Annals of Computer Science and Information Systems
Volume 44

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS)

September 14–17, 2025. Kraków, Poland

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

PTI

Annals of Computer Science and Information Systems, Volume 44

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS)

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

POLSKIE TOWARZYSTWO INFORMATYCZNE
POLISH INFORMATION PROCESSING SOCIETY

2025, Warszawa, Polskie Towarzystwo Informatyczne

Annals of Computer Science and Information Systems, Volume 44

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS)

ISBN 978-83-973291-8-8

ISSN 2300-5963
DOI: 10.15439/978-83-973291-8-8

Ul. Solec 38/103
00-394 Warsaw
Poland

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

Cover art:
Grzegorz Lechwar,
Elbląg, Poland

Also in this series:

Volume 45: Communication Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN 978-83-973291-9-5**

Volume 43: Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN WEB: 978-83-973291-6-4, ISBN ART 978-83-973291-7-1**

Volume 42: Proceedings of the Ninth International Conference on Research in Intelligent Computing in Engineering **ISBN 978-83-973291-5-7**

Volume 41: Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN WEB: 978-83-973291-0-2, ISBN USB: 978-83-973291-1-9**

Volume 40: Position Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN WEB: 978-83-969601-9-1, ISBN USB: 978-83-969601-0-8**

Volume 41: Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN WEB: 978-83-973291-0-2, ISBN USB: 978-83-973291-1-9**

Volume 39: Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS), **ISBN WEB: 978-83-969601-6-0, ISBN USB: 978-83-969601-7-7, ISBN ART 978-83-969601-8-4**

Volume 38: Proceedings of the Eighth International Conference on Research in Intelligent Computing in Engineering, **ISBN WEB: 978-83-969601-5-3**

Volume 37: Communication Papers of the 18th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-969601-3-9, ISBN USB: 978-83-969601-4-6**

Volume 36: Position Papers of the 18th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-969601-1-5, ISBN USB: 978-83-969601-2-2**

Volume 35: Proceedings of the 18th Conference on Computer Science and Intelligence Systems, **ISBN WEB 978-83-967447-8-4, ISBN USB 978-83-967447-9-1, ISBN ART 978-83-969601-0-8**

Volume 34: Proceedings of the Third International Conference on Research in Management and Technovation **ISBN 978-83-965897-8-1**

Volume 33: Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering, **ISBN WEB: 978-83-965897-6-7, ISBN USB: 978-83-965897-7-4**

DEAR Reader it is our pleasure to present to you the Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS 2025), which took place on September 14-17, 2025, in Kraków, Poland.

Position papers comprise two categories of contributions – challenge papers and emerging research papers. *Challenge papers* propose and describe research challenges in theory, or practice, of computer science and intelligence systems. Papers in this category are based on deep understanding of existing research or industrial problems. Based on such understanding and experience, they define new exciting research directions and show why these directions are crucial to the society at large. *Emerging research papers* present preliminary research results from work-in-progress, based on sound scientific approach but presenting work not completely validated as yet. They describe precisely the research problem and its rationale. They also define the intended future work including the expected benefits from solution to the tackled problem. Subsequently, they may be more conceptual than experimental.

FedCSIS 2025 was chaired by Jarosław Wąs. Moreover, Tomasz Hachaj was the Chair, while Marian Bubak, Marek Grzegorowski and Łukasz Rauch, were the Co-Chairs of the Organizing Committee.

This year, FedCSIS was organized by the Polish Information Processing Society (Mazovia Chapter), IEEE Poland Section Computer Society Chapter, Systems Research Institute of Polish Academy of Sciences, The Faculty of Mathematics and Information Science Warsaw University of Technology, The Faculty of Electrical and Computer Engineering of the Rzeszów University of Technology and The Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering AGH in cooperation with The Faculty of Metals Engineering and Industrial Computer Science AGH, The Faculty of Materials Science and Ceramics AGH, and Centre for Computational Personalised Medicine SANO.

FedCSIS 2025 was technically co-sponsored by IEEE Poland Section, IEEE Poland Section Computer Society (Gdańsk) Chapter, IEEE Czechoslovakia Section Computer Society Chapter, IEEE Poland Section Systems, Man, and Cybernetics Society Chapter, IEEE Serbia and Montenegro Section Computational Intelligence Society Chapter, IEEE Serbia and Montenegro Section Young Professionals Affinity Group, Committee of Computer Science of the Polish Academy of Sciences and Mazovia Cluster ICT.

FedCSIS 2025 was organized in collaboration with the Strategic Partner QED Software, and sponsored by Intel+Lenovo, Jupiter as well as MDPI Electronics, MDPI Applied Sciences and MDPI AI journals. Moreover, FedCSIS 2025 has been conducted under Honorary Patronages of Professor Jerzy Lis, Rector of the AGH University of Kraków and of Aleksander Miszalski, Mayor of Krakow, as well as under patronages of the Ministry of Digital Affairs of the Republic of Poland, Polish Artificial Intelligence Society (PSSI), Forum Akademickie and Naukowe Towarzystwo Informatyki Ekonomicznej. Finally, media patronage was provided by Krakow.pl, TVP Info, TVP3 Kraków, and Kraków Convetion Bureau.

During FedCSIS 2025 four keynote speakers delivered lectures providing a broader context for the conference participants. These presentations were:

- Damaševičius, Robertas, Kaunas University of Technology, Lithuania
  Keynote title: AI-Driven Innovations in Brain Cancer Research
- Dustdar, Schahram, TU Wien, Austria
  Keynote title: Active Inference for Distributed Intelligence in the Computing Continuum
- Jonker, Catholijn, TU Delft (main affiliation), Leiden University, Vrije Universiteit Amsterdam, Netherlands
  Keynote title: Hybrid Human-AI Intelligence to Strengthen the Reflective and Learning Capacity of Organisations
- Plank, Barbara, LMU Munich, Germany
  Keynote title: Human-centered LLMs for Inclusive Language Technology

Moreover, four past FedCSIS keynote speakers have been invited to prepare and deliver special contributions, which refer to the core focus of the conference series. These were:

- Atiquzzaman, Mohammed, University of Oklahoma, USA
  Contribution title: Q-ID: A Reinforcement Learning Framework for Adaptive Intrusion Detection
- Blum, Christian, Artificial Intelligence Research Institute, Spain
  Contribution title: Optimizing the Optimizer: An Example Showing the Power of LLM Code Generation
- Luković, Ivan, University of Belgrade, Serbia
  Contribution title: New Education Challenges in Profiling Digital Experts for a Digital Economy Era
- Skowron, Andrzej, Systems Research Institute Polish Academy of Sciences, Poland
  Contribution title: Interactive Granular Computing: Toward Computing Model for Complex Intelligent Systems

At the time, when you are reading this text, videos of the keynote presentations and of invited contributions, delivered during the FedCSIS 2025 conference, are already available on the official conference website (www.fedcsis.org). We warmly encourage you to visit the website and watch these recordings to gain additional insights and perspectives shared by distinguished speakers.

Finally, as a part of official Conference Opening, a special presentation, entitled: *Paths to Zero Emission Computing – Reducing Energy Consumption, and carbon emissions in HPC and AI environments*, was delivered by Tikiri Wanduragala, Technology Leader Lenovo Infrastructure Solutions Group (ISG), Lenovo UK and Ireland. An extended abstract, outlining main pints of this presentation can be found in this volume.

FedCSIS 2025 consisted of Main Track, with five Topical Areas, and 12 Thematic Sessions. Some of Thematic Sessions have been associated with the FedCSIS conference series for

many years, while some of them were relatively new. The role of the Thematic Sessions is to focus and enrich discussions on selected areas, pertinent to the general scope of the conference, i.e. intelligence systems.

Each contribution, found in this volume, was refereed by at least two referees. They are presented in alphabetic order, according to the last name of the first author. The specific Topical Area or Thematic Session that given contribution was associated with is listed in the article metadata.

The delivery of FedCSIS 2025 required a dedicated effort of many people. We would like to express our warmest gratitude to all Topical Area Curators, Thematic Session organizers, members of the FedCSIS 2025 Senior Program Committee and members of the FedCSIS 2025 Program Committee (a total of more than 600 individuals), for their hard work in attracting and reviewing all submissions. We thank the authors of papers for their great contribution to the theory and practice of Computer Science and Intelligence Systems. We are grateful to Keynote and Invited Speakers for sharing their knowledge and experiences with the participants. Last, but not least, we acknowledge, one more time, Jarosław Wąs, Tomasz Hahaj, Łukasz Rauch, Anna Smyk, Anna Stolarczyk Piotrowska, Marian Bubak and Marek Grzegorowski, and their Team, Anastasiya Danilenka and Paweł Szmeja, as well as a fantastic group of student helpers. We are very grateful for your efforts!

We also hope to meet you again for the 21st Conference on Computer Science and Intelligence Systems (FedCSIS 2026) which will take place in Riga, Latvia, on August 23-26, 2026.

Co-Chairs of the FedCSIS Conference Series

Bolanowski, Marek, *Rzeszów University of Technology, Poland*

Ganzha, Maria, *Warsaw University of Technology, and Systems Research Institute Polish Academy of Sciences, Poland*

Maciaszek, Leszek, (Honorary Chair), *Macquarie University, Australia and Wrocław University of Economics, Poland*

Paprzycki, Marcin, *Systems Research Institute Polish Academy of Sciences, Poland*

Ślęzak, Dominik, *QED Software and University of Warsaw, Poland*

Position Papers of the 20th Conference on Computer Science and Intelligence Systems

September 14–17, 2025. Kraków, Poland

TABLE OF CONTENTS

POSITION PAPERS

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 1–8
DOI: 10.15439/2025F2032
ISSN 2300-5963 ACSIS, Vol. 44

Enhancing Arabic ASR in Noisy and Transcoding EVS Conditions: A Multimodal Deep Learning Study

Lallouani Bouchakour
0000-0003-2070-5115
Scientific and Technical Research Center for the Development of the Ar- abic Language (CRSTDLA) Algiers, Algeria.
l.bouchakour@crstdla.dz
lbouchakour@usthb.dz

Khaled Lounnas
0000-0003-2649-4419
University of Sciences and Technol- ogy Houari Boumediene (USTHB) Speech Communication and Signal Processing Laboratory (LCPTS), P.O. Box 32, Bab Ezzouar, 16111 Algiers, Algeria.
k.lounnas@crstdla.dz

Ahmed Krobba
0000-0002-7197-1870
University of Sciences and Technol- ogy Houari Boumediene (USTHB), Speech Communication and Signal Processing Laboratory (LCPTS), P.O. Box 32, Bab Ezzouar, 16111 Algiers, Algeria.
akrobba@usthb.dz

***Abstract*—In this paper, we investigate the impact of speech transcoding and noise on the performance of Arabic automatic speech recognition (ASR) systems based on deep learning. We apply Non-negative Matrix Factorization (NMF) as a denoising preprocessing step to enhance robustness to noise. Three deep architectures—CNN-LSTM, LSTM, and DNN—are evaluated using fused acoustic features including MFCCs, Mel- spectrograms, and Gabor filter representations. Experiments are conducted under four signal-to-noise ratio (SNR) conditions (−5 dB, 0 dB, 5 dB, and 10 dB) on both transcoded and non-transcoded speech. Results show that the CNN-LSTM model achieves the highest accuracy of 87% at 10 dB SNR on clean (non-transcoded) speech using multimodal features. However, speech recognition performance degrades by 2–4% when using the Enhanced Voice Services (EVS) codec, especially in high-noise environments. Specifically, accuracy drops from 65.00% to 61.43% at −5 dB SNR, and from 87.00% to 84.00% at 10 dB SNR due to transcoding. These findings highlight the negative impact of mobile codec compression on ASR systems, particularly under low-SNR conditions. Our study confirms the effectiveness and stability of NMF-based feature fusion and denoising in improving recognition, offering insights into deploying Arabic ASR in real-world scenarios such as mobile and VoIP communications.**

***Index Terms*—Audio transcoding, Noise, Arabic speech, NMF; CNN-LSTM, LSTM, DNN, SNR.**

I. Introduction

AUTOMATIC Speech Recognition (ASR) technologies have achieved remarkable performance in clean, controlled environments with the advancement of deep learning and sophisticated feature extraction techniques. Their prowess in real-world environments under hostile conditions such as mobile communication, Voice over IP (VoIP) services, and low-bandwidth channels remains an enduring challenge. In these situations, speech signals are usually distorted by not only background noise but also compression distortions due to speech codecs, such as those employed in Enhanced Voice Services (EVS). The dual distortions greatly impair speech intelligibility and acoustic coherence, leading to drastic degradation of ASR performance. Traditional automatic speech recognition (ASR) systems, being predominantly Hidden Markov Model (HMM)- and Gaussian Mixture Model (GMM)- based [1][2], are plagued with limited robustness in mildly noisy environments. Their accuracy significantly with nonlinear distortions via lossy speech compression. Deep learning models—Deep Neural Networks (DNNs), Long Short-Term Memory (LSTM) networks, and hybrid Convolutional Neural Network–LSTM (CNN-LSTM) architecture—have overwhelmed such traditional practices in recent years due to their strong ability to learn complicated speech patterns [16]. Due to all these developments, current state-of-the-art ASR engines are still very susceptible to non-stationary noise and encoding artifacts, particularly in the absence of any special preprocessing. A hard problem arises in mobile and internet communication systems, where speech signals are typically compressed by low-bitrate codecs (EVS), perceptually optimized rather than acoustically faithful [18,19]. The compression causes time-frequency distortions that mask important phonetic information, significantly degrading ASR performance. Moreover, these distortions become exacerbated under low signal-to- noise ratio (SNR) conditions, such as –5 dB, significantly making it difficult to obtain correct speech recognitionIn order to address these challenges, a speech enhancement method based on Non-negative Matrix Factorization (NMF) is put forward in this research. As an unsupervised learning algorithm, NMF decomposes the magnitude spectrogram of noisy speech into low-rank, non-negative bases and temporal activations [8,9,10,11]. With separate modeling of speech and noise components, efficient noise reduction can be achieved without prior noise training. This feature makes the approach extremely adaptive to dynamic and changing acoustic environments. Moreover, we investigate the impact of Enhanced Voice Services (EVS) transcoding on the performance of Arabic automatic speech recognition (ASR), which is an under investigated area considering the widespread use of EVS deployment in mobile wireless networks. In this regard, we compare the performances of three deep learning- based architectures: deep neural networks (DNNs), long short-term memory (LSTM) networks, and a hybrid convolutional-LSTM (CNN-LSTM) network [16]. These models are acquired on the basis of a multimodal fusion of acoustic features like Mel-frequency Cepstral Coefficients (MFCCs), Mel-spectrograms, and Gabor filter-based descriptors.

Topical area: Network Systems and Applications

By fusing complementary spectral and temporal representations of speech, our work achieves increased robustness in adverse acoustic conditions.

This paper advances the understanding of Automatic Speech Recognition (ASR) robustness in challenging acoustic environments through the following key contributions:

- A thorough analysis of ASR performance under simulated combined noise and Enhanced Voice Services (EVS)-induced distortions, considering serious degradation trends under actual-like unfavorable conditions.
- A novel preprocessing system with NMF as the underlying framework to enhance the quality of noisy and transcoded speech, significantly enhancing downstream ASR accuracy.
- A comparative study of deep learning-based ASR models through integrated acoustic representations, demonstrating their ability to successfully counter compounded speech distortions.

The remainder of this paper is organized as follows: Section II presents speech enhancement techniques based on Non-Negative Matrix Factorization (NMF) in order to establish the theoretical framework for our preprocessing approach. Section III presents the feature extraction methods investigated in this work, noting their suitability for noise and transcoded speech. Section IV describes the deep learning-based Automatic Speech Recognition (ASR) models used in this study. Section V describes the speech corpus, experiment setup, and discusses the results in various degradation conditions. Section VI summarizes the paper with the most significant results and future directions of work.

II. Automatic Speech Recognition Over Mobile Network And Speech Enhancement

Today, with rapid expansion of *cellular networks* for voice services, system design for making speech recognition systems reliable and solid in the environment is a paramount issue of research.

Noise is introduced by cellular network transmission, bandwidth constraint, signal degradation, all of which are certain to impact recognition. In an effort to combat these factors, strategies from effective robust automatic speech recognition techniques to advanced speech enhancement approaches have been developed. This section explains these strategies in depth, beginning with the exploration of how the performance of speech recognition systems under mobile network conditions, followed by implementing techniques such as Non-negative Matrix Factorization for enhancing the intelligibility and quality of speech signals

A. Speech recognition over mobile Network

The incredible developments in computing and networking have spurred a huge interest in deploying Automatic Speech Recognition on Mobile Devices and Over Communication Networks, and this trend is growing.

B. Client-server architectures for Speech Recognition

Many studies focus on architectures where speech recognition is performed on a remote server, while the mobile device acts as a lightweight client. For instance, Aggarwal et al. [1] proposed optimized protocols for real-time transmission of compressed audio streams, reducing latency and bandwidth consumption. These architectures leverage the computational power of cloud data centers to run complex recognition models

C. Audio Compression and Transmission

Recognition accuracy heavily depends on the quality of the transmitted audio signal. Research has explored compression methods tailored for speech recognition, such as specialized codecs (AMR-WB, Opus) that preserve essential speech features while minimizing bitrate. Lukas et al. [2] studied the impact of different codecs on recognition performance over mobile networks.

D. Robustness to Variable Network Conditions

Mobile networks (3G, 4G, 5G) experience fluctuating bandwidth, latency, and packet loss. Kumar et al. (2020) proposed adaptive mechanisms that dynamically adjust audio quality and recognition model complexity based on network conditions to ensure smooth user experience.

Cloud-Based Speech Recognition Models

With cloud computing advances, platforms like Google Speech-to-Text, Microsoft Azure Speech Services, and IBM Watson provide APIs accessible via mobile networks. These services utilize deep learning models trained on large multilingual datasets, offering high accuracy even in noisy environments.

E. On-device vs Network-Based Recognition

Research comparing on-device and network-based speech recognition highlights trade-offs. Chen et al. (2021) showed that on-device recognition reduces latency and enhances privacy but is limited by mobile hardware constraints, justifying cloud usage for more demanding applications.

F. Speech enhancement

Speech signals under real acoustic conditions are mostly corrupted by forms of acoustic interference. Speech enhancement techniques, particularly those using spectral subtraction, have proved to significantly improve the performance of Automatic Speech Recognition (ASR) systems under noisy conditions. The observed noisy speech signal can be modeled in the time domain as:

$$y(t) = x(t) + n(t) \tag{1}$$

where $x(t)$ denotes the clean speech signal, $y(t)$ represents the observed noisy speech, and $n(t)$ is the additive noise component. By applying the Short-Time Fourier Transform (STFT).

The signals are represented in the time-frequency domain as y(f, m), $\hat{x}$(f,m), and $n(f, m)$, corresponding to the noisy speech, estimated clean speech, and noise spectrum, respectively. The basic spectral subtraction method estimates the clean speech spectrum as follows:

$$x(f,m) = y(f,m) - n(f,m) \quad (2)$$

G. Non-negative matrix factorization

Non-negative Matrix Factorization (NMF) is a widely used technique for speech enhancement that decomposes the training data of noisy speech— typically represented as a magnitude or power spectrogram—into the product of two non-negative matrices: a basis matrix and an activation (or weight) matrix. This decomposition enables the independent reconstruction of the magnitude spectrograms of both speech and noise components [8]-[9]-[10]-[11]. Formally, given a non-negative matrix $V \in R \geq 0\ n \times m$ NMF seeks to find two non-negative matric$W \in R \geq 0\ n \times r$ and $H \in R \geq 0\ r \times m$ such that:

$$V = W * H \quad (3)$$

Here, W contains the basis vectors (e.g., spectral patterns), and H contains their corresponding activations over time. The rank r is typically chosen such that $r < min(n, m)$, resulting in a low-rank approximation of the original matrix $\boldsymbol{V}$. This decomposition allows for the modeling and separation of speech and noise components in the spectrogram domain using NMF-based reconstruction techniques [9]. After segmenting the time-domain signal, each segment is transformed into the frequency domain using the Fast Fourier Transform (FF).

III. HYBRID DEEP LEARNING ARCHITECTURES FOR ASR

In this study, the Deep Neural Network (DNN) architecture comprises three hidden layers, following the design proposed in [10]. The network is trained to perform speech enhancement by mapping noisy speech inputs to their clean counterparts. Each input sample consists of a log-magnitude spectrogram computed over a window of consecutive frames, providing temporal context. The dimensionality of the input layer corresponds directly to the size of the feature vector. The output layer generates an estimated log-magnitude spectrogram of clean speech, aiming to suppress noise components effectively. Each hidden layer activation hi is calculated through a linear transformation of the input, using a weight matrix , followed by a nonlinear activation function. This layer- wise transformation allows the DNN to learn complex mappings between noisy and clean speech spectra. The network is trained using a mean squared error loss between the predicted and target clean spectrograms.

where. $Z(v) = (w)^{\mathrm{T}}\, v + a$, and W and a represent the weight matrix.
respectively.

$$h_i^l = \sigma\left(\left(w_i^l\right)^T v^l + a_i^l\right) \quad (5)$$

where w^l and a^l are the weight matrix and bias, respectively, at the hidden layer l, h_i^l is the output of the neuron.

A. LSTM (Long Short-Term Memory) Model for Speech Recognition

The Long Short-Term Memory (LSTM) network is a highly evolved version of the recurrent neural network (RNN) that was originally created to mitigate the short comings of standard RNNs—most notably the vanishing and exploding gradient issues hindering learning over long sequences. LSTM architecture consists of a memory cell and three gate mechanisms—input, forget, and output gates—which manage the flow of information into, through, and out of the cell. This architecture enables the network to retain meaningful information on large time steps and thus is most appropriate for sequence data modeling of long-term dependencies such as speech. This gating architecture allows the model to effectively extract long-term temporal relationships by discarding or main-training them suitably. Because of this capability, LSTM networks have proven to be particularly beneficial in sequential data modeling applications such as voice processing, where retaining context over time is critical. In speech recognition, it is essential to preserve the temporal context of phonemes and words to correctly interpret them. The Long Short-Term Memory (LSTM) model meets this need by processing input sequences of acoustic feature vectors, for instance, Mel-Frequency Cepstral Coefficients (MFCCs), spectrogram slices, or Gabor-based features, that represent the speech signal as a function of time. Using its internal memory characteristics, the LSTM effectively captures dynamic temporal patterns and transitions of spoken language without any need for spatial structure analysis.

Major advantages of the LSTM architecture are:

Ability to manage long-term temporal dependencies, which play a significant role in the context of continuous speech understanding.

Noise and variability insensitivity in the speech sequence length, enhancing performance under real-world conditions.

A reasonably simple and computationally efficient architecture, and thus suitable for real-time and embedded speech processing tasks.

Overall, LSTM networks continue to offer a robust and interpretable approach to sequence modeling in speech recognition tasks.

B. CNN-LSTM Model for Speech Recognition

Convolutional Long Short-Term Memory (CNN- LSTM) is a deep learning hybrid architecture which integrates Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to efficiently learn and represent speech signals. In this case, CNNs operate on time-frequency representations such as spectrograms to learn spatial features—identifying significant acoustic patterns such as formants, harmonics, and local frequency changes. These high-level feature maps are then fed into the LSTM, where it extracts their temporal dynamics and sequential dependencies inherent in natural speech. This combined architecture has strong points, particularly in noisy acoustic scenarios. CNNs are insensitive to noise and local deformations, whereas LSTMs preserve long temporal dependencies well. In contrast to traditional models relying on hand-designed features, CNN-LSTM models learn discriminative feature representations automatically from raw inputs, reducing the demands of manual feature engineering [17]. Generally, the CNN-LSTM architecture demonstrates superior performance in speech recognition tasks by leveraging spatial and temporal modeling capabilities in combination. It is well suited for application in visual time-series input tasks and has potential in real-world and multilingual speech processing.

IV. FEATURES EXTRACTION (FRONT-END)

The front-end analysis is the preliminary step of Automatic Speech Recognition (ASR), wherein the acoustic in- put signal is mapped into a series of acoustic feature vectors. This typically involves inspection of the short-term signal spectrum, which effectively characterizes the acoustic realizations of phonetic events. The optimal front- end analysis method must be able to retain all perceptually pertinent information needed for phonetic discrimination while remaining tolerant of variations that are linguistically or phonetically insignificant. We utilize two techniques for feature extraction in this paper. One technique is perceptually motivated representations of speech that we use to align the extracted features with human perception. The second is the utilization of Gabor filter-based representations because such representations suit extracting localized spectro-temporal patterns from the speech signal [7].

A. Perceptual Speech Approach

This approach is perceptually centered on speech modeling, with the focus laid on how humans interpret and process auditory signals. Methods such as: Fourier Analysis: Used to decompose the speech signal into its frequency constituents, providing a spectral description over time. Mel-Frequency Cepstral Coefficients (MFCCs): A widely employed feature extraction algorithm that maps frequencies to the Mel scale—a more perceptually human auditory scale. MFCCs capture perceptually relevant spectral information and perform best at phoneme-level discrimination. In parallel, Gabor filter banks are used as a second alter- native, particularly for extracting spectro-temporal features from time-frequency representations. Originally designed for image analysis, Gabor filters mimic the response characteristics of visual cortex neurons by extracting local frequency, orientation, and texture details. In speech processing, they are employed to promote feature representation by identifying fine-grained spectrogram patterns for better classification performance in both clean and noisy conditions [7]. The Gabor features are employed here to retrieve robust spectro-temporal information from the speech signal. Two-dimensional (2-D) Gabor modulation filters are employed to manipulate the input spectro- gram. These filters operate in frequency and time domains and produce 2-D feature vectors that capture the patterns of localized modulation. Gabor representation describes the envelope width as a function of modulation frequency in order to possess the same number of periods at every frequency. It possesses this property so that Gabor features can be used as a wavelet-like representation in frequency and time domains too [13]-[14]. The convolution of the Gabor functions $g_{u,v}(t, f)$ with the power spectrum $X(t, f)$ is given by:

$$G_{u,v}(t, f) = |X(t, f) * g_{u,v}(t, f)| \tag{6}$$

where $*$ represents the 2-D convolution operation. These resulting feature maps constitute a collection of image-like representations, each for different time-frequency modulations and filter parameters. The underlying spectro-temporal representation utilized for Gabor filtering is often obtained from the Short-Time Fourier Transform (STFT) or Mel spectrogram. The STFT is widely used for speech analysis, where the signal is segmented into overlapping frames and transformed via the Discrete Fourier Trans- form (DFT). This complex-valued STFT obtained has both magnitude and phase. The magnitude spectrogram is created by computing the absolute value of each STFT coefficient. In situations where the amplitude spectrum is modified—e.g., by masking methods—reconstruction of the time-domain signal will typically involve retaining the original phase and applying the inverse DFT. Alternatively, Non-negative Matrix Factorization (NMF) is more likely to be applied in the Mel-frequency spectral domain, which offers a frequency resolution inspired by perception aligned with human hearing.

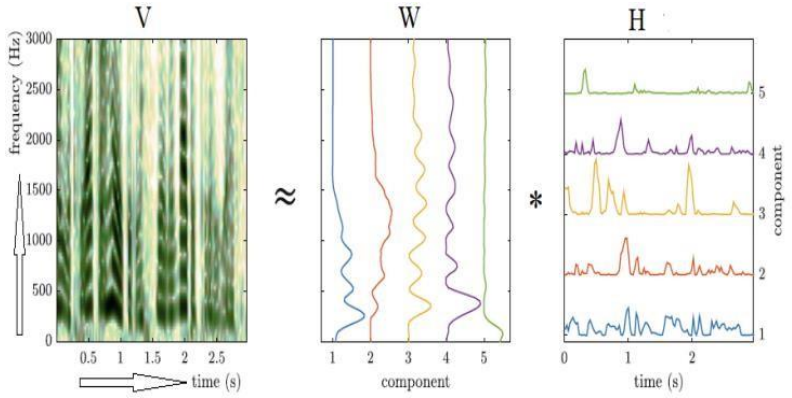


Fig. 1. The NMF model, represents the magnitude spectrum matrix V as the product of basis matrix W and H

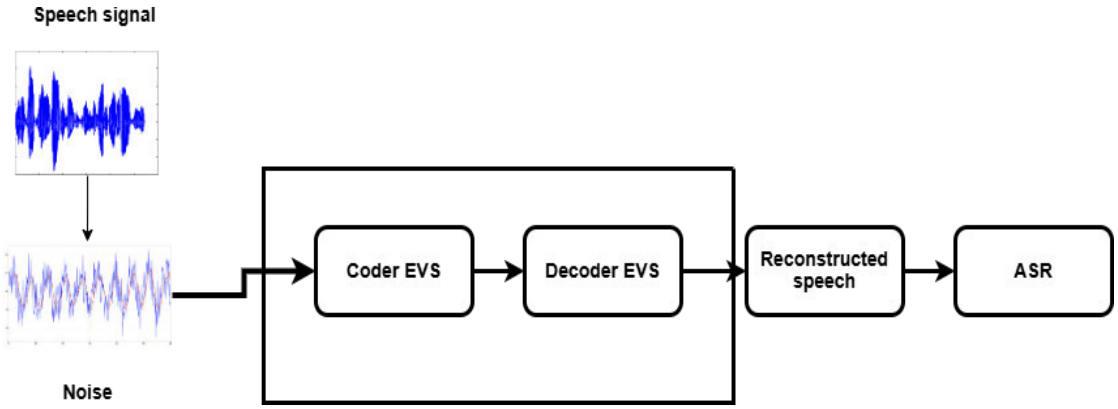


Fig.2. Speech recognition in mobile communication

TABLE. 1. SUMMARY OF ARADIGITS-BASED SPEECH DATABASES

Attribute	ARADIGIT_NOISE_NMF	ARADIGIT_EVS_NOISE_NMF
Content	Arabic digits (0 to 9)	Arabic digits (0 to 9)
Speakers	110 Algerian speakers (both genders)	Same as ARADIGIT_NOISE_NMF
Repetitions	3 repetitions per digit	Same as ARADIGIT_NOISE_NMF
Speaker Age Range	18 to 50 years	Same as ARADIGIT_NOISE_NMF
Recording Environment	Quiet room, ambient noise < 35 dB	Same as ARADIGIT_NOISE_NMF
File Format	WAV, sampled at 16 kHz, downsampled to 8 kHz	Same as ARADIGIT_NOISE_NMF
Developed By	LCPTS Laboratory	LCPTS Laboratory
Noise Type	Babble noise	Babble noise
Processing Steps	Noise added + NMF-based noise removal	Noise added + EVS transcoding + NMF-based noise removal

V. EXPERIMENTAL SETUP

In this section, we introduce the datasets, evaluation metrics.

A. Datasets

This section describes the database used to train the speech recognition models. The speech database used in this paper is the ARADIGITS database [4]. It consists of a set of 10 digits of the Arabic language (zero to nine) spoken by 110 speakers of both genders with three repetitions for each digit. This database was recorded by Algerian speakers from different regions aged between 18 and 50 years in a quiet environment with an ambient noise level below 35 dB, in .wav format, with a sampling frequency equal to 16 kHz and converting to 8kHz. We used two datasets:

1. ARADIGIT_NOISE_NMF

- ***Content***: Arabic digits from 0 to 9.
- ***Creation***: Developed at the LCPTS laboratory.
- ***Processing***:
  - This database is contaminated with various levels of babble noise.
  - The noise is then estimated and removed using the Non-negative Matrix Factorization (NMF) technique.

2. ARADIGIT_EVS_NOISE_NMF

- ***Content***: Arabic digits from 0 to 9.
- ***Creation***: Developed at the LCPTS laboratory.
- ***Processing***:
  - This database is also contaminated with various levels of babble noise.
  - It is then transcoded using an EVS.
  - Finally, the noise is estimated and removed using the NMF technique.

These databases (as illustrated in table 1) are used to evaluate the performance of our feature extraction approache under various noisy conditions by implementing advanced noise reduction techniques. We used EVS (Enhanced Voice Services) as the speech codec standardized by 3GPP for voice communication over LTE networks (VoLTE) [18]-[19]. It was developed to significantly improve audio quality compared to earlier codecs like AMR- NB and AMR-WB, while offering greater robustness to packet loss and more efficient compression.

B. Recognition Accuracy (RA)

A set of experiments was conducted to test the Recognition Accuracy (RA) by measuring the ASR performance. The Recognition Accuracy is calculated by the following equation

$$RA(\%) = \frac{N - D - S}{N} \times 100$$

where N is the total number of units (words), D is the number of deleted errors, S is the number of substituted.

IV. RESULTS OF SPEECH RECOGNITION USING HYBRID DEEP LEARNING ARCHITECTURES

The following table presents the speech recognition results for speech corrupted by different levels of SNR with babble noise and estimated using the NMF technique. Two parameterization approaches are used: MFCC representing the perceptual approach and Gabor filter representing the approach. The recognition system used is based on DNNs (Deep Neural Networks).

TABLE. 2. DNN RECOGNITION ACCURACY

Model and features	**Signal**	**SNR (-5dB)**	**SNR (0dB)**	**SNR (5Db)**	**SNR (10dB)**
MFCC	Non-trans-coded	62%	69.21%	75.65%	83.05%
MFCC	Transcoded	**54.39%**	**62.15%**	**69.82%**	**77.08%**
GFMFCC	Transcoded	**55.34%**	**64.28%**	**73.17%**	**78.77%**

This table provides an overview of the speech recognition system's performance under various noise conditions, highlighting a comparison between MFCC and Gabor filter-based feature extraction methods. The use of Non-negative Matrix Factorization (NMF) for noise reduction is essential for enhancing recognition accuracy in noisy environments. SNR Level (dB): This column denotes the Signal-to-Noise Ratio levels at which babble noise was introduced. MFCC (Perceptual Approach): This column shows the recognition accuracy achieved using Mel-Frequency Cepstral Coefficients, which capture the perceptual features of speech. Gabor Filter: This column presents the recognition accuracy achieved with Gabor features, which are designed to improve the representation of speech signal parameters by analyzing time-frequency resolution.

A. Description and Analysis of Results

The Table.3 presents a comparative analysis of the performance of three speech recognition models— CNN-LSTM, LSTM, and DNN—using various feature sets (MFCC, Mel spectrogram, and Gabor filter) under different noise conditions. The models are evaluated on both non-transcoded and transcoded speech signals, with the Signal-to-Noise Ratio (SNR) and Noise-to-Speech Ratio (NSR) values reported at -5 dB, 0 dB, 5 dB, and 10 dB for each condition. Non-Transcoded Speech:

The CNN-LSTM model, using the combination of MFCC, Mel spectrogram, and Gabor filter, shows the best performance across all noise levels, achieving a significant improvement in recognition accuracy, particularly under higher noise conditions (NSR -5 dB to 5 dB), with the highest recognition accuracy of 87.00 at SNR 10 dB.

