. During training, the expected output was concatenated into one sequence. Thus, the model was trained with the sequences in the format presented in Table II. The output of the model was then split into domain, intent and slot values.

In order to take full advantage of a sequence to sequence model it was necessary to use one created with sizeable training data. This was done in hope of achieving the best possible results especially in the task of filling the slot values. Apart from that, the model had to be multilingual as the data set contained sentences in English, Spanish and Polish. The language of the prompt was in fact annotated in the data but author's solution was aimed at providing a language-independent NLU module. The use of a large-scale model was also motivated by the fact that the input is sometimes distorted. Such models are known to deal well with the task of text generation even for noisy prompts.

At first, the FLAN-T5 [8] model was used. It was observed, however, that the results it renders fall below expectations for a specific technical reason. For Polish prompts, the outputs rendered by FLAN-T5 had erroneous Polish character encoding. This caused a significant and unnecessary drop in the quality measures of the CAICCAIC challenge. This motivated the switch to the Facebook mBART [9] model which yielded much better results altogether.

IV. Evaluation of the solution

As all submissions to the CAICCAIC challenge, the author's challenge was evaluated according to a detailed procedure described in [1]. The metric used to rank the submissions was Exact Match Accuracy (EMA), i.e. "the percentage of utterance-level predictions in which domain, intent, and all the slots are correct". Apart from that, the following additional metrics were reported for each submission:

- Domain accuracy (the percentage of utterances with correct domain prediction)
- Intent accuracy (the percentage of utterances with the correct intent prediction)
- Slot Word Recognition Rate (Word Recognition Rate calculated on slot annotations which is the percentage of correctly annotated slot values).

The evaluation scores for top five submissions are presented in Table III.

V. Error analysis

This section presents the analysis of some of the errors that the author's solution has committed. An error is counted when as per the Exact Match Accuracy metric. This means that error is reported when any of the labels is predicted incorrectly by the system.

A. Example: cold/weather

Command:

```
it is too cold in here
```

Expected output:

```
Airconditioner ChangeTemperature {}
```

System's output:

```
Weather OpenWeather {}
```

This example shows incorrect attribution of domain, intent and slot values. This error is caused by the confusion related to the word "cold" which can be associated with both air-conditioning and weather.

B. Example: temperature

Command:

```
20 degrees celsius would be ideal
temperature because it is too cold in here
```

Expected output:

```
Airconditioner SetTemperatureToValue
{'value': '20 degrees celsius'}
```

System's output:

```
Weather SetTemperatureToValue
{'value': '20 degrees celsius'}
```

This example shows incorrect attribution of the domain only. This is caused by the association of the word "temperature" with weather instead of air-conditioning. Based on this and the previous error example it is possible to conclude that the author's solution could benefit from a separate model to predict the domain only. Such model would have higher probability of predicting the domain correctly in these cases and this information could be used to affect the predictions of intent and slot values.

C. Example: expected result is not the only correct

Command:

```
give me information about my events
```

Expected output:

```
Calendar CheckCalendarOnDate {}
```

System's output:

```
Calendar OpenCalendar {}
```

The system's output is also acceptable.

TABLE II
EXAMPLE DATA FOR A TEXT2TEXT MODEL TRAINING

Input	Output
change the maximum temperature on my thermostat	Airconditioner ChangeTemperature {}
check the temperature on the keeping room sensor	Airconditioner GetTemperatureFromDevice {'device_name': 'keeping room'}

TABLE III
TOP FIVE SUBMISSIONS IN THE CAICCAIC CHALLENGE

Description	Slot WRR	Intent	Domain	EMA
author	0.87	0.92	0.96	0.75
flanT5-large	0.80	0.92	0.97	0.69
baseline	0.75	0.95	0.98	0.68
flanT5-large	0.77	0.90	0.95	0.67
flanT5-large	0.74	0.82	0.93	0.57

D. Example: minor mistake

Command:

```
update me when an appointment
in the calendar in location thornton begins
```

Expected output:

```
Calendar NotifyOnEventInLocation
{'location': 'thornton'}
```

System's output:

```
alendar NotifyOnEventInLocation
{'location': 'tornton'}
```

This error is a minor phonetic mistake.

VI. CONCLUSIONS

In general, the results achieved by the first five submissions prove the hypothesis that the problem of domain classification was relatively the easiest, intent label classification slightly more difficult and slot filling significantly more challenging than the first two problems. Also, the FLAN-T5 language model was a popular choice among the participants. This followed from the fact that the idea behind the Gonito.net platform is full collaboration between participants. Taking the solution of another participant, improving it even slightly and then uploading it as one's own is not only not forbidden but encouraged.

It can be observed that the author's solution scored significantly better in the most important EMA metric and on par with the best solutions in all other metrics. Best results in intent and domain classification individually were achieved by the baseline provided by the organizers of the challenge.

A good idea for future experiments would be a combination of solutions from the baseline and from the winning submission. Some errors identified during the error analysis process could also be corrected.

REFERENCES

[1] M. Kubis, P. Skórzewski, M. Sowański, T. Ziętkiewicz, "CAICCAIC: Centre for Artificial Intelligence Challenge on Conversational AI Correctness", *Proceedings of FedCSIS 2023*, 2023

[2] F. Graliński, R. Jaworski, Ł. Borchmann and P. Wierzchoń, "Gonito.net - Open Platform for Research Competition, Cooperation and Reproducibility" *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language / Branco António, Calzolari Nicoletta* Paris, France, European Language Resources, 2016, pp. 13–20

[3] F. Graliński, R. Jaworski, Ł. Borchmann and P. Wierzchoń, "A semi-automatic method for thematic classification of documents in a large text corpus" *Mambrini Francesco, Passarotti Marco, Sporleder Caroline : Proceedings of the Workshop on Corpus-Based Research in the Humanities (CRH)*, Warszawa, Institute of Computer Science, Polish Academy of Sciences, 2015, pp. 13–21

[4] F. Graliński, Ł. Borchmann, R. Jaworski and P. Wierzchoń, "The RetroC challenge: How to guess the publication year of a text?" *Anatonacopoulos Apostolos (red.): DATeCH 2017: Proceedings of the 2nd International Conference on Digital Access to Textual Cultural Heritage,*New York, Association for Computing Machinery, 2017

[5] M. Faruqui and D. Hakkani-Tür, "Revisiting the Boundary between ASR and NLU in the Age of Conversational Dialog Systems". *Computational Linguistics* vol. 48 (1), 2022, pp. 221–232

[6] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar and S. Tong. "Corpora generation for grammatical error correction." *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* Volume 1, 2019, pp. 3291–3301. https://doi.org/10.18653/v1/N19-1333

[7] L. Velikovich, I. Williams, J. Scheiner, P. Aleksic, P. Moreno, and M. Riley. 2018. "Semantic lattice processing in contextual automatic speech recognition for Google Assistant." *Proceedings of Interspeech*, 2018, pp. 2222–2226.