TABLE. 3. PERFORMANCE COMPARISON OF SPEECH RECOGNITION MODELS WITH DIFFERENT FEATURE S AND RECOGNITION MODEL UNDER VARYING NOISE CONDITIONS

Model and features	Signal	SNR (-5dB)	SNR (0 dB)	SNR (5 dB)	SNR (10dB)
CNN-LSTM (MFCC+mel_db+Gabor)	Non-transcoded	**65.00**	**71.00**	**77.00**	**87.00**
LSTM (MFCC+Gabor)	Non-transcoded	63.78	70.12	75.55	86.74
DNN (MFCC+Gabor)	Non-transcoded	63.50	70.00	74.33	85.00
CNN-LSTM (MFCC+mel_db+Gabor)	Transcoded	61.43	68.00	75.00	84.00
LSTM (MFCC+Gabor)	Transcoded	60.00	67.32	73.95	82.74
DNN (MFCC+Gabor)	Transcoded	59.50	65.00	70.33	82.00

This suggests that the inclusion of Mel spectrogram and Gabor filter features provides enhanced robustness against noise. The LSTM model with MFCC and Gabor filter features also demonstrates good performance, but it falls behind the CNN-LSTM model in terms of recognition accuracy, particularly as the noise level increases. Its best performance is 86.74 at SNR 10 dB. The DNN model, while still effective, shows the lowest performance compared to the CNN-LSTM and LSTM models across all noise conditions. This model achieves its best result (85.00) at SNR 10 dB. Transcoded Speech: when the speech signal undergoes transcoding, performance degrades across all models. The CNN-LSTM model still outperforms the other two models but with a notable drop in accuracy, especially under lower noise conditions (NSR -5 dB to 5 dB). It achieves a maxi- mum recognition accuracy of 84.00 at SNR 10 dB. In the similarly, the LSTM and DNN models exhibit reduced accuracy in the transcoded speech condition, with the LSTM reaching a maximum of 82.74 at SNR 10 dB, and the DNN reaching 82.00.

Overall, the CNN-LSTM model with the combination of MFCC, Mel spectrogram, and Gabor features offers the best performance across both non-transcoded and trans- coded speech, showing strong resilience against noise. However, the performance degradation with transcoding highlights the impact of signal distortion on model effectiveness, and future work could explore improving robust- ness under transcoding scenarios.

V. Conclusion

In this study, we evaluated the robustness of Arabic automatic speech recognition (ASR) systems under challenging conditions, focusing on the combined effects of noise and speech transcoding using the Enhanced Voice Services (EVS) codec. The proposed approach incorporated Non- negative Matrix Factorization (NMF)-based denoising and multi-acoustic feature fusion as a preprocessing strategy. Experimental results demonstrated that the hybrid CNN- LSTM model, combined with the proposed preprocessing pipeline, achieved the highest recognition accuracy of 87% at 10 dB SNR on clean speech. However, EVS transcoding led to a performance drop of 2–4%, particularly in low- SNR scenarios. These findings underscore the effective- ness of NMF-based denoising and the benefit of combining multiple spectral representations to enhance ASR robust- ness in real-world environments. Future work will explore advanced speech enhancement techniques and more sophisticated architectures, including self-supervised learning models, to further improve robust-ness especially in mobile telephony and multilingual con- texts.

REFERENCES

[1] C. Aggarwal, D. Olshefski, D. Saha, Zon-Yin Shae and P. Yu, "CSR. (2005): Speaker Recognition from Compressed VoIP Packet Stream,". IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands,, pp. 970-973.

[2] Drude, L., Heymann, J., Schwarz, A., & Valin, J. M. (2021). Multi- channel Opus compression for far-field automatic speech recognition with a fixed bitrate budget. arXiv preprint arXiv:2106.07994.

[3] Dong, P., Wang, S., Niu, W., Zhang, C., Lin, S., Li, Z., ... & Tao, D. (2020). Rtmobile: Beyond real-time mobile acceleration of rnns for speech recognition. In 2020 57th ACM/IEEE Design Automation Con- ference (DAC) (pp. 1-6). IEEE

[4] Amrouche, A., Debyeche, M., Taleb Ahmed, A., Rouvaen, J. M., & Ya- goub, M. C. E. (2010). Efficient system for speech recognition in ad- verse conditions using nonparametric regression. Engineering Applica- tions of Artificial Intelligence, 23(1), 85–94.

[5] Ryumin, D., Ivanko, D., & Ryumina, E. (2023). Audio-visual speech and gesture recognition by sensors of mobile devices. Sensors, 23(4), 2284.,

[6] Bouchakour, L., & Debyeche, M. (2022). Noise-robust speech recogni- tion in mobile network based on convolution neural networks. Interna- tional Journal of Speech Technology, 25(1), 269-277.

[7] Bouchakour, L., Debyeche, M., & Krobba, A. (2024). Robust Features in Deep Neural Networks for Transcoded Speech Recognition DSR and AMR-NB. In 8th International Conference on Image and Signal Pro- cessing and their Applications (ISPA) (pp. 1-5). IEEE.

[8] M. Schmidt and R. Olsson, (2006).“Single-channel speech separation using sparse non-negative matrix factorization,” in Proc. Interspeech, pp. 3111–3119.

[9] R. J. Weiss and D. P. Ellis, (2006). “Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking,” in Proc. SAPA,, pp. 31–36.

[10] Rohlfing, C., Becker, J. M., & Wien, M. (2016,). NMF-based informed source separation. In IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 474-478). IEEE.

[11] Tuomas Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 15, no. 3, pp. 1066–1074, 2007.

[12] Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recogni- tion. IEEE/ACM Transactions on audio, speech, and language pro- cessing, 22(10), 1533-1545.

[13] Schädler, M. R., & Kollmeier, B. (2012) Normalization of Spectro- Temporal Gabor Filter Bank Features for Improved Robust Automatic Speech Recognition Systems. In : In Thirteenth Annual Conference of the International Speech Communication Association.

[14] Schädler, Marc René; Meyer, Bernd T.; Kollmeier, Birger (2012) Spec- tro-temporal modulation subspace-spanning filter bank features for ro- bust automatic speech recognition. In : The Journal of the Acoustical Society of America, vol. 131, n° 5, p. 4134–4151. DOI: 10.1121/1.3699200.

[15] Zhao, J., Li, R., Tian, M., & An, W. (2024). Multi-view self-supervised learning and multi-scale feature fusion for automatic speech recogni- tion. Neural Processing Letters, 56(3), 168.

[16] A. Mahmoudi and M. Deriche, (2004). "CNN-BiLSTM Architectures for Arabic Speech Recognition under Noise and Compression," Neural Computing and Applications, 2024.

[17] Djeffal, N., Addou, D., Kheddar, H., & Selouani, S. A. (2023). Noise- robust speech recognition: A comparative analysis of

LSTM and CNN approaches. In 2023 2nd International Conference on Electronics, En- ergy and Measurement (IC2EM) (Vol. 1, pp. 1-6). IEEE.

[18] Dietz, M., Multrus, M., Eksler, V., Malenovsky, V., Norvell, E., Pobloth, H., ... & Zhu, C. (2015). Overview of the EVS codec architecture. In 2015 IEEE International Conference on Acoustics, Speech and Sig- nal Processing (ICASSP) (pp. 5698-5702). IEEE.

[19] Wankhede, N., & Wagh, S. (2023). Enhancing biometric speaker recog- nition through MFCC feature extraction and polar codes for remote ap- plication. IEEE Access, 11, 133921-133930.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 9–14
DOI: 10.15439/2025F7856
ISSN 2300-5963 ACSIS, Vol. 44

Examining the Increasing Use of Artificial Intelligence in Education, A step Closer to Personalized Learning

Matteo Ciaschi
National Research Council (CNR)
Boglonia, Italy
matteo.ciaschi@cnr.it

Marco Barone
Universtiy of Studies of Foggia
Foggia, Italy
marco.barone@unifg.it

***Abstract*—This study, conducted within the Erasmus Programme "Language, Education and Society," investigates the growing use of Artificial Intelligence (AI) technologies in education and explores the future of learning through the lens of AI and advanced Machine Learning (ML) methods i.e. Reinforcement Learning (RL) and deep learning. AI is the automation of cognitive processes traditionally associated with human intelligence. It encompasses the development of computational systems capable of performing tasks that require knowledge, reasoning, learning, and decision-making when carried out by humans. In the educational context, AI offers transformative potential by enabling personalized learning pathways, automating instructional processes, and enhancing the adaptability and effectiveness of pedagogical strategies. This research explores how AI technologies, including ML and RL, are currently being leveraged to optimize educational practices, and it highlights the growing intersection between AI advancements and the evolving demands of the educational sector.**

***Index Terms*—Artificial Intelligence, Machine Learning, Reinforcement Learning, Personalized Learning, EdTech, Intelligent Tutoring Systems, Smart Content, Educational Innovation, Digital Education.**

I. Introduction

ARTIFICIAL Intelligence (AI) and Machine Learning (ML) have profoundly transformed nearly every aspect of modern life, including healthcare [14], transportation [1], resource management [2], agriculture [3], autonomous systems, and self-organizing processes [4]. In recent years, education has emerged as one of the most dynamic and rapidly evolving domains for AI integration. These technologies are reshaping traditional educational paradigms by enabling personalized learning experiences, intelligent tutoring systems, automated assessment, and adaptive learning environments [5].

Another highlighting factor is the growing commercial demand for AI in education, as evidenced by substantial investments from both public institutions and private enterprises. Leading technology companies—such as Microsoft, Google, Meta (formerly Facebook), and Amazon—have been investing billions of dollars into the development of AI-powered tools that span a broad range of applications, including computer vision, natural language processing, predictive analytics, and virtual assistants. Notably, many of these investments also target the education sector [7]. For instance, Google's AI-driven "Read Along" app helps young learners improve reading fluency using real-time speech recognition, while Microsoft's "Immersive Reader" enhances reading comprehension across multiple languages and learning abilities. A notable case study is IBM's Watson Education platform, which leverages AI to provide teachers with data-driven insights into student performance, helping educators tailor instruction to meet individual needs. Similarly, platforms like Carnegie Learning and Squirrel AI in China utilize AI algorithms to provide adaptive learning pathways that respond to each student's pace and level of understanding.

The COVID-19 pandemic [8] significantly accelerated the adoption of AI and EdTech solutions globally. With the abrupt shift to remote learning, educational institutions faced an urgent need for scalable and effective digital tools. During this period, AI-based solutions saw a surge in demand for supporting virtual classrooms, automating administrative tasks, and facilitating online assessments [9]. A 2021 survey conducted by the University Professional and Continuing Education Association (UPCEA) revealed that 51% of American faculty members became more optimistic about the future of online learning compared to their pre-pandemic views, signaling a long-term shift in the perception of technology-enhanced education. Furthermore, the rise of "edutainment" — the blending of education and entertainment — has fueled greater acceptance of AI in learning environments. Educational apps, games, and interactive platforms powered by AI, such as Duolingo and Khan Academy's Smart Feedback system, are increasingly popular for engaging learners of all ages. The convergence of AI and education represents a paradigm shift in how knowledge is delivered and consumed. As the demand for lifelong learning and skill acquisition continues to grow, AI is poised to play a pivotal role in shaping the future of education, making it more personalized, inclusive, and data-informed [10].

The rest of the paper is organized as follows. The next section provides a brief technical introduction to AI technologies enabling readers to grasp the argument. The section III is the main section that presents use cases to understand the

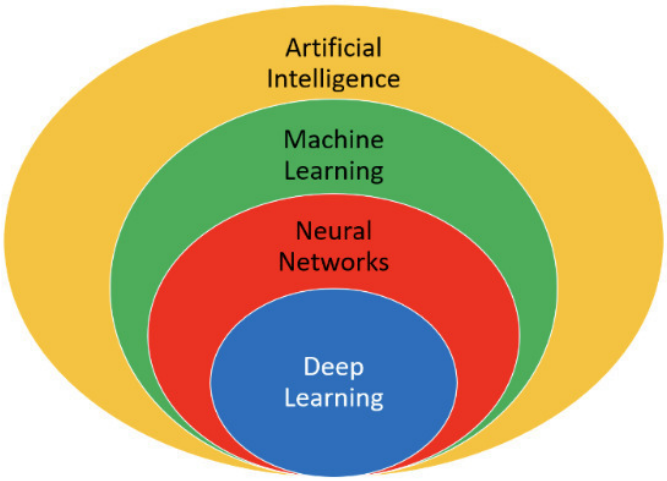


Fig. 1: Overview of AI technologies

increasing use of AI and its impact in Education. We discussed the challenges, limitations and future directions in section IV while we conclude the study in section V.

II. Technical Background

This section presents a brief introduction to Artificial Intelligence (AI) and Machine Learning (ML) technologies Figure 1.

AI is the technology that enables machines to think like humans. It can be a computer or robot able to learn, reason, solve complex problems, and can understand languages. AI tools have powerful features to recognize patterns much faster than use that help AI based systems to make decisions and this happens as AI mimics the cognitive abilities of human brain. AI-powered systems learn how humans think and process information, hence enabling them to perform tasks smartly and more efficiently. It is important to understand how AI is being implemented. AI is practically the sum of many technologies including machine learning, computer vision and natural language processing. Similarly, machine learning is the sum of many categories including supervised learning, unsupervised learning, semi supervised learning, Reinforcement Learning and Deep Learning. Among these types, RL and DL are the most advanced form of AI which enables AI to mimic a human brain's neural network. Reinforcement learning a branch of machine learning, is goal-directed learning from interaction. Reinforcement learning involves improving performance through trial-and-error experience [14]. A method with a software agent that interacts with an unknown environment, selects actions dynamically and discovers which action yields more reward [11]. Reinforcement learning focuses on teaching algorithms to make choices by providing positive feedback for preferred actions and negative feedback for unwanted ones. Similarly to how behavior is influenced by rewards and consequences in psychology, this method allows systems to gradually develop the best strategies through a process of trial and error as shown in Figure 2. The reward system

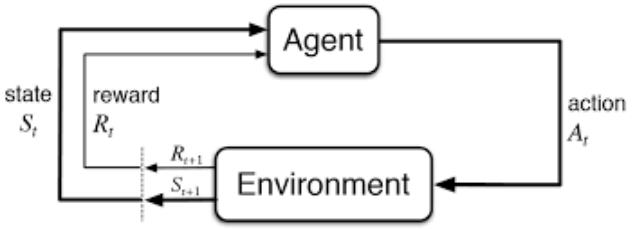


Fig. 2: Working of an RL agent presenting agent-environment interaction [11].

is crucial for guiding the agent's actions toward achieving the final goal. It serves as a feedback mechanism, clearly indicating whether a chosen action has led to a positive or negative outcome. By understanding this, the agent can adjust its strategies effectively, ensuring progress and success in reaching its objectives.

Similarly, deep learning is the latest AI tool which has brought transformation how machines train, learn and interact with environment and complex data. It is the type of learning which mimics Neural Networks (NNs) of the human brain and thus enabling machines to autonomously uncover patterns and make informed decisions from huge amounts of data [6]. NN is the main part of the deep learning algorithm, consists of layers of interconnected neurons working in collaboration to process input data. In a fully connected Deep Neural network (DNN) data flows via multiple layers and every neuron do nonlinear transformations, permitting the model to learn intricate representations of the given data. In a DNN the input layer receives data and this data is then passes through hidden layers and these central layers further transform the data using nonlinear functions. At the final stage, the output layer generates the model's prediction or output.

III. Decoding AI in Education

This section presents the various use cases that we consider to explain the increasing use and impact of AI solutions in education.

A. Personalized Learning

The first case is personalized learning that we can say is one of the best and innovative uses of AI in education. The concept of personalized learning is getting attention worldwide and it can be realized with the help of modern AI and ML tools as demonstrated in Figure 3. Personalized learning is a learning method to employ AI and ML specially Reinforcement Learning (RL) that considers the requirements of every individual student. Personalized learning means that each student's learning experience and skills are customized to adopt their needs. Personalized learning provides an opportunity to grow using their own skills and learning experience.

The AI-powered personalized learning gives flexibility to students in various aspects like: the use of material, quality of material, speed to learn, and way of teaching. Although there are various benefits of personalized learning as highlighted,

Fig. 3: Benefits of AI on personalized learning

there are also some limitations. For example, the implementation of personalized learning is a time taking task and it is difficult without the use of AI technology. Similarly, the cost associated with the implementation of technology-based infrastructure is another issue. Lastly and importantly, the training of teachers and relevant persons on the use of modern technology is another challenge that needs to be addressed.

B. Task Automation

The second case is how AI methods can be useful for task automation. The presence of adaptive learning platforms based on AI technology, can analyze student data, for example, their learning pace, strengths, weakness and performance. This information is feedback to AI systems for task automation to obtain personalized pathways for every student, providing suitable and adaptive activities, resources and contents based on their specific needs. These systems are also known as Intelligent Tutoring Systems as they offer individualized support and guidance to students [19]. Another advantage of these systems is their ability to assess an individual student's understanding, identify areas of weakness, and provide corresponding feedback, and exercises for practice. The intelligent tutoring systems adapt to every student's progress and adjust the learning material accordingly.

C. Smart Content Creation

The next use case is about the innovative use of AI technology for smart content creation in the context of education and learning. There are many examples like Information visualization, digital lesson generation and frequent content updates. Moreover, AI algorithms are also helpful in content optimization and content curation as illustrated in Figure 4. AI technology provides huge potential in improving content creation processes, assisting students to develop engaging and suitable content for their study objectives and tasks.

In addition, content creation using AI tools also saves time and offers an effective way to generate relevant contents in a short time. AI-based content optimization ensures that the content resonates with the study goals. At the same time students get valuable contents creation experience with the use of AI for content creation curation and optimization. It is important to embrace AI tools and students can use them to their learning advantage. Smart content creation based on AI technologies is the way forward for content creators students who seek to develop impactfull and engaging content in a time-oriented task.

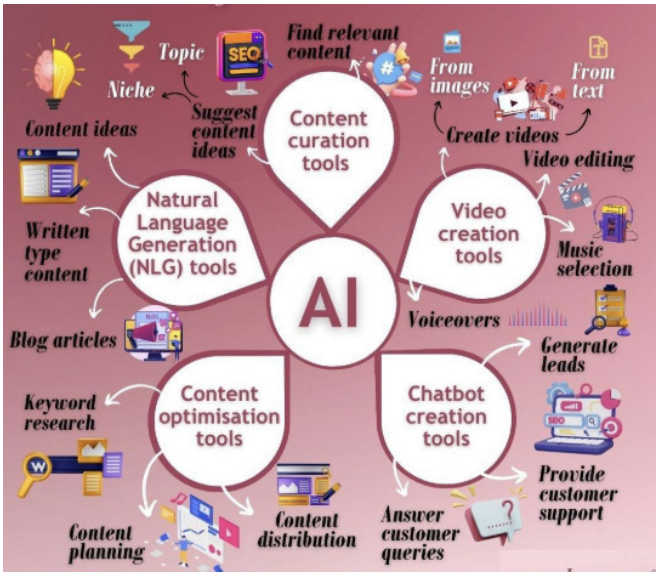


Fig. 4: AI tools for contents creation

D. AI in Examinations

An important phase of the education system and learning process is the examination. AI technology can assist both teachers and students in the examination process because it is possible to track the performance of students in examinations. The AI-enabled systems then help teachers by providing them detailed analytics of each students' performance and the performance of the whole class as well. These analytics will assist teachers in understanding which arguments or concepts are difficult for students and consequently can develop new strategies to help students in grasping highlighted topics. Similarly, AI tools are also useful to students by providing them feedback over examinations. These modern technologies not only help students in pointing out their weak parts but also assist them with personalized schemes to understand a specific topic with maximum attention and retention. Moreover, AI-powered systems can alert teachers if a student or group of students is lagging behind others in some subjects.

The scenario or problem as shown in Figure 5, can be considered as the problem of personalized learning, task management or subject selection. We have different tasks and we have to make task selection using RL policy and feedback to RL agent after selection of the particular task. This is an emulated environment where different students have to learn and perform different tasks. The performance of a student varies from task to task and similarly, the outcome of each task in terms of score (S1,S2,S3,S4) may be different for different people. The probability distribution for the reward corresponding to each task is different and is unknown. The problem for an AI agent is to learn which task to select in order to get the maximum score in a given amount of

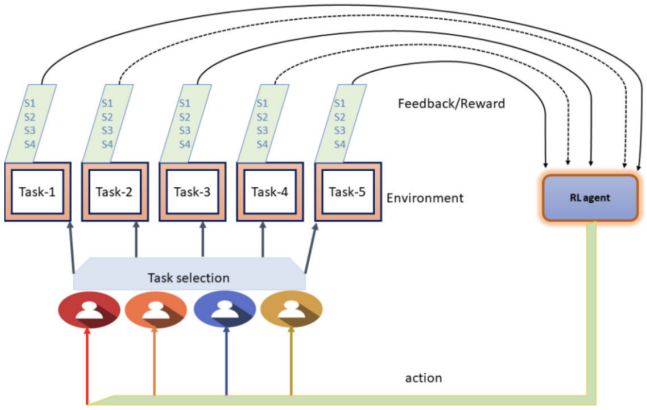


Fig. 5: An AI based subject selection system [20]

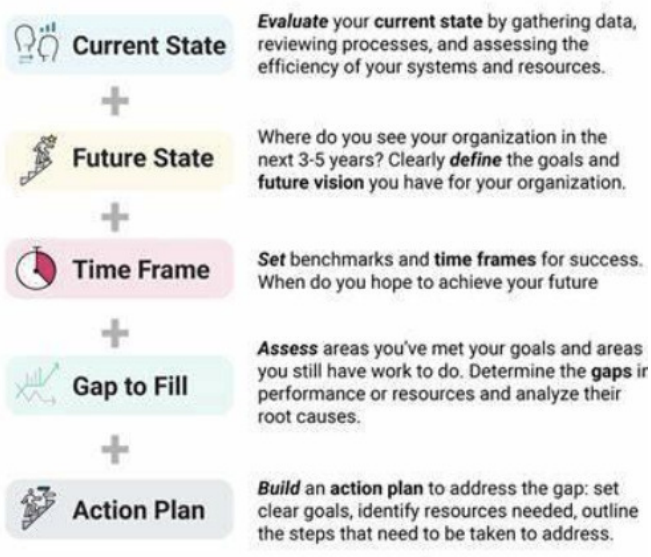


Fig. 6: An AI based subject selection system [20]

time. This problem statement is identical to a single step Markov Decision Process (MDP). The score list (S1,S2,S3,S4) measures different skills of a person during the execution of a task. The better score in skills indicates more interest and better performance for a particular task and lower score indicates that a particular task is unsuitable for a person. After a lot of interaction with the environment, the RL agent learns the most suitable task for a student.

This scenario can be modeled as a problem as a MMDP with a single state. There are in general K number of tasks and it is possible to select anyone and each task has a certain probability of returning a reward (score). Therefore, we have a single state and K possible actions (one action for each task). At each time period the agent selects one task and it receives feedback in terms of different scores (reward). The goal of the agent is to learn the best task/subject for each student in in order to maximise its long term reward. A suitable machine learning algorithm like Boltzman sampling, Epsilon decreasing, Random, Softmax, and Thompson sampling algorithms can be used to solve this MDP problem.

E. Secure and Decentralized Learning Systems

Artificial Intelligence, when integrated with emerging technologies such as blockchain, can contribute to the development of secure and decentralized learning systems [21]. One of the major concerns in digital education environments is the security and privacy of learners' data. AI-driven platforms, enhanced with decentralized technologies, can provide a transparent and tamper-proof infrastructure for storing educational records, certificates, and learning progress. This ensures that students have full control over their data and can securely share their academic achievements with educational institutions or employers without relying on centralized authorities.

Furthermore, decentralization promotes inclusivity and accessibility by enabling peer-to-peer learning networks, where educational content and credentials are distributed across secure nodes rather than hosted on a single centralized server. AI algorithms can monitor and verify these exchanges, ensuring content quality and relevance while maintaining integrity and trust in the system. As a result, secure and decentralized learning systems not only protect student data but also foster global collaboration, democratizing access to education across borders and socioeconomic boundaries.

F. Customized Data-Based Feedback, Closing the Skill Gap

The final use case examined the role of AI in delivering customized feedback to students by analyzing large datasets of their learning behavior, performance trends, and engagement metrics. This real-time, data-driven feedback is crucial for identifying individual strengths and weaknesses, which helps educators design tailored learning pathways for each student as also indicated in Figure 6. AI systems can generate predictive analytics that forecast student outcomes and suggest timely interventions, thereby preventing learning delays or dropouts.

Moreover, this personalized feedback mechanism plays a vital role in addressing the skill gap between academic training and real-world job market needs. AI can map students' learning progress against industry requirements and recommend specific skills or courses to align their competencies with emerging market demands. Educational institutions and employers can also benefit from this data, as it enables more accurate student profiling, workforce readiness assessments, and targeted curriculum development. In conclusion, AI-enabled customized feedback serves as a bridge between education and employment, helping students acquire relevant skills and enhancing the overall effectiveness of learning systems.

IV. Discussion and Future Work

The integration of AI in education signals a paradigm shift in how learning is designed, delivered, and evaluated. Reinforcement Learning, with its capacity for modeling sequential decision-making, offers substantial potential for tailoring educational experiences to individual learner behaviors. Similarly, the implementation of AI-powered tools such as chatbots

and virtual tutors allows for scalable, continuous support that can simulate human-like interaction, providing learners with instant feedback and guidance. The adoption of AR and VR technologies introduces immersive learning environments that enhance conceptual understanding through experiential simulation. However, the widespread implementation of AI in education also raises important considerations. These include ethical issues surrounding data privacy and algorithmic bias, the digital divide that limits access to advanced technologies, and the preparedness of educators and institutions to adopt AI-based methodologies. There is a clear need for policy frameworks, teacher training programs, and interdisciplinary collaboration to ensure that the benefits of AI are equitably distributed and effectively managed.

A. Future Directions

Although, artificial intelligence and machine learning tools have shown significant applications in almost all aspects of human life and education sector is one of them. but still there are many challenges and limitations that need to be addressed as a way forward [15]. In this study, we highlight some of the key points that need to be considered to translate the potential of AI technologies into effective educational practice and learning:

Teacher Training: As we discussed, teachers skills in efficient use of modern technologies is key to implement AI tools in education and learning. Educators must be equipped with the skills and tools to understand and integrate AI technologies into their pedagogical practice [17]. Therefore, it is essential and strategic to start investing in teachers and educational staff. This process requires mandatory training and equipped classrooms and labs with necessary equipment. The training should include both technical training and the development of critical perspectives on AI's role in education.

Ethical Guidelines and Data Governance: One of the most critical aspects of the use of AI technologies in each sector is the lack of ethical guidelines and lack of formal protocols. As in any other area, there should be formal ethical guidance on the use of AI and machine learning tools in education . In addition, it is necessary to have data governance and protocols to protect both teachers and students privacy and security. In conclusion, educational institutions should make and implement policies regarding data usage, user privacy, and transparency in AI decision-making processes.

Easy Access to AI Tools: When we talk about AI applications, it is normally discussed more about its usage, benefits and drawbacks but one aspect that is comparatively discussed less is the accessibility of these technologies to the masses. We all know that in the education sectors we have students as well as teachers from diverse backgrounds in terms of many factors. Therefore, it is very essential to ensure uniform and easy access to modern resources to everyone. So it is recommended that AI-empowered educational technologies should be designed and deployed with a focus on accessibility to students from different backgrounds and regions.

Cross-sector Collaboration: We highlighted the need of teachers and educators training for better use of AI in education and quipping classrooms, labs with modern infrastructures. The training and infrastructure purchase require a considerable investment and it is important to have a strong collaboration with private sector, companies and other stakeholders. Secondly, we discussed ethical and data governance protocols which is not possible without the involvement of government institutions. In summary, partnerships between educational institutions, policymakers, the AI community, and the private sector are necessary for the responsible scaling and innovation of AI solutions in education.

Support Continuous Evaluation: To sum up all previous arguments, we can state that it is important we support the positive use of AI technologies in education and learning. In conclusion, implementations should be subject to ongoing assessment to ensure they fulfill educational goals, bring innovations, meet students and teachers needs, and adapt to emerging challenges.

V. Conclusion

In conclusion, artificial intelligence particularly reinforcement learning offers a mathematically sound and practically effective framework for optimizing learning decisions and customizing educational experiences. This study provided a comprehensive exploration of AI applications across several key areas, including personalized learning, immersive technologies, and intelligent tutoring systems. The convergence of AI with education holds transformative potential, yet it also necessitates thoughtful consideration of the ethical, infrastructural, and pedagogical dimensions involved. Future research and development efforts should focus on creating inclusive, transparent, and adaptive AI systems that complement human teaching and foster lifelong learning. With strategic planning, stakeholder collaboration, and evidence-based implementation, AI can serve as a powerful catalyst in shaping the future of education

References

[1] Jamal, M., Ullah, Z., Naeem, M., Abbas, M. and Coronato, A., 2024. A hybrid multi-agent reinforcement learning approach for spectrum sharing in vehicular networks. Future Internet, 16(5), p.152.

[2] Quilliot, A. and Mombelli, A., 2024. Handling Lot Sizing/Job Scheduling Synchronization through Path Search Algorithms. Annals of Computer Science and Information Systems, 41, pp.131-138.

[3] Kepka, M., Jedlicka, K. and Charvát, K., 2024. Combining Local and Global Weather Data to Improve Forecast Accuracy for Agriculture.

[4] Andrey, S., Truscan, D., Schneider, M., Mallouli, W., Cavalli, A., Seceleanu, C. and Ahmad, T., 2024. Smart Assistants for Enhancing System Security and Resilience. In Conference on Computer Science and Intelligence Systems (pp. 151-158). Polskie Towarzystwo Informatyczne.

[5] Abomelha, F. and Newbury, P., 2024. A VARK learning style-based Recommendation System for Adaptive E-learning. Annals of Computer Science and Information Systems, 41, pp.1-8.

[6] Shah, S.I.H., Naeem, M., Paragliola, G., Coronato, A. and Pechenizkiy, M., 2023. An ai-empowered infrastructure for risk prevention during medical examination. Expert Systems with Applications, 225, p.120048.

[7] Gerhards, C., Allee, F.E. and Baum, M., 2024. AI in the Workplace: Who Is Using It and Why? A Look at the Driving Forces Behind Artificial Intelligence in German Companies. Annals of Computer Science and Information Systems, 41, pp.45-52.

[8] Qayyum H., Rizvi S.T.H., Naeem M., Khalid U.b., Abbas M., and Coronato A., "Enhancing Diagnostic Accuracy for Skin Cancer and COVID-19 Detection: A Comparative Study Using a Stacked Ensemble Method," *Technologies*, vol. 12, no. 9, p. 142, 2024. doi: https://doi.org/10.3390/technologies1209014210.3390/technologies12090142

[9] Jovanovic, J., 2024, September. The Interplay of Learning Analytics and Artificial Intelligence. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 35-44). IEEE.

[10] Barone, M., Ciaschi, M., Ullah, Z. and Piccardi, A., 2024, September. Reinforcement Learning based Intelligent System for Personalized Exam Schedule. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 549-553). IEEE.

[11] Barto, A.G., 2021. Reinforcement learning: An introduction. by richard's sutton. SIAM Rev, 6(2), p.423.

[12] Naeem, M., Coronato, A., Ullah, Z., Bashir, S. and Paragliola, G., 2022. Optimal user scheduling in multi antenna system using multi agent reinforcement learning. Sensors, 22(21), p.8278.

[13] Wang, Z., Xu, Y., Wang, D., Yang, J. and Bao, Z., 2022. Hierarchical deep reinforcement learning reveals a modular mechanism of cell movement. Nature machine intelligence, 4(1), pp.73-83.

[14] Ismail, A., Naeem, M., Syed, M.H., Abbas, M. and Coronato, A., 2024. Advancing Patient Care with an Intelligent and Personalized Medication Engagement System. Information, 15(10), p.609.

[15] Müller, J., Würth, S., Schäffer, T. and Leyh, C., 2024, September. Toward a Framework for Determining Methods of Evaluation in Design Science Research. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 231-236). IEEE.

[16] Fiorino, M., Naeem, M., Ciampi, M. and Coronato, A., 2024. Defining a metric-driven approach for learning hazardous situations. Technologies, 12(7), p.103.

[17] Kosar, T., Bjeladinović, S., Ostojić, D., Škembarević, M.S., Leber, Ž., Jejić, O.A., Furtula, F., Ljubisavljević, M.D., Luković, I.S. and Mernik, M., 2024, September. Teaching Beginners to Program: should we start with block-based, text-based, or both notations?. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 395-403). IEEE.

[18] Ismail, A., Naeem, M., Khalid, U.B. and Abbas, M., 2025. Improving adherence to medication in an intelligent environment using reinforcement learning. Journal of Reliable Intelligent Environments, 11(1), pp.1-10.

[19] Luţan, E.R. and Bădică, C., 2024, September. Literature Books Recommender System using Collaborative Filtering and Multi-Source Reviews. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 225-230). IEEE.

[20] Muddasar Naeem, Antonio Coronato, Valeriano Fabris, RL-Based Model for Improving Human Task Management Performance, IE2022 18th International Conference on Intelligent Environments 2022

[21] Saritas, H.B. and Kardas, G., 2024, September. A blockchain-based transaction verification infrastructure in public transportation. In 2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 169-176). IEEE.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 15–21
DOI: 10.15439/2025F7538
ISSN 2300-5963 ACSIS, Vol. 44

Enhancing Socio-Emotional Skills in Children with Autism through AI-Powered Serious Games: A Narrative Review

Enza Curcio
Giustino Fortunato University
Benevento, Italy
Email:
e.curcio@unifortunato.eu;
https://orcid.org/0009-0008-3541-7996

Fabrizio Stasolla
Giustino Fortunato University
Benevento, Italy
Email:
f.stasolla@unifortunato.eu

Antonio Zullo
Universitas Mercatorum
Rome, Italy
Email:
a.zullo@unifortunato.eu

Mariacarla Di Gioia
Universitas Mercatorum
Rome, Italy
Email: m.digioia@unifortunato.eu

Anna Passaro
Giustino Fortunato University
Benevento, Italy
Email:
a.passaro@unifortunato.eu

***Abstract*—This narrative review explores the integration of Artificial Intelligence (AI) and Serious Games (SGs) as a novel, interdisciplinary approach to fostering socio-emotional skills in children with Autism Spectrum Disorder (ASD). As ASD is characterized by persistent challenges in emotional understanding, social communication, and behavioral regulation, there is a growing need for interventions that are both effective and personalized. SGs provide structured, interactive environments where children can practice skills such as emotion recognition, joint attention, and empathy in a safe and motivating way. When augmented with AI, these games offer real-time feedback, dynamic personalization, and adaptive learning experiences tailored to individual cognitive and emotional profiles. This review synthesizes recent empirical evidence on AI-powered SGs targeting socio-emotional development in children with ASD. It examines the design strategies, targeted competencies, and evaluation methods used across current literature. The integration of SGs and AI is positioned as a promising and scalable tool to promote autonomy, emotional well-being, and social inclusion in neurodiverse children.**

***Index Terms*—Serious Games, Artificial Intelligence, Autism Spectrum Disorder, Socio-emotional skills, Emotion Recognition, Personalized Intervention**

I. Introduction

AUTISM Spectrum Disorder (ASD) is a complex, lifelong neurodevelopmental condition characterized by persistent difficulties in social communication and interaction, along with restricted and repetitive patterns of behavior, interests, or activities [1]. With the growing global prevalence of ASD, there is an urgent need for effective, individualized interventions that address the unique profiles and evolving needs of autistic children and their families. Among the most critical domains for intervention is the development of socio-emotional competencies, which are foundational for overall well-being, meaningful relationships, and successful participation in school and community life. This need is supported by developmental frameworks such as Bandura's Social Learning Theory [2], which emphasizes learning through observation and interaction, and cognitive-behavioral models that highlight the role of emotional awareness and regulation in adaptive functioning. These perspectives provide a theoretical basis for designing digital tools that scaffold emotional development and promote meaningful social engagement. Difficulties in emotional understanding, behavioral regulation, and interpersonal engagement often lead to heightened anxiety, social withdrawal, and limited opportunities for inclusion [3]. In response to these challenges, Serious Games (SGs) have emerged as innovative tools in therapeutic and educational contexts. Designed with specific learning or clinical objectives, SGs provide structured, immersive environments where children with ASD can safely practice and generalize socio-emotional skills. These game-based interventions leverage the affinity many autistic children have for digital and rule-based systems, increasing engagement and retention while reducing the unpredictability of real-life interactions [4]. At the same time, Artificial Intelligence (AI) has emerged as a powerful tool for personalizing and optimizing interventions. Through machine learning, natural language processing, and real-time feedback systems, AI can monitor behavior, assess progress, and adapt content to individual needs [5]. When combined with SGs, AI enhances their adaptability, responsiveness, and effectiveness, providing tailored support that evolves with the user [6]. Together, SGs and AI offer a promising framework for delivering targeted, engaging, and personalized socio-emotional interventions for children with

ASD. This narrative review aims to explore key trends and thematic insights into the integration of Serious Games and Artificial Intelligence in the promotion of socio-emotional development in children with ASD. It examines key applications, benefits, and evaluation strategies, and addresses the challenges, ethical considerations, and future directions in this rapidly evolving field [7]. Building on the author's previous research on AI-driven autism interventions and VR-based serious games for adolescents [6, 7], the present work narrows its focus to the use of AI-powered SGs to enhance socio-emotional competencies in children with ASD. The aim is to synthesize current evidence and identify future directions for research, development, and implementation of these promising tools. This review is guided by the following questions: (1) How are Serious Games enhanced by AI used to foster socio-emotional development in children with ASD? (2) What evidence exists regarding their effectiveness, design strategies, and implementation challenges? These questions inform the structure and scope of the present synthesis.

II. Background: Serious Games and Autism

An expanding body of research highlights the diverse applications of SGs in supporting children with ASD. These tools have been used to target a wide range of skills, including social communication, turn-taking, empathy, and collaboration. Rather than aiming for exhaustive coverage, this section provides a thematic overview of key developments in the use of Serious Games for children with ASD, as reported in selected studies and literature reviews. Recent developments in reinforcement learning systems for healthcare support have shown how AI can provide adaptive decision-making in real time, adjusting treatment pathways based on patient interaction patterns [10]. This work illustrates how reinforcement learning frameworks, although not ASD-specific, can be repurposed for responsive intervention delivery in SG contexts. Such mechanisms could be leveraged in SGs to optimize emotional feedback loops and behavioral reinforcement in children with ASD.

Many SGs leverage immersive technologies such as virtual reality to simulate realistic environments where children can safely practice social interactions. Other games are designed to support emotion recognition and regulation by helping users identify facial expressions, vocal cues, and contextual emotional indicators [11]. SGs have also been applied to areas like attention control, executive functioning, and language development, including vocabulary building and vocalization. In addition to core developmental skills, SGs have been shown to support learning in academic and functional life domains. Some games teach numeracy, literacy, and problem-solving skills, while others focus on everyday tasks such as navigating public transportation, understanding health and safety practices, or applying basic first aid. These experiences not only promote cognitive development but also enhance independence and self-confidence in real-world situations [12].

SGs offer multiple advantages for children with ASD. Their structured, predictable, and customizable nature aligns well with the preferences of many autistic learners, helping to reduce anxiety and sensory overload. They provide opportunities to engage in repeated practice at an individualized pace, with feedback tailored to specific learning profiles. The interactive nature of SGs increases motivation and engagement, which are crucial for the success of therapeutic and educational programs. Moreover, the use of multisensory feedback and realistic simulations supports memory, self-regulation, and emotional awareness, encouraging the transfer of learned skills to everyday settings [13]. While SGs provide a structured and engaging medium for practicing skills, AI introduces the capability to personalize and dynamically adapt these interventions. The next section explores how AI technologies contribute to the broader landscape of autism interventions, laying the groundwork for their integration within game-based contexts.

III. Background: Artificial Intelligence in Interventions for Autism

AI has found a wide range of applications in tools and interventions developed to support children with ASD. One of its most impactful uses is in the personalization of learning experiences through machine learning algorithms and adaptive platforms that provide real-time feedback. These systems can monitor user behavior, track progress, and dynamically adjust content, making learning more responsive and individualized [14].

AI is also used to support the development of social skills. It powers interactive systems such as virtual agents, conversational chatbots, and social robots that simulate real-life interactions. These tools support turn-taking, nonverbal cue recognition, and social problem-solving in controlled environments [15].

In communication, AI enhances Augmentative and Alternative Communication (AAC) tools by incorporating features like predictive text, intelligent voice recognition, and adaptive vocabulary suggestions. These capabilities help users with limited verbal communication express themselves more efficiently, and systems improve over time by learning from usage patterns [16].

Emotional understanding is another key domain where AI contributes significantly. Intelligent systems can detect and interpret facial expressions, vocal tone, and body language using computer vision and affective computing. Children receive immediate feedback from avatars or virtual tutors, helping them improve emotion recognition and regulation [17].

AI also supports the analysis of social behavior through natural language processing and multimodal data interpretation. These systems assess user engagement, attention, and response patterns, providing valuable insights for clinicians and educators. In some cases, AI can offer autonomous support during learning sessions without direct human supervision.

Another important application is in early screening and diagnosis. By analyzing data such as gaze patterns, movement, vocalizations, and neuroimaging, AI systems can help detect

early signs of ASD, improving the timeliness and accuracy of assessments [18].

Overall, AI enables highly adaptive and scalable interventions that align with the diverse learning needs of children with ASD. When integrated into virtual or gamified environments, AI increases engagement, promotes independent learning, and supports the acquisition of social, emotional, and cognitive skills. Its ability to process large datasets and refine its responses over time makes it a valuable tool in both educational and therapeutic contexts [19]. This section highlights recurring applications of AI identified across various studies and conceptual papers, illustrating the breadth of its contribution to autism interventions.

IV. Method

A computerized search was carried out using the Scopus database to identify studies on the integration of SGs and AI aimed at supporting socio-emotional development in children with ASD. The search was limited to publications in English, published between 2014 and 2024, and focused on intervention-based studies involving SGs designed to improve emotional and social functioning in children with ASD. The following search string was used in the TITLE-ABS-KEY field: "serious games" AND "autism" AND "emotions". This query returned thirty-one records. After an initial screening of titles and abstracts, articles that did not meet the preliminary inclusion criteria were excluded. These included studies focusing solely on physical rehabilitation, motor coordination, sensory processing, or diagnostic methods without the use of game-based or AI-enhanced interventions. From this initial set, six studies were identified as directly meeting the eligibility criteria for detailed narrative synthesis. These included five empirical interventions and one systematic review, all focusing on the explicit integration of Serious Games with Artificial Intelligence to promote socio-emotional development in children with ASD.

The six included studies were:

1. Zirkus Empathico 2.0 [20]: RCT – multiplayer SG with adaptive feedback for emotion recognition and empathy;
2. EMOCASH [21]: pilot study – intelligent agent-based multiplayer game using the ASPECTS™ model;
3. JeStiMulE [22]: pre-post study – multimodal facial expression training game with adaptive feedback;
4. Game-Based Social Interaction Platform [23]: pilot study – integrated eye-tracking with emotion recognition tasks;
5. Interactive game using physiological sensors [24]: pilot study – real-time biofeedback for emotional state classification;
6. Emotion Detectives [25]: quasi-experimental ABA design – SG with adaptive learning for emotion discrimination.

Eligible articles addressed key domains such as emotion recognition, emotional regulation, empathy, joint attention, and social interaction, and employed AI features such as real-time feedback, gaze tracking, physiological sensing, and adaptive personalization algorithms. For example, the Game-Based Social Interaction Platform [23] utilized real-time eye-tracking combined with facial emotion recognition tasks to assess user engagement and emotional responses.

V. Narrative review

This narrative review aims to synthesize and critically compare recent literature on the use of AI-powered SGs to foster socio-emotional development in children with ASD. The originality of this contribution lies in: (a) the focus on SGs specifically enhanced by AI components such as adaptive feedback, emotion recognition, and personalization systems; and (b) the thematic analysis of socio-emotional outcomes such as empathy, joint attention, emotion regulation, and prosocial behavior.

A total of twenty-two studies were identified through the initial screening. Among these, six empirical interventions were retained for in-depth synthesis, based on their integration of AI-driven components and their targeted impact on socio-emotional development in ASD. The remaining studies were used to support background discussion on broader trends, implementation challenges, and design principles.

Zirkus Empathico 2.0 [20], a bilingual mobile serious game tested in Germany and Pakistan, significantly improved emotional awareness and empathy in children with ASD after an 8-week randomized controlled trial. Notably, participants were able to apply learned emotional skills in real-world contexts, showing potential for generalization beyond the digital environment.

Similarly, EMOCASH [21], a virtual agent-based multiplayer game, was designed to teach both financial literacy and emotion recognition within a 3D virtual shop. The game, tailored to Egyptian children with ASD using the ASPECTS™ design index, demonstrated high usability and educational impact by facilitating real-life skill transfer in a socially simulated environment. Another line of research explored emotion recognition via multimodal feedback systems.

The game JeStiMulE [22], focused on improving facial expression recognition, showed significant pre-post improvements in accuracy on standardized emotion tasks in a sample of Moroccan children with ASD. These outcomes highlight the importance of integrating multimodal feedback, repetitive training, and adaptive interfaces based on functioning level.

A fourth study [23] the Game-Based Social Interaction Platform, integrated real-time eye-tracking with facial emotion recognition tasks. Reduced fixation on positive expressions was interpreted as a digital biomarker of engagement.

An additional study [24] used physiological sensors within an interactive game to assess and classify emotional states, laying the groundwork for future emotionally responsive game-based interventions.

Finally, the Emotion Detectives [25] game demonstrated improvements in emotion discrimination and self-regulation in children with neurodevelopmental conditions, including ASD, with gains maintained at a one-month follow-up.

From a design perspective, the reviewed literature emphasizes the relevance of co-design practices involving both

autistic users and key stakeholders such as parents and educators. Games developed with Tangible User Interfaces (TUIs) [26] proved particularly effective in maintaining attention and facilitating emotional understanding. Despite encouraging outcomes, the field still suffers from limited clinical validation, small sample sizes, and short follow-up periods. Furthermore, most reviewed games were developed for high-functioning children with ASD, revealing a need for more inclusive designs. Overall, the integration of AI in SGs provides a scalable and adaptable framework for delivering engaging, personalized, and evidence-informed socio-emotional interventions in autistic populations.

VI. Discussion

This discussion integrates both the main empirical findings and broader implications of the reviewed studies. It begins by synthesizing key socio-emotional outcomes across interventions, then explores design considerations, methodological limitations, and opportunities for future research in the field of AI-powered Serious Games for children with ASD.

The reviewed studies collectively illuminate promising directions in the development and application of SGs and immersive technologies to support socio-emotional learning in children with ASD. While the integration of such tools has not yet reached full methodological maturity, emerging patterns suggest tangible benefits for emotion recognition, behavioral regulation, and engagement in children with ASD.

Recent evidence underscores the pivotal role of immersive environments in modulating emotional activation and improving performance in emotion recognition tasks. The pilot study utilizing Unreal Engine 4 [27], although conducted on neurotypical adults, demonstrated heightened emotional engagement in 3D environments compared to traditional settings. These findings are especially relevant for ASD interventions, where attention and motivation are often reduced. It is plausible that immersive graphics and interactive feedback may scaffold attentional focus and facilitate deeper emotional processing in children with ASD - a hypothesis that warrants further empirical validation in clinical populations.

Games leveraging multisensory tools—such as real-time eye-tracking, physiological sensors, and spatialized audio—demonstrated potential for broader accessibility and improved user engagement.

The adaptive capabilities of SGs, powered by AI, are increasingly recognized as essential to their efficacy. Games such as JeStiMulE [22] and Emotion Detectives [25] integrated feedback mechanisms that responded dynamically to user behavior, allowing for personalized pacing and reinforcement. These features align well with the cognitive and emotional heterogeneity characteristic of ASD. Intervention outcomes from these studies demonstrated not only significant improvements in targeted emotion discrimination tasks but also observable behavioral gains in naturalistic settings, suggesting generalization beyond the digital context.

For instance, in the Emotion Detectives study [25], the system adjusted the difficulty and type of emotion recognition tasks in real time based on the child's performance, offering immediate visual and auditory reinforcement when correct responses were detected. This dynamic feedback loop helped maintain engagement and reinforce emotional learning in a personalized manner.

Personalization remains a critical factor in intervention success. The variability in cognitive profiles among children with ASD - particularly between high- and low-functioning individuals - necessitates differentiated user interfaces and multimodal content. For example, the JeStiMulE [22] study revealed that children with high-functioning autism significantly outperformed their lower-functioning peers in emotion recognition tasks, highlighting the need for adaptable systems. Multisensory feedback tools, including eye-tracking (as seen in the Game-Based Social Interaction Platform [23]), spatialized audio, and physiological sensors, may enhance accessibility and foster engagement across a broader segment of the autism spectrum.

Beyond empirical outcomes, several design and usability insights were noted.

A recurring theme is the divergence between user motivation and therapeutic intention. Studies grounded in user-centered frameworks, such as the one referencing Whyte et al.'s model [28], showed that autistic youth prioritize engaging, visually rich gameplay, while professionals emphasize generalizable skill acquisition. Bridging this divide requires participatory design approaches that incorporate the lived experiences and preferences of children with ASD, ensuring both usability and therapeutic relevance.

Despite encouraging findings, methodological limitations remain. Sample sizes across studies were often small, and long-term assessments were rarely conducted. Additionally, there is a notable gap regarding the use of AI to dynamically adapt emotional feedback and narrative progression within SGs.

In practical terms, an emotionally attuned system would use multimodal inputs - such as facial expression analysis, vocal tone monitoring, and physiological sensors - to detect a child's emotional state and adapt gameplay accordingly. For example, if signs of frustration or disengagement are detected, the system could simplify tasks, slow down interactions, or introduce calming stimuli to re-engage the user.

Nevertheless, the convergence of AI, adaptive learning environments, and game-based delivery represents a compelling frontier for inclusive and scalable ASD interventions.

VII. Future Directions and Research Gaps

Despite the growing interest and positive initial findings regarding SGs and AI for children with ASD, emerging themes and conceptual gaps identified across the literature suggest several promising areas for future exploration.

One promising future direction involves the development of more advanced AI algorithms capable of deeper personalization and nuanced real-time adaptation to a child's learning style, emotional state, and progress. Such systems could provide finely tuned interventions that evolve continuously based

on user interaction [29]. Another emerging area is the creation of hybrid AI-human learning models, where AI supports but does not replace human instruction or therapy. Hybrid AI-human models have been tested in platforms like Woebot [30] or Replika [31], where automated responses are supported by clinician supervision or feedback. These models may combine the scalability of digital tools with the irreplaceable relational and contextual insight of human facilitators. Related to this is the need to leverage multimodal data inputs - such as text, audio, facial expressions, gesture, and biosignals - to design more natural and emotionally attuned learning experiences [32].

Designing SGs that address sensory sensitivities and support imaginative play is another crucial area for development. These capabilities could significantly expand the emotional and behavioral range of digital interventions, particularly for children who have trouble with unstructured or abstract tasks. Furthermore, the field would benefit from interdisciplinary design frameworks that bring together educators, clinicians, game developers, families, and neurodiverse individuals to co-create content. This participatory approach would ensure that SGs reflect real-world needs and diverse lived experiences [33].

Advanced time-series prediction models, such as GLinear [34], offer new possibilities for decoding physiological and behavioral signals in real time. GLinear and similar architectures have been proposed for modeling arousal and engagement in real time using physiological data streams [34]. These architectures could enhance SG responsiveness by enabling more accurate modeling of attention, arousal, and engagement patterns.

Generative AI, which allows for the creation of dynamic narratives, characters, and interactive environments based on user input, represents another exciting frontier. Generative AI systems, such as GPT-based narrative engines, can dynamically adjust storylines and dialogue based on user preferences or detected emotions. By generating personalized stories or scenarios, such systems may enhance engagement, emotional learning, and long-term retention. On the research side, longitudinal studies are essential to evaluate the sustained impact of AI-integrated SGs on socio-emotional development, academic outcomes, and real-life functioning. Additional studies should examine the role of social interactions within multiplayer SGs, particularly how these experiences transfer to offline settings [35].

There are several critical gaps that must be addressed to strengthen the field. These include the limited number of long-term and ecologically valid studies, and the lack of research examining how cultural and socioeconomic factors affect the success and accessibility of SG and AI interventions. Furthermore, there is a pressing need for more inclusive sampling to represent the full diversity of the autism spectrum, as well as greater focus on underexplored domains such as emotional regulation in high-stress or unpredictable contexts. Another persistent issue is the generalization gap - that is, the challenge of ensuring that skills practiced within SGs translate meaningfully to everyday environments. Lastly, there remains an absence of standardized frameworks for evaluating and comparing different AI-enhanced SG interventions, which hinders both replication and broader implementation [36].

Emerging metric-driven approaches designed for the recognition of hazardous or high-stakes situations offer promising frameworks for modeling stress responses and predicting behavioral escalations [37]. These could inform the design of emotionally aware SGs capable of anticipating distress and dynamically modulating difficulty or content.

Future research must address these shortcomings through rigorous, long-term, and multisite studies. There is also a need to explore emerging AI applications and to develop standardized, reliable outcome measures. Doing so will help establish stronger evidence base and improve the design and delivery of effective, inclusive, and sustainable digital interventions for children with ASD.

VIII. Conclusion

The integration of SGs and AI represents a rapidly growing and highly promising frontier in the promotion of socio-emotional skills among children with ASD. This combined approach leverages the immersive, interactive nature of game-based learning and the adaptive, data-driven capabilities of AI to deliver interventions that are not only engaging but also finely tailored to the unique and heterogeneous profiles of autistic learners [38].

SGs provide structured, low-risk environments where children can repeatedly practice and reinforce critical skills such as emotion recognition, turn-taking, and social communication. These environments benefit learners who experience anxiety or sensory overload in real-life contexts. AI further enhances these games by enabling dynamic personalization - adjusting content and difficulty in real time based on the child's behavioral patterns, emotional cues, and performance data. When combined, SGs and AI support a wide array of socio-emotional competencies, including self-regulation, empathy, joint attention, and behavioral flexibility. As such, this integrated approach serves as a powerful complement to traditional therapeutic and educational interventions [39].

Research to date has yielded encouraging results, with numerous studies documenting measurable improvements in specific target areas following the use of AI-enhanced SGs. However, the field is still evolving, and there remains considerable variation in study methodologies, sample sizes, and assessment tools. These inconsistencies limit the ability to draw firm conclusions about generalizability and long-term effectiveness. A stronger evidence base is needed to guide the design, implementation, and evaluation of these tools across diverse populations and real-world settings. In particular, future studies should address the generalization gap - ensuring that skills learned in digital contexts transfer meaningfully to everyday social environments. Future research must focus on longitudinal studies involving larger and

more diverse participant groups to better capture the full spectrum of ASD and the contextual factors that influence intervention outcomes [40].

At the same time, the field must prioritize transparency and fairness in algorithm design to avoid reinforcing existing disparities or excluding vulnerable users. By balancing innovation with ethical responsibility, AI-powered Serious Games can evolve into truly inclusive and impactful tools for supporting the emotional well-being and social inclusion of children with ASD.

Despite its potential, this approach also presents a number of critical challenges. Key concerns include the need for inclusive, sensory-accessible design; transparent and ethical use of personal data; algorithmic fairness; and equitable access across cultural and socio-economic backgrounds. Additionally, developers and practitioners must guard against the risk of over-reliance on technology, ensuring that these tools supplement, rather than replace, essential human interactions and relationships.

To ensure sustainable progress, interdisciplinary collaboration will be essential - bringing together children with ASD, caregivers, educators, clinicians, AI developers, and game designers throughout the research and development process [41].

Looking forward, the field should adopt a participatory and interdisciplinary framework that brings together children with ASD, families, educators, clinicians, designers, and researchers. Such collaboration is vital to creating tools that are relevant, user-centered, and grounded in real-life experiences. Advances in generative AI and hybrid human-AI learning models offer exciting possibilities for deepening engagement, promoting emotional insight, and crafting personalized narratives that resonate with each child's developmental needs. Furthermore, the integration of multimodal data - such as facial expressions, speech, biosignals, and gaze - can pave the way for more emotionally responsive and adaptive learning environments [42]. In parallel, future systems must adopt transparent, explainable AI protocols and ensure that personalization algorithms do not inadvertently reinforce cognitive or socio-economic disparities in access or engagement. To fully realize their potential, AI-powered SGs must be embedded in broader clinical frameworks and supported by policies that ensure ethical deployment, accessibility, and cross-sector integration in health and education systems.

This work is positioned as a conceptual and narrative synthesis aimed at fostering discussion on the integration of AI and Serious Games for autism intervention. As such, it contributes to ongoing dialogue in the interdisciplinary community and aligns with the goals of conferences like FedCSIS.

REFERENCES

[1] American Psychiatric Association, *Diagnostic and Statistical Manual of Mental Disorders*, Fifth Edition. American Psychiatric Association, 2013. doi: 10.1176/appi.books.9780890425596.

[2] A. Bandura, «Social Cognitive Theory: An Agentic Perspective», *Annu. Rev. Psychol.*, vol. 52, fasc. 1, pp. 1–26, feb. 2001, doi: 10.1146/annurev.psych.52.1.1.

[3] F. Xu *et al.*, «The Use of Digital Interventions for Children and Adolescents with Autism Spectrum Disorder—A Meta-Analysis», *J Autism Dev Disord*, set. 2024, doi: 10.1007/s10803-024-06563-4.

[4] C. Eichenberg e M. Schott, «Serious Games for Psychotherapy: A Systematic Review», *Games for Health Journal*, vol. 6, fasc. 3, pp. 127–135, giu. 2017, doi: 10.1089/g4h.2016.0068.

[5] F. Petcusin, C. S. Spahiu, e L. Stanescu, «A machine learning approach for automatic testing», in *Annals of Computer Science and Information Systems*, PTI, ott. 2023, pp. 215–220. doi: 10.15439/2023f4426.

[6] S. D'Alfonso, «AI in mental health», *Current Opinion in Psychology*, vol. 36, pp. 112–117, dic. 2020, doi: 10.1016/j.copsyc.2020.04.005.

[7] S. Kewalramani, K.-A. Allen, E. Leif, e A. Ng, «A Scoping Review of the Use of Robotics Technologies for Supporting Social-Emotional Learning in Children with Autism», *J Autism Dev Disord*, vol. 54, fasc. 12, pp. 4481–4495, dic. 2024, doi: 10.1007/s10803-023-06193-2.

[8] F. Stasolla, E. Curcio, A. Passaro, M. Di Gioia, A. Zullo, e E. Martini, «Exploring the Combination of Serious Games, Social Interactions, and Virtual Reality in Adolescents with ASD: A Scoping Review», *Technologies*, vol. 13, fasc. 2, p. 76, feb. 2025, doi: 10.3390/technologies13020076.

[9] F. Stasolla, E. Curcio, A. Zullo, A. Passaro, e M. D. Gioia, «Integrating Artificial Intelligence-based programs into Autism Therapy: Innovations for Personalized Rehabilitation», presentato al 19th Conference on Computer Science and Intelligence Systems (FedCSIS), nov. 2024, pp. 169–176. doi: 10.15439/2024F6229.

[10] A. Coronato e M. Naeem, «A Reinforcement Learning Based Intelligent System for the Healthcare Treatment Assistance of Patients with Disabilities», in *Pervasive Systems, Algorithms and Networks*, vol. 1080, C. Esposito, J. Hong, e K.-K. R. Choo, A c. di, in Communications in Computer and Information Science, vol. 1080. , Cham: Springer International Publishing, 2019, pp. 15–28. doi: 10.1007/978-3-030-30143-9_2.

[11] H. M. Zakari, M. Ma, e D. Simmons, «A Review of Serious Games for Children with Autism Spectrum Disorders (ASD)», in *Serious Games Development and Applications*, vol. 8778, M. Ma, M. F. Oliveira, e J. Baalsrud Hauge, A c. di, in Lecture Notes in Computer Science, vol. 8778. , Cham: Springer International Publishing, 2014, pp. 93–106. doi: 10.1007/978-3-319-11623-5_9.

[12] K. Martinez, M. I. Menéndez-Menéndez, e A. Bustillo, «Awareness, Prevention, Detection, and Therapy Applications for Depression and Anxiety in Serious Games for Children and Adolescents: Systematic Review», *JMIR Serious Games*, vol. 9, fasc. 4, p. e30482, dic. 2021, doi: 10.2196/30482.

[13] J. Wolstencroft, L. Robinson, R. Srinivasan, E. Kerry, W. Mandy, e D. Skuse, «A Systematic Review of Group Social Skills Interventions, and Meta-analysis of Outcomes, for Children with High Functioning ASD», *J Autism Dev Disord*, vol. 48, fasc. 7, pp. 2293–2307, lug. 2018, doi: 10.1007/s10803-018-3485-1.

[14] E. Ferrari, «Artificial Intelligence for Autism Spectrum Disorders», in *Artificial Intelligence in Medicine*, N. Lidströmer e H. Ashrafian, A c. di, Cham: Springer International Publishing, 2021, pp. 1–15. doi: 10.1007/978-3-030-58080-3_249-1.

[15] S. S. Sethi e K. Jain, «AI technologies for social emotional learning: recent research and future directions», *JRIT*, vol. 17, fasc. 2, pp. 213–225, ago. 2024, doi: 10.1108/JRIT-03-2024-0073.

[16] M. Wang, B. Muthu, e C. B. Sivaparthipan, «Smart assistance to dyslexia students using artificial intelligence based augmentative alternative communication», *Int J Speech Technol*, vol. 25, fasc. 2, pp. 343–353, giu. 2022, doi: 10.1007/s10772-021-09921-0.

[17] S. Poria, N. Majumder, R. Mihalcea, e E. Hovy, «Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances», *IEEE Access*, vol. 7, pp. 100943–100953, 2019, doi: 10.1109/ACCESS.2019.2929050.

[18] M. Mengi e D. Malhotra, «Artificial Intelligence Based Techniques for the Detection of Socio-Behavioral Disorders: A Systematic Review», *Arch Computat Methods Eng*, vol. 29, fasc. 5, pp. 2811–2855, ago. 2022, doi: 10.1007/s11831-021-09682-8.

[19] N. Wankhede *et al.*, «Leveraging AI for the diagnosis and treatment of autism spectrum disorder: Current trends and future prospects», *Asian Journal of Psychiatry*, vol. 101, p. 104241, nov. 2024, doi: 10.1016/j.ajp.2024.104241.

[20] A. Hassan, N. Pinkwart, e M. Shafi, «Zirkus Empathico 2.0: a multiplayer serious mobile game for children with autism spectrum

disorder (ASD), with a focus on enhancing social and emotional development», *Multimed Tools Appl*, apr. 2025, doi: 10.1007/s11042-025-20826-x.

[21] H. K. H. A. El-Sattar, «EMOCASH: An Intelligent Virtual-Agent Based Multiplayer Online Serious Game for Promoting Money and Emotion Recognition Skills in Egyptian Children with Autism», *IJACSA*, vol. 14, fasc. 4, 2023, doi: 10.14569/IJACSA.2023.0140414.

[22] M. Elhaddadi *et al.*, «SERIOUS GAMES TO TEACH EMOTION RECOGNITION TO CHILDREN WITH AUTISM SPECTRUM DISORDERS (ASD)», *Acta Neuropsychologica*, vol. 19, fasc. 1, pp. 81–92, gen. 2021, doi: 10.5604/01.3001.0014.7569.

[23] Y.-L. Chien *et al.*, «Game-Based Social Interaction Platform for Cognitive Assessment of Autism Using Eye Tracking», *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 749–758, 2023, doi: 10.1109/TNSRE.2022.3232369.

[24] S. Baldassarri, L. Passerino, S. Ramis, I. Riquelme, e F. J. Perales, «Toward emotional interactive videogames for children with autism spectrum disorder», *Univ Access Inf Soc*, vol. 20, fasc. 2, pp. 239–254, giu. 2021, doi: 10.1007/s10209-020-00725-8.

[25] J. Löytömäki, P. Ohtonen, e K. Huttunen, «Serious game the Emotion Detectives helps to improve social–emotional skills of children with neurodevelopmental disorders», *Brit J Educational Tech*, vol. 55, fasc. 3, pp. 1126–1144, mag. 2024, doi: 10.1111/bjet.13420.

[26] J. M. Garcia-Garcia, V. M. R. Penichet, M. D. Lozano, e A. Fernando, «Using emotion recognition technologies to teach children with autism spectrum disorder how to identify and express emotions», *Univ Access Inf Soc*, vol. 21, fasc. 4, pp. 809–825, nov. 2022, doi: 10.1007/s10209-021-00818-y.

[27] G. Quirantes-Gutierrez, Á. F. Estévez, G. Artés Ordoño, e G. López-Crespo, «Design of an Emotional Facial Recognition Task in a 3D Environment», *Computers*, vol. 14, fasc. 4, p. 153, apr. 2025, doi: 10.3390/computers14040153.

[28] J. S. Y. Tang, M. Falkmer, N. T. M. Chen, S. Bölte, e S. Girdler, «Designing a Serious Game for Youth with ASD: Perspectives from End-Users and Professionals», *J Autism Dev Disord*, vol. 49, fasc. 3, pp. 978–995, mar. 2019, doi: 10.1007/s10803-018-3801-9.

[29] S. S. Joudar *et al.*, «Artificial intelligence-based approaches for improving the diagnosis, triage, and prioritization of autism spectrum disorder: a systematic review of current trends and open issues», *Artif Intell Rev*, vol. 56, fasc. S1, pp. 53–117, ott. 2023, doi: 10.1007/s10462-023-10536-x.

[30] K. K. Fitzpatrick, A. Darcy, e M. Vierhile, «Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial», *JMIR Ment Health*, vol. 4, fasc. 2, p. e19, giu. 2017, doi: 10.2196/mental.7785.

[31] M. Saha, S. Lindsay, D. Varghese, T. Bartindale, e P. Olivier, «Benefits of Community Voice: A Framework for Understanding Inclusion of Community Voice in HCI4D», *Proc. ACM Hum.-Comput. Interact.*, vol. 7, fasc. CSCW2, pp. 1–26, set. 2023, doi: 10.1145/3610174.

[32] T. Liu, J. Huang, T. Liao, R. Pu, S. Liu, e Y. Peng, «A Hybrid Deep Learning Model for Predicting Molecular Subtypes of Human Breast Cancer Using Multimodal Data», *IRBM*, vol. 43, fasc. 1, pp. 62–74, feb. 2022, doi: 10.1016/j.irbm.2020.12.002.

[33] P. Zemliansky e D. Wilcox, A c. di, *Design and Implementation of Educational Games: Theoretical and Practical Perspectives*. IGI Global, 2010. doi: 10.4018/978-1-61520-781-7.

[34] S. T. H. Rizvi, N. Kanwal, M. Naeem, A. Cuzzocrea, e A. Coronato, «Bridging Simplicity and Sophistication using GLinear: A Novel Architecture for Enhanced Time Series Prediction», 2025, *arXiv*. doi: 10.48550/ARXIV.2501.01087.

[35] J. Pérez, M. Castro, e G. López, «Serious Games and AI: Challenges and Opportunities for Computational Social Science», *IEEE Access*, vol. 11, pp. 62051–62061, 2023, doi: 10.1109/ACCESS.2023.3286695.

[36] C. Kasari, S. Shire, W. Shih, e D. Almirall, «Getting SMART About Social Skills Interventions for Students With ASD in Inclusive Classrooms», *Exceptional Children*, vol. 88, fasc. 1, pp. 26–44, ott. 2021, doi: 10.1177/00144029211007148.

[37] M. Fiorino, M. Naeem, M. Ciampi, e A. Coronato, «Defining a Metric-Driven Approach for Learning Hazardous Situations», *Technologies*, vol. 12, fasc. 7, p. 103, lug. 2024, doi: 10.3390/technologies12070103.

[38] F. Abomelha e P. Newbury, «A VARK learning style-based Recommendation system for Adaptive E-learning», in *Annals of Computer Science and Information Systems*, PTI, nov. 2024, pp. 1–8. doi: 10.15439/2024f5253.

[39] W. Westera *et al.*, «Artificial intelligence moving serious gaming: Presenting reusable game AI components», *Educ Inf Technol*, vol. 25, fasc. 1, pp. 351–380, gen. 2020, doi: 10.1007/s10639-019-09968-2.

[40] S.-J. Eun, E. J. Kim, e J. Kim, «Artificial intelligence-based personalized serious game for enhancing the physical and cognitive abilities of the elderly», *Future Generation Computer Systems*, vol. 141, pp. 713–722, apr. 2023, doi: 10.1016/j.future.2022.12.017.

[41] E. Smith, Y. Wang, e E. Matson, «Psychological Needs as Credible Song Signals: Testing Large Language Models to Annotate Lyrics», in *Annals of Computer Science and Information Systems*, PTI, nov. 2024, pp. 159–168. doi: 10.15439/2024f7168.