[8] H. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W Fedus et al. "Scaling Instruction-Finetuned Language Models", https://arxiv.org/pdf/2210.11416.pdf, 2022

[9] Y. Tang, C. Tran, X. Li, P. Chen, N. Goyal, V. Chaudhary et al., "Multilingual Translation with Extensible Multilingual Pretraining and Finetuning", https://arxiv.org/abs/2008.00401, 2022

[10] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, A. Ng, "Neural Language Correction with Character-Based Attention. ", https://arxiv.org/abs/1603.09727, 2016

Temporal Image Caption Retrieval Competition

TEMPORAL Image Caption Retrieval Competition was organized as part of the 1st Symposium on Challenges for Natural Language Processing. The goal of the competition was, given a picture from a newspaper and the newspaper's publication daily date, to retrieve a picture caption from a given caption set.

Multimodal models, especially combining vision and text, are gaining great recognition. One such multimodal challenge is Text-Image retrieval, which is to retrieve an image for a text query or retrieve a text for a given image. In this challenge, we introduce a task in the Text-Image retrieval setup, additionally extending the modalities with temporal data.

Language models rarely utilize any input information except for text. E.g additional data could be a text domain, document timestamp, website URL, or other metadata information. However, models trained solely on text data may be limited in usage. Additional temporal information is useful when factual knowledge is required, but the facts change over time.

The presented task is based on the Chronicling America [1] and Challenging America [2] projects. Chronicling America is an open database of over 16 million pages of digitized historic American newspapers covering 274 years. Challenging America is a set of temporal challenges built from the Chronicling America dataset.

This chapter includes the paper discussing the objectives, evaluation rules and results of the competition, authored by the organizers followed by the detailed description of the leading solution contributed by the winners of the challenge.

References

[1] Lee, B. C. G., Mears, J., Jakeway, E., Ferriter, M., Adams, C., Yarasavage, N., . . . , Weld, D. S., "The Newspaper Navigator Dataset: Extracting Headlines and Visual Content from 16 Million Historic Newspaper Pages in Chronicling America." in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM '20).* Association for Computing Machinery, New York, NY, USA, pp. 3055–3062, 2020.

[2] Pokrywka, J., Gralinski, F., Jassem, K., Kaczmarek, K., Jurkiewicz, K., Wierzchoń, P., "Challenging America: Modeling language in longer time scales," in *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 737–749, 2022.

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1331–1336
DOI: 10.15439/2023F7280
ISSN 2300-5963 ACSIS, Vol. 35

Temporal Image Caption Retrieval Competition – Description and Results

Jakub Pokrywka, Piotr Wierzchoń, Kornel Weryszko, Krzysztof Jassem
Adam Mickiewicz University
Faculty of Mathematics and Computer Science,
Emails: jakub.pokrywka@amu.edu.pl, wierzch@amu.edu.pl
korwer@st.amu.edu.pl, jassem@amu.edu.pl

***Abstract*—Multimodal models, which combine visual and textual information, have recently gained significant recognition. This paper addresses the multimodal challenge of Text-Image retrieval and introduces a novel task that extends the modalities to include temporal data. The Temporal Image Caption Retrieval Competition (TICRC) presented in this paper is based on the Chronicling America and Challenging America projects, which offer access to an extensive collection of digitized historic American newspapers spanning 274 years. In addition to the competition results, we provide an analysis of the delivered dataset and the process of its creation.**

I. Introduction

MULTIMODAL models are gaining great recognition, especially those combining image and text. A recent example is the image generation model, DALL·E 2 [1]. Tasks executed by such multimodal models usually consist of text-image retrieval, namely, either retrieving an image from its text description or retrieving a text caption for a given image. In this challenge, we introduce a task in the caption retrieval setup, additionally extending the model with temporal data.

Language models rarely utilize metadata, such as text domain, timestamp, or website URL. Additional temporal information may prove helpful when factual knowledge is required, and the facts rely on time (e.g., the answer to the question: "Who is the president of the U.S.A?" depends on the date). Temporal information may also be relevant in case of language semantic changes (e.g., the meaning of the word "gay" has shifted from "cheerful" to referring to homosexuality).

The presented task is based on the projects: Chronicling America [2] and Challenging America [3]. Chronicling America is an open database of over 16 million pages of digitized historic American newspapers covering 274 years. Challenging America is a set of temporal challenges based on the Chronicling America dataset.

The described competition was conducted using the Gonito platform [4], and its results are available at https://gonito.csi.wmi.amu.edu.pl/challenge/cnlps-ticrc. The competitions started on Feb 20, 2023, and ended on June 14, 2023. The training dataset was published in two batches (train and train2). Participants were allowed to use the delivered development dataset (dev) for training. The preliminary testing dataset (test-A) was available from the beginning of the competition. The final testing dataset (test-B) was released in the last two weeks of the competition. The golden truth for the testing datasets has not been made public. The Gonito platform is open to post-competition submissions.

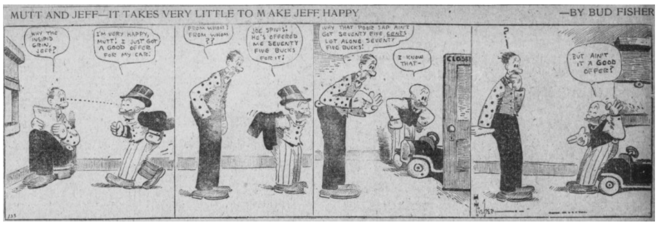

MUTT AND JEFF—IT TAKES VERY LITTLE TO MAKE JEFF HAPPY
—BY BUD FISHER
WHY THE INSIPID GRIN, JEFF?
I'M VERY HAPPY, MUTT! I JUST GOT A GOOD OFFER FOR MY CAR!
FROM WHOM? FROM WHOM ??
JOE SPIVIS! HE'S OFFERED ME SEVENTY FIVE BUCKS FOR IT!
WHY THAT POOR SAP AIN'T GOT SEVENTY FIVE CENTS LET ALONE SEVENTY FIVE BUCKS!
I KNOW THAT-
CLOSET
?!
BUT AIN'T IT A GOOD OFFER?

Fig. 1. Sample picture with a caption above. This picture comes from a newspaper issued dated Jan 11, 1928.

II. Motivation

From a linguistic and historical standpoint, Temporal Image Caption Retrieval (TICRC) holds significant value and brings various benefits. Firstly, TICRC facilitates the analysis of language evolution over time by associating image captions with specific temporal periods. Through this approach, researchers can investigate changes in vocabulary, grammar, and linguistic styles, thereby gaining insights into the adaptation and evolution of language across different historical contexts.