[42] S. Rossi, M. Rossi, R. R. Mukkamala, J. B. Thatcher, e Y. K. Dwivedi, «Augmenting research methods with foundation models and generative AI», *International Journal of Information Management*, vol. 77, p. 102749, ago. 2024, doi: 10.1016/j.ijinfomgt.2023.102749.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 23–30
DOI: 10.15439/2025F4971
ISSN 2300-5963 ACSIS, Vol. 44

Practical security of evidence for regulated artificial intelligence modules

Marko Esche, Levin Ho, Martin Nischwitz
0009-0001-7110-5665
0000-0002-7560-5456
0009-0002-5488-5820
Physikalisch-Technische Bundesanstalt,
Abbestraße 2-12, 10587 Berlin, Germany
Email: marko.esche@ptb.de, levin.ho@ptb.de, martin.nischwitz@ptb.de

Sabine Glesner
0009-0003-6946-3257
Technische Universität Berlin
Straße des 17. Juni 135
Email: sabine.glesner@tu-berlin.de

***Abstract*—Artificial intelligence has recently led to numerous new applications in various industry sectors. Whenever artificial intelligence modules are used in a black-box setting, quality monitoring of such modules remains an open challenge. This implies that users of such modules cannot predict the modules' performance following software updates or retraining. Specifically for regulated devices, keeping track of an artificial intelligence module's behavior and compliance with requirements is crucial. To this end, existing methods for monitoring the functional behavior of software are investigated and evaluated regarding their practical usability in this paper. Based on the results of the investigation, a proposal for a new adaptive quality monitoring scheme for artificial intelligence modules is made.**

I. Introduction

ARTIFICIAL intelligence (AI) modules are becoming increasingly common in private and public sectors. [1] Such applications range from classical machine learning algorithms, e.g. object detection and classification [2], to generative AI systems such as chatGPT [3] which can be seen as a first step towards a general-purpose AI. In the European Union (EU) the AI Act [4] provides terms, definitions and requirements for AI models and systems. It also introduces general transparency requirements to be met by any AI system and establishes a conformity assessment framework for so-called "high-risk scenarios" and general-purpose AI systems, aiming to ensure that all high-risk AI systems and general-purpose AI systems used within the EU ensure a minimum level of customer protection. While these high-risk applications are limited to law enforcement, health etc., the AI Act also stresses that AI modules within regulated products still need to pass conformity assessment according to the relevant directives. [4] One such regulated sector is legal metrology, covering measuring instruments used for commercial transactions or official measurements. There exist already multiple signs indicating that the integration of AI modules into regulated measuring instruments is imminent. In the EU, the Measuring Instruments Directive (MID) Annex I lays down essential requirements for regulated measuring instruments which apply to ten different types of instruments, e.g., length measuring devices, taximeters, across all EU member states. As an example, requirement 8.3 imposes the following, "[...] Software identification shall be easily provided by the measuring instrument. Evidence of an intervention shall be available for a reasonable period of time." WELMEC Guide 7.2 [5] provides harmonized technical guidance regarding the interpretation of the software-related essential requirements for all EU members. From essential requirement 8.3, two deductions can be drawn: Firstly, since an AI module must be interpreted as software, it shall be possible to identify a specific version of the AI module for control purposes. Secondly, any change to the AI module (including its parameters) shall be traceable to provide evidence of an intervention. This requirement aims to ensure continued compliance of the instrument by providing traceability of modifications. Typically, users of AI modules do not have access to the actual executable code of the module but either use it as a remote service or as part of a device with limited interaction capabilities. Given the potential adaptability of AI modules, existing static solutions for providing traceability of modifications, e.g., hashes over executable files [5], will likely reach their limits quickly, if the frequency of modifications increases. In particular, any approach should not be dependent on manual interference or classification of changes. Therefore, a solution should be able to automatically differentiate between simple bugfixes that do not affect a module's intended behavior and more fundamental modifications such as the addition of new functionality. To this end, classical and alternative approaches for identification and traceability of software modules are investigated in this paper, resulting in a new proposal which aims to be generally applicable for all types of AI modules subject to similar requirements. The resulting use case can be summarized as follows: An AI module is used for data processing purposes, e.g., within a cyber-physical system for classification of input data. Any change to the module shall be traceable, either by providing evidence of an intervention or through demonstrating continued compliance with predefined requirements. The remainder of the paper is structured as follows: Section II provides an overview of different existing methods to identify and monitor modifications in software modules. In Section III one selected method will be extended to deal with the potential behavior of AI modules. The proposal will be practically tested and compared with the

Topical area: Advanced Artificial Intelligence in Applications

current state of the art in Section IV. Section V concludes the paper and provides suggestions for further work.

II. Related Work

Numerous methods exist for identifying software modules and providing traceability of changes in real-world scenarios. These range from simple version control systems to automata learning approaches. While version control can be seen as a static approach that does not allow automatic distinction between minor bugfixes and major changes, automata learning can be used to quantify the scope of modifications for certain types of automata. Subsection II-A will cover existing active automata learning methods that require bi-directional communication with a system under learning (SUL). Methods that operate passively and are also applicable in black-box scenarios will be addressed in Subsection II-B. Classical static approaches of identifying software through hashes over binaries will be revisited in Subsection II-C. In Subsection II-D the differences between the approaches are discussed and a candidate for the described use case is selected.

A. Active automata learning

In the 1987 paper "Learning Regular Sets from Queries and Counterexamples" [6], Angluin outlined the now well-known L^* algorithm. The algorithm uses a learner L which sends consecutive queries to a teacher T to construct and update an automata representation of the SUL. The teacher acts as an interface to the SUL and provides necessary abstraction for the learner. Thus, the approach is only applicable in a white-box or gray-box scenario. Since the learner paradigm plays a central role in the developed method in Section III, the main aspects of the L^* algorithm will be reiterated here. L^* was originally developed for learning the behavior of deterministic finite automata (DFAs), which are 5-tuples $(Q, \Sigma, \delta, q_0, F)$ [7]:

$$\begin{aligned} &Q \text{ is a finite non-empty set of states.} \\ &\Sigma \text{ is a finite input alphabet.} \\ &\delta : Q \times \Sigma \to Q \text{ is the transition function.} \\ &q_0 \in Q \text{ is the initial state of the DFA.} \\ &F \subset Q \text{ is the set of accepting states of the DFA.} \end{aligned} \tag{1}$$

During execution of L^*, the learner L sends membership and equivalence queries to the teacher T. If the accepted language of the SUL A is $L(A)$ and $Aut(A)$ is the set of all DFAs with the same input alphabet Σ, the two queries are defined in the following manner:

- Membership query $Q_M : \Sigma^* \to \{0, 1\}$ where the learner asks the teacher to test if a given string x is part of the language $L(A)$. If $x \in L(A)$, the teacher's response is 1, 0 otherwise.
- Equivalence query $Q_E : Aut(\Sigma) \to \Sigma^* \cup \{\text{true}\}$ where the learner asks the teacher to test equivalence between the SUL A and the current learned automaton representation $A' \in Aut(\Sigma)$.

With the aim of constructing an internal observation table for storing the results of the queries in systematic fashion, the learner issues membership queries until an initial model A' is obtained. The learner then performs an equivalence query for A'. The teacher subsequently either acknowledges correspondence between the learned and the true model or supplies a counterexample $c \in \Sigma^*$ fulfilling the condition

$$c \in L(A) \wedge c \notin L(A') \text{ or } c \notin L(A) \wedge c \in L(A').$$

Windmüller, Neubauer, Steffen, Howar and Bauer showed in [8] and [9] that the L^* algorithm can be adapted to large-scale software applications with varying degrees of complexity. However, they also noted that this requires a lot of adaptation by the developer within the teacher to properly abstract the behavior of the SUL to the needs of the learner. Furthermore, while AI modules can be interpreted as deterministic software modules, their output for arbitrary, unknown input generally cannot be predicted due to the complexity of implementations such as Artificial Neural Networks (ANN). [10]

B. Passive automata learning

If white-box access to the SUL is not possible, passive automata learning algorithms can be used as an alternative for learning the behavior of a SUL. These algorithms usually obtain a set of traces $S = \{S_+, S_-\}$, where S_+ are positive traces describing the correct behavior of the SUL and S_- are traces that contain known errors that contradict the behavior of the SUL [11]. A trace itself is a list of input symbols and subsequently reached states represented by the corresponding observed output symbols. The Regular Positive Negative Inference algorithm (RPNI) [11] can be used to learn a model of an SUL from such traces. While the approach can correctly learn the behavior of complex software systems given sufficient time [12], it lacks the possibility of mapping the potentially arbitrary response of an AI module for unknown input to a DFA. A common limitation for typical automata learning algorithms is the inefficiency of handling so-called deeply nested states, especially when the SUL is highly complex. On the other hand, although passive automata learning algorithms allow efficient learning of an SUL's behavior from a provided set of traces, testing the conformance of the learned model, e.g., checking the completeness and/or the consistency, could result in relatively long time because the tests are usually performed via repeatedly checking different input combinations [12]. While [12] puts the emphasis on inferring a complete automaton of a black-box system, this paper focuses on monitoring a task learned by an AI module without prior knowledge.

C. Integrity protection through hashes

As mentioned in Section I, [5] provides harmonized technical guidelines for all EU member states regarding the application of securing and protection requirements for software in regulated measuring instruments. However, the Guide currently relies on static methods such as cryptographic hashes for providing evidence of interventions for all types of software modules. If an AI module contains a learning facility for adaptation in the field, any change would result in a new

hash over binary code and necessitate a new conformity assessment. To remedy this problem, the International Organization for Legal Metrology (OIML) published a revised version of the OIML Document D31 [13] in 2023. This document is, in theory, applicable to all regulated measuring instruments world-wide and addresses provision of evidence of intervention in a meaningful manner for AI modules. D31 treats AI modules as software modules with a predefined structure that are controlled by a (potentially very large) set of parameters that can be modified by means of a learning facility. Clause 6.2.3.1 of D31:2023, for instance, provides an example of a large ANN that uses version control for identification of the network topology and a cryptographic hash over the network weights in predefined order for tracing changes to the ANN's behavior. To avoid having to re-certify the adapted ANN, D31 recommends providing fingerprints of the network weights and storing the actual configuration of the weights externally. While this method ensures that an AI module within a regulated measuring instrument can continue to be used even after a learning cycle, the size of the externally stored weight configuration will increase linearly over time. It also places the burden of monitoring compliance with legal requirements on inspectors and market surveillance authorities. To check a specific result of the AI module, the authority has to verify if the ANN together with the stored network weight configuration is suitable to produce measurement results within the legally required limits. While the method does not require access to the development environment of an AI module, it does require access to the weights of the ANN and thus only works in a white-box scenario.

D. Admissibility as evidence

It should be noted that the development processes for 'classical' software and AI modules are very similar. For classical software, the developer constructs an initial concept based on known requirements and available data. This concept is implemented and tested to ensure compliance with requirements [14]. Figure 1 provides a visual representation of this workflow for a measuring instrument. During use, changes to the software can be made via updates. In the use case investigated here, any change to the software shall produce evidence of intervention. Consequently, any software update must either result in a broken physical seal or a permanent logbook entry with the same legal consequences.

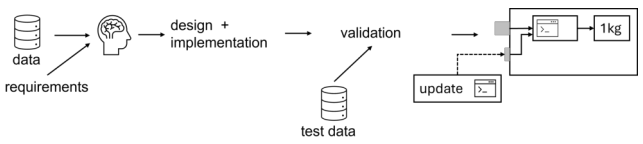


Fig. 1: Classical workflow for software development: Taking into account predefined requirements, the developer uses the available data to create an initial implementation, which is then validated using independent test data. During use, software modules can be modified by means of an update.

TABLE I: Overview of the different monitoring approaches for AI modules and their properties.

approach	black-box support	memory requirements	suitable for AI
active automata learning	no	size of learning table	no
passive automata learning	yes	size of learning table + size of saved traces	no
hash comparison	no	size of AI model per update	yes
remote quality control	yes	size of AI monitor model	yes

Development of an AI module follows a similar pattern [15]. Initially, the developer selects an AI model (such as a decision tree or a deep ANN) taking into account known requirements. The model is then trained to learn a certain behavior based on the available pre-processed training data. Prior to the release of the model, it is validated using a validation data set which is disjunct from the data used for training, see Figure 2. During use of the model, different scenarios for updating it are possible:

1) Updates can follow a pattern similar to 'classical' software products, where the entire trained model is replaced by a new one.
2) A modification of the AI module can be realized by providing it with new training data and initiating another training procedure during use.
3) A learning algorithm as part of the AI module could use observed real-world data together with an externally provided reference for improving its configuration. In this scenario, all individual serial devices in the field will demonstrate different behavior.

These three variants will be revisited in Section IV.

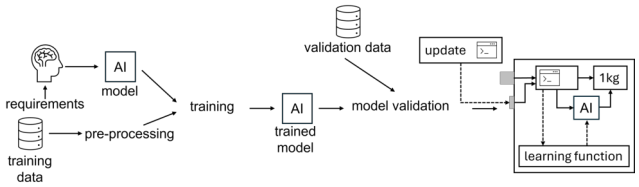


Fig. 2: Workflow for development of AI modules: Based on predefined requirements, the developer selects an initial AI model, which is then trained using pre-processed training data. The trained model is validated using independent test data. During use, the AI module can either be replaced during an update or re-trained using external or internal reference data.

Development of 'classical' software and AI modules both use a two-step approach that first produces an initial implementation which is then validated using an independent test dataset not included during development of the initial implementation. Also, changes to the final implementation can occur during use. While a software update can affect both types of modules, the source for modifications can also be an

internal learning procedure for an AI module. Regardless of the fact if such a learning procedure uses a supervised or an unsupervised training method, the main distinction compared to 'classical' software thus becomes the ability to dynamically change, potentially without an external trigger. It is this property that makes continuous monitoring of AI modules a necessity. Table I provides a summary of the aforementioned different approaches for providing security of evidence for interventions. As can be seen from rows 1, 2, and 3, only the hash comparison between different binary images of an AI module can actually be used for providing evidence of intervention while potentially needing linearly increasing chunks of memory per modification of the AI module. At the same time, the passive automata learning approach also supports black-box scenarios combined with a significantly smaller memory footprint, but is not originally able to monitor the underlying massively complex models behind an AI module. Thus, an extension of the passive automata learning approach to adaptive AI modules will be investigated in the subsequent section to derive an optimized solution with smaller memory usage and black-box applicability.

III. Remote quality control approach

As has been demonstrated in [12], passive automata learning algorithms can be used to learn the behavior of the software of complex cyber-physical systems in a quasi black-box scenario given sufficient learning time. To this end, the learning algorithm generates prefixes from the observed positive and negative traces S_+ and S_-. In the case of an SUL containing an AI model, such as a deep ANN, the notion of traces (consisting of input symbols and triggered state changes) has to be replaced by observing pairs of input datasets I and corresponding output datasets O. The mapping between the two will be denoted as $\{I, O\}$. As such, the approach developed here shows some similarity with the learner/teacher approach from the L^* algorithm, see Section II-A: The central aim of the approach will be to approximate an SUL's behavior by the learner. To this end, a teacher instance is added to the SUL, transforming its input I and output O into a data format compatible with the learner. Since a specific model structure needs to be selected prior to training of the learner, it shall be initially assumed that an oracle exists that the learner can use to select a specific model type. The consequences of this restriction will be examined and discussed in Section IV-E. It is assumed that a sufficiently complex learner can properly monitor compliance of a given AI module with predefined requirements, thus providing functional identification of software as defined in [12]. Once an initial version of the trained SUL exists, it is used as a teacher for a subsequent second AI learner model. This second model will be referred to as the AI monitor in the following text. It will be assumed that the SUL is not modified during an initial stabilization period t_S. A graphical representation of the dataflow during the stabilization period is shown in Figure 3. During this stabilization period, the AI monitor will be trained using $\{I, O\}$ observed during t_S. For the purpose of the experiments

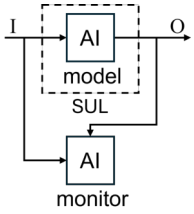


Fig. 3: Stabilization phase of the proposed quality control approach: A second AI module is fed input data I and output data O of the SUL to perform passive learning of the SUL's AI module.

described in Section IV, t_S was selected so that during initial training of the AI monitor/learner, the same amount of input data I was used as for the SUL. It should be noted that this leads to a configuration, where the groundtruth G reference data that corresponds to the input data I of the SUL is not the same as the reference data O used by the AI monitor for initialization and continuous training. After the stabilization period, the trained model of the AI monitor will be used to calculate an individual prediction $p \in P$ for each new input symbol $i \in I$. This will be compared with the corresponding output symbol $o \in O$ of the SUL. i, o and p thus represent individual symbols observed by the AI monitor during normal operation. Over a sliding window of length w matches and mismatches between predictions P and observed output O are monitored. If the resulting prediction accuracy is above a threshold a_{min}, the model of the AI monitor will be updated using i and o. If the threshold is violated, the monitor triggers a compliance warning to all concerned parties. The intended workflow of the method is shown in Figure 4. The restriction regarding the stabilization period is not strictly necessary, but will avoid triggering a large amount of compliance warnings at the beginning of the monitoring process. For the purpose of experimental evaluation, a_{min} was set here to allow a 3% accuracy decrease relative to the initially trained learner. The properties of the approach are shown in row 4 of Table I.

The intention behind the proposed remote quality monitoring approach is to ensure continued compliance of the SUL with requirements while reducing memory consumption and reducing the need for manual interventions. Compared to the hash comparison described in OIML D31 [13] the approach loses some resolution regarding the SUL's behavior since inaccurate predictions of the monitor are tolerated to a certain extent. To check their real-world applicability, both approaches will be described in more detail in Section IV. The section will also perform an in-depth analysis using real-world data.

IV. Experimental evaluation

For the purpose of this evaluation, SULs will be seen as compliant as long as they perform an originally acquired task correctly. Due to the different scenarios of modifying an AI module, this section is divided as follows: Section IV-A describes the algorithms used for evaluation as well as the

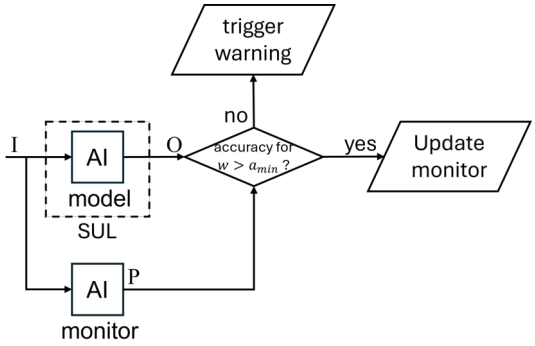
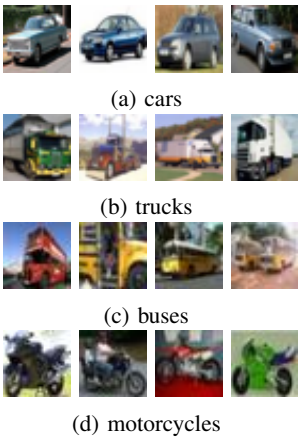


Fig. 4: Monitoring phase of the proposed quality control approach: The second AI module (the AI monitor) continues to monitor the behavior of the SUL and calculates its own prediction accuracy over a sliding window w. If the prediction accuracy drops below a predefined threshold a_{min}, the AI monitor triggers a warning. Otherwise, the newly observed pair of i and o is used to update the AI monitor.

datasets used for experiments. Section IV-B describes how iterative additions of new reference datasets were used to update the SUL and the reaction of the examined methods. Section IV-C extends the use case of providing new global reference datasets to individual reference data for each AI module in use. A replacement of the SUL is addressed in Section IV-D. Section IV-E presents a completely different type of SUL to investigate potential bias in the experiments. Section IV-F discusses the results.

A. Utilized algorithms and datasets

With the aim of testing the applicability of the proposed method of providing evidence of intervention for AI modules, a convolutional neural network (CNN) with six convolutional layers was selected to perform a typical classification task in legal metrology, for which CNNs have already proven their suitability: If speed measurements are performed by law enforcement personell, the used measuring instruments usually incorporate a feature for automatic vehicle classification since different types of vehicles, i.e., cars, trucks, buses, and motorcycles may be subject to different speed limits. As such, this example fits into the EU legislation on measuring instruments and the applicable requirements. It also includes aspects of object detection and recognition, applications for which AI modules have already demonstrated their suitability [2]. This CNN shall serve as the SUL for the remainder of Section IV, except for the use case with CNN3, see Table II. The CNN was implemented using PyTorch and Tensorflow libraries. Training and validation data were obtained from the publicly available CIFAR-10 dataset for images of cars and trucks as well as the CIFAR-100 dataset [16] for images of buses and motorcycles. CIFAR-10 contains 6000 images for 10 different types of objects, whereas the CIFAR-100 dataset contains 100 different classes of objects with 600 images each. The combined dataset used here, thus contained 12000 images of cars and trucks and 1200 images of buses and motorcycles, each image being labeled as belonging to one of the vehicle classes. Exemplary images from each class are shown in Figure 5. For the purpose



(a) cars

(b) trucks

(c) buses

(d) motorcycles

Fig. 5: Exemplary images from the combined CIFAR-10 and CIFAR-100 datasets [16] for the classes ”car/automobile“, ”truck“, ”motorcycle“.

of this experimental evaluation, the input data I are thus the individual training images, whereas the groundtruth G are the corresponding classification labels assigned to those images.

The remote quality control approach described in Section III was similarly implemented using Python Tensorflow and PyTorch libraries. Initially, the internal structure of the AI monitor's model was chosen to be a CNN identical to the one of the SUL. Unless mentioned otherwise, this internal model was used for all subsequent experiments. The AI monitor was used to iteratively learn the behavior of the SUL for all use cases described in Sections IV-B to IV-E using the available input and output data $\{I, O\}$ of the SUL only. Stabilization time t_S and lower accuracy bound a_{min} were configured as described in Section III. The hash comparison algorithm described in [13] was implemented as a reference method in the following manner: The network structure given above was automatically translated to an identifier string that uses a single character to denote the type of the layer, e.g. 'c' for 'convolutional', 'l' for 'linear', followed by the output shape for each layer, where individual dimensions are separated by slash symbols. This results in the string `c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1l128/4` for the described CNN. Similarly, SHA256 hashes were calulated over the exported parameter sets of the CNN for the initial trained network. Both are given in row 1 of Table II.

B. Iterative provision of new external reference data

For the initial model, 500 training images + 100 validation images from each of the four classes were used for its first training. To determine how the two evaluated algorithms react to a modified more precise SUL, CNN1 was updated after its initial training. The update was performed iteratively by adding 1000 images to the training dataset with each new

TABLE II: Identifiers produced by the hash comparison described in [13] for the various SULs used for experimental evaluation.

no.	AI model	classes	topology ID	parameter hash digest
1	**original CNN1**	car, truck, bus, motorcycle	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	53899e22277658092a576cb65f29e443 64107c39080c687cbc29e9afe392d69f
2	**CNN1 update 1**	car, truck, bus, motorcycle	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	9aaba668b4cb6f99d608e54b7fa051de 1cf3465994dc2722888ec2db9df1855d
3	**CNN1 update 2**	car, truck, bus, motorcycle	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	b8e431a95e7f1b335f1779a61854fd8f dee9de46e416aa56864de3c045175422
4	**CNN1 update 3**	car, truck, bus, motorcycle	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	09143768afae4bff1b814de3777a7185 72445e807728b75a1e056362d059eb4f
5	**CNN1 update 4**	car, truck, bus, motorcycle	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	4c8f57d109b4e4deabf8f53b1be4f730 126fc5f505e43fa67572eb1e4cc7eed7
7	**CNN2**	car, truck, bus, motorcycle	c128/512/64/1c128/512/64/1c128/256/64/1 c128/256/64/1c128/128/64/1c128/128/64/1/l128/4	c5e303dcb9d30729cc5e65ef44054283 1650ff6cc9490132ff4ce90744347ca3
8	**CNN3**	horse, bird, car, truck	c128/256/64/1c128/256/64/1c128/128/64/1 c128/128/64/1c128/64/64/1c128/64/64/1/l128/4	3cd2c07c569770c7f5bdbae50007bc7f 08f00102f6f0678e56832d7e44790dce
9	**ResNet50**	car, truck, bus, motorcycle	c128/256/1/1c128/256/1/1c128/256/1/1 c128/512/1/1c128/512/1/1c128/512/1/1 c128/512/1/1c128/512/1/1c128/1024/1/1...	98318830b1eccd5a51422c5c5cf11c4f 7428e0f664dc0cda43e8a76732980ac1

dataset pair $\{I_\Delta, G_\Delta\}$ being used to retrain the SUL. The resulting classification accuracy for each incremental modification is given in Figure 6. At the same time, the AI monitor was fed in-between classification output of the SUL for smaller chunks of added images consisting of 250 images each, to continually observe the SULs behavior within a sliding window. Figure 7 illustrates this continuous monitoring for an excerpt of Figure 6 between the original CNN1 and its first update. In the excerpt, the AI-monitor's accuracy changes slightly with each new batch of images, as these are unknown to both SUL and monitor. Nevertheless, a_{min} is not violated within this excerpt. It should be noted that the accuracy of the SUL is measured between its classification output O and the groundtruth G. For the AI monitor however, accuracy is measured between its own prediction P and the SUL's classification output O. Thus, the AI monitor's accuracy can, theoretically, be higher than that of the SUL. This would indicate that the AI monitor has learned the SUL's behavior correctly, even when the SUL itself performs false classifications. As can be seen from Figure 6, the AI monitor continually achieves an accuracy above the threshold a_{min}. Consequently, the AI monitor's model is updated to include new classification output from the SUL for the observed chunks of images.

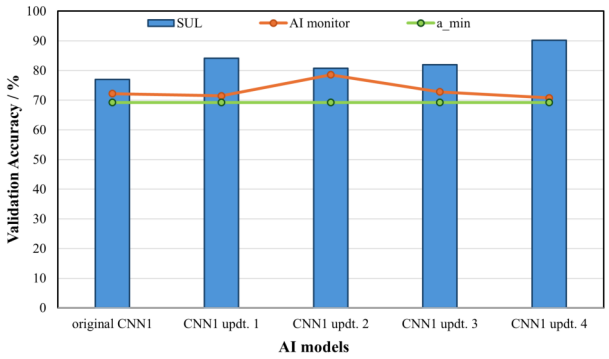


Fig. 6: Timeseries of the AI monitor's classification accuracy for iterative updates. The AI monitor's accuracy is measured relative to the classification output O of the SUL. The SUL's accuracy is measured relative to the groundtruth G.

Fig. 7: Timeseries of the AI monitor's classification accuracy for chunks of new images between updates of the SUL (CNN1 to CNN1 update 1). The AI monitor's accuracy is measured relative to the classification output O of the SUL. The SUL's accuracy is measured relative to the groundtruth G.

C. Iterative provision of individual reference data

From the point of view of both the D31 method [13] and the remote quality control method, no distinction can be made between AI modules being updated with new common external reference data and provision of individual reference data per AI module. Thus, all results from Section IV-B also apply for this use case. The main distinction lies in the required memory capacity needed for storing the configuration of the CNNs for later manual inspection: If each AI module can change independently, such traceability data also must be provided for each module, thus increasing memory requirements linearly with the number of AI modules used in the field.

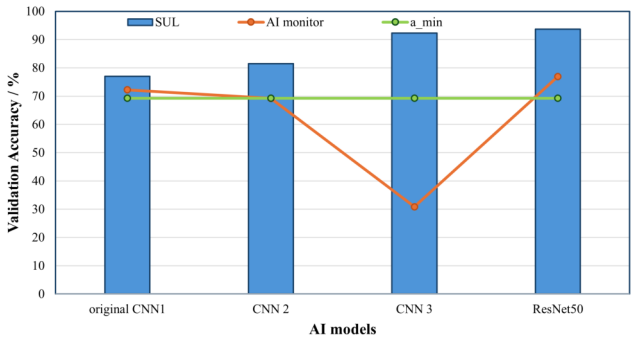


Fig. 8: Timeseries of the AI monitor's accuracy of classification results for the replaced SUL. The AI monitor's accuracy is measured relative to the classification output O of the SUL. The SUL's accuracy is measured against the groundtruth G.

D. Replacement of the CNN

To test the provision of evidence of an intervention by the two algorithms for a modified SUL, the CNN1 was replaced by a different CNN2 with six convolutional layers and one linear layer, where each layer has twice the number of neurons. CNN2 was trained using the aforementioned combined vehicle training dataset from CIFAR-10 and CIFAR-100. Consequently, the topology identification string changed to `c128/512/64/1c128/512/64/1c128/256/64/1c128/256/64/1c128/128/64/1c128/128/64/1l128/4`, and all weights within the CNN of the SUL were abruptly changed, too. The new identifications provided by the D31 method are shown in row 7 of Table II. The classification accuracy of the AI monitor for CNN1 and CNN2 is shown in Figure 8. Even though CNN2 uses a different network topology, the AI monitor still achieves an accuracy similar to the one for the original SUL CNN1.

In order to compare observations made for the classification task performed by CNN1 and CNN2 with a common reference, a third CNN3 was trained to detect birds, horses and cars from the CIFAR-10 dataset, i.e., with different groundtruth data. The identifications obtained by the D31 method for this new SUL CNN3 are shown in row 8 of Table II. Similarly, the reaction of the AI monitor was tested by providing it input data and the SUL's output classification for CNN3 for the image classification task. The resulting change in classification accuracy is shown in Figure 8. As anticipated, the accuracy drops to below 40%, indicating that there is a mismatch between behavior of SUL and AI monitor. How modifications of t_S can influence the detection rate, will be discussed in Section IV-F. At the same time, the output of the D31 method does indicate a modification, but fails to illustrate the impact of the modifications compliance with requirements.

E. Comparison with a reference classifier

To avoid bias because of the known network structure of CNN1 during experiments, a generic KERAS ImageClassifier with the preset "resnet_50_imagenet" (afterwards referred to as ResNet50) was used as an additional reference. It consists of a total of 49 convolutional layers and one output layer. The resulting data are shown in row 9 of Table II, where the topology ID produced by the D31 method had to be truncated since it is too long to be repeated here. In practice, this corresponds to a scenario where the internal topology of the SUL is unknown and the oracle introduced in Section III is no longer needed. However, the learner still needs to know the general task performed by the SUL, i.e., the classification of images into predefined classes. Thus, the oracle from Section III is reduced to providing a general task description which can easily be done by a human expert with knowledge of the data types for I and O. The corresponding prediction accuracy of the AI monitor after stabilization was 76.95%. As can be seen from Figure 8, the AI monitor successfully learned the behavior of ResNet50 despite its unknown internal structure.

F. Analysis

From Table II, it should be clear that due to the avalanche effect in cryptographic hashes [17], even slight modifications of the network weights result in a completely different hash digest. Differentiating between incremental improvements and the complete replacement of the SUL from the hash digest alone thus becomes impossible. Even when combined with the topology ID, the hash digest only provides general information on whether or not any parameter of the ANN was changed. The topology identification on the other hand does allow a human expert to evaluate if the network is still able to perform a certain classification task after a modification. As described in Section II-C, OIML Document D31 solves this issue of differentiating between two identical networks trained for different tasks by imposing storage requirements on the parameter set used for each instance of the ANN. In the particular setup, the CNNs 1 and 3 have 1,153,114 parameters due to the number of connections between successive layers and the ouput layers respectively. CNN2 similarly uses 4,591,642 parameters. For later manual inspection of a specific instance of the CNNs, CNN1 and CNN3 thus require 92.83 MB storage capacity, whereas CNN2 requires 370.92 MB. The remote monitoring approach proposed here, however, does not detect minor gradual changes of an AI module, see Figure 6. In fact, a complete replacement of the SUL with another CNN performing an identical task does not result in any compliance warnings. Only the abrupt modification of the SUL, in case it is replaced with a CNN that portrays a different behavior for similar input data, is automatically detected, see Figure 8. However, if a replacement is done gradually, without violating a_{min} within t_S, the AI monitor would similarly fail to trigger any warnings. The detection of compliance violations is influenced significantly by the allowed stabilization time t_S for training an AI monitor and by the allowed lower accuracy bound a_{min}. While both can be fine-tuned for a specific task, an inadequate selection of one or the other can result in either too many or too few compliance. While the default values chosen in Section IV-A may have worked for the examined use cases, they are not guaranteed

to work as well in other settings. In practice, t_S could be set to a default value, such as one day and then decreased iteratively, if too many changes occur. Similarly, $a_{\min}$ could be initialized with a value of 90% and then be reduced to fit the particular application. The applicability of the method developed here does not only depend on the parameter settings, but also on the size of available observed data. Minimum requirements for these data for different use cases and AI models still remain to be formally specified. A full oracle, which can tell the topology of a SUL, is not always needed, see Section IV-E. In fact, one could envision a scenario where, in Legal Metrology in particular, the usage of certain types of ANNs is prescribed for specific tasks, thus eliminating the need for an oracle altogether. At the moment, it appears unfeasible to use a general-purpose AI model and attempt to learn an SUL's behavior using an extremely large observation dataset $\{I, O\}$. It does, however appear plausible that the setup and AI monitor used here, can also be applied to different kinds of image or more general signal classification tasks. Nevertheless, there is the potential bias in the findings shown here since image classification is particularly suitable for the remote monitoring approach proposed here. Therefore, further work will include extending the method developed here to audio classification and other common AI tasks.

V. Summary

As stated in Section IV the D31 method described in [13] appears to be realizable for providing evidence of intervention for AI modules in practice if a white-box scenario is given. However, the method has very high memory consumption if traceability of individual changes is required. Nevertheless, this paper can be seen as a first proof of concept of the method from [13]. In addition, the AI monitor introduced here was able to correctly identify compliance violations as required, for example, for regulated measuring instruments in the EU. [18] As demonstrated in Section IV-F, the AI monitor can be seen as another form of the functional identification of a software module introduced in [12]. It would allow inspectors or market surveillance authorities to remotely monitor the compliance of AI modules in quasi black-box settings in regulated industry sectors, such as legal metrology. Moreover, the method could equally be applied by users of AI services in other industry sectors to determine if a service quality is reduced without prior warning. Of course, the monitoring approach requires resources similar to those for operating the actual SUL. The obvious advantage of the approach, however, is that the monitoring does not have to be conducted permanently. The monitoring can also be carried out at a later point when resources are available as long as observed data can be buffered for an intermediate timespan. Further work will address the tradeoff between resources and algorithmic complexity as well as the impact of t_S and $a_{\min}$ on monitoring accuracy. It will also include application of the remote monitoring method to other use cases and investigations into minimal observable data requirements for different use cases. Such investigations would also provide some insight into the applicability of general-purpose AI modules as AI monitors for a larger range of tasks. Additionally, benchmark tests comparing actual memory usage between the remote monitoring system and traditional techniques will be performed.