Secondly, TICRC contributes to the preservation and documentation of historical knowledge. Image captions accompanying visual content often contain valuable historical information. By leveraging TICRC, historians and researchers can effectively search and analyze these image captions, enabling a deeper understanding of specific historical periods, events, or cultural contexts. This process enhances the documentation of historical knowledge and enriches our comprehension of the past.

Furthermore, TICRC facilitates cross-referencing and integration of visual and textual sources. By associating image captions with specific temporal intervals, the competition makes it possible to establish connections between relevant textual documents, such as diaries, newspapers, or historical records. The interlinking of visual and textual data enhances contextualization and aids in interpreting and analyzing visual content from a historical perspective.

Moreover, TICRC offers valuable contextual information regarding the depicted scenes, individuals, or objects in images. By retrieving relevant captions based on temporal information, researchers gain a more comprehensive understanding of the context in which the images were captured. This contextualization further strengthens the interpretation and analysis of visual content within its historical framework.

In summary, Temporal Image Caption Retrieval enables the analysis of language evolution, enhances historical documentation and preservation, facilitates the integration of visual and textual sources, provides contextualization of visual content, and supports the study of cultural and societal changes over time.

III. Related Work

A. Temporal language datasets and models

Several textual benchmarks concerning the date of text publication have been published in recent years. Challenging America [3] presents a set of three temporal tasks. Authors of [5] introduce a temporal question answering task and dataset, in which the query's answer depends on a year, e.g., *Who is the current president of the USA?*. Both benchmarks contain a baseline temporal language model trained on a text with a date timestamp prepended as text. In [6], the authors propose another text classification task, including temporal information. In addition to the timestamp in the textual form the model is also trained on temporal input embeddings. The authors of [7] modify the transformer architecture, proposing a temporal attention component.

B. Multimodal vision-language models

Recently, the quality of vision-language models has improved greatly thanks to introducing models such as CLIP [1], EVAL-CLIP [8], ALIGN [9], BASIC [10], LiT [11], Flamingo [12], or GPT-4 [13] and [14].

MS COCO [15] and Visual Genome [16] are two large-scale, high-quality vision datasets annotated by humans. YFCC-100M [17] is an even larger dataset that contains user data collected from Flickr, not specifically designed for model training. Authors of CC12M [18] and LAION-5B [19] apply cleaning procedures to adapt user data for the purpose of model training. The works mentioned did not prioritize the importance of temporal data.

IV. Task Definition

The task here is to retrieve a relevant caption from a caption set for the given picture from a newspaper and the newspaper's publication daily date. For each picture, only one caption is relevant.

The dataset is provided on the challenge GitHub repository https://github.com/kubapok/cnlps-ticrc.

Figure 2 presents an example source picture with a caption.

A. Sample Data

In this section, we provide sample data. A picture and the publication date (in the YYYY-MM-DD format) of a given newspaper issue are given, as well as the collection of all captions for the given dataset type (train, train2, dev-0, test-A, or test-B). In the caption collection, a newline character is represented as \n. The challenge participant is supposed to return the list of captions from the given dataset in descending probability order.

Picture: Figure 2

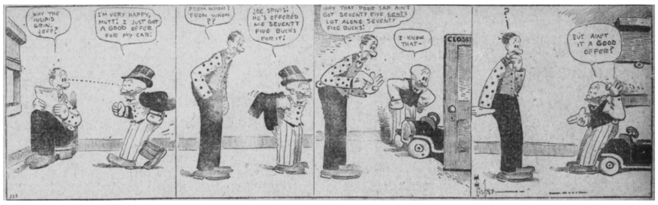

WHY THE INSIPID GRIN, JEFF?
I'M VERY HAPPY, MUTT! I JUST GOT A GOOD OFFER FOR MY CAR!
FROM WHOM! FROM WHOM ??
JOE SPIVIS! HE'S OFFERED ME SEVENTY FIVE BUCKS FOR IT!
WHY THAT POOR SAP AIN'T GOT SEVENTY FIVE CENTS LET ALONE SEVENTY FIVE BUCKS!
I KNOW THAT-
CLOSET
?!
BUT AIN'T IT A GOOD OFFER?
Fisher

Fig. 2. Sample input picture

Date timestamp: 1928-01-11

Set of all possible captions:

- "China Dinner Sets."
- "MUTT AND JEFF — IT TAKES VERY LITTLE TO MAKE JEFF HAPPY"
- "PARIS MILLINERY\nfrom every Parisian modiste,\nof note - embracing every \nstyle tendency of the fall \nand winter season \nand \n GOWNS COATS WRAPS \nTAILORED SUITS AND \nDRESSES"
- ...

Correct Output: "MUTT AND JEFF -– IT TAKES VERY LITTLE TO MAKE JEFF HAPPY"

More examples are provided in Figure 8.

B. Metric

The metric for the competition is Mean Reciprocal Rank:

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i},$$

where: |Q| — number of queries, $rank_i$ — rank position of the relevant document for the i-th query. The metric is implemented in the GEval evaluation tool [20] and available for offline use (details are provided on the competition page).

V. Data Annotation Process

The data was taken from the Challenging America project, according to the data processing rules provided there. The annotation was done manually in the Doccano [21] system, which helped effective processing of annotation pairs: image and text. The annotation platform required the annotation of the entire newspaper pages. A sample page from which a picture was selected is presented in Figure 3. The annotation of images was carried out according to given guidance rules divided into three aspects: Objects to be annotated (what to annotate), technical parameters of the image area (what technical requirements are imposed on annotated objects), and rules of text transcription (how to transcript caption texts).

These were the annotation guidance rules:

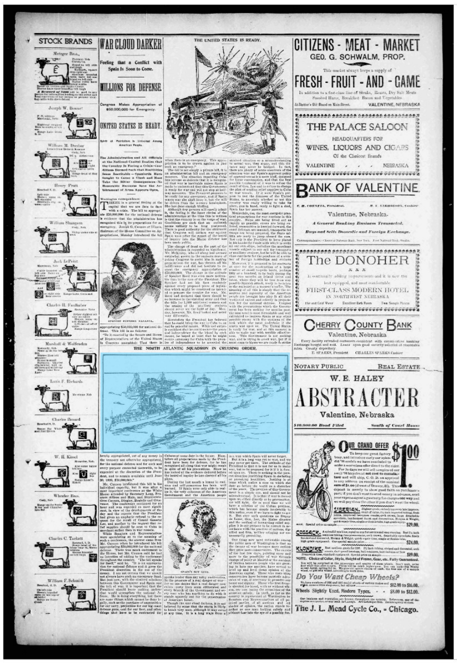


Fig. 3. Picture selected on the whole page.

a) Objects to be annotated:

- Images may be selected for annotation only if they occur along with the corresponding caption.
- The caption text should be maximum a few sentences long. In case of longer captions, the annotator should select and mark the most relevant fragment of the caption.
- The caption text should – at the discretion of the annotator – be relevant to the image in content.
- The annotator should select at most one image per page.
- If the annotator has already encountered the same image on one of the previously annotated pages, the image should not be annotated again.
- The annotator should minimize the number of portraits.

b) Technical requirements for the image area (bbox):

- The picture frame should encompass the image in its entirety (the picture should not be cut off).
- The image frame should not cover more area than the image.
- The frame must not cover the caption text.

c) Rules for text transcription:

- The transcription should preserve the character size of the original
- Punctuation and line-break characters should be preserved as in the original.
- Paragraph indentation in the text should be ignored. If the words are divided by a hyphen or line break, the original spelling (separated words) should be preserved.

The dataset was annotated mainly by one annotator, and his work took 70 hours.

TABLE I
DATA SPLIT STATISTICS

Type	**Name**	**Instances**	**Ratio**
Training	train	675	70.0
	train2	2054	
Development	dev-0	646	16.6
Testing	test-A	92	13.4
	test-B	435	

VI. Data Analysis

The dataset comprises 3902 instances, each consisting of a picture, a caption, and a date timestamp. The pictures and corresponding captions were extracted from scans of newspapers dating back to 1853, which appends the element of fuzziness in image recognition to the challenge and makes the temporal aspect even more relevant (as the image quality depends on the publication date).

A. Data Split

Five datasets have been prepared for the competition – two training sets (train, train2), a development set (dev-0), and two test sets (test-A, test-B). The final split ratio is illustrated in Table I. Precautions similar to those described in [3] have been taken to ensure that there is no detrimental overlap between the datasets.

B. Datasets Statistics

For the sake of statistical analysis, the two testing datasets and the development dataset have been combined into one dataset, referred to as the testing dataset in this section. Similarly, the two training datasets have been combined into one.

Figures 4 and 5 provide insight into the temporal variance in the frequency distributions of the instances. Whereas both datasets are negatively skewed (as suggested by the mean ≈ 1895.82 and median $= 1897.0$ of the testing dataset and mean ≈ 1903.52, median $= 1905.0$ in the case of the training dataset), the latter covers a significantly greater period containing data points between 1853 and 1922. The testing dataset spans from 1880 to 1900. Moreover, the testing dataset's standard deviation ≈ 4.18 is also less than $\frac{1}{3}$ of the training dataset's standard deviation ≈ 12.97.

The captions are measured in the number of words and characters. The captions from the testing dataset captions tend to be longer, with mean ≈ 11.77 and median $= 8.0$ words per caption and mean ≈ 66.79, median $= 44.0$ characters per caption. The respective parameters for captions from the training dataset have the following values: mean ≈ 9.80, median $= 7.0$ and mean ≈ 56.54, median $= 43.0$. There is no significant difference in the corresponding frequency distributions, as can be seen in Figures 6 and 7.

VII. Baselines

The official competition baseline is included in the competition repository and relies on the transformer model clip-ViT-B-32 [14] model without fine-tuning. The secondary baseline is the randomized caption order.

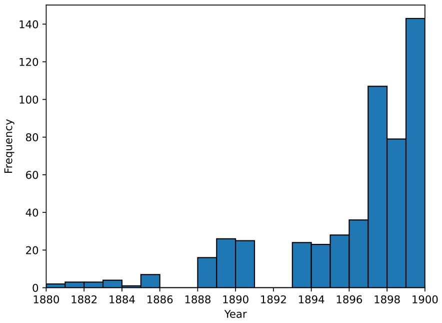


Fig. 4. Testing distribution over the years

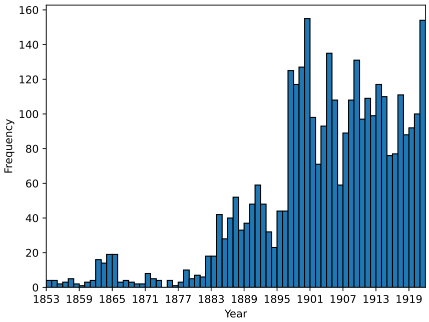


Fig. 5. Training distribution over the years

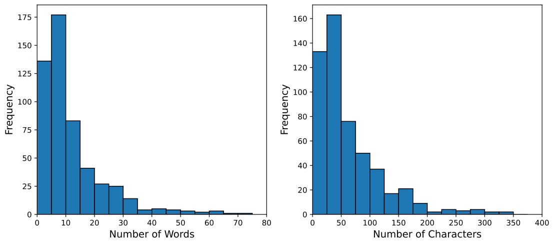


Fig. 6. Word and character per caption statistics in testing dataset

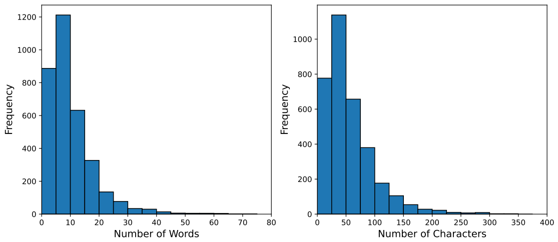


Fig. 7. Word and character per caption statistics in the training dataset

VIII. Shared Task Results

Five teams participated in the competition. Three solutions scored above the official competition baseline. The final results are provided in Table II.

TABLE II
Final competition results. The test-B dataset is used for winner determination, whereas the test-A dataset is only preliminary.

place	submitter	test-A MRR	test-B MRR	submissions
1	Kaszuba	0.6059	0.3444	6
2	s478846	0.5529	0.33850	11
3	Serba	0.3506	0.2283	1
-	**transformer baseline**	0.2697	**0.1710**	-
4	Szyszko	0.0887	0.0621	1
-	**random baseline**	0.0513	**0.0193**	-
5	s478855	0.0514	0.0137	3

The competition's winner is Patryk Kaszuba, who was invited to prepare a report for publication in the conference proceedings and presentation at FedCSIS 2023. His solution is based on EVA02_CLIP_E_psz14_plus_s9B model [8]. The model was used without fine-tuning to the competition dataset.

IX. Conclusions

In this paper, we introduced a new benchmark for temporal image caption retrieval, called TRIC (Temporal Image Caption Retrieval). TRIC includes a three-modal (vision-language-time) dataset, divided into two train sets, two test sets and a development set. The proposed task consists in selecting a caption relevant for a given image, from a given set. The temporal information is significant for the task as the data comprise scanned texts spanning the period of 274 years.