References

[1] K. McElheran, J. F. Li, E. Brynjolfsson, Z. Kroff, E. Dinlersoz, L. Foster, and N. Zolas, "Ai adoption in america: Who, what, and where," *Journal of Economics & Management Strategy*, vol. 33, no. 2, pp. 375–415, 2024.

[2] Y. Wei, N. Song, L. Ke, M.-C. Chang, and S. Lyu, "Street object detection / tracking for ai city traffic analysis," in *2017 IEEE SmartWorld*, 2017. doi: 10.1109/UIC-ATC.2017.8397669 pp. 1–5.

[3] G. Rani, J. Singh, and A. Khanna, "Comparative analysis of generative ai models," in *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, 2023. doi: 10.1109/ICAICCIT60255.2023.10465941 pp. 760–765.

[4] EC, "Regulation (eu) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence," European Union, Council of the European Union; European Parliament, Directive, February 2024.

[5] "WELMEC 7.2 Software Guide," European cooperation in legal metrology, WELMEC Secretariat, Braunschweig, Standard, Mar. 2022.

[6] D. Angluin, "Learning regular sets from queries and counterexamples," *Information and Computation*, vol. 75, no. 2, pp. 87–106, 1987. doi: 10.1016/0890-5401(87)90052-6

[7] M. Sipser, *Introduction to the theory of computation*, 2nd ed. Boston, Massachusetts: Thomson, 2006. ISBN 0-534-95097-3

[8] S. Windmüller, J. Neubauer, B. Steffen, F. Howar, and O. Bauer, "Active continuous quality control," in *Proceedings of the International Symposium on Component-Based Software Engineering*. ACM, Jun. 2013. doi: 10.1145/2465449.2465469 pp. 111–120.

[9] J. Neubauer, S. Windmüller, and B. Steffen, "Risk-based testing via active continuous quality control," *International Journal on Software Tools for Technology Transfer*, vol. 16, pp. 569–591, 2014. doi: 10.1007/s10009-014-0321-6

[10] C. Oliva and L. F. Lago-Fernández, "On the interpretation of recurrent neural networks as finite state machines," in *Proceedings of ICANN 2019: Theoretical Neural Computation: 28th International Conference on Artificial Neural Networks,*, vol. I 28. Munich, Germany: Springer International Publishing, September 2019, pp. 312–323.

[11] B. Aichernig, E. Muskardin, and A. Pferscher, "Active vs. Passive: A Comparison of Automata Learning Paradigms for Network Protocols," in *Formal Methods for Autonomous Systems and Automated and verifiable Software system development*, ser. Electronic Proceedings in Theoretical Computer Science, EPTCS, vol. 371. National ICT Australia Ltd, September 2022. doi: 10.4204/EPTCS.371.1 pp. 1–19.

[12] L. C. X. Ho, M. Esche, M. Nischwitz, and S. Glesner, "Black-box conformity tests on regulated measuring instruments: A machine learning approach," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference*. Chemnitz, Germany: IEEE, 05 2025, to be published.

[13] "OIML D 31: General requirements for software controlled measuring instruments," International Organisation of Legel Metrology, Paris, France, Tech. Rep., 2023.

[14] M. Gorrod, *The software development lifecycle*. London: Palgrave Macmillan UK, 2004, pp. 93–120. ISBN 978-0-230-51029-6. [Online]. Available: https://doi.org/10.1057/9780230510296_4

[15] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable ai systems," *ACM Trans. Interact. Intell. Syst.*, vol. 11, no. 3–4, Sep. 2021. doi: 10.1145/3387166. [Online]. Available: https://doi.org/10.1145/3387166

[16] A. Krizhevsky, "Learning multiple layers of features from tiny images," *University of Toronto*, 05 2012.

[17] D. Upadhyay, N. Gaikwad, M. Zaman, and S. Sampalli, "Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications," *IEEE Access*, vol. 10, pp. 112 472–112 486, 2022. doi: 10.1109/ACCESS.2022.3215778

[18] EC, "Directive 2014/32/EU of the European Parliament and of the Council of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of measuring instruments," European Union, Council of the European Union; European Parliament, Directive, February 2014.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 31–36
DOI: 10.15439/2025F6762
ISSN 2300-5963 ACSIS, Vol. 44

SIG Denúncia - Web GIS of Popular Participation in the Public administration

Henrique Pereira de Freitas Filho
Instituto Federal de Brasília
Campus Taguatinga
Brasília, Brasil
Email: henrique.filho@ifb.edu.br

Thiago Oliveira de Freitas, Johnny Evangelista Figueiredo
Instituto Federal de Goiás
Campus Luziânia
Goiás, Brasil
Email: thiago.oliveira.f@outlook.com, johnnyestudos@gmail.com

***Abstract*—The participation of the population in public management results in different solutions from those that can be obtained without their involvement, because it is the inhabitants who know the local problems and can present details that are not usually obtained from other sources. In order to receive and manage denunciations related to public services, focusing on the involvement of society, it is necessary to create systems capable of registering, organizing and presenting this information covering the geographic position of each occurrence. In this sense, a computational solution called SIG Denúncia was developed, which includes an Android application, a Web Geographic Information System (Web GIS) and a geographic database, which together allow the collection, visualization and storage of the alphanumeric and geographic data of each denunciation received regarding public services.**

***Index Terms*—Web Geographic Information System (Web GIS); Geographic Database; SIG Denúncia; Public Services; Public Administration; Popular Participation.**

I. Introduction

PUBLIC services are provided directly by a country's government or by affiliated entities with the aim of meeting the needs of the population and their habitat. Among the most important services are hospital care, police patrols, educational activities, public transportation, maintenance of recreational areas, environmental preservation, urban works, and others. These services, when delivered with quality, ensure the well-being of citizens and the preservation of urban spaces and nature. However, when these services are executed irregularly, they can negatively impact people's daily lives, resulting in serious social problems, particularly in Brazil, where the majority of individuals have low incomes and rely on these services daily [1].

The collection of data on complaints related to public services in Brazil is generally conducted through phone calls or in-person visits. A citizen calls or goes to the public agency responsible for overseeing the public activity to report their dissatisfaction. This process often leads to data loss and inconsistency, as the collected information is typically stored on paper, which can easily be misplaced or lost due to its fragility. Consequently, this method does not provide a reliable and consistent database. The difficulty faced by the population in participating in public matters fosters public dissatisfaction and results in low data collection, which could be crucial for improving the services provided by each Brazilian municipality [2].

Public participation in the management of public services is mandated by Brazilian law to improve the quality of public administration [3]. The residents of each city are the ones who truly understand the local realities and issues and are able to provide details that are often not found in other sources [4].

With the digital transformation that has occurred in Brazil in recent years, citizens in many cities have gained a new means of reporting issues related to public services through the web systems of public agency ombudsman offices. While this innovation partially addresses the previously mentioned database issues and offers greater convenience and satisfaction to the population, challenges remain. For each area or type of complaint, citizens must access a specific system of a specific agency to file their report; there is no unified system for submitting all types of complaints. After a complaint is made, only the individual who submitted it receives feedback and has access to the status of the complaint. Other citizens are unaware of the complaints that have already been filed, resulting in a lack of transparency. Additionally, these ombudsman systems do not utilize georeferencing, making it difficult to pinpoint locations and have an overarching view of the issues presented in the complaints [5].

Despite the widespread availability of mobile and web technology in Brazil, public administration in several Brazilian cities still lacks the digitization of many of its work processes, including the collection and management of information on the quality of public services, particularly those involving public participation [6].

To address this issue, a computational system was developed, consisting of a mobile application, a Web Geographic Information System (Web GIS), and a geographic database. With the mobile application, citizens can register their com-

The authors of this article would like to thank the Research Support Foundation of the Federal District (FAP-DF) for your support.

Topical area: Information Technology for Business and Society

plaints in text format and include a photo to verify the existence of the reported problem.

The Web GIS will provide visualization and management of the occurrences received by the system, ensuring that city administrators can respond to complaints received through the application, and allowing the population to view a map containing the data of each reported issue, thereby ensuring transparency of information for society.

The geographic database will be used by both the mobile and web applications to store and query the alphanumeric and geographic data of complaints related to health, safety, education, transportation, recreation, the environment, and urban development projects. This database can be extended to persist data from various other areas of public services.

The system was developed using modern software development technologies that allow for the maintenance of up-to-date information. It highlights the issues present in each environment within the municipal, state, and federal spheres, as well as the expectations of its users.

II. Popular Participation in Public Administration

The concept of public administration has two meanings. One is objective, encompassing the idea of action, activity, and task, encapsulating the very function of administering. The other is subjective, as it refers to the universe of administrative sectors and the people who perform management work collectively, sharing the same function [1].

Popular participation refers to the involvement of citizens or their representatives from social groups in public management with the aim of seeking improvements within the administration of the State, implemented to favor the collective interest [7]. This term is used when a citizen or their social representative seeks the common good without pursuing personal interest. It conceptualizes the exercise of society's power over Brazilian politics and expresses the democracy mandated by law in the contemporary era [8] [9].

In Brazil, societal participation gained greater emphasis following the 1988 Brazilian Constitution, which granted a series of rights and duties to Brazilians, including the right to human dignity and citizenship [1]. Article 1 of the Constitution emphasizes that all power emanates from the people, who exercise it directly or through democratically elected representatives. Article 37 of the Constitution outlines the right of Brazilian citizens to participate directly or indirectly in public governance [10].

In recent decades, there has been enhancement in discussions and legislation addressing the participation of Brazilian society. However, the achieved outcome is not yet considered fully democratic due to low popular engagement, as the government has not managed to include the majority of the urban population, attributed to three main factors described below [7][9]:

1) Political apathy: occurs when the population lacks information about their rights and responsibilities, receives no feedback through public communication channels with government officials, does not receive responses to their inquiries, or experiences excessive delays in response times, leading to low levels of popular participation;
2) Political abolition: becomes evident when citizens choose not to engage in public management due to their disbelief in being heard by the government and having their requests addressed;
3) Political acracy: is the issue that arises when citizens' level of education is low and the participation tools provided by the government address complex terms and data, which hinders the involvement of individuals who lack sufficient education to engage in public management affairs.

According to [4], the Brazilian government still requires effective tools to ensure the rights established by laws for citizens, aiming to enhance the quality of popular participation. There remains a challenge in the daily lives of public service users to efficiently engage in matters related to their areas of life.

III. The SIG Denúncia

A. Architecture

The architecture of the SIG Denúncia solution shown in Figure 1 was designed following the thick client model.

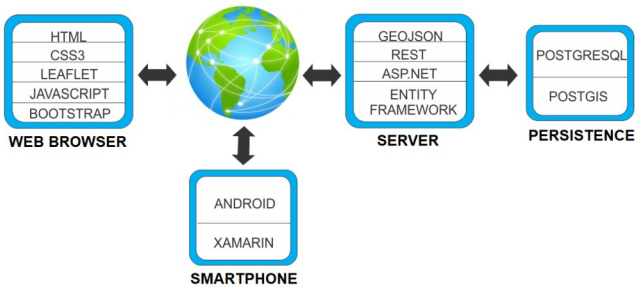


Fig 1. Architecture of the SIG Denúncia

The SIG Denúncia is a computational solution (system) that consists of two applications sharing the same geographic database:

- Mobile Application: The mobile application is responsible for collecting and submitting citizens' complaints to the server. The framework Xamarin was used for the development of this system. The communication between the application and the server is conducted using RESTful, with data transmitted in JSON format;
- Web GIS: It was built using HTML5, CSS3, JavaScript, Bootstrap, and Leaflet technologies. The web system is divided into two parts:
  - Map: This is the homepage of the web system, responsible for displaying georeferenced complaints from citizens. In this layer of the system, Leaflet was utilized as the primary component of the Web GIS. Leaflet is responsible

for generating and managing the map within the client's browser. Data communication between the server and Leaflet is performed using RESTful, with information transmitted in GeoJSON format;

- Administrative: The administrative area manages the system's forms, utilizing the Bootstrap framework to ensure a responsive interface. This allows the interface to adapt seamlessly to the client's browser screen size.

In the Server layer, technologies such as ASP.NET MVC, RESTful, GeoJSON, and Entity Framework were employed. ASP.NET manages the operations of the system's forms, while the RESTful communication layer returns responses in JSON and GeoJSON formats. ASP.NET utilizes Entity Framework for mapping, controlling, and accessing persistence.

The Persistence layer is responsible for storing registration information. The PostgreSQL database management system with the PostGIS geographic extension was used to store geographic information.

B. Technologies Used

The following are the technologies used in the SIG Denúncia:

- ASP.NET MVC: Part of the .NET framework, ASP.NET MVC was designed for creating websites. Websites built with ASP.NET run on Internet Information Services (IIS) and can be developed using C#, F#, and VB.NET languages. ASP.NET MVC implements the Model-View-Controller (MVC) pattern, a software architecture pattern created to abstract the complexity of information systems by separating development into layers [11];
- C#: It is an interpreted, multi-paradigm programming language rooted in C and easily assimilated by developers familiar with C, C++, and Java [12]. C# is compiled and interpreted within the .NET framework;
- GeoJSON: It is a data interchange format for various geographical data structures based on the JSON format. GeoJSON supports geometric types such as Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection [13];
- Leaflet: It is a robust open-source library developed in JavaScript with a simple design. It operates on all major browsers and mobile platforms, supporting CSS3 and HTML5 [14]. Developers need to connect Leaflet to a map server, which can be public like OpenStreetMap [15], private, or personal using a geographic database management system (GIS) such as PostgreSQL together with PostGIS. Georeferenced objects are added to the map through the GeoJSON layer. Thus, Leaflet can manage interactive layers;
- PostgreSQL: It is an open-source data storage system. PostgreSQL is a powerful framework for data management and processing, as it allows the use of multiple languages to execute functions and triggers, making it flexible. It features dynamic loading of user-defined functions, eliminating the need for recompiling the database, and includes automatic actions to update changed data in the database. The system is available for MacOS, Linux, and Windows operating systems [16];
- PostGIS: It is an open-source spatial extension for the PostgreSQL database management system (DBMS). This extension enables PostgreSQL to support spatial types such as Point, LineString, Polygon, Multipoint, MultiLineString, MultiPolygon, and GeometryCollection. PostGIS provides numerous functions for spatial queries, allowing geographic queries to be performed using SQL;
- Xamarin Community 2015: It is a framework used for developing cross-platform mobile applications. It allows developers to create native apps for iOS, Android, and Windows Phone using a single language, C#. This means a team of C# developers can produce an application for three different platforms without needing to learn a new language [17].

IV. Systems Interface

The SIG Denúncia consists of a mobile application and a Web GIS. This section presents the interfaces of these systems.

A. Mobile App

This section demonstrates the interface of the mobile application and its functionalities. To make a complaint, users need to log into the system. Therefore, users must enter their email address and password. If they have not registered yet, they can click on "Register" on the home screen of the system.

The application has two main tabs: "Report" and "Complaint History". The "Report" tab consists of four sub-tabs (as shown in Figures 2 and 3), which represent the steps required to make a complaint, described as follows:

- Area: The citizen must choose the area where they will make the complaint;
- Type: After selecting the area, the user must choose the type of complaint. Each type of complaint is listed under categories; Figure 2 displays the categories within the "Infrastructure and Urbanization" area and the types of complaints that can be made under the "Lighting and Energy" category;
- Additional Information: The citizen must attach an image to their complaint by clicking "Attach a photo to your Complaint" and enter a comment in the field below;
- Submit: On this screen, all complaint details are displayed for the user to confirm. The user must

have the GPS location active on their Android device; if it is disabled, the app will request the user to activate it. The "Submit" button is only enabled after the device's location is determined. After submitting the complaint, the app returns to the "Area" tab.

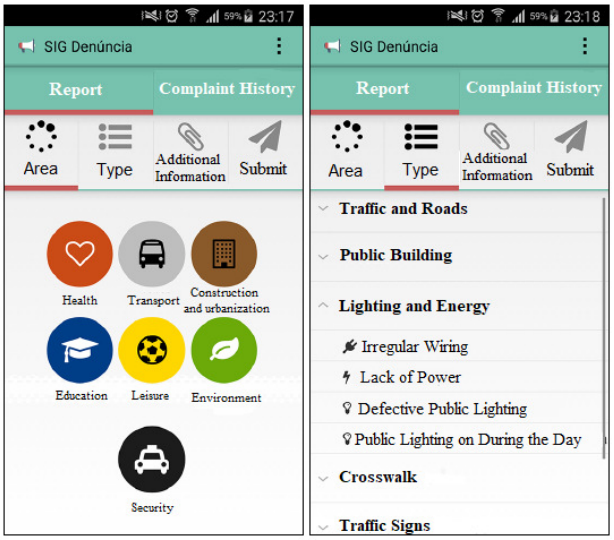


Fig 2. Screens of the "Report" tab - Part 1

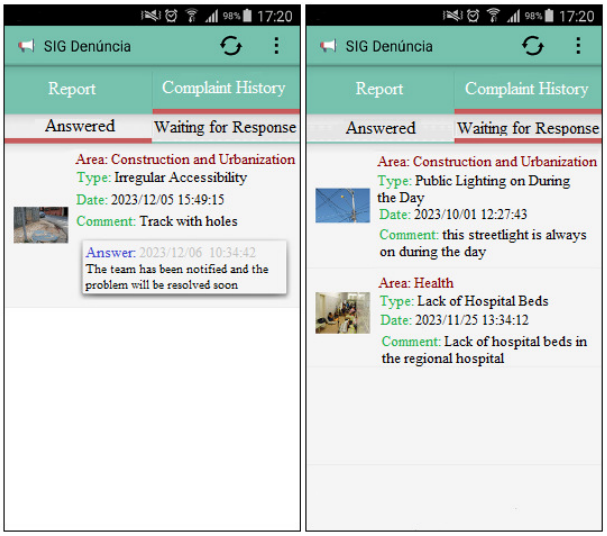


Fig 3. Screens of the "Report" tab - Part 2

The "Complaint History" tab displays the complaints submitted by the user regarding public services and the responses provided by public agencies, as shown in Figure 4.

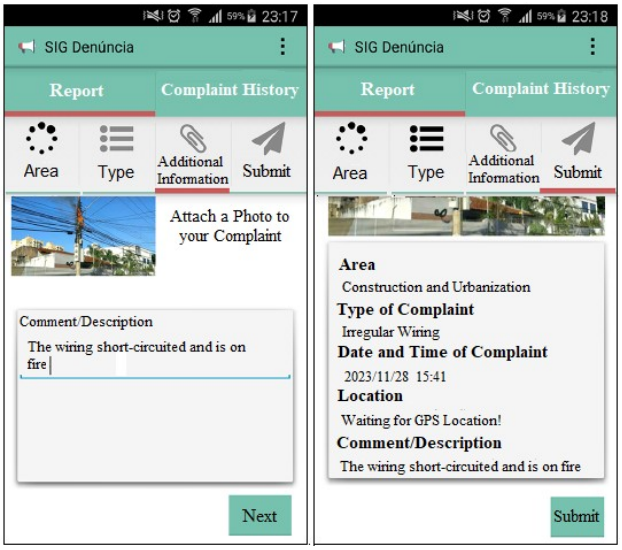


Fig 4. Screens of the "Complaint History" tab

B. SIG Web

The Web GIS has two main areas: the map area (georeferenced) and the administrative area. In the map area, all complaints are displayed at their respective locations. Figure 5 depicts the initial screen of the Web GIS.

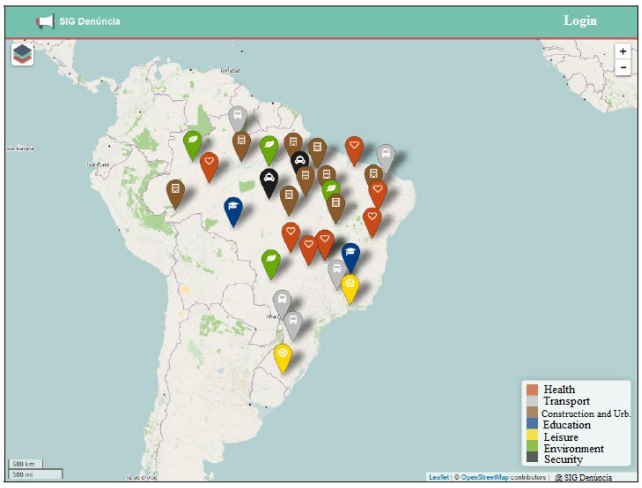


Fig 5. Screens of the Web GIS - Part 1

Clicking on a complaint icon displays a tooltip containing information about the complaint, such as the citizen's name who submitted the complaint, type, category, city name, comment, date, latitude, longitude, and the image uploaded by the citizen, as shown in Figure 6.

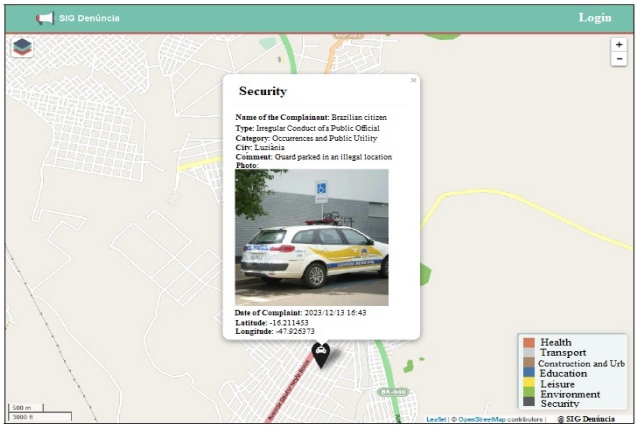


Fig 6. Screens of the Web GIS - Part 2

In the administrative area, the administrator user has functionalities to respond to complaints, as well as to add, edit, and delete areas and users. Figures 7 and 8 depict some screens from the administrative area.

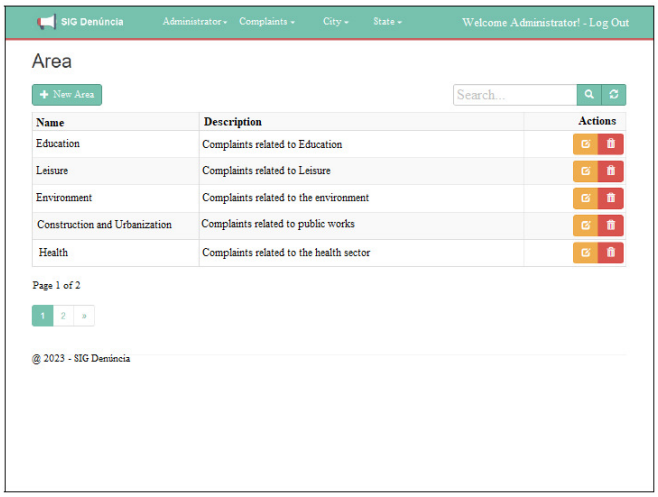


Fig 7. Screens of the administrative area of the Web GIS - Part 1

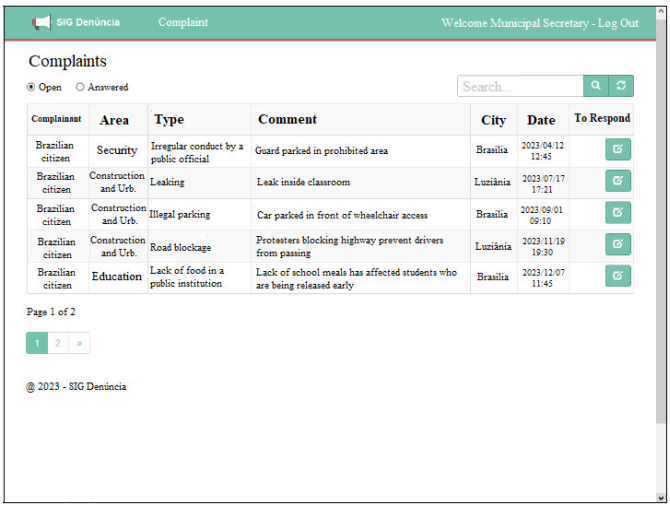


Fig 8. Screens of the administrative area of the Web GIS - Part 2

V. Related Work

Similar works to the SIG Denúncia solution also encompass the issue of public participation in public services.

A. Colab.re

Colab.re is a digital platform that provides government services to citizens through a web interface and a free mobile application available for Android and iOS devices. Its primary objective is to improve Brazilian cities by fostering collaboration between the population and public authorities. The platform enables users to report urban issues, submit evaluations, and suggest solutions, with complaints being forwarded to the appropriate government agencies. While it offers functionalities such as photo submission, idea sharing, and public discussion, Colab.re does not incorporate georeferencing features to locate reports, which limits the spatial accuracy of the submitted information [18].

B. SP156

SP156 is a public service platform developed by the São Paulo City Hall to encourage citizen participation in municipal management. Similar to Colab.re, its primary focus is the provision of public services to citizens through a web portal, a mobile application, and a telephone service center (accessible via 156). Although the systematic collection of complaints is not its core objective, the system allows users to register and track requests related to urban issues, such as waste management or street lighting. However, SP156 does not utilize georeferencing to locate reported occurrences, which limits the spatial accuracy of the information and reduces its potential for territorial analysis [19].

C. The Differentiator of SIG Denúncia

SIG Denúncia stands out from other similar platforms by being specifically designed for the management of urban complaints, with a strong emphasis on the spatialization and georeferenced monitoring of reported occurrences. While many existing solutions prioritize the provision of public services or the simple forwarding of individual requests, SIG Denúncia distinguishes itself by integrating georeferencing capabilities, which add substantial value to the decision-making processes of public administration.

The use of spatially referenced data enables precise visualization of the territorial distribution of citizen demands, facilitating the identification of patterns, critical areas, and recurring problems. This approach supports more efficient urban planning, strategic resource allocation, and the development of public policies grounded in territorial evidence. Furthermore, it allows municipal authorities to adopt a more proactive and coordinated approach, guided by the geographic context of reported issues.

Additionally, the system is characterized by its configurational flexibility, enabling administrators to easily define new categories of reports and designate specific geographic areas of interest. This versatility ensures that the platform can be adapted to the particular needs of diverse urban settings, ranging from large metropolitan centers to small mu-

nicipalities. In this way, SIG Denúncia establishes itself as a robust, scalable, and effective tool for strengthening participatory governance and integrated territorial management in cities.

VI. Conclusion and future work

Currently, in several cities across Brazil, for a citizen to file a complaint or express dissatisfaction to the government, they must physically visit the responsible public agency or make a phone call. However, even after going through this process, their reports may get lost because they are stored on paper, which is prone to misplacement or loss due to its fragility. Therefore, it is evident that individuals face difficulties in presenting local issues to public administration and tracking their submitted requests.

To address this issue and improve public participation in governance matters, the development of a computational solution called "SIG Denúncia" has been proposed. This solution comprises a mobile application, a Web GIS (Geographic Information System), and a geographic database. Together, these components enable better interaction between public service users and city administrators for tracking incidents and potential resolutions.

Unlike other similar systems, which often focus primarily on the provision of public services and are developed for specific contexts, SIG Denúncia was designed with an emphasis on the structured management of urban complaints, enabling the registration, consultation, and systematic monitoring of reported occurrences. One of the main distinguishing features of the solution is the integration of georeferencing capabilities, which allow for the spatial mapping of complaints and significantly enhance the system's potential to support decision-making processes within public administration. The spatial analysis of data facilitates the identification of critical areas, the detection of recurring patterns, and the strategic prioritization of interventions based on evidence. Moreover, SIG Denúncia offers a high degree of configurability, allowing for customization across different categories of complaints and geographic areas, making it suitable for a wide range of urban contexts—from large metropolitan centers to small municipalities.

The mobile and web applications were developed in strict accordance with the proposed architecture, fully fulfilling their intended role as integral components of a Web Geographic Information System (Web GIS) designed to promote Popular Participation. Although the system has not yet been deployed in an operational environment, its design and implementation demonstrate substantial potential to significantly enhance the interaction between citizens and public administration. By fostering greater transparency, facilitating more efficient responses from public agencies, and encouraging active citizen engagement, these functionalities contribute meaningfully to the strengthening of participatory urban governance.

As part of future work, several enhancements are planned, including the addition of features such as audio and video messaging, data caching to enable offline functionality, and mechanisms for information synchronization when an internet connection becomes available. The system is also intended to be deployed in a Brazilian city, with a subsequent evaluation of the outcomes to complete and validate the research. Additionally, tools and dashboards may be developed to provide strategic insights to public administrators, following the principles of Business Intelligence (BI).

Acknowledgment

The authors of this article would like to thank everyone who contributed to this work, as well as the Federal Institute of Education, Science and Technology of Goiás (IFG), the Federal Institute of Education, Science and Technology of Brasília (IFB), and the Research Support Foundation of the Federal District (FAP-DF) for their support.

References

[1] Filho, J. d. S. C.: Manual of Administrative Law. Atlas, (2015).
[2] Guide for Complaints about Public Services – MPGO, https://www.mpgo.mp.br/portal/arquivos/2013/06/10/09_56_26_826_cao_consumidor.pdf, last accessed 2023/11/09.
[3] Costa, W. L.: Participatory Management in Public Administration in the Current Scenario, https://wesley18.jusbrasil.com.br/artigos/226084652/gestao-participativa-na-administracao-publica-no-cenario-atual, last accessed 2023/08/20.
[4] Bugs, G., Reis, A. T.: Popular Participation in Urban Planning: Interactive Maps and GIS Tools on the Internet and Cognitive Aspects. Proceedings: National Meetings of ANPUR, v. 14, in press, (2013).
[5] Transformação Digital, https://www.gov.br/governodigital/pt-br/estrategias-e-governanca-digital/sisp/guia-do-gestor/ptd, last accessed 2024/04/12.
[6] Branco, W. G., Holanda, M. T.: Comune — An android application for applying surveys to and collecting reports from public service users.12th Iberian Conference on Information Systems and Technologies – CISTI, Lisbon, Portugal (2017).
[7] Modesto, P.: Popular Participation in Public Administration: Operational Mechanisms. Jus Navigandi, Teresina, v. 6, (1999).
[8] Silva, I. V. S.: Social Control in Public Administration and the Principle of Popular Participation, https://openrit.grupotiradentes.com/xmlui/bitstream/handle/set/1533/monografia%20biblioteca.pdf?sequence=1&isAllowed=y, last accessed 2023/10/11.
[9] Gavronski, A. A.: Popular participation. Dictionary of Human Rights, http://www.esmpu.gov.br/dicionario, last accessed 2023/10/05.
[10] Constitution of the Federative Republic of Brazil, http://www.planalto.gov.br/ccivil_03/Constituicao/Constituicao.htm, last accessed 2023/09/23.
[11] Galloway, J., Haack, P., Wilson, B., Allen, K. S.: Professional ASP. NET MVC 4. John Wiley & Sons, (2012).
[12] Hejlsberg, A., Wiltamuthi, S., Golde, P.: The C# programming language. Adobe Press, (2006).
[13] BUTLER, H., DALY, M., et. al.: GeoJSON Format Specification, https://datatracker.ietf.org/doc/html/rfc7946, last accessed 2023/10/13.
[14] Leaflet, http://leaflet.org/, last accessed 2023/10/10.
[15] OpenStreetMap, https://www.openstreetmap.org, last accessed 2023/12/17.
[16] Krosing, H., Mlodgenski, J.: PostgreSQL server programming. Packt Publishing Ltd, (2013).
[17] Hermes, D.: Developing Mobile Applications with Xamarin. Apress, (2015).
[18] Colab.re, https://pages.colab.re/governodigitalsolucoes, last accessed 2023/11/10.
[19] Sp156, https://sp156.prefeitura.sp.gov.br/portal/servicos, last accessed 2023/08/28.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 37–40
DOI: 10.15439/2025F5892
ISSN 2300-5963 ACSIS, Vol. 44

Constructive genetic algorithm with penalty function for a concurrent real-time optimization in embedded system design process

Adam M. Górski
0000-0003-3821-5333
Jagiellonian University
Faculty of Physics, Astronomy
and Applied Computer Science
ul. Prof. Stanisława
Łojasiewicza 11,
30-348 Kraków, Poland
Email: a.gorski@uj.edu.pl

Maciej J. Ogorzałek,
fellow IEEE
0000-0003-3314-269X
Jagiellonian University
Faculty of Physics, Astronomy
and Applied Computer Science
ul. Prof. Stanisława
Łojasiewicza 11,
30-348 Kraków, Poland
Email: maciej.ogorzalek@uj.edu.pl.

***Abstract*—In this paper we present a genetic programming based constructive algorithm with penalty function for a concurrent real-time optimization in embedded system design process. Proposed approach uses genetic programming mechanism to optimize detecting and assignment of unexpected tasks process in embedded system design. Unlike others methodologies the approach described in this paper uses a penalty in objective function in optimization process. As a result during the evolution generations of individuals can also contain solutions which violate time constraints. Thus the approach is more proof to stop in local minima of optimizing parameters. Therefore the final result could be better adapted to the environment and the optimization process can be cheaper and more effective.**

***Index Terms*—Genetic Programming, Concurrent Real-Time Optimization, Embedded Systems, Artificial Intelligence.**

I. Introduction

EMBEDDED system design process [1] can be split on four phases[2]: modeling, implementation, validation and assignment of unexpected tasks. Unexpected tasks [2][3] can appear when the architecture of embedded system is produced, all known tasks are assignment to available resources and the system works in a target environment. In [4] authors proposed a methodology for assignment of unexpected tasks for a group of embedded systems. Unexpected tasks that appeared were the result of cooperation of the systems in bigger environment. The first methodologies [2][5] proposed for assignment of unexpected tasks have one major weakness – unexpected tasks needed to be detected externally. All values of time and cost of execution needed to be given for every task. In [6] the authors proposed an algorithm which was able to detect unexpected tasks and assign them to appropriate Processing Element (PE). The authors indicated that some of unexpected situations can be solved as a result of connection of some number of subtasks of known tasks. However not only one connection of subtask leads to solve unexpected situation. On the other side not every connection give the solution. Connection of a subtasks that gives an appropriate solution needs to be assignment on one of available resources. The problem is to find which connection of subtasks is better. Such a problem was called picking an apple problem. Generally the optimization process can be split into two phases. Each phase impacts another in a real-time. That is why this type of optimization was named concurrent real-time optimization. Further information about the problem and are given in next section. In [7] the authors proposed the solution of such a problem in IoT design. In [6] genetic algorithm was proposed to solve the problem in embedded system design. Genetic programming methodology [8] was also presented for such a problem. The biggest disadvantage of the methodology was a constructive nature of the algorithm. Such group of methodologies [9] [10] have low complexity but are prone to stop in local minima of optimizing parameters. It is caused because such methodologies construct the system by making decisions step by step for every task separately. Iterative improvement algorithms [11] [12] start from suboptimal solutions, usually the fastest, and by local changes try to improve the system quality. Such algorithms can escape local minima however the results are still suboptimal. In [13] the authors provided the genetic programming based iterative improvement method for the problem. However the biggest disadvantages of the methodology was that only valid individuals could be investigated in the evolution process. Therefore some of the solutions could be unobtainable. Concurrent real-time optimization occurs not only in hardware design. The solution for such kind of optimization was also proposed in game theory [14]. Proposed methodology belonged to metaheuristics group. The authors

·The publication has been supported by a grant from the Faculty (Faculty of Physics, Astronomy and Applied Computer Science) under the Strategic Programme Excellence Initiative at Jagiellonian University.

proposed a grey wolf optimizer to find an automatic solution of computer games.