We organised the competition based on the benchmark. Five participants participated, with three of them scoring above the baseline. The benchmark is still open for further improvement of the obtained results.

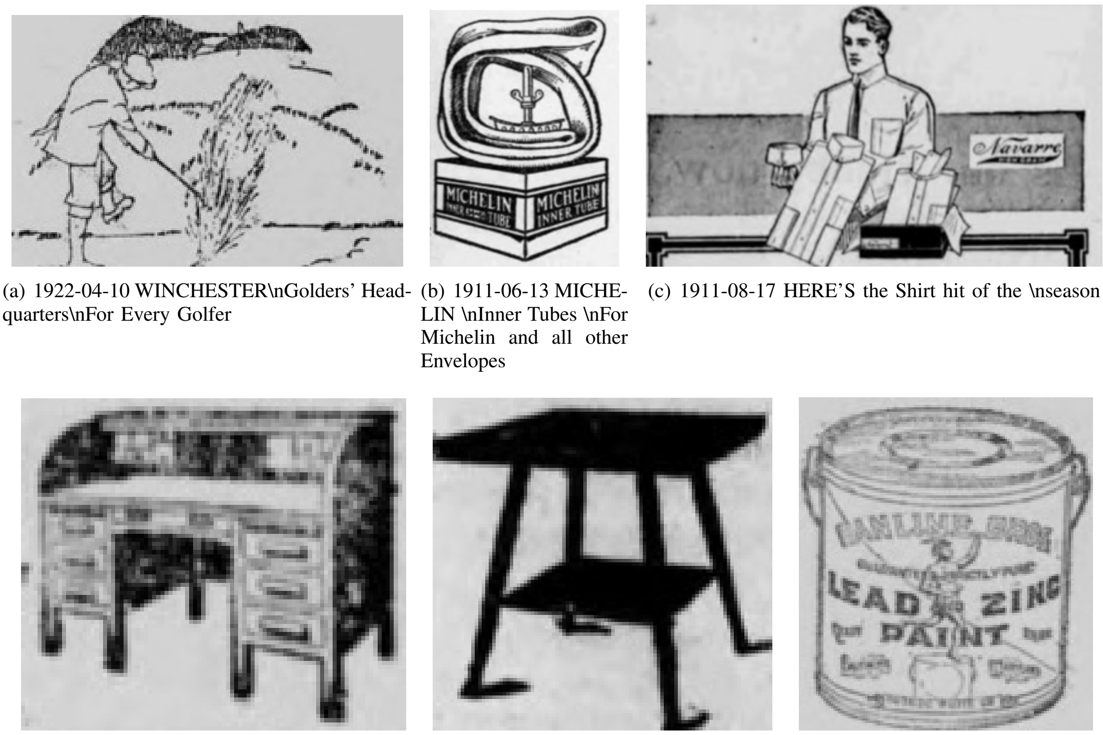
We believe that TRIC will have a positive impact on the analysis of language evolution and support the study of cultural and societal changes over time.

References

[1] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[2] B. C. G. Lee, J. Mears, E. Jakeway, M. Ferriter, C. Adams, N. Yarasavage, D. Thomas, K. Zwaard, and D. S. Weld, "The newspaper navigator dataset: Extracting headlines and visual content from 16 million historic newspaper pages in chronicling america," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, ser. CIKM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3055–3062. [Online]. Available: https://doi.org/10.1145/3340531.3412767

[3] J. Pokrywka, F. Graliński, K. Jassem, K. Kaczmarek, K. Jurkiewicz, and P. Wierzchon, "Challenging America: Modeling language in longer time scales," in *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 737–749. [Online]. Available: https://aclanthology.org/2022.findings-naacl.56



(a) 1922-04-10 WINCHESTER\nGolders' Headquarters\nFor Every Golfer

(b) 1911-06-13 MICHELIN \nInner Tubes \nFor Michelin and all other Envelopes

(c) 1911-08-17 HERE'S the Shirt hit of the \nseason

(d) 1913-02-12 DESKS AND OFFICE FURNITURE.

(e) 1902-10-16 M. O'NEIL & CO. \nFurniture \nAND... \nCarpets \nWe are now showing one of the \nmost complete lines of \nParlor \nand \nLibrary \nTables

(f) 1907-06-27 Guaranteed \nPURE. \nLEAD AND ZINC PAINTS. \n"Made in BALTIMORE"

Fig. 8. Sample images from the training dataset with the corresponding date of publication caption. The images were not selectively chosen.

[4] F. Graliński, R. Jaworski, Ł. Borchmann, and P. Wierzchoń, "Gonito.net – open platform for research competition, cooperation and reproducibility," in *Proceedings of the 4REAL Workshop: Workshop on Research Results Reproducibility and Resources Citation in Science and Technology of Language*, A. Branco, N. Calzolari, and K. Choukri, Eds., 2016, pp. 13–20.

[5] B. Dhingra, J. R. Cole, J. M. Eisenschlos, D. Gillick, J. Eisenstein, and W. W. Cohen, "Time-aware language models as temporal knowledge bases," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 257–273, 2022. [Online]. Available: https://aclanthology.org/2022.tacl-1.15

[6] J. Pokrywka and F. Graliński, "Temporal language modeling for short text document classification with transformers," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, 2022, pp. 121–128.

[7] G. D. Rosin and K. Radinsky, "Temporal attention for language models," in *Findings of the Association for Computational Linguistics: NAACL 2022*. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 1498–1508. [Online]. Available: https://aclanthology.org/2022.findings-naacl.112

[8] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv preprint arXiv:2303.15389*, 2023.

[9] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*. PMLR, 2021, pp. 4904–4916.

[10] H. Pham, Z. Dai, G. Ghiasi, K. Kawaguchi, H. Liu, A. W. Yu, J. Yu, Y.-T. Chen, M.-T. Luong, Y. Wu *et al.*, "Combined scaling for zero-shot transfer learning," *arXiv preprint arXiv:2111.10050*, 2021.

[11] X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, "Lit: Zero-shot transfer with locked-image text tuning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 123–18 133.

[12] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.

[13] OpenAI, "Gpt-4 technical report," 2023.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[16] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[17] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[18] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut, "Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3558–3568.