In this paper we propose a genetic algorithm based methodology [6][16] with penalty function for concurrent real-time optimization in embedded system design process. Unlike other approaches we investigate in evolution process not only valid solutions. Therefore the algorithm is more able to escape local minima of optimizing parameters.

The paper is organized as follows: next section are preliminaries, then the algorithm is described. The fourth section contains experimental results. At the and the conclusions and directions of future work are presented.

II. Preliminaries

A. Embedded systems

Embedded systems are computer systems mostly microprocessor or microcontroller based. They were created to execute some special group of tasks. most of modern systems are solved as distributed once. Such kind of systems are consisted of two kinds of resources: processing elements (PEs), responsible for executing the tasks, and communication links (CLs) responsible for providing communication between PEs. There are two basic kinds of PEs: programmable processors (PPs) and hardware cores (HCs). PPs are universal resources able to execute more than one task. HCs are specialized resources dedicated to execute only one task. Therefore unexpected tasks can be executed only by PPs. The behaviour of the system is specified by an acyclic directed graph called an extended task graph G = (V, E). Each node $v_i \in V$ in the graph is a task, each edge $e_{ij} \in E$ describes the amount of data transferred between two connected tasks. The transmission time t_{ij} is equal to:

$$t_{ij} = \frac{e_{ij}}{b} \quad (1)$$

where b is a bandwidth of a communication link. Fig. 1 below presents the example of a task graph.

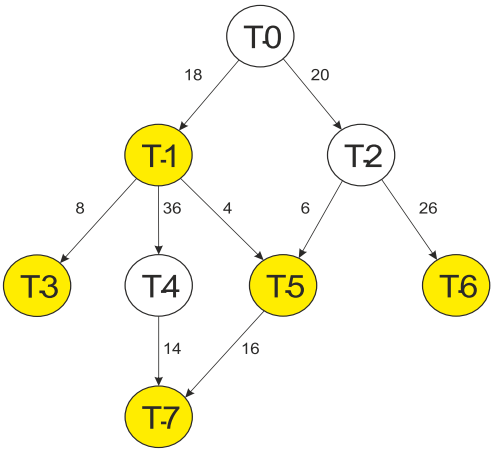


Fig 1. Example of an extended task graph

The graph contains eight tasks. The nodes with yellow color (T1, T3, T5, T5, T7) marks the tasks that can be split on subtasks. The overall cost of a system C_o is described by the following formula:

$$C_o = C_a + \sum_{i=1}^{n} C_i + k * (t - t_{max}) \quad (2)$$

where n is a number of tasks in an extended task graph, t_{max} is a time constrain, k is a parameter given by the designer which decides about the penalty function and therefore how is the weight of violation of time constraints. The unit of k is [c/t] where c is a unit of cost and t is a unit of time. The goal of the optimization is to find the solution with the lowest value of C_o.

B. Concurrent real-time optimization – picking an apple problem

It is possible to pick up an apple on many ways. Each of them demands different parameters to optimize and different tasks to execute. The question is how to find out which way is better without picking up the apple. The problem can be split into two phases. The first phase is responsible for the choice of optimizing parameters. The second phase makes the optimization and verifies the choice made in the first phase. During the process it can be found out that some of the ways of solving the problem do not lead to success. It is not always known at the beginning. In such a case after executing some of the tasks and starting optimization process the tasks to execute and optimization parameters are changed. Therefore each of the phases can impact another in real time. The change of one phase changes another. Some unexpected situations can also happen which demands to execute some unexpected tasks. The most important issue is how to compare the results if every can demand different parameters to optimize or optimizing parameters can change during the process. Every solution can be characterized by global common parameters which can be used to compare the results. Such parameters can be for example cost of the solution or time of execution of all the tasks.

Such a problem occurs in embedded system design. If system meets unexpected situation it can be solved on many ways. Each way demands different tasks to execute. The problem is to find the optimal way to execute unexpected tasks and to find the appropriate hardware components to execute them. The solutions can be compared using two global parameters: time and cost of execution of all the tasks. As it can be easily observed, if the time is getting lower the cost is rising. Such relation is not proportional. Therefore, designing of embedded systems belongs to pareto group of problems [15].

III. The Algorithm

Unexpected tasks can appear in every moment of system life, after every task. After appearing of such a task it needs to be inserted on extended task graph as a separate task. Then all the tasks need to be split on possible number of subtasks. The algorithm starts with generating the initial population. The number of individuals in population is dependent on

a number of programmable processors p and a number of tasks n in the extended task graph. It is equal to:

$$\Pi = \alpha * p * n \quad (3)$$

where α is given by a designer. It controls the size of the population. In each of the individuals unexpected tasks are solved as a random connection of randomly chosen number of subtasks. Not every connection gives the solution of unexpected situation. Such solutions are not passed over. Next generations of individuals are created using standard genetic operators: crossover, mutation, cloning and selection. In this paper we decided to choose rank selection. After generating each population the genotypes are ranked by cost. All of the individuals on a rank list have probability of being chosen during the evolution process. The probability P depends on a position r of an individual in a rank list. It is described by the following equation:

$$P = \frac{\Pi - r}{\Pi} \quad (4)$$

Crossover selects Ψ individuals and randomly connects them in pairs. Then for each genotype in each pair randomly a cutting point is chosen. The genes of two parents are swapped. The number of individuals created by crossover is presented on equation 4 below:

$$\Psi = \gamma * \Pi \quad (5)$$

where γ is a parameter given by the designer, γ ϵ (0,1).

Mutation selects Ω genotypes. Then randomly a gene is chosen. The number of a PE in the gene is substituted by another. The mutation can also change the connection of subtasks solving unexpected situation. The number of individuals created using mutation operator is equal to:

$$\Omega = \beta * \Pi \quad (6)$$

where β ϵ (0,1) and is given by a designer.

Cloning copies Φ individuals to a new population without any changes. Φ is equal to:

$$\Phi = \delta * \Pi \quad (7)$$

where δ ϵ (0,1). It is given by a designer.

To have the same number of individuals in all of the populations the sum of the parameters β, γ and δ needs to be equal to 1:

$$\beta + \gamma + \delta = 1 \quad (8)$$

The algorithm finishes its execution after ε generations without a better result.

IV. Experimental results

In this section the results of the experiments are presented. The results were compared with genetic programming methodology [13] proposed by Górski and Ogorzałek (GP 2025). Table 1 contains the results. The results were made for benchmarks with 10, 20 and 30 nodes. The parameters were set as follows for both of algorithms: α = 100, γ = 0,7, β = 0,2, δ = 0,1 and ε = 5. The first results seem promising. However it is needed to underline that presented results are first obtained and the algorithm needs further investigation with different parameters, time constrains and bigger graphs.

Algorithm presented in this paper (GA 2025) was able to provide better results for every benchmarks. For the graph with ten nodes the difference between the best results obtained by GP 2025 and GA 2025 was the lowest – it was equal only 15 units of cost. The cost of the best solutions generated by both algorithms was the same and equal to 100. That could be an effect of a small size of the graph. For a such a graph the search space is smaller and maybe it could be reasonable to decrease the value of α parameter. The difference between obtained values of cost for a graph with 20 nodes was the greatest – more than 700 cost units (1643 for GA 2025 and 2358 for GP 2025). Such a difference is surprising however we cannot forgot about probabilistic nature of the algorithms. As a consequence of such a type of algorithms one or a few results can be very different from the majority. For a graph with 30 nodes there was not such a big difference of costs – it was appropriately 1643 for GA 2025 and 2358 for GP 2025. It is worth to mention that even that GP 2025 produced the results which were more expansive the time of the solutions was faster in most of cases than the time of results generated by GA 2025. Such a situation was expected because, as it was mentioned before, investigated problem in hardware design belongs to pareto group of problems. It also can be observed that GA 2025 generated less populations than GP 2025 – 17, 14 and 18 for graphs with 10, 20 and 30 nodes meanwhile GP 2025 produced appropriately 22, 15 and 23 generations.
graph.

Table I.
Experimental results

Graph	GA 2025			GP 2025		
	cost	time	generation	cost	time	generation
10	197	100	17	212	100	22
20	1643	2497	14	2358	2394	15
30	1997	2956	18	2244	2873	23

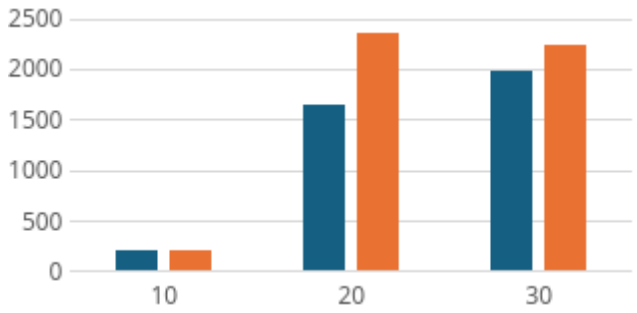


Fig 2. Comparison of obtained results

In fig. 2 the graphical comparison of the results were presented. It contains best obtained results for every used benchmark. Algorithm GP 2025 was compared with GP 2024 [8] and was more efficient.

V. Conclusions

In this paper a constructive genetic algorithm for a concurrent real-time optimization in embedded systems design process was proposed. Solving the problem in hardware design can make the design faster and cheaper. It can also help with adapting embedded systems to a changing environment and thus making the systems more universal.

In the paper only first results were presented. Therefore the algorithm needs more examination. The experiments should be made using bigger graphs, different parameters and time constrains.

In the future we plan to deliver more algorithms to solve the problem investigated in hardware design. It seems that good direction is to develop genetic programming solutions. We also plan to propose new solutions to concurrent real-time optimization problem in other areas too, not only in embedded system design. Therefore the future work will be divided into two directions. The first direction will include improvement of proposed algorithms for hardware design. The second direction will be concentrated on investigated problem – its constrains, areas of appearance and searching its different solutions.

Acknowledgment

The publication has been supported by a grant “Solving real-time optimization problems” from the Faculty of Physics, Astronomy and Applied Computer Science under the Strategic Programme Excellence Initiative at Jagiellonian University.

References

[1] G. C. Duarte, D. S. Loubach and I. Sander, "High-Level Reconfigurable Embedded System Design Based on Heterogeneous Models of Computation," in *IEEE Access*, vol. 13, 2025, pp. 63918-63934,

[2] A. Górski and M., J. Ogorzałek, “Assignment of unexpected tasks in embedded system design process”. *Microprocessors and Microsystems*, Elsevier vol. 44, 2016 pp. 17– 21.

[3] A. Górski and M., J. Ogorzałek, “Auto-detection and assignment of unexpected tasks in embedded systems design process*”. Proceedings of the 23rd International Workshop of the European Group for Intelligent Computing in Engineering*, 2016, pages 179 – 188.

[4] A. Górski and M., J. Ogorzałek, “Assignment of unexpected tasks for a group of embedded systems. *IFAC-PapersOnLine*, vol. 51, Issue 6, 2018, pp. 102-106.

[5] A. Górski and M., J. Ogorzałek, “Assignment of unexpected tasks in embedded system design process using genetic programming”. *Proceedings of the 6th International Conference on the Dynamics of Information Systems (DIS 2023)*, Lecture Notes in Computer Science, vol. 14321, Springer, Cham., 2024, pp 93 – 101.

[6] A. Górski and M., J. Ogorzałek, “Concurrent real-time optimization in embedded system design process using genetic algorithm”. *Progress in Polish Artificial Intelligence Research 5: proceedings of the 5th Polish Conference on Artificial Intelligence* (PP-RAI’2024), 2024, pp. 331-337.

[7] A. Górski and M., J. Ogorzałek, “Concurrent Real-Time optimization of detecting unexpected tasks in IoT design process using GA”. *Late Breaking Papers from the IEEE 2023 Congress on Evolutionary Computation*, Chicago, IL, USA, IEEE, 2023, pp. 74 – 77.

[8] A. Górski and M., J. Ogorzałek, “Detecting and assignment of unexpected tasks in SoC design process using genetic programming”. *Proceedings of the 21st International SoC Design Conference (ISOCC),* Sapporo, Japan, august 2024 pp. 398-399.

[9] B. P. Dave, G. Lakshminarayana and N. K. Jha, "COSYN: Hardware-software co-synthesis of heterogeneous distributed embedded systems," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, March 1999, pp. 92-104.

[10] S. Q. Liu and E. Kozan, “A hybrid metaheuristic algorithm to optimise a real-world robotic cell”, *Computers & Operations Research.* Vol. 84, Elsevier, 2017, pp. 188-194.

[11] J. A. Austin, M. A. Barras, and C. M. Sullivan. "Safe and effective digital anticoagulation: a continuous iterative improvement approach." *ACI open* 5.02 (2021), pp. e116-e124.

[12] H. Lu, X. Zhang, S. Yang, A, “earning-based iterative method for solving vehicle routing problems” *International Conference on learning Representations*, 2020.

[13] A. M. Górski and M., J. Ogorzałek, “Genetic programming iterative improvement algorithm for a concurrent real-time optimization in embedded system design process” *Proceedings of the 6th Polish Conference on Artificial Intelligence* (PP-RAI’2025), 2025 (in press).

[14] A. M. Górski and M., J. Ogorzałek, “Grey wolf optimization algorithm for a concurrent real-time optimization problem in game theory” *Proc. Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 19 no. 2, 2025, pp. 65-72.

[15] S. Mahajan, A. Chauhan, and S. K. Gupta “On Pareto optimality using novel goal programming approach for fully intuitionistic fuzzy multi-objective quadratic problems”. *Expert Systems with Applications*, vol. 243, Elsevier, 2024.

[16] K. Gmyrek, M. Antkiewicz and P. Myszkowski ” Genetic Algorithm for Planning and Scheduling Problem -- StarCraft II Build Order case study*” Proceedings of the 18th Conference on Computer Science and Intelligence Systems, Annals of Computer Science and Information Systems*, vol. 35, 0223 pp. 131–140.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 41–48
DOI: 10.15439/2025F7399
ISSN 2300-5963 ACSIS, Vol. 44

RAG^4-Unet: An Approach for Recognition and Segmentation of Brain Tumor in MRI Scans

Ameer Hamza
Centre of Real Time Computer Systems
Kaunas University of Technology
Kaunas, Lithuania
ameer.hamza@ktu.edu

Robertas Damaševičius
Centre of Real Time Computer Systems
Kaunas University of Technology
Kaunas, Lithuania
robertas.damasevicius@ktu.lt

***Abstract*—We propose a novel U-net architecture, RAG^4-Unet, based on residual attention gated for brain tumor segmentation, Swin transformer for classification task, and Yolo11 for tumor detection. For the experiments, the Figshare dataset is employed and the proposed architecture achieved 91.37% Dice for tumor segmentation task, and Swin transformer achieved 91.74% classification accuracy. The Yolo11 gained 89.6% of detection precision. Comparative evaluation with the SOTA techniques reveals that the proposed architecture outperformed the existing methods and Yolo11. The proposed architecture improved the tumor boundary detection, making it a promising solution for brain tumor recognition and segmentation.**

***Index Terms*—Tumor Segmentation, Residual Attention Gated, Unet, Yolo11, Attention Maps.**

I. Introduction

BRAIN tumors are a major health challenge, characterized by abnormal cell growth in the brain, which can affect its vital functions [1]. These tumors, from benign to malignant, are often associated with persistent headaches, seizures, cognitive impairments, and neurological diseases and have a negative impact on the quality of life of patients [2]. Manual diagnostic methods such as visual inspection of histological slides and radiological imaging have traditionally been used, but they take time, are subjective, and can cause human error [3]. Radiologists are imaging modalities more frequently because they tend to be more accurate and put patients at far lower risk. Medical imaging data can be recorded using a variety of techniques, such as tomography [4], magnetic resonance imaging (MRI) [5], radiography [6], and echocardiography [7].

The introduction of artificial intelligence (AI) has transformed the detection of brain tumors through automation and improved diagnostic accuracy [8]. Machine Learning ML) approaches depend on methods for gathering features, selecting features, and classification [9]. Deep Learning (DL) models learn by extracting features from images. Particularly, Convolutional Neural Networks (CNNs) are widely used in medical imaging analysis and show vital achievements in the identification of brain tumors, enabling advances in classification and segmentation. Several studies have proposed innovative methods for the segmentation and classification of brain tumors from MRI images. Zhang et al. [10] introduced a modified U-net method with an attention mechanism for improving segmentation accuracy. Their approach focused on addressing limitations of traditional U-net models, such as difficulties in handling small tumor regions and blurry tumor boundaries. By incorporating multi-scale feature fusion and attention mechanisms, their method demonstrated enhanced efficiency and achieved Dice coefficients of 0.876, 0.868, and 0.814 for tumor subregions.

Ahsan et al. [11] compared object detection algorithms (YOLOv5, Faster R-CNN, SSD) for brain tumor. They used Figshare dataset and paired YOLOv5 with 2D U-Net for segmentation. Yolov5 gained the highest mAP of 89.5%, and Yolov5+2D U-Net achieved 88.1% DSC. However, the dual-model framework increased learning complexity.

Arumaiththurai et al. [12] proposed two methods for classifying brain tumors using ML and DL algorithms. The first method used decision trees and SVM, while the second used pre-trained VGG19 and ResNet152 models. Figshare brain tumor dataset assessed the effectiveness of these approaches. The CNN-based method performed better in classification and attained an accuracy rate of 94.67%.

Alyami et al. [13] employed AlexNet and VGG19 models for feature extraction and the slap swarm algorithm for feature selection. They used Kaggle brain tumor dataset and achieved an accuracy of 99.1% with a cubic SVM using 4111 best selected features out of 8192.

Asiri et al. [14] introduced a customized CNN model for classification brain tumor, focusing on hyperparameter tuning of kernel size, strides, activation, and learning rates. The model was evaluated on two MRI datasets: a four-class dataset with 7,023 images and a binary dataset with 253 images. This method achieved 88% accuracy.

These studies demonstrate the growing use of deep learning models, particularly U-net, transfer learning, and attention mechanisms, to enhance the accuracy and efficiency of brain tumor segmentation and classification. The incorporation of explainable AI such as LIME, attention maps, also adds a layer of transparency, which is crucial for the deployment of these models in clinical settings.

However, challenges such as variability in growth patterns, textures, and irregularity in tumors across patients, and different tumors have overlapping visual features and irregular

Topical area: Advanced Artificial Intelligence in Applications

boundaries, especially when the tumors are in early stages. Addressing these challenges remains a critical focus of ongoing research on brain tumor analysis.

To address these challenges, we introduce an innovative architecture, RAG^4-Unet, for the segmentation process, and this framework incorporates Swin transformer and Yolo11 for precision detection of the tumor region. The key contributions of this work is summarized as follows:

- We introduce a novel Residual Attention Gated (RAG) module to focus on significant spatial and contextual features to enhance detection of brain tumor boundaries.
- We employ a Swin Transformer to leverage shifting window sizes, utilizing its attention mechanism to learn features hierarchically.
- We integrate YOLO11 for detection of growth regions in brain tumors, enhancing accuracy of tumor detection.

II. Methodology

A. Data Collection and Augmentation

The FigShare Brain tumor dataset is utilized for experiments. The dataset is available at https://Figshare.com/articles/dataset/brain_tumor_dataset/1512427. This dataset includes 233 patients with three types of tumors: glioma, meningioma, and pituitary tumor. The glioma category contains 1426 slices, meningioma has 708 slices, and the pituitary tumor has 930 slices. Each image has a dimension of 512×512 with a depth resolution of 96 dpi. The dataset is imbalanced and that there were not enough samples for the efficient learning of the deep learning model. Therefore, we performed an augmentation process to increase the diversity in the dataset. For the augmentation process, four basic transformations are utilized: horizontal flip, rotation by 10°, vertical flip, and solarization. After augmentation process, the samples in each class are 4120. The augmentation process is visually presented in Fig. 1.

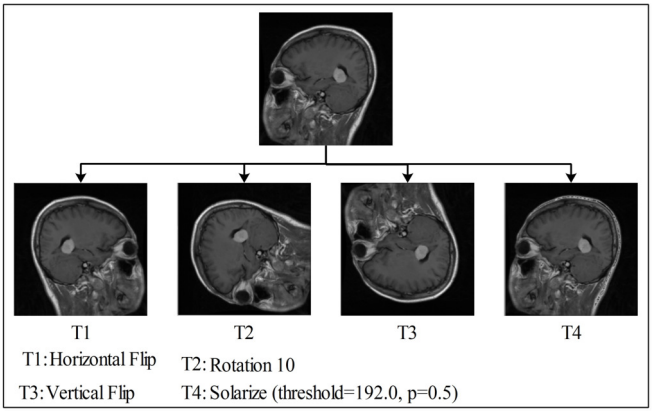


Fig. 1. Sample of augmentation operation on brain tumor dataset.

B. Overview of Swin Transformer

Swin Transformer is an enhanced version of the transformer that boosts computational effectiveness and capacity for high-resolution images. Similar to conventional CNNs, the Swin Transformer gradually reduces the image size by introducing a hierarchical structure that reflects images at various sizes. It limits attention to small windows and shifts these windows at every level. The input RGB image is separated into non-overlapping patches using a patch-splitting module like ViT. Each patch is handled as a "token" and its feature is configured as a concatenation of raw pixel RGB values. After that, many Swin Transformer blocks are applied to these patch tokens.

a) Swin Transformer Stages:

The Swin Transformer block, known as "Stage 1," maintains a token count of $\frac{\phi_h}{4} \times \frac{\phi_w}{4}$ when used with linear embedding. Hierarchical representation is achieved by reducing the number of tokens using a patch merging technique as the depth of the neural network grows. The initial patch merging layer concatenates the features of neighboring 2×2 patches and then applies a linear layer to the $4C$-dimensional features produced. This procedure reduces the token count by a factor of $2 \times 2 = 4$, while changing the output dimension to $2C$. Swin Transformer blocks are added to transform features while keeping a resolution of $\frac{\phi_h}{8} \times \frac{\phi_w}{8}$. Stage 2 begins with patch merging and feature transition. The procedure is performed twice, resulting in "Stage 3" and "Stage 4", with output resolutions of $\frac{\phi_h}{16} \times \frac{\phi_w}{16}$ and $\frac{\phi_h}{32} \times \frac{\phi_w}{32}$ correspondingly. The four stages work together to provide a hierarchical representation with feature map resolutions equivalent to those of typical CNNs. Swin Transformer replaces multi-head self-attention (MSA) module in a transformer block with a module based on the shifted window, while the other layers remain unchanged. The Swin Transformer block consists of a shifted window-based MSA module, a 2-layer MLP with GELU activation in between. The layer normalization is placed before each MSA and MLP module, followed by a residual connection.

b) Hierarchical Feature Learning:

The self-attention within localized windows enables effective modeling. The windows are positioned such that they do not overlap and divide the image equally. The computational complexity of a global MSA module and a window-based one, based on an image of $\phi_h \times \phi_w$ patches, assuming each window has $k \times k$ patches:

$$\Omega(MSA) = 4\phi_h\phi_w C^2 + 2(\phi_h\phi_w)^2 C \tag{1}$$

$$\Omega(MSA)_w = 4\phi_h\phi_w C^2 + 2k^2\phi_h\phi_w C \tag{2}$$

The computation of the Swin Transformer has linear complexity when fixed, but the computational cost of traditional ViT increases quadratically with the number of patches. Although the W-MSA of the Swin Transformer decreases the computational cost from quadratic to linear, its modeling capability may be limited by the absence of links and communication between many windows. To overcome this restriction, the Swin Transformer adds a shifted window divider that makes it easier for nearby non-overlapping windows to share information. In two successive Swin Transformer blocks, this method alternates between using W-MSA and a modified SW-MSA. By connecting adjacent non-overlapping windows, the shifted window partitioning greatly expands the receptive field. After

employing the shifted window divider, the computation within two consecutive Swin Transformers is followed as:

$$\hat{\Phi}^b = (MSA)_w(\text{LN}(\Phi^{b-1})) + \Phi^{b-1} \tag{3}$$

$$\Phi^b = \text{MLP}(\text{LN}(\hat{\Phi}^b)) + \hat{\Phi}^b \tag{4}$$

$$\hat{\Phi}^{b+1} = (MSA)_{sw}(\text{LN}(\Phi^b)) + \Phi^b \tag{5}$$

$$\Phi^{b+1} = \text{MLP}(\text{LN}(\hat{\Phi}^{b+1})) + \hat{\Phi}^{b+1} \tag{6}$$

Where $(MSA)_w$ and $(MSA)_{sw}$ represent window-based multi-head self-attention and shifted window divider, respectively. $\hat{\Phi}^b$ and Φ^b denote resultant features of the $(MSA)_{sw}$ and MLP module for block b.

Swin Transformer introduces the relative position biases for every head during the similarity calculation, which is formulated as:

$$Att(Q, K, V) = \text{Soft}\left(\frac{QK^T}{\sqrt{d}} + \psi_b\right)V \tag{7}$$

Where Q, K, V are the query, key, and value vectors, and d denotes the dimension of Q, K, V, and ψ_b is the bias vector.

The motivation behind choosing the Swin Transformer for brain tumor analysis is its ability to process high-resolution images and its window-based attention mechanism, which can learn fine-grained details about the tumor region, such as tiny tumor boundaries and growth patterns in the local context.

C. Proposed RAG^4-Unet Architecture

U-Net is a deep learning architecture proposed for image segmentation. It consists of three steps: encoder, bridge, and decoder. In this work, we proposed a Residual Attention-Gated U-Net (RAG^4-Unet) for the segmentation of tumors from brain MRI scans. RAG^4-Unet consists of four residual encoders, one bridge, and four residual decoders. All the residual encoders are connected to the attention gate to generate the attention maps, and the attention gates are concatenated with the decoders. The RAG^4-Unet accepts the input of size $256\times256\times3$.

a) Encoder Phase:

The first encoder consists of one residual block, max-pooling with stride 2, and one dropout layer. The dropout factor is 0.1. The residual block contains two convolutional layers with a 3×3 filter size, 64 filters, and a stride of 1. The residual encoder is desribed as follows:

$$\partial_1 = \emptyset_1(I) \tag{8}$$

$$\partial_1^{\psi} = \psi_1(\partial_1) \tag{9}$$

$$\partial_2 = \emptyset_2(\partial_1^{\psi}) \tag{10}$$

$$\partial_2^{\psi} = \psi_2(\partial_2) \tag{11}$$

$$\partial_3 = \emptyset_3(\partial_2^{\psi}) \tag{12}$$

$$\partial_3^{\psi} = \psi_3(\partial_3) \tag{13}$$

$$\partial_{\text{skip}} = \partial_3^{\psi} + \partial_2^{\psi} \tag{14}$$

$$\partial_{\text{ReLU}} = \lambda(\partial_{\text{skip}}) \tag{15}$$

$$\partial_{\boxplus} = \boxplus_{\text{Mpool}}(\partial_{\text{ReLU}}, s = 2) \tag{16}$$

$$\partial_{\boxminus} = \boxminus_{\text{drop}}(\partial_{\boxplus}, f = 0.1) \tag{17}$$

Where the $\emptyset_c$ represents the convolutional operation, ψ is the batch normalization, λ represents the ReLU activation, $\boxplus_{\text{Mpool}}$ is max pooling, and $\partial_{\boxminus}$ represents the dropout layer. The second and third encoders also consist of one residual block, max-pooling with stride 2, and one dropout layer with a 0.1 dropout factor. In the second residual block, the convolutional layer is configured with a 3×3 filter size, 128 filters, and a stride 1. In the third residual block, the convolutional operation is performed by employing a 3×3 kernel size, 256 filters, and a stride 1. In the last encoder, dropout factor is 0.2, and convolutional inside the fourth residual block is configured with a 1×1 kernel size, 512 filters, and a stride 1.

b) Bridge Phase:

The bridge between the encoder and decoder is the deepest point in the network. The bridge is configured by employing a residual block with 1024 depth and one dropout layer with a 0.3 drop factor.

c) Decoder Phase:

After the Bridge, the first decoder consists of one transpose convolutional layer configured with a 2×2 filter size, 512 depth, and 2×2 stride, one attention gate that is applied on the fourth encoder and transpose layer. The resultant feature map of attention-gated and transpose layers is further combined using the concatenation layer. After that, one residual block and dropout layer with a 0.2 drop factor is employed. The mathematical representation is:

$$\partial_{\text{Tconv}} = \emptyset^T(\beta, k = 2, s = 2, \text{ch} = 512) \tag{18}$$

$$\partial_{\text{AG}} = \text{AttGate}(\partial_{\text{end4}}, \partial_{\text{Tconv}}) \tag{19}$$

$$\partial_{\text{Con}} = \biguplus(\partial_{\text{AG}}, \partial_{\text{Tconv}}) \tag{20}$$

$$\partial_{d1} = \emptyset(\partial_{\text{Con}}) \tag{21}$$

$$\partial_{d1}^{\psi} = \psi_{d1}(\partial_{d1}) \tag{22}$$

$$\partial_{d2} = \emptyset_{d2}(\partial_{d1}^{\psi}) \tag{23}$$

$$\partial_{d2}^{\psi} = \psi_{d2}(\partial_{d2}) \tag{24}$$

$$\partial_{d3} = \emptyset_{d3}(\partial_{d2}^{\psi}) \tag{25}$$

$$\partial_{d3}^{\psi} = \psi_{d3}(\partial_{d3}) \tag{26}$$

$$\partial_{\text{skip}}^{d} = \partial_{d3}^{\psi} + \partial_{d2}^{\psi} \tag{27}$$

$$\partial_{\lambda}^{d} = \lambda(\partial_{\text{skip}}^{d}) \tag{28}$$

$$\partial_{\boxminus}^{d} = \boxminus_{drop}(\partial_{\lambda}^{d}, f = 2) \tag{29}$$

Where $\biguplus$ is the concatenation layer, AttGate is the attention mechanism, and $\emptyset^T$ represents the transpose convolutional operation. In the second decoder, the transpose convolutional has a 2×2 filter size, 256 depth, and 2×2 stride, and the remaining mechanisms are the same. The implementation phenomena of the third and fourth decoders are the same. However, the configurations of transpose convolutional and

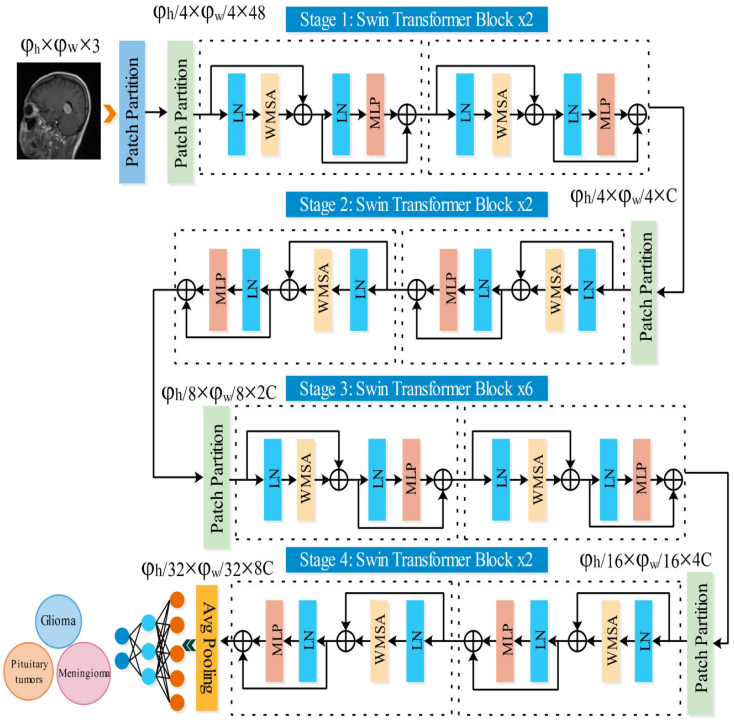


Fig. 2. Architecture of Swin Transformer for the classification of brain tumor

dropout layers are updated. The updated configurations are a 2×2 filter size, 128, 64 depth, and 2×2 stride, and the dropout factor is 0.1, respectively. The architecture of the proposed RAG^4-Unet is presented in Fig. 3.

The proposed model is developed using the TensorFlow framework and The proposed model has 99.45M parameters bringing the model size to around 379.37 MB of memory. 33.14M are trainable and 17.66K non-trainable parameters kept by the optimizer in memory 252.87 MB while training. The model inference complexity is evaluated with GFLOPS. The overall computation cost of a single forward pass is approximately 106.25 GFLOPs.

D. Novelty: Proposed RAG Module

In this work, We designed a novel hybrid feature enhancement module based on Residual and Attention gated mechanism. This module synergistically combines the residual learning to stabilizes the gradient flow, with attention gating, which focused on salient regions of the interest within the brain MRI image.The tumor regions often confused with healthy tissues. the RAG module addresses this problem by filtering irrelevant and low importance features while enhancing the high relevance activations related to tumor boundaries and cores. It enhances the boundary detail and localization of tumor objects, because the attention gated mechanism reduces irrelevant activations that strengthen the task of boundary detail and out-of-distribution activations that strengthens spatial detail of the tumor region when propagating features and helping to ensure gradient stability.