[19] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[20] F. Graliński, A. Wróblewska, T. Stanisławek, K. Grabowski, and T. Górecki, "GEval: Tool for debugging NLP datasets and models," in

Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 254–262. [Online]. Available: https://www.aclweb.org/anthology/W19-4826

[21] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, and X. Liang, “doccano: Text annotation tool for human,” 2018, software available from https://github.com/doccano/doccano. [Online]. Available: https://github.com/doccano/doccano

Proceedings of the 18th Conference on Computer Science and Intelligence Systems pp. 1337–1340

DOI: 10.15439/2023F4192
ISSN 2300-5963 ACSIS, Vol. 35

Multimodal Neural Networks in the Problem of Captioning Images in Newspapers

Patryk Kaszuba
Faculty of Mathematics and Computer Science
Adam Mickiewicz University
Poznań, Poland

***Abstract*—This paper presents the effectiveness of different multimodal neural networks in captioning newspaper scan images. These methods were evaluated on a dataset created for the Temporal Image Caption Retrieval Competition, which is a part of the FedCSIS 2023 conference. The task was to predict a relevant caption for a picture taken from a newspaper, chosen from a given list of captions. The results we obtained show the promising potential of image captioning using CLIP architectures and emphasize the importance of developing new multimodal methods for problems that combine multiple disciplines, such as computer vision with natural language processing.**

I. Introduction

IMAGE captioning is the task of transforming the visual information of an image into a natural language description of the image. This process combines the fields of natural language processing and computer vision. Artificial intelligence models, similarly to humans, can describe images with varying levels of detail. The variation in image descriptions generated by different models is due to differences between model architectures and training data sets. These factors affect the models' ability to extract different image features and focus attention on different aspects, resulting in diverse interpretations and semantics in the generated descriptions. Early methods were based on feature extraction techniques in which low-level visual features such as Histogram of Oriented Gradients (HOG) descriptor [1], attribute representation [2] or Support Vector Machine (SVM) [3] were combined with language models to generate captions. These methods had difficulties capturing higher-level semantic terms and processing images with varying content. The development of neural networks in the past decade led to the development of more successful methods in image captioning. Using deep neural networks eliminated the need for manual feature extraction, which resulted in the automatic creation of better representations and improved results. The first models used a combination of Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs) containing layers such as Long Short-Term Memory (LSTM) [4] or Gated Recurrent Units (GRU) [5]. Later models used attention mechanisms [6] or reinforcement learning [7] [8].

In this paper, we will focus on the use of multimodal neural networks in the problem of image captioning. In the following sections, we will discuss in detail the competition in which we participated, describe the methods that utilized three popular pre-trained neural network models CLIP, and in the last sections, present the results and describe the conclusions.

II. Related work

Understanding and interpreting the meaning of the content in an image based on the image itself is one of the more challenging problems in the field of artificial intelligence. However, in recent years, the development of deep neural networks has brought remarkable advancements in this field, and as a result, multimodal neural networks have emerged. Combining text and image representations in a joint embedding space results in significant improvements in image captioning, as demonstrated by methods such as those described in [9] or [10]. Nevertheless, the most significant results have been achieved using contrastive learning in papers presenting methods such as VILLA [11], ERNIE-ViL [12], Oscar [13], ALIGN [14] and CLIP [15].

III. FedCSIS 2023 Competition

A. Problem description

In the Temporal Image Caption Retrieval Competition, organized during FedCSIS 2023, the goal is to select the correct caption for the image. The dataset contains temporal information along with images, which can be used to accurately assign the most relevant captions to each image based on historical data.

The evaluation metric for this competition is Mean Reciprocal Rank (MRR).

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

where $|Q|$ is the total number of images in the dataset and $rank_i$ is the position of the correct caption in the ranked list for each image.

B. Dataset description

The competition dataset is based on the project “Challenging America” [16], which was initially created for three tasks. The first task, known as “RetroTemp”, focused on temporal classification. The objective was to predict the publication date based on given newspaper titles and text excerpts. In the second task, “RetroGeo”, the goal of the task was to predict the latitude and longitude coordinates of the place of issue using normalized newspaper titles, text excerpts, and fractional publishing dates. The last task, "RetroGap", involved predicting the missing word within a provided normalized newspaper title, text excerpt, and year of publication in fractional format.

For the competition, the organizers expanded the original dataset with test sets that had never been published before. The purpose of this action was to prevent participants from accessing the data during the competition.

All the collected data for the dataset comes from the "Chronicling America" [17] database, which contains digitized newspapers from 1690 until now, encompassing approximately 150,000 bibliographic title entries, as well as 600,000 library holdings records.

C. Dataset structure

The organizers of the competition split the dataset into 5 sets as follows: two training sets *train* and *train2*, a development set *dev-0*, and two test sets *test-A* and *test-B*. The total number of samples in the entire dataset was 3902 samples.

Each of the splits contained the following amounts of data:

- train - 675 samples
- train2 - 2054 samples
- dev-0 - 646 samples
- test-A - 92 samples
- test-B - 435 samples

Every single record consisted of three features: a picture, a caption text, and a publication date. The images were in grayscale, with a minimum and maximum width of 2 and 1162 pixels, respectively. For the height, the minimum value was 5 pixels, and the maximum was 1592 pixels. The header text contained both lowercase and uppercase letters, as well as symbols and special characters. The number of words in the headers varied, with the shortest containing 1 word, and the longest containing 83 words. For the publication date, the ISO 8601 (YYYY-MM-DD) format was used. The oldest publication date was 1853, and the latest one was 1922.

IV. Methods

Our solution is based on three different multimodal neural networks: **CLIP-ViT**, **OpenCLIP** [18] and **EVA-CLIP** [19]. We used the above pre-trained models for a zero-shot classification task, experimenting with their various parameter variants. As part of data preprocessing, we converted newline characters to spaces. The solution is described in the form of pseudocode in Algorithm 1.

In the initial step, we preprocess all the captions and extract their embedded vector representations obtained from the neural network's output. Then, for each image, we execute the same process to obtain its embedded vector representation. Finally, we calculate cosine similarity between an individual embedded image vector and all embedded caption vectors to determine the most similar images with captions, which we then sort based on their similarity values in descending order.

Algorithm 1 Pseudocode of our solution for image captioning

```
Require: Image vector $I = (I_1, I_2, ..., I_n)$, caption vector $T = (T_1, T_2, ..., T_m)$
  for each $t \subset T$ do
    $t \leftarrow preprocess(t)$
    $Emb_t \leftarrow CLIP(t)$
  end for
  for each $i \subset I$ do
    $Emb_i \leftarrow CLIP(i)$
    for each $t \subset Emb_t$ do
      $sim \leftarrow cosinesimilarity(Emb_i, t))$
      $Y_i.insert(sim)$
    end for
  end for
  for each $c \subset Y$ do
    $Y_c \leftarrow sort(c)\ descending$
  end for
```

A. CLIP-ViT models

We utilize the CLIP-ViT pre-trained models, based on the Vision Transformer architecture. These models were pre-trained by OpenAI on a set derived from a subset of the YFCC100M [23] dataset, with four different model parameters:

- **ViT-B-16** - 12 vision layers, 12 text layers, 512 embedding dimensions, image patch size 16x16, image resolution 224^2
- **ViT-B-32** - 12 vision layers, 12 text layers, 512 embedding dimensions, image patch size 32x32, image resolution 224^2
- **ViT-L-14** - 24 vision layers, 12 text layers, 768 embedding dimensions, image patch size 14x14, image resolution 224^2
- **ViT-L-14-336** - 24 vision layers, 12 text layers, 768 embedding dimensions, image patch size 14x14, image resolution 336^2

B. OpenCLIP models

The main difference between CLIP-ViT by OpenAI is that these models were pre-trained on the LAION-2B [24] dataset. Three new models have been created with the following parameters:

- **ViT-H-14** - 32 vision layers, 24 text layers, 1024 embedding dimensions, image patch size 14x14, image resolution 224^2
- **ViT-g-14** - 40 vision layers, 24 text layers, 1024 embedding dimensions, image patch size 14x14, image resolution 224^2
- **ViT-G-14** - 48 vision layers, 32 text layers, 1280 embedding dimension, image patch size 14x14, image resolution 224^2

C. EVA-CLIP models

The models differ from the previous ones by the implied techniques, such as the LAMB [20] optimizer, random input token dropping [21], and flash attention [22]. The **EVA02_CLIP_E_psz14_plus_s9B** model, just like the previous OpenCLIP models, was pre-trained on the LAION-2B dataset, but in the case of models **EVA02_CLIP_B_psz16_s8B** and **EVA02_CLIP_L_psz14_s4B**, they were pre-trained on the Merged-2B dataset, which combines 1.6 billion samples from the LAION-2B dataset with 0.4 billion samples from the COYO-700M dataset. The models have the following parameters:

- **EVA02_CLIP_B_psz16_s8B** - 12 vision layers, 12 text layers, 512 embedding dimension, image patch size 16x16, image resolution 224^2
- **EVA02_CLIP_L_psz14_s4B** - 24 vision layers, 12 text layers, 768 embedding dimension, image patch size 14x14, image resolution 224^2
- **EVA02_CLIP_E_psz14_plus_s9B** - 64 vision layers, 32 text layers, 1024 embedding dimension, image patch size 14x14, image resolution 224^2

V. Results

The results from the evaluated models on three subsets are presented in Table I. The metric provided in the results is the same as the one used in the competition ranking. All models evaluated by us achieved a higher score than the baseline. The best result on the test-B set, which was 0.344423 MRR, was achieved by the **EVA02_CLIP_E_psz14_plus_s9B** model, due to having the highest number of parameters among all the other models.

We also conducted an error analysis for images on which our top model struggled the most. The four images that achieved the worst MRR score are shown in Fig1. The model had difficulty choosing the correct caption for the images in cases where the caption was the author's subjective interpretation of the image and did not directly relate to the description of the elements in the photo. This can be observed in Fig. 1a and Fig. 1b. Another problem related to the model was low-resolution images, which could result in difficulties in object detection and, consequently, making inferior decisions regarding the accurate labeling of the image, as seen in Fig. 1c and Fig. 1d.

VI. Conslusion

In this paper, we presented our solution for the Temporal Image Caption Retrieval Competition. We evaluated various multimodal pre-trained models with different parameter sizes. The model with the highest Mean Reciprocal Rank metric on the dev-0 set was submitted to the competition system and ranked first place. Our approach indicates that multimodal neural networks are effective for image captioning in newspapers. For future work, we suggest improving results by fine-tuning the pre-trained models using the training data provided by the organizers. Additionally, better results may be achieved by using temporal data as an extended input for the neural network and making predictions based on historical information.

TABLE I: Experiment results

Model	**MRR**		
	dev-0	**test-A**	**test-B**
Baseline	0.156270	0.269739	0.171050
ViT-B-32 *openai*	0.162395	0.328729	0.171469
ViT-B-16 *openai*	0.193840	0.389401	0.201968
ViT-B-32 *laion2b_s34b_b79k*	0.208152	0.436798	0.221351
EVA02_CLIP_B_psz16_s8B	0.221110	0.395650	0.229678
ViT-L-14 *openai*	0.243495	0.466656	0.242640
ViT-B-16 *laion2b_s34b_b88k*	0.239205	0.430418	0.255294
ViT-L-14-336 *openai*	0.259092	0.459236	0.255777
ViT-L-14 *laion2b_s32b_b82k*	0.273631	0.505207	0.291728
ViT-g-14 *laion2b_s34b_b88k*	0.296378	0.485058	0.300874
ViT-H-14 *laion2b_s32b_b79k*	0.275778	0.490147	0.313473
ViT-bigG-14 *laion2b_s39b_b160k*	0.308845	0.572111	0.319987
EVA02_CLIP_L_psz14_s4B	0.321763	0.503575	0.332623
EVA02_CLIP_E_psz14_plus_s9B	**0.339309**	**0.605919**	**0.344423**

References

[1] A. Farhadi, S. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences from Images. ECCV (4) , volume 6314 of Lecture Notes in Computer Science, page 15-29. Springer, (2010)

[2] Vicente Ordonez, Girish Kulkarni, Tamara L. Berg. Im2Text: Describing Images Using 1 Million Captioned Photographs. Neural Information Processing Systems(NIPS), 2011.

[3] A. Farhadi, S. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. ECCV (4) , volume 6314 of Lecture Notes in Computer Science, page 15-29. Springer, (2010)

[4] A. Karpathy, and F. Li. Deep visual-semantic alignments for generating image descriptions. CVPR , page 3128-3137. IEEE Computer Society, (2015)

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. CVPR, "Show and tell: A neural image caption generator.", page 3156-3164. IEEE Computer Society, (2015)

[6] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , page 6077-6086. IEEE Computer Society, (2018)

[7] N. Xu, H. Zhang, A. Liu, W. Nie, Y. Su, J. Nie, Y. Zhang, "Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning," in IEEE Transactions on Multimedia, vol. 22, no. 5, pp. 1372-1383, May 2020

[8] Zhang, L; Sung, F; Liu, F; Xiang, T; Gong, S; Yang, Y; Hospedales, TM, Actor-Critic Sequence Training for Image Captioning. ; Volume. abs/1706.09601 ; Journal. CoRR

[9] Madhyastha et al., "End-to-end Image Captioning Exploits Distributional Similarity in Multimodal Space", EMNLP, pages 381–383, 2018

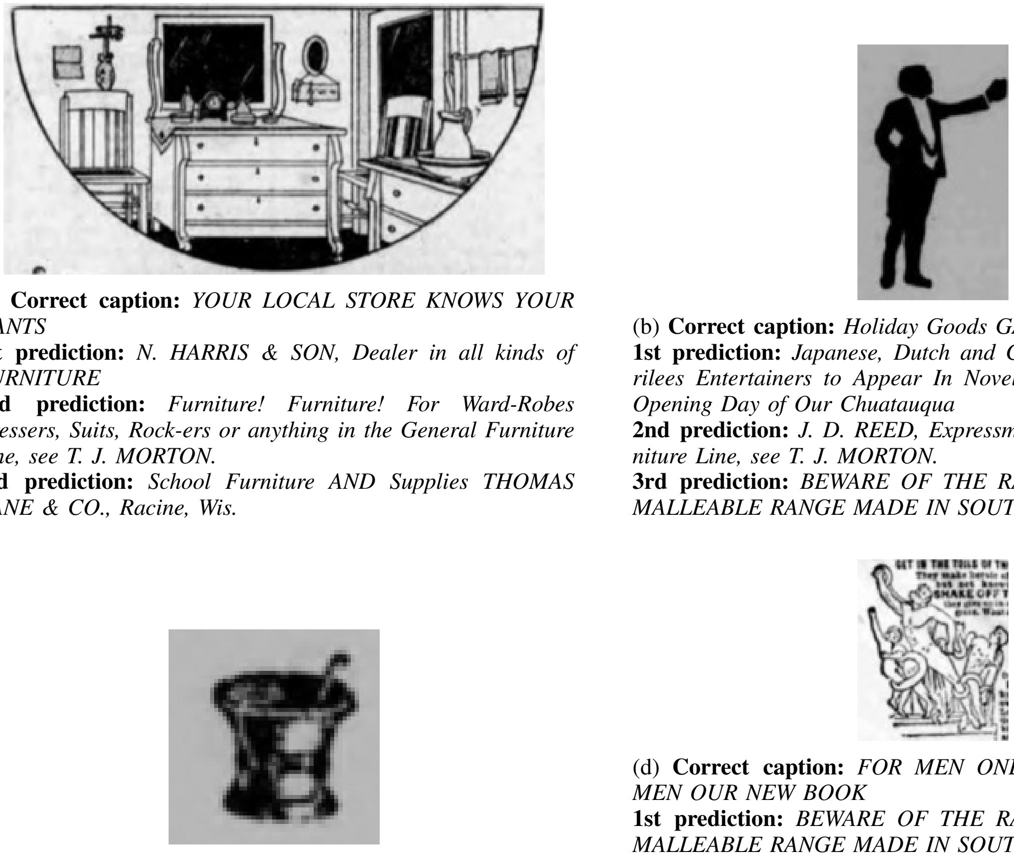
[10] YC Chen, L Li, L Yu, A El Kholy, F Ahmed, Z Gan, Y Cheng, J Liu, UNITER: universal image-text representation learning. In ECCV, vol. 12375, pages 104–120. 2020

[11] Z. Gan, Y.-C. Chen, L. Li, C. Zhu, Y. Cheng, and J. Liu, "Large-scale adversarial training for vision-and-language representation ". In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 555, 6616–6628, 2020.

[12] Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, & Haifeng Wang. (2021). ERNIE-ViL: Knowledge Enhanced Vision-Language Representations Through Scene Graph

[13] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, & Jianfeng Gao. (2020). Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks.

[14] Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. ICML, 2021



(a) **Correct caption:** *YOUR LOCAL STORE KNOWS YOUR WANTS*
1st prediction: *N. HARRIS & SON, Dealer in all kinds of FURNITURE*
2nd prediction: *Furniture! Furniture! For Ward-Robes Dressers, Suits, Rock-ers or anything in the General Furniture Line, see T. J. MORTON.*
3rd prediction: *School Furniture AND Supplies THOMAS KANE & CO., Racine, Wis.*

(b) **Correct caption:** *Holiday Goods GALORE*
1st prediction: *Japanese, Dutch and Colonial Sketches Merrilees Entertainers to Appear In Novel Musical Program on Opening Day of Our Chuatauqua*
2nd prediction: *J. D. REED, Expressman and Drayman Furniture Line, see T. J. MORTON.*
3rd prediction: *BEWARE OF THE RANGE PEDLER! THE MALLEABLE RANGE MADE IN SOUTH BEND*

(c) **Correct caption:** *Down go the Prices AT THE Drug Store!*
1st prediction: *C. C. HURLEY, Hardware, Agricultural Implements, Paints OILS, GLASS, CUTLERY, GUNS, ETC*
2nd prediction: *HOUSEHOLD WARE*
3rd prediction: *PORTABLE MILLS For Corn Meal STRAUR & CO., P. O. Box 1430, Cincinnati.*

(d) **Correct caption:** *FOR MEN ONLYYOUNG MEN OLD MEN OUR NEW BOOK*
1st prediction: *BEWARE OF THE RANGE PEDLER! THE MALLEABLE RANGE MADE IN SOUTH BEND*
2nd prediction: *HOSTETTER'S CELEBRATED STOMACH BITTERS*
3rd prediction: *Doctor Henderson OVER 27 YEARS OF SPECIAL PRACTICE Seminal Weakness & Sexual Debility,Syphilis,Book Free Museum of Anatomy,Stricture Rheumatism*

Fig. 1: The figure presents four images from the dev-0 set in which the model achieved the worst results in predicting the correct caption

[15] A. Radford et al. "Learning Transferable Visual Models from Natural Language Supervision". In: Int. Conf. Mach. Learn. PMLR, 2021, pp. 8748–8763.

[16] Pokrywka, J., Gralinski, F., Jassem, K., Kaczmarek, K., Jurkiewicz, K., & Wierzchoń, P. (2022, July). Challenging America: Modeling language in longer time scales. In Findings of the Association for Computational Linguistics: NAACL 2022 (pp. 737-749).

[17] Lee, B. C. G., Mears, J., Jakeway, E., Ferriter, M., Adams, C., Yarasavage, N., Thomas, D., Zwaard, K., & Weld, D. S. (2020). The Newspaper Navigator Dataset: Extracting Headlines and Visual Content from 16 Million Historic Newspaper Pages in Chronicling America. Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 3055–3062.

[18] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, J. Jitsev, Reproducible Scaling Laws for Contrastive Language-Image Learning , Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 2818-2829

[19] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao EVA-clip: improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389.

[20] You, Y., Li, J., Reddi, S. J., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., & Hsieh, C.-J. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.

[21] Li, Y., Fan, H., Hu, R., Feichtenhofer, C., & He, K., "Scaling Language-Image Pre-Training via Masking." CVPR, 2023.

[22] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. 2022.

[23] Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D. & Li, L.-J. (2016). YFCC100M: the new data in multimedia research.. Commun. ACM, 59, 64-73.

[24] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, J. Jitsev, LAION-5B: An open large-scale dataset for training next generation image-text models, NIPS 2022, pp. 25278-25294.

Author Index