The sequence of RAG module begins with a series of convolutions to extract features from the input tensor, with the output then entering more convolutions and subsequently Attention Gated module, when performing spatial attention analysis, attention maps are created using extracted features and a gating signal is produced from a feature map. The attention maps are resampled, and modify the original feature map, it allows the network to learn where to increase and where to decrease specific spatial regions and a residual connection allows the network to skip non-linearity,if needed, thus minimizing the possibility of vanishing gradients strengthening the source of information and allowing for richer contextual experience for the network over numerous forward passes. The output of this module contains local enriched features and global semantic guiding features useful for precise identification of the tumor edges. the proposed RAG module is presented in Figure 4

III. Results

A. Experimental Setup

The dataset is divided into training, testing, and validation. 70% data is employed for training, 10% data is employed for the validation during the training process, and the 20% data is utilized for the testing process. The hyperparameters selected for Swin Transformer are batch size, number of workers, selected optimizer ADAM, learning rate, momentum, and epochs having values are 8, 4, 0.0004, 0.9, and 250. For RAG^4-Unet the utilized hyperparameters are learning rate is 0.0001, epochs is 100, optimizer is ADAM, batch size is 8,

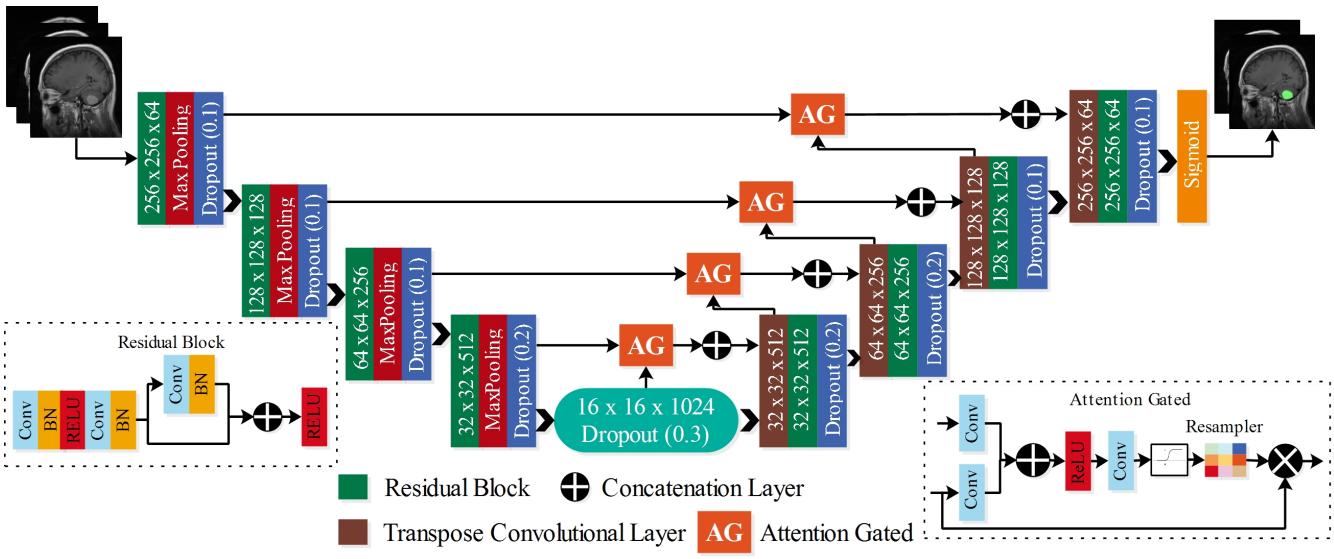


Fig. 3. Architecture of proposed RAG^4-Unet for brain tumor segmentation

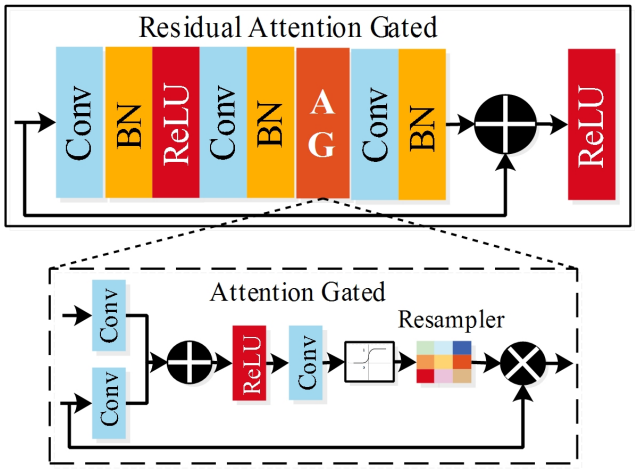


Fig. 4. Architecture of Residual Attention Gated mechanism (RAG) module

and early stopping is employed with learning decay is 0.2, patience is 5, min_lr is 0.00001. The evaluation metrics are accuracy, recall, precision, f1-score, Dice, Jaccard loss, and IoU for the segmentation and classification.

The experiments are conducted on MSI GL75 Leopard model configured with Core–i7 10 generation 2.59GHz processor, 16 GB of RAM, 512GB of SSD storage, and GTX GeForce 1660ti 6GB graphics card.

B. Results of Swin Transformer

The classification results of Swin Transformer on Figshare dataset has been presented in Table I. The model achieved 91.74% accuracy, 91.64% precision, 91.73% recall, 91.52% f1-score, 98.33% AUC, and 86.91% kappa index. The performance across the individual classes such as pituitary tumor gained the highest accuracy of 97.85%, precision of 95.13%, recall of 97.85%, and f1-score of 96.48% with 2.87 (sec) inference time. The confusion matrix gives more comprehensive details about the class's performance, as shown in Fig. 5. Glioma and pituitary tumor have the highest accuracy of 95.79%, and 97.85% respectively, because 205 samples of glioma and 137 samples of pituitary tumor class are correctly classified and 9 samples from the glioma and only 3 samples from the pituitary tumor are misclassified. The meningioma class has 75.47% of accuracy, 88.88% precision, 75.47% recall, and 81.63% f1-score. Meningioma tumor suffers from the considerable misclassification, the 5 samples are incorrect classified as pituitary tumor and 21 samples are misclassified as glioma. The overall misclassification rate of the meningioma class is higher than the other two classes. The overall confidence index of model is quite better which is 97.22%.

TABLE I
RESULTS OF SWIN TRANSFORMER ON FIGSHARE DATASET

Class-wise	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Glioma	95.794	90.707	95.794	93.181
Meningioma	75.471	88.888	75.471	81.632
Pituitary Tumor	97.857	95.138	97.857	96.478
Overall Performance				
Accuracy (%)	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**AUC (%)**
91.74	91.64	91.73	91.52	98.33
Kappa (%)	**CI**	**Inference Time (sec)**		
86.91	97.22	2.306		

C. Results of proposed RAG^4-Unet

The segmentation is implemented using the proposed RAG^4-Unet model. The images and their masks are provided as input to the proposed model. After training the RAG^4-Unet model, the model is evaluated on the test data. The

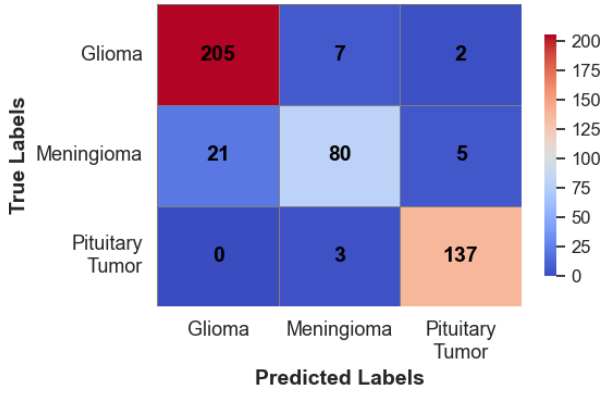


Fig. 5. Confusion matrix of Swin Transformer on Figshare dataset

overall performance and sample-wise results of the proposed RAG^4-Unet are presented in Table II. The proposed model achieved 91.37% Dice, 94.74% precision, 96.23% sensitivity, and 98.46% specificity. Some of the testing sample results are presented in Table II. Most test samples have a high Dice of 0.90, with consistent IoU and low Jaccard loss. The model could segment the tumor regions accurately and clearly distinguish the tumor portion from the surrounding information, such as samples 1, 2, 3, 4, 7, 8, 9, and 12 have more than 90% Dice score, 87-94% IoU, due to the clear and uniform morphology of the tumor region and a few samples, like 10, 11, 13, and 14, have quite better Dice scores and IoU with the small size of the tumor. However, the model is struggling with samples that do not clear the tumor boundary because the results are leading to under or over-segmentation, like in samples 5 and 6.

TABLE II
SEGMENTATION RESULTS OF PROPOSED RAG^4-UNET BASED ON FIGSHARE DATASET

Sr.	**Dice**	**Jaccard Loss**	**IoU**	**Sr.**	**Dice**	**Jaccard Loss**	**IoU**
1	0.906	0.171	0.828	2	0.931	0.127	0.872
3	0.943	0.106	0.893	4	0.948	0.097	0.902
5	0.649	0.519	0.481	6	0.782	0.357	0.642
7	0.971	0.055	0.944	8	0.945	0.10	0.896
9	0.950	0.094	0.905	10	0.957	0.081	0.918
11	0.960	0.076	0.923	12	0.971	0.054	0.945
13	0.929	0.131	0.868	14	0.970	0.056	0.943
Overall Performance							
Dice	**Dice Loss**	**Precision**	**Sensitivity**	**Specificity**			
0.9137	0.0863	0.9474	0.9623	0.9846			

Fig. 6 presents a visual comparison of the original ground truth and predicted ground truth with the overlap maps for further investigation of the above samples. In overlapping maps, the green region indicates the original mask, the red region demonstrates the predicted mask, and the yellow region indicates the perfect match between the predicted and original masks. In addition, in the last column of Fig. 6, the attention maps are generated by the proposed model to further evaluate the transparency. The generated attention maps highlighted the focus of the RAG^4-Unet during the segmentation process. These maps show the areas of the segmentation process where the model concentrates. The tumor locations are prominently highlighted in the attention maps, signifying that the model effectively suppresses background noise and prioritizes relevant areas. For samples 5 and 6, as shown in Fig. 6, the model generated a weaker or scattered focus, which indicated low performance. For the overall performance, generating attention maps are a suitable instrument for interpreting the decision-making process of the model.

D. Results of Yolo11 model

In this section, the Yolo11 model is implemented for the detection of tumor region from the brain MRI and the metrics are presented in Table III. The Yolo11 model achieved 89.6% boundary box precision, demonstrates that the model has high rate of correct detections with the less false positive. While, the recall box is 87.4% indicates that the model has quite number of missed detections and the mAP50 and mAP50-95 are 86.4% and 81.76% respectively. The inference time is also measure for the Yolo11 which is 1.3 (sec), reveals that the model is fast and responsive. The fitness score which is 0.7443 exposes the balance among the accuracy and computational cost. The overall pre and post processing of Yolo11 is 9.5 and 18.6265 (sec) respectively, reflecting the computation strength to arranged the brain MRI for detection. Table III also presents the speed, preprocessing, inference time, and confidence of the few individual cases. The each individual case the preprocessing time is lies between the 7.8 to 11.3 (sec) and the 1.3 (sec) is need for all the most cases. Few of samples such as 1,4,5, and 14 have high confidence scores which is 0.88, 0.96, 0.91, and 0.90, respectively, highlighting the effective predictions with tumor localization while the sample 3 achieved the confidence score of 0.00 which indicates the complete failure of detection of tumor region. Similarly, the samples that have overlapping visual features tend to results in low confidence score.

Fig. 6 shows visual comparison results between the Yolo11 detection and proposed RAG^4-Unet model. In this figure, Yolo11 model fails to align with the tumor boundaries, evidently, clearly visual in samples such as 5,6,and 9 and in some samples the boundary boxes has missed of the tumor and include non-tumor regions, indicating that the model faces the challenges when tumor has complex and irregular shape. In contrast, the proposed RAG^4-Unet segmentation maps indicating the higher boundary alignment. The segmented region by the RAG^4-Unet model are more closely to the original ground truth. In addition, the proposed model provides a detailed representation of tumor boundaries that Yolo11 boundary boxes cannot match and the Yolo11 is unable in detecting and localizing the tumor regions with the high precision such as in sample 3 and 10, the Yolo11 model missed and incorrectly detect the tumor region.

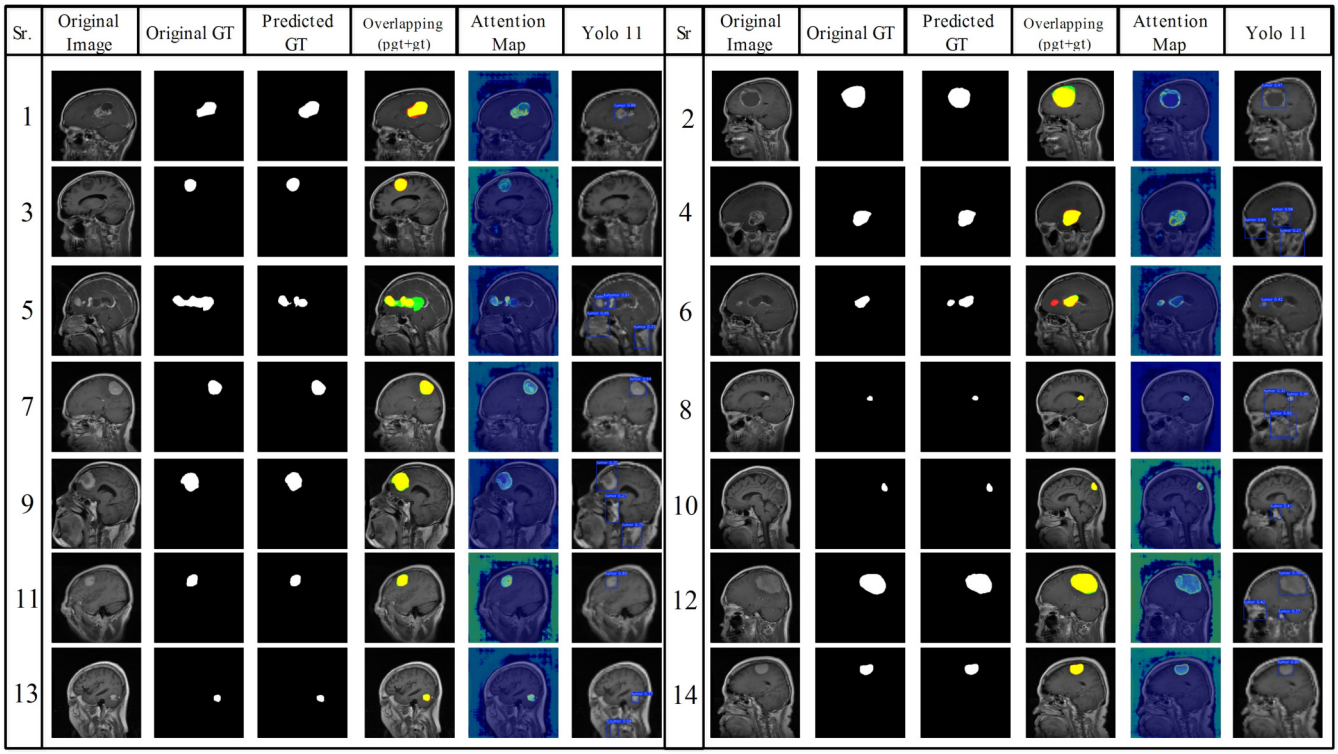


Fig. 6. Segmentation visualizations of predicted mask, overlapping maps, attention maps, and yolo 11 detection for analyzing the RAG^4-Unet

TABLE III
DETECTION RESULTS OF YOLO11 MODEL ON FIGSHARE DATASET

Sr.	Speed (ms)	Preprocess (ms)	Inference (ms)	Confidence (%)
1	2.4	7.8	1.2	0.88
2	2.5	8.7	1.3	0.47
3	2.7	10.4	0.6	0.00
4	2.5	9.8	1.5	0.96
5	2.9	10.6	1.3	0.91
6	2.6	8.9	1.4	0.42
7	2.6	9.8	1.3	0.64
8	2.7	8.2	1.2	0.60
9	2.7	9.5	1.3	0.30
10	2.6	10.0	1.4	0.41
11	2.7	10.7	1.3	0.80
12	2.7	9.5	1.3	0.78
13	2.6	11.3	1.3	0.77
14	2.5	9.3	1.4	0.90
Overall Performance				
Precision(B)	**Recall(B)**	**mAP50(B)**	**mAP50-95(B)**	
0.896	0.876	0.864	0.8176	
Preprocessing	**Inference**	**Fitness**	**Post process**	
0.494	4.260	0.7443	18.6265	

E. Comparison with SOTA

The comprehensive comparison has been conducted between the proposed and state-of-the-art methods, as shown in Table IV. Authors in [11] employed U-net architecture for the segmentation and conducted experiments on Figshare dataset. They achieved 88.1% of accuracy. In [14], the authors proposed customized CNN for the classification of tumor types using the Figshare dataset and they achieved 88% accuracy. Authors in [15] implemented ResNet50 model using deep transfer learning method on private dataset and they gained 90% of accuracy. In [16], the authors employed semi deep learning framework based on customized Unet and histogram features. The performed experiments on BITE dataset and they achieved 91%. However, our proposed methods achieved the highest accuracy of 91.74% using the swin transformer and 91.37% Dice score using proposed RAG^4-Unet in segmentation task.

TABLE IV
COMPREHENSIVE COMPARISON BETWEEN THE PROPOSED FRAMEWORK AND STATE-OF-THE-ART METHOD

Ref	Year	Dataset	Methodology	Accuracy
Ahsan et al. [11]	2024	Figshare	Unet architecture	88.1%
Asiri et al. [14]	2024	Figshare	Customized CNN	88%
Rajput et al. [15]	2024	Private	ResNet50	90%
Shiny et al. [16]	2024	BITE	Semi Deep learning	91%
Proposed Work		**Figshare**	**Swin Transformer**	**91.74%**
		Figshare	**RAG^4-Unet**	**91.37%**

IV. STATISTICAL ASSESSMENT

In order to fully assess the consistency and reliability of the proposed RAG^4-Unet model, we utilized Z-score method of the Dice similarity produced from the 14 test samples. the z-score for the each dice score is calculated using the equation 38.

$$\partial_z = \frac{d_i - \mu}{\sigma} \tag{30}$$

where d_i represents the Dice score of each sample, μ denotes the mean across all samples, and σ is the standard deviation. The value of the standard deviation is 0.0853 and the mean is 0.9216.

All samples (12 out of 14) showed Z-scores that fell within -1.0 and +1.0 which means that the most of the Dice values are close to the mean and demonstrate consistent segmentation performance across the samples in the test data. Sample 5 with a Dice score of 0.649 produced a Z-score of -3.20 indicating it was a significant outlier case, as shown in Figure 7. This Z-score indicated a material drop in performance relatively speaking for that one case, which could have been due to noise, complicated tumor morphology. Sample 6 had a moderately low Z-score of -1.635 which indicates that it did perform below the mean relative to the remaining samples. Sample 7, Sample 12 and Sample 14 had Z-scores that were above +0.5, indicating those samples performed above and beyond the average segmentation performance.

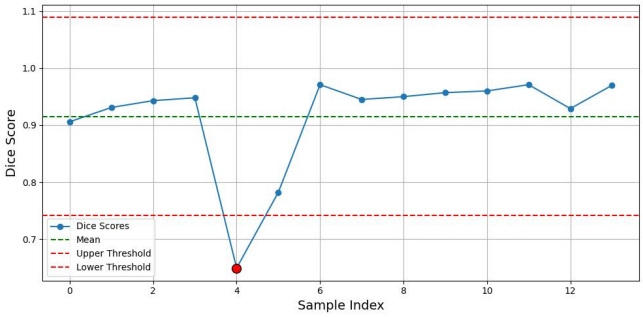


Fig. 7. Z-score Analysis of Dice Coefficients

V. Conclusion

In this work, we proposed a novel RAG^4-Unet architecture for the segmentation task integrated with swin transformer and Yolo11 for the classification and detection task. The proposed RAG^4-Unet architecture addresses the challenges of irregular shapes of boundaries and intersecting visual features of tumors by employing the residual attention gated mechanism. The proposed model achieved 91.73% of Dice coefficient, 94.74% of precision, 96.23% of sensitivity, and 98.46% specificity and swin transformer achieves 91.74% of accuracy, 91.64% of precision, 91.73% of recall, 91.52% of f1-score, 98.33 AUC, 86.91 kappa index, and 97.22% of confidence index with 2.306 (sec) inference time. The Yolo11 model achieves a boundary precision o 86.6%. The limitation of the proposed work is the proposed model goes under segmentation and low Dice when tumor size are small and Yolo11 lead to inaccurate boundary boxes when the tumor are complex.

In future work, we will focus on addressing these limitations using more diverse datasets and we will further explore and refine the attention mechanism to improve the tumor boundary delineation.

Acknowledgements

We acknowledge the support from COST Action "A Comprehensive Network Against Brain Cancer" (Net4Brain - CA22103).

References

[1] T. Rahman, M. S. Islam, and J. Uddin, "Mri-based brain tumor classification using a dilated parallel deep convolutional neural network," *Digital*, vol. 4, no. 3, pp. 529–554, 2024.

[2] H. A. Munira and M. S. Islam, "Hybrid deep learning models for multi-classification of tumour from brain mri," *J Inf Syst Eng Bus Intell*, vol. 8, pp. 162–74, 2022.

[3] N. Elazab, W. A. Gab-Allah, and M. Elmogy, "A multi-class brain tumor grading system based on histopathological images using a hybrid yolo and resnet networks," *Scientific Reports*, vol. 14, no. 1, p. 4584, 2024.

[4] P. Kuppler, P. Strenge, B. Lange, S. Spahr-Hess, W. Draxinger, C. Hagel, D. Theisen-Kunde, R. Brinkmann, R. Huber, V. Tronnier, *et al.*, "Microscope-integrated optical coherence tomography for in vivo human brain tumor detection with artificial intelligence," *Journal of Neurosurgery*, vol. 1, no. aop, pp. 1–9, 2024.

[5] Z. Rasheed, Y.-K. Ma, I. Ullah, M. Al-Khasawneh, S. S. Almutairi, and M. Abohashrh, "Integrating convolutional neural networks with attention mechanisms for magnetic resonance imaging-based classification of brain tumors," *Bioengineering*, vol. 11, no. 7, p. 701, 2024.

[6] S. Saket, Y. Nilipour, R. Taherian, and N. F. Marnaanni, "Evaluation of radiographic, neuropathological, and demographic findings in children aged 1 to 18 years with brain tumor," *Novelty in Biomedicine*, vol. 12, no. 2, pp. 55–59, 2024.

[7] M. S. I. Khan, A. Rahman, T. Debnath, M. R. Karim, M. K. Nasir, S. S. Band, A. Mosavi, and I. Dehzangi, "Accurate brain tumor detection using deep convolutional neural network," *Computational and Structural Biotechnology Journal*, vol. 20, pp. 4733–4745, 2022.

[8] D. Reyes and J. Sánchez, "Performance of convolutional neural networks for the classification of brain tumors using magnetic resonance imaging," *Heliyon*, vol. 10, no. 3, 2024.

[9] K. Singh, A. Kaur, and P. Kaur, "Computer aided detection of brain tumors using convolutional neural network based analysis of mri data," 2023.

[10] Y. Zhang, H. C. Ngo, Y. Zhang, N. F. A. Yusof, and X. Wang, "Imaging segmentation of brain tumors based on the modified u-net method," *Information Technology and Control*, vol. 53, no. 4, p. 1074 – 1087, 2024.

[11] R. Ahsan, I. Shahzadi, F. Najeeb, and H. Omer, "Brain tumor detection and segmentation using deep learning," *Magnetic Resonance Materials in Physics, Biology and Medicine*, pp. 1–10, 2024.

[12] T. Arumaiththurai and B. Mayurathan, "The effect of deep learning and machine learning approaches for brain tumor recognition," in *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 185–190, IEEE, 2021.

[13] J. Alyami, A. Rehman, F. Almutairi, A. M. Fayyaz, S. Roy, T. Saba, and A. Alkhurim, "Tumor localization and classification from mri of brain using deep convolution neural network and salp swarm algorithm," *Cognitive Computation*, vol. 16, no. 4, pp. 2036–2046, 2024.

[14] A. A. Asiri, A. Shaf, T. Ali, M. Aamir, M. Irfan, and S. Alqahtani, "Enhancing brain tumor diagnosis: an optimized cnn hyperparameter model for improved accuracy and reliability," *PeerJ Computer Science*, vol. 10, p. e1878, 2024.

[15] I. S. Rajput, A. Gupta, V. Jain, and S. Tyagi, "A transfer learning-based brain tumor classification using magnetic resonance images," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 20487–20506, 2024.

[16] K. Shiny, "Brain tumor segmentation and classification using optimized u-net," *The Imaging Science Journal*, vol. 72, no. 2, pp. 204–219, 2024.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 49–54
DOI: 10.15439/2025F7347
ISSN 2300-5963 ACSIS, Vol. 44

Evaluating Depression and Stress Among Young Adults Using DASS-21: Towards Personalized Intervention Strategies

Umamah Bint Khalid
Department of Electronics
Quaid-I-Azam University
Islamabad, Pakistan
umamahkhalid@ele.qau.edu.pk

Mario Fiorino
*D*ipartimento di Automatica
e Informatica (DAUIN)
Politecnico di Torino, 10129 Torino, Italia
mario.fiorino@polito.it

Madiha Haider Syed
Institute of Information Technology
Quaid-I-Azam University
Islamabad, Pakistan
madiha@qau.edu.pk

Musarat Abbas
Department of Electronics
Quaid-I-Azam University
Islamabad, Pakistan
mabbas@qau.edu.pk

***Abstract*—Depression, anxiety, and stress are commonly studied in the elderly, often manifesting as a loss of interest in previously enjoyed activities, disrupted sleep patterns, and other emotional or behavioral changes. However, with rapid technological advancements, young adults particularly those between the ages of 20 and 40 are emerging as a highly vulnerable group. This demographic faces a unique psychological burden, as they attempt to navigate the cultural and generational gap between two vastly different worlds: an older generation that often resists or struggles to adapt to revolutionary technologies, and a younger generation having a grip on modern technology. This generational divide can create a sense of isolation and pressure for young adults especially those people living in developing countries where open conversations about mental health still remain stigmatized and difficult to initiate.**

This research aims to develop a mental health app that can evaluate depression, and stress among young adults using the DASS-21 self-assessment test and suggest a personalized intervention keeping in view the level and severity of depression and stress. For personalized interventions, upper confidence bound algorithm is used to maintain a balance between exploration and exploitation. Agent's performance and effectiveness of intervention is evaluated by a post-test.

***Index Terms*—depression, stress, personalized intervention, dass-21, reinforcement learning, UCB**

I. Introduction

ARTIFICIAL Intelligence (AI) is an umbrella term use to describe technological advancements. Everyone is familiar with AI, especially those who are familiar with self-adaptive, self-learning systems. Reinforcement learning (rl), a type of machine learning, comes under the umbrella of AI. RL is the soul of self-adaptive and self-learning systems, where the system learns patterns and adjusts itself according to requirements. Implications of AI in the field of health care are remarkable [1]. These implications not only involve diagnosing a disease but also recommend treatment plans. Moreover, technological advancements of AI have been reported for personalized medication adherence for elderly people [2], for the analysis of the living cell mechanism [3], for therapeutic treatment interventions of Alzheimer's Disease [1], Multiple Sclerosis [4], Autism Spectrum Disorder (ASD) [5], and other mental disorders.

Mental health is a fundamental concept and is closely related to the overall well-being of mankind [6]. Sartorius has defined mental health in three different manners: i) as the absence of any illness or disease; ii) a state of an organism helps to perform all functions at their best; and iii) a balanced state of an organism to maintain a healthy relationship with others and with its surroundings [7]. All these definitions are directly dependent on the fundamental needs of an individual. The degree to which these fundamental needs are met determines the mental health condition. The fundamental needs of any individual include food, shelter, family support, social circle, unnecessary stress, survival, security, freedom from pain, and environmental hazards [8]. Reflecting on these aspects highlights the complexity involved in determining which definition applies in varying contexts.

Mental health, like mental illness, is also affected by biological, social, psychological and environmental factors. Mental health, similar to mental illness, is shaped by a combination of biological, psychological, social, and environmental influences. The ability of a person to function is deeply influenced by their family, close friends, colleagues, and peers and, in the broader context, influenced by society and culture [9] [10]. Social relationships, in both contexts, play a significant role in psychological well-being. Positive social interactions can help reduce symptoms of depression and stress, while isolation or negative social conditions can increase vulnerability to mental health issues.

Depression is a complex, multifaceted disorder. It comes under the category of mood disorders and is considered one of the primary contributors to disability across the world [11]. A depressed person feels tired most of the time, struggles with sleep, irritability, sadness, and headaches. Depression should be taken seriously before it turns into a clinical depression or major depressive disorder (MDD) [13]. Major depressive disorder can lead to neurological disorders i.e., multiple sclerosis (MS) or Alzheimer's disease (AD).

Stress is another state of mind that every human being faces most of the time. Stress is a mental pressure that arises as a response to some external stimuli (tough situation, threat, challenge etc). Positive stress is helpful and plays an important role in handling different situations. Almost everyone experiences stress at some point, but what makes it worse is the way every human handles it. The stress response is different for different people and plays a crucial role in maintaining our overall health and well-being. Stress affects all body systems including the musculoskeletal, respiratory, cardiovascular, endocrine, gastrointestinal, nervous, and reproductive systems [12].

A lot of pharmacological and non-pharmacological treatments are available for depression and stress. This research aims to address the level of depression and stress among young adults living in a developing country like Pakistan and its treatment using technology-based personalized interventions. The level of depression and stress has been measured using dass-21 self assessment test. Dass-21 self assessment test measures depression, anxiety and stress into normal, mild, moderate and severe level. After the severity of results the user will be assigned an intervention through the RL agent, the correctness of choice of intervention depends on post-test results.

The rest of the paper is organised in the following manner: Section II contains literature review, section III is the research background , Section IV Proposed methodology, section V is Results and Discussion and section VI conclusions and future work.

A. Research Objectives

A lot of work has been done about mental health of older adults and adolescents. Very few worked in mental health evaluation among young adults. In the ongoing socio-economic environment and lifestyle, almost every individual is facing mental health problems. Keeping these factors in mind, proposed research aims to provide a mental health app to measure and manage depression and stress among young adults. The objectives of this research include:

- Construction of mental health app for people aged 20-40.
- Use of DASS-21 test to measure level of three main mental problems: Depression, Anxiety, Stress.
- Use of RL agent to suggest an intervention that best suits the needs and demands of a person.
- Evaluation of intervention decision by an agent through post-test.

II. Literature Review

A. DASS-21

To understand the effectiveness of the 21-item Depression Anxiety Stress Scales (DASS-21) for individuals with mild traumatic brain injury (mTBI), the study in [14] performed a psychometric evaluation. Through Rasch analysis, the researchers examined the scale's underlying structure, consistency, ability to differentiate among individuals, and item fairness. Findings suggest that the DASS-21 is a psychometrically robust instrument for gauging distress and stress in adults receiving care for mTBI. For specific depression assessment, the study advises using a shortened six item subscale for depression. [15] investigated the practical utility of the DASS-21 in elderly populations across various nations, examining sample demographics, application goals, and recruitment sites. Researchers screened 855 studies from EMBASE, PubMed, and SciELO, ultimately analyzing 22 involving 14,339 participants (predominantly women aged 60-91) from 13 countries. The review concluded that the DASS-21 is a valuable instrument for tracking depression, anxiety, and stress in diverse elderly groups globally.

This study [16] evaluates the DASS-42 and DASS-21 for assessing depression, anxiety, and stress in hematologic cancer patients. Analyzing data from 452 patients, the research shows both scales have strong psychometric properties, with the bifactor model fitting better. The results support using these scales for reliable assessment in Turkish hematologic cancer patients, aiding clinical evaluations and interventions. Evaluating the DASS-21's psychometrics in Spanish and Chinese primary school teachers, [17] underscores the significance of educators' psychological health. The study revealed cross-cultural measurement invariance issues, with the DASS-21 best fitting a one-factor model for Chinese teachers and a three-factor model for Spanish teachers. Notably, it demonstrated concurrent validity with emotional exhaustion in both samples.

B. Personalized Interventions

The systematic review [18] investigates the cutting-edge developments in Next-Generation Cognitive Behavioral Therapy (NG-CBT) for depression, with a particular emphasis on how digital tools, teletherapy, and individualized treatment approaches are being integrated. The findings indicate that NG-CBT interventions significantly enhance treatment accessibility and patient engagement. Specifically, personalized digital tools contribute to improved treatment adherence and can be cost-effective alternatives to traditional therapy.

The article [19] thoroughly examines current tools and technologies leveraging artificial intelligence (AI) in the management of anxiety and depression. It highlights the growing integration of AI applications, such as chatbots, mobile health applications, wearables, virtual reality, and large language models (LLMs), into mental health. These tools facilitate accessible, personalized, and immediate support for individuals experiencing anxiety and depression. Researchers suggest that AI interventions are good for underserved populations,

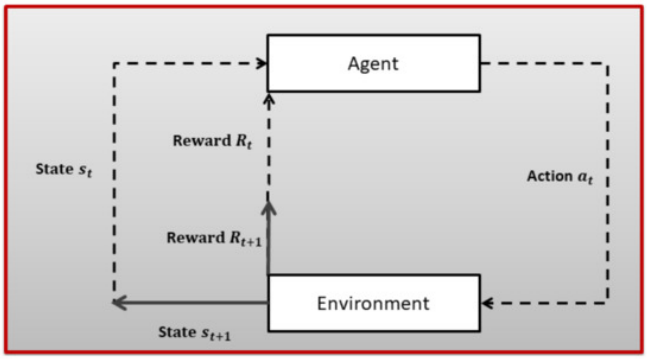


Fig. 1: Working of an RL agent presenting agent-environment interaction. a_t = action performed for state S_t, S_t = current state after performing an action at, Rt=reward received after at, $S_(t+1)$ = next state, $r_(t+1)$= reward received at a state $S_(t+1)$ [24].

but should not replace human professionals. [20] reviews the rise of anxiety among college students, by the COVID-19 pandemic, identifying risk factors across societal, institutional, familial, and individual levels. While traditional therapies are useful, they face accessibility and stigma issues. The review highlights digital interventions (apps, chatbots, VR) as scalable, cost-effective, and less stigmatizing solutions, particularly for tech-savvy students. Successful implementation requires collaborative efforts from governments, colleges, families, and students.

III. Research Background

A. Reinforcement Learning

Reinforcement learning (RL), a branch of machine learning, is goal-directed learning from interaction. Reinforcement learning involves improving performance through trial-and-error experience [21]. A method with a software agent that interacts with an unknown environment, selects actions dynamically and discovers which action yields more reward [22]. Reinforcement learning focuses on teaching algorithms to make choices by providing positive feedback for preferred actions and negative feedback for unwanted ones [23]. Similarly to how behavior is influenced by rewards and consequences in psychology, this method allows systems to gradually develop the best strategies through a process of trial and error [24] as shown in Fig 1. The reward system is crucial for guiding the agent's actions toward achieving the final goal. It serves as a feedback mechanism, clearly indicating whether a chosen action has led to a positive or negative outcome. By understanding this, the agent can adjust its strategies effectively, ensuring progress and success in reaching its objectives.

1) Upper Confidence Bound (UCB): Upper Confidence Bounds (UCB) are statistical techniques applied in decision-making under uncertainty, especially when there is a need to balance exploration of new choices with exploitation of known favorable ones. This method is commonly used in multi-armed bandit problems, where it aids in selecting actions by considering both their estimated rewards and the uncertainty around those estimates.
The UCB method works by computing an upper confidence limit for the expected reward of each option. This involves combining the estimated reward with an additional term that captures the level of uncertainty or variability in that estimate. The option with the highest resulting upper bound is chosen for the next decision step [25].
Exploration: Exploration enables the agent to gather more information about the available actions, by exploring all possible states in a given environment.
Exploitation: Exploitation allows the agent to select the action it currently believes will yield the highest reward, aiming to maximize short-term gains based on existing knowledge.

2) UCB Action Selection: UCB is favorable for uncertain conditions to balance exploration and exploitation. Mathematically, a UCB agent selects an action using the following equation:

$$A_t = argmax_a(Q_t(a) + c(\sqrt{ln(t)/N_t(a)})$$

Here t presents timesteps, Q_t is expected reward (Exploitation) at time t and last term is exploration reward.

The main objective of using the UCB here is to maximize the cumulative reward across multiple trials. UCB algorithms seek to minimize regret (the difference between the actual rewards earned and the estimated rewards. Unlike other RL algorithms, it does not require extensive training data to learn. It's exploration-exploitation property helps in minimizing regret by choosing optimal intervention available.

B. Personalized Interventions

Personalized interventions are care strategies that adapt to the specific circumstances of each individual, especially in mental health settings. Rather than relying on broad, generalized solutions, personalized interventions emphasize tailored strategies that align with the unique needs, contexts, and characteristics of an individual, a group, or a system. By aligning the intervention closely with the individual's needs, the aim is to improve the effectiveness and engagement of the treatment process, ultimately leading to better therapeutic outcomes [1].
When implementing personalized interventions, it is essential to choose them carefully, guided by the individual's specific needs and preferences. This strategy encourages shared decision-making between the individual and the healthcare provider. Research suggests that personalized interventions are especially beneficial for complex mental health conditions such as depression and anxiety, as these disorders often present differently in different individuals [26].
The emergence of AI and digital technology has revolutionized the field of healthcare especially mental health by offering tailored treatment regimes. Machine learning made significant advancements in this field, but reinforcement learning helped

to turn traditional treatment methods into personalized and adaptive treatment interventions [1].

C. Self Assessment Test

Different self-assessment tools are available online to measure depression and anxiety. Patient Health Questionnaire (PHQ-9) consists of 9 self-reporting items, one of the free online available tests used to measure depression [27]. Back Depression Inventory (BDI), Back Anxiety Inventory (BAI) consists of 21 self-reporting items, each with 4 options. BDI is used for depression and BAI is used for anxiety [28]. Depression, Anxiety, Stress Scales (DASS) is most commonly used self-assessment test for three different but interrelated variables [29]. DASS comes in two versions, DASS-42 is long item scale and DASS-21 is short item scale with 21 reporting items, each having 4 options. The DASS-21 is widely recognized for its brevity and reliability. It is available in more than 40 languages, making it a popular choice in both research and clinical contexts worldwide [30]. DASS-21 has three subscales and contains seven items for each subscale (i.e. depression, anxiety and stress). Each item has four options, and each option shows level of severity from 0-3. Level 0 shows do not apply to me at all, 1 shows sometimes applied to me, 2 shows applied a good part of the time and 3 shows strongly applied to me.

IV. Proposed Methodology

This research aims to measure level of depression, anxiety and stress in young adults. The main steps involved in the proposed research are mentioned in Fig 2. This research aims to develop a web app to evaluate and manage depression and stress among young adults. The system has been made using Python Flask and consists of three main steps:

- Pre-Test
- RL Agent for Treatment Intervention
- Post-Test

DASS-21 appears as a pre-test as the user logged in using few demographics (name, age, gender, occupation). To track the outcomes user data is stored. The interface of pre-test can be seen in Fig 3a. Dass-42 covers every minor and major detail in a behavior over a specific period like a week or two. So, the selection of items for dass-21 was critical. We made sure to include those items in dass-21 that a user (a young adult) can associate himself/herself with.

The interface of a pre-test contains a few guidelines for the. Selection of an option for each item is mandatory. Pre-test will not be submitted until all the items are responded by selecting the severity option. In case of missing item, alert message will appear on the screen reminding the user to respond to each item. After submitting the pre-test as shown in Fig 3b, results will appear on the screen. Results contains both scores and severity of each subscale (depression, anxiety, stress). Results of a random user can be seen in Fig 4. (*Note:* The user in Fig 4 selected options on purpose to obtain mild to moderate results for each subscale). It is to be mentioned here that the primary focus is to manage depression and stress, keeping anxiety apart. The reason is that managing depression and stress simultaneously with a single intervention is quiet challenging. Once the agent is trained to manage these two subscales, it will be easy to manage the interventions for the third one.

Keeping anxiety apart does not mean that anxiety is not worth cured or managed. Anxiety is as important as depression and stress. The reason behind selection of DASS-21 in this research is to provide a single platform to manage all three subscales of DASS. As discussed in section III-C each subscale contains four levels of severity ranges from 0-3. These levels represent normal, mild, moderate and severe levels of each subscale. The range for each level is different for different subscales and is quite challenging to treat at the same time.

A. Interventions Selection

After completion of the first step (pre-test), the next step is the management of depression and stress. The management or intervention selection is done through RL agent. Interventions are assigned as actions of the agent, and reward is collected through post-test results. If post-test results are better than pre-test results, then the action performed by the agent is correct. The agent receives a positive reward. If the post-test results are equal to the pre-test, then there is a capacity to evaluate another action or to try another intervention. And, in the third case, if post-test results are less than the pre-test results, the agent gets a negative reward. Hence, the user was unable to manage the problem with the selected intervention. The agent will select another intervention to obtain better results.

Selected interventions to manage depression include:

- Mindfulness meditation
- Behavioral activation
- Cognitive restructuring
- Benson relaxation technique
- Empty chair technique

Interventions used to manage stress include:

- Box breathing exercise
- Time management techniques
- Physical exercise
- Guided imagery
- Improving sleep cycle

To overcome the unavailability of training data, threshold values are fixed to help agent maintain a balance between exploration and exploitation. These threshold values play an important role in selecting intervention. Agent selects an intervention that could treat both depression and stress of user at the same time. The decision made by an agent depends on subscale with high severity. This decision is then evaluated with a post-test at the end of the session as discussed before.

V. Discussion

This section presents the results, advantage and limitation of the work and possible directions for future work.

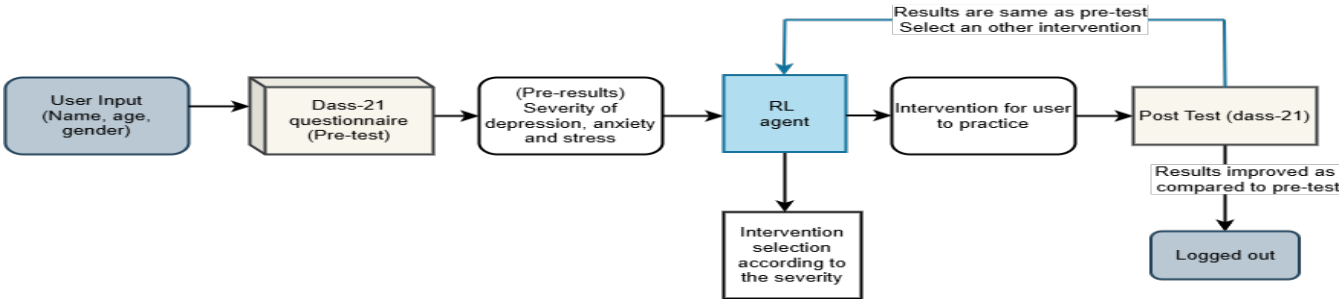
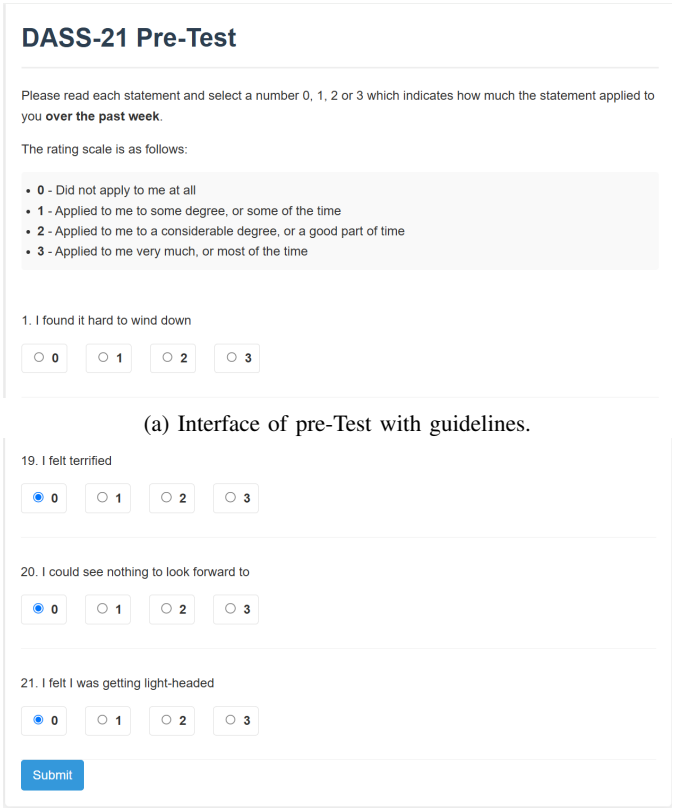


Fig. 2: Workflow diagram



(a) Interface of pre-Test with guidelines.

(b) Submission of DASS-21 test.

Fig. 3: Some visuals of DASS-21 items with levels of severity ranges from 0-3.

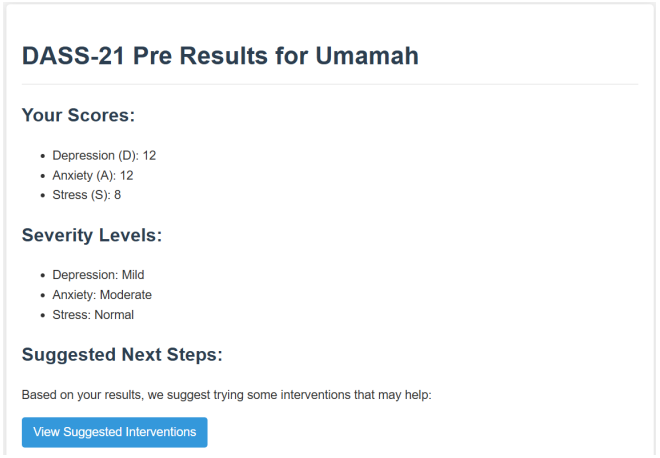


Fig. 4: Level and severity for each subgroup evaluated by selecting options given with each item.

A. Results

Use of reinforcement learning to generate adaptivity in AI-based digitial interventons is not new. Reinforcement learning has the potenial to do a lot for personalized interventions related to mental health problems. Use of DASS-21 to measure level of depression, anxiety and stress among young adults and suggest intervention according to their needs is promising. University students show a lot of interest in the proposed web app based digital intervention. The UCB algorithm used in this case does not have any prior data to train upon. But it maintained a good balance to train itself on the threshold values. Few university students volunteered for the proposed web app. Initially, the results of post-test remained same after practicing the suggested intervention for a week. The performance of the agent gets better after obtaining the same results and suggesting another intervention. To obtain satisfactory results, UCB requires more data for training. Another Problem faced by users is the understanding of reporting items. Few items are hard to interpret, and takes time to understand the context of the query being asked.

B. Conclusions and Future Work

The idea of a personalized interventions to treat depression and stress among young adults looks promising. It is important for the cultures where discussing one's mental health is difficult. The model used here requires a lot of prior knowledge to perform better. The proposed research focuses on management of depression and stress. But there is a need to manage anxiety along with depression and stress. Our future work includes intervention for all three subscales depression, anxiety and stress along with an additional feature of DASS-21 version in the native language.

ACKNOWLEDGEMENT

This project has been partially supported by the "Programma Nazionale Ricerca, Innovazione e Competitività per la transizione verde e digitale 2021/2027 destinate all'intervento del FCS "Scoperta imprenditoriale" - Azione 1.1.4 "Ricerca collaborativa" - with the project SIAMO (Servizi Innovativi per l'Assistenza Medica a bOrdo) project number F/360124/01-02/X75.

REFERENCES

[1] Khalid U, Naeem M, Stasolla F, Syed MH, Abbas M, Coronato A. Impact of AI-Powered Solutions in Rehabilitation Process: Recent Improvements and Future Trends. Int J Gen Med. 2024;17:943-969. https://doi.org/10.2147/IJGM.S453903.

[2] Ismail, A., Naeem, M., Syed, M.H., Abbas, M. and Coronato, A., 2024. Advancing Patient Care with an Intelligent and Personalized Medication Engagement System. Information, 15(10), p.609.

[3] Naeem, Muddasar. Fiorino, Mario. Addabbo, Pia. Coronato, Antonio. 2024. Integrating Artificial Intelligence Techniques in Cell Mechanics. Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS). 2024. 111–116. DOI: http://dx.doi.org/10.15439/2024F4351.

[4] Floriano Zini, Fabio Le Piane, and Mauro Gaspari. 2022. Adaptive Cognitive Training with Reinforcement Learning. ACM Trans. Interact. Intell. Syst. 12, 1, (February 2022), 29 pages. https://doi.org/10.1145/3476777.

[5] Stasolla, Fabrizio. Curcio, Enza. Zullo, Antonio. Passaro, Anna. Gioia, Maricarla. Integrating Artificial Intelligence-based programs into Autism Therapy: Innovations for Personalized Rehabilitation. Communication Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS). 2024. 169-176. 10.15439/2024F6229.

[6] Cinque, M., Coronato, A. and Testa, A. 2012. Dependable services for mobile health monitoring systems. International Journal of Ambient Computing and Intelligence (IJACI), 4(1), pp.1-15. DOI: 10.4018/jaci.2012010101.

[7] Sartorius, Norman. Fighting for mental health: a personal view. Cambridge University Press, 2002.

[8] Maslow, Abraham H. Toward a psychology of being. Simon and Schuster, 2013.

[9] SHRUTHI S. A Critical Study On Socializing And Its Benefits On Mental Health. ScienceOpen Preprints. 2022. DOI: 10.14293/S2199-1006.1.SOR-.PPJQN0I.v1.

[10] Bhugra D, Till A, Sartorius N. What is mental health?. International Journal of Social Psychiatry. 2013;59(1):3-4. doi:10.1177/0020764012463315.

[11] Sampogna, Gaia; Toni, Claudia; Catapano, Pierluigi; Rocca, Bianca Della; Di Vincenzo, Matteo; Luciano, Mario; Fiorillo, Andrea. New trends in personalized treatment of depression. Current Opinion in Psychiatry 37(1):p 3-8, January 2024. — DOI: 10.1097/YCO.0000000000000903.

[12] American Psychological Association. Stress effects on the body. (2023, March 8). https://www.apa.org/topics/stress/body.

[13] Medical News Today. Situational vs clinical depression: Differences and diagnoses. (2017). Medical News Today. https://www.medicalnewstoday.com/articles/314698.

[14] Faulkner, J. W., Snell, D. L., Siegert, R. J. (2024). Rasch analysis of the depression anxiety stress scales-21 (DASS-21) in a mild traumatic brain injury sample. Brain Injury, 39(2), 136–144. https://doi.org/10.1080/02699052.2024.2411297.

[15] Morero, Juceli., Esteves, Rafael., Verderoce Vieira, Mariana., Park, Tanya., Hegadoren, Kathleen., Cardoso, Lucilene. (2024). Systematic Review of the use of Depression Anxiety Stress scale 21 (Dass-21) in the elderly: Practical applicability across countries. Research Society and Development. 10.33448/rsd-v13i2.45107.

[16] Güven, S.; Şahin, E.; Topkaya, N.; Aydın, Ö.; Aktimur, S.H.; Turgut, M. Psychometric Properties of the Depression Anxiety Stress Scales (DASS-42 and DASS-21) in Patients with Hematologic Malignancies. J. Clin. Med. 2025, 14, 2097. https://doi.org/10.3390/jcm14062097.

[17] Wang, X., Cao, CH., Liao, XL. et al. Comparing the psychometric evidence of the Depression, Anxiety, and Stress Scale-21 (DASS-21) between Spanish and Chinese primary schoolteachers: insights from classical test theory and Rasch analysis. BMC Psychol 13, 450 (2025). https://doi.org/10.1186/s40359-025-02728-7.

[18] Gkintoni, E., Vassilopoulos, S. P., Nikolaou, G. (2025). Next-Generation Cognitive-Behavioral Therapy for Depression: Integrating Digital Tools, Teletherapy, and Personalization for Enhanced Mental Health Outcomes. Medicina, 61(3), 431. https://doi.org/10.3390/medicina61030431.

[19] Pavlopoulos, A., Rachiotis, T., Maglogiannis, I. (2024). An Overview of Tools and Technologies for Anxiety and Depression Management Using AI. Applied Sciences, 14(19), 9068. https://doi.org/10.3390/app14199068.

[20] Liu XQ, Guo YX, Xu Y. Risk factors and digital interventions for anxiety disorders in college students: Stakeholder perspectives. World J Clin Cases 2023; 11(7): 1442-1457. https://dx.doi.org/10.12998/wjcc.v11.i7.1442.

[21] Fiorino, M., Naeem, M., Ciampi, M. and Coronato, A., 2024. Defining a metric-driven approach for learning hazardous situations. Technologies, 12(7), p.103.

[22] Coronato, A., Naeem, M. (2019). A Reinforcement Learning Based Intelligent System for the Healthcare Treatment Assistance of Patients with Disabilities. Communications in Computer and Information Science, vol 1080. 2019. https://doi.org/10.1007/978-3-030-30143-9-2.

[23] Barto, Andrew G. "Reinforcement learning: An introduction. by richard's sutton." SIAM Rev 6.2 (2021): 423.

[24] Naeem, M. and Coronato, A., 2022. An AI-empowered home-infrastructure to minimize medication errors. Journal of Sensor and Actuator Networks, 11(1), p.13.

[25] Ismail, A., Naeem, M., Khalid, U.B. et al. Improving adherence to medication in an intelligent environment using reinforcement learning. J Reliable Intell Environ 11, 3 (2025). https://doi.org/10.1007/s40860-024-00242-y.

[26] Fiveable. Personalized interventions – Abnormal Psychology. (2024, August 1). https://library.fiveable.me/key-terms/abnormal-psychology/personalized-interventions.

[27] Kroenke, Spitzer., Williams. (2001). The PHQ-9. Journal of General Internal Medicine 16(9), 606-613. Retrieved from http://onlinelibrary.wiley.com/doi/10.1046/j.1525-1497.2001.016009606.x/pdf.

[28] Dale A. Halfaker, Steven T. Akeson, Danielle R. Hathcock, Curtis Mattson, Ted L. Wunderlich. Psychological Aspects of Pain. Hanley & Belfus. 2011. 13-22. ISBN 9781416037798. https://doi.org/10.1016/B978-1-4160-3779-8.10003-X.

[29] MORERO, J. A. P.; ESTEVES, R. B.; VIEIRA, M. V.; PARK, T.; HEGADOREN , K. M.; CARDOSO, L. Systematic Review of the use of Depression Anxiety Stress scale 21(Dass-21) in the elderly: Practical applicability across countries. Research, Society and Development, [S. l.], v. 13, n. 2, p. e10613245107, 2024. DOI: 10.33448/rsd-v13i2.45107.

[30] Bibi A, Lin M, Zhang XC, Margraf J. Psychometric properties and measurement invariance of Depression, Anxiety and Stress Scales (DASS-21) across cultures. Int J Psychol. 2020 Dec;55(6):916-925. doi: 10.1002/ijop.12671. Epub 2020 Apr 6. PMID: 32253755.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 55–60
DOI: 10.15439/2025F8934
ISSN 2300-5963 ACSIS, Vol. 44

Detecting Spatial Ordering of Nanoparticles with Geometric Deep Learning

Jan Krupiński
0009-0001-0267-7387
Cracow University of Technology
Faculty of Electrical and Computer Engineering
ul. Warszawska 24, 31-155 Kraków, Poland
Email: jan.krupinski@pk.edu.pl

Kazimierz Kiełkowicz
0000-0001-5791-6069
Cracow University of Technology
Faculty of Electrical and Computer Engineering
ul. Warszawska 24, 31-155 Kraków, Poland
Email: kazimierz.kielkowicz@pk.edu.pl

***Abstract*—Nanoparticle dispersion in heterogeneous catalysts plays a critical role in catalytic performance. We propose a robust and generalizable graph neural network (GNN) approach that combines the edge convolutional operator (EdgeConv) with a graph attention (GAT) layer to classify dispersion patterns in palladium on carbon (Pd/C) catalysts. Our method leverages GNNs to operate directly on particle location data extracted from scanning electron microscopy (SEM) images, thereby avoiding reliance on image features that may introduce bias. The proposed method offers an advantage over traditional image-based approaches, which risk overfitting to visual characteristics of the image that are unrelated to the spatial distribution of the nanoparticles. We validate the performance of our GNN architecture on multiple Pd/C samples with distinct carbon support types, achieving an accuracy of 89.84%, and demonstrate that our approach can reliably identify dispersion defects. The results highlight the potential of GNNs as a promising alternative for structure-based analysis and quality assessment of nanomaterial-based catalysts.**

***Index Terms*—Heterogeneous Catalysts, Graph Neural Networks, Deep Learning, Scanning Electron Microscopy**

I. Introduction

NANOMATERIAL-BASED catalysts are a major class of heterogeneous catalysts, widely used in chemistry, industry and medicine [1], [2]. They primarily consist of metal nanoparticles dispersed on a solid support, forming active sites that facilitate organic chemical reactions. Differentiated by the type of metal or metal alloy nanoparticle, as well as by the nature of the support material (ranging from carbon to various oxides such as silica), they exhibit tunable catalytic properties that can be optimized and tailored for specific reactions.

Here, we focus on palladium metal on carbon support (Pd/C) catalysts, which are primarily used in carbon-carbon coupling reactions (C-C coupling) and hydrogenation processes to efficiently synthesize a wide range of organic compounds [3], [4]. There exists a great variability among Pd/C catalysts, depending on both the metal and the characteristics of activated charcoal support used. Such factors as palladium oxidation, dispersion of the nanoparticles, water content and the support structure play a great role in the reactivity of the catalyst [5]. Optimization of these parameters can lead to improved reaction efficiency, greater selectivity, and enhanced catalyst stability across a great range of synthetic applications.

The dispersion of nanoparticles on the carbon support can be analyzed using scanning electron microscopy (SEM) [6]. The Pd/C morphology may range from a mostly uniform distribution of palladium nanoparticles to the formation of ordered structures arising from imperfections in the support [7]. In such cases, nanoparticles can, for example, nucleate along grain boundaries or pore edges, leading to non-uniform distributions of active sites. These structural irregularities can negatively impact the catalytic efficiency and consistency.

This work proposes a generalizable and robust deep learning approach for detecting nanoparticle dispersion defects in Pd/C catalysts based on a novel graph neural network with edge convolutional operator (EdgeConv) [8] and a graph attention (GAT) layer [9]. We address limitations of existing techniques by leveraging graph neural networks to classify catalysts based on nanoparticle spatial arrangements and geometrical patterns. By working solely with particle location data, we aim to eliminate potential biases introduced during image acquisition or sample preparation. Additionally, we evaluate our model on Pd/C samples with an alternative carbon support type to demonstrate its effectiveness under real-world variations in material composition.

This paper is structured as follows. Section II reviews related research involving deep neural networks. Section III introduces the dataset and data preparation process, followed by a description of the graph neural network (GNN) architectures and the training scheme. In Section IV, we outline the evaluation metrics, present model performance, and compare our approach with alternative methods. Section V concludes our paper and provides a brief discussion.

II. Related Work

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have been successfully applied to problems with an underlying grid-like (i.e., Euclidean) data structure, particularly in image processing, speech recognition, classification, and image segmentation [10]. Recent applications of CNNs include bone age evaluation from X-ray images to support diagnosis and treatment planning for various disorders [11]; olive disease classification using an adaptive ensemble of two EfficientNet-B0 models,

Topical area: Advanced Artificial Intelligence in Applications

which improves state-of-the-art accuracy on a publicly available dataset [12]; and semantic segmentation of complex urban street scenes for autonomous driving, where models such as MobileNet and ResNet50 are used as encoders in the U-Net architecture [13].

In the field of heterogeneous catalysis, CNNs have been applied to nanoparticle segmentation and tracking under reactive conditions [14]. CNN architectures such as U-Net [15], which consists of two paths (a contracting path and an expansive path) and employs a training strategy that heavily relies on data augmentation, have been shown to make more efficient use of limited annotated samples. U-Net has been used to analyze transmission electron microscopy (TEM) images and videos, along with other architectures [16], [17]. Additionally CNNs have been used for automated analysis to identify the number of defects and to define anchoring and segmentation in the study of high-entropy metal nanoparticles [16]. More recent state-of-the-art models, such as Segment Anything Model (SAM) [18], have also been employed to aid in the quantification and analysis of nanoparticles [19].

CNNs have been further used to analyze SEM images of Pd/C catalysts for the purpose of classification of nanoparticle dispersion defects, distinguishing between defective and non-defective morphology [20]. While high classification accuracy ($\geq 90\%$) was reported, the dataset was limited in size, and the models were not evaluated on independent samples. As a result, the models were shown to differentiate between specific sample identities rather than between dispersion patterns themselves. In this work, we aim to overcome these limitations by developing a more generalizable method.

B. Graph Neural Networks

For problems involving data that does not exhibit a regular grid-like structure, such as point clouds or molecular structures, the data can instead be modeled as graphs [21]. Graph Neural Networks (GNNs) represent each data point as a vertex in a graph and construct edges based on neighborhood relationships. Spatial GNNs define message-passing and aggregation operations directly over a node's neighborhood in the input or feature space.

In the message-passing mechanism, each node updates its representation by receiving and aggregating information (or "messages") from its neighboring nodes, often using learnable functions such as multilayer perceptrons (MLPs). This process allows nodes to iteratively encode both local geometric structure and other features from their spatial context. Pooling is often used to coarsen the graph or summarize neighborhood information, followed by task-specific layers - such as fully connected layers for classification or regression tasks [22].

Simonovsky and Komodakis [21] generalized the convolution operation from traditional CNNs on regular grids to arbitrary graphs. Their approach constructs deep neural networks for graph classification by treating each point as a vertex and connecting it to its neighbors via directed edges.

Building on this idea, Wang et al. [8] proposed the Dynamic Graph CNN (DGCNN) architecture, where the graph is constructed in the feature space and dynamically updated at each network layer. At its core, DGCNN uses the edge convolutional operator (EdgeConv), which applies an MLP as a feature learning function over edges, followed by an aggregation function that combines information from each point's neighbors.

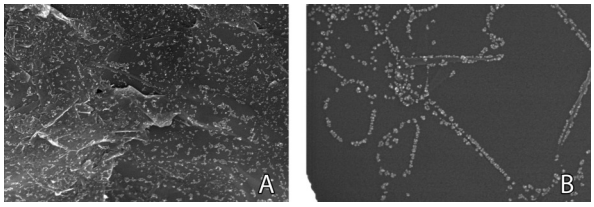


Fig. 1. Image A (on the left) shows an example of a disordered nanoparticle distribution, while Image B (on the right) illustrates nanoparticles forming ordered patterns.

While GNNs are primarily used for analyzing point clouds [22], they have also found broader application in fields such as molecular modeling and physical system prediction [23]. GNNs have recently been applied to heterogeneous catalysis, including reaction modeling [24] and catalyst screening via binding energy prediction [25]. GNNs have also been applied to graph classification tasks, particularly for predicting the overall toxicity of molecular structures. To improve generalization with limited labeled data, Few-Shot Learning techniques have been incorporated alongside models such as Graph Isomorphism Networks (GINs) and multi-headed attention mechanisms [26].

III. Materials and Methods

This section describes and analyzes the dataset, and introduces the proposed graph neural network (GNN) architecture used in this work.

A. Dataset

The dataset used in this study [7] consists of 1000 scanning electron microscopy (SEM) images collected from five different Pd/C catalyst samples. Despite its limited sample size, the dataset encompasses a range of support materials, imaging magnifications, and spatial sampling regions, aiming to provide representative variability across Pd/C catalysts. The dataset includes three types of carbon supports: graphite powder, nanoglobular carbon, and pressed graphite bars. Each image was labeled as either containing ordered or disordered nanoparticle distributions. An example of both distributions is presented on Figure 1. A summary of the images per sample in the dataset is shown in Table I.

TABLE I
Overview of the Pd/C samples in the dataset [7].

Sample	Images	Ordered	Support	Subset
S1	687	Yes	Graphite powder	Train. / Val.
S2	63	Yes	Graphite powder	Testing
S3	185	No	Nanoglobular C	Train. / Val.
S4	50	No	Graphite bars	Testing
S5	15	No	Graphite bars	Testing

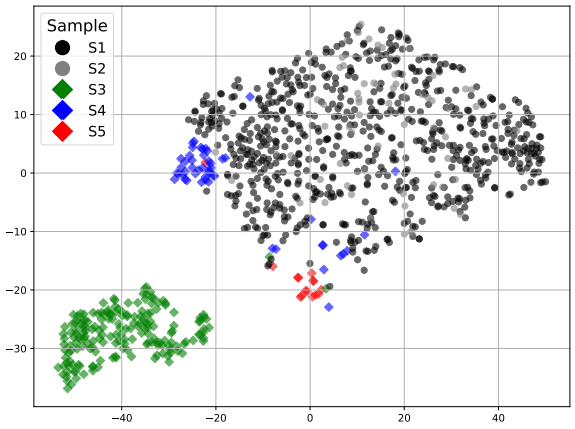


Fig. 2. T-SNE visualization of features extracted by ViT-L from the images in the dataset. Samples marked with dots contain spatially ordered nanoparticles, while samples marked with diamonds are disordered. While most samples are relatively similar visually, S3 is a clear outlier.

To explore visual similarities within the dataset, we extract image features using a deep learning model pretrained on a large-scale image dataset. These features are then projected into a lower-dimensional space using t-distributed stochastic neighbor embedding (t-SNE), a non-linear dimensionality reduction technique commonly used for visualizing high-dimensional data [27]. For this purpose, we use a Vision Transformer (ViT) [28] pretrained using the self-supervised DINOv2 method on LVD-142M [29]. DINOv2 trains vision models without the need for labeled data by encouraging consistency between different augmented views of the same image, enabling the model to learn embeddings that capture semantic and structural information in its latent feature space. This approach is particularly suitable for our task, as it enables meaningful feature extraction without requiring task-specific fine-tuning.

The resulting t-SNE visualization of the dataset is shown in Figure 2. The images are clustered primarily based on their visual characteristics, rather than the spatial arrangement of the nanoparticles. These visual characteristics are primarily determined by the type of sample support (background). At this scale, graphite powder (samples S1 and S2) appears visually similar to graphite bars (samples S4 and S5), while nanoglobular carbon (sample S3) looks distinctly different and forms a separate cluster. This highlights how deep learning models trained on the dataset can be influenced by the visual features introduced by the support material, rather than focusing solely on the spatial distribution of nanoparticles.

B. GNN Architecture

Since our data consists of two-dimensional projections of nanoparticle positions extracted from SEM images, we base our GNN models on architectures commonly used for point cloud analysis. Although the actual particle arrangements are three-dimensional, their projections still encode meaningful spatial structure.

The first step in such models is the construction of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from the given positions. Here $\mathcal{V} = \{1, \ldots, n\}$ denotes the n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. We construct a k-nearest neighbors (k-NN) graph to accomplish this task, setting $k = 6$ to balance graph complexity with computational efficiency.

Our base model employs the edge convolutional operator (EdgeConv) [8], which computes edge features for each node $\mathbf{x}_i$ by aggregating information from its neighborhood $\mathcal{N}(i)$:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i)} h_{\mathbf{\Theta}}(\mathbf{x}_i \,\|\, \mathbf{x}_j - \mathbf{x}_i), \tag{1}$$

Here the $\mathbf{x}$ variables are the 2D spatial coordinates of particles and $h_{\mathbf{\Theta}}$ is a learnable function implemented as a multilayer perceptron (MLP). Max pooling is used to aggregate information across neighbors. Following the Dynamic Graph CNN (DGCNN) architecture [8], multiple EdgeConv layers can be stacked to learn hierarchical features. However, to avoid problems caused by excessive Laplacian smoothing [30], we use a shallow model with only two EdgeConv layers. As more layers are added, the features of neighboring nodes become increasingly similar, and the representation across the entire graph converges to a single value. This erases important local differences, harming performance. After the EdgeConv layers, the outputs are concatenated and pooled globally before passing through a final MLP for binary classification.

To mitigate the effects of Laplacian smoothing in deeper graph architectures, we also propose a hybrid model architecture where the deeper EdgeConv layer is replaced by a graph attention (GAT) layer [9]:

$$\mathbf{x}'_i = \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{i,j} \mathbf{\Theta}_t \mathbf{x}_j, \tag{2}$$

In this operator, the transformed features $\mathbf{\Theta}_t \mathbf{x}_j$ of neighboring nodes are weighted by attention coefficients $\alpha_{i,j}$, which are learned via an additive attention mechanism. In our architecture, the first EdgeConv layer captures local geometric relationships by operating on an initial k-NN graph. A new k-NN graph is then reconstructed based on the learned features. The subsequent GAT layer models higher-order dependencies on this updated graph by assigning learnable, context-aware weights to neighboring nodes. This dynamic attention mechanism reduces over-smoothing and improves the model's ability to focus on the most informative neighbors. The features are then concatenated and pooled as before. An overview of both models is provided in Table II.

IV. Experimental Evaluation

A. Data Preprocessing

While both the image resolution (1280×890) and overall dataset size were sufficient for training, validation, and testing of deep learning models, the number of distinct samples was relatively limited. To ensure that the models learn to recognize nanoparticle ordering (rather than relying on sample-specific characteristics or substrate structure) we partitioned the data by

TABLE II
PROPOSED GNN ARCHITECTURES, WITH LAYER SHAPE DESCRIBING THE NUMBER OF NEURONS PER LAYER.

DGCNN Model	
Layer Type	**Layer Shape**
EdgeConv	4, 64, 64, 128
EdgeConv	256, 128, 128, 256
Concat. + Pooling	-
Global MLP	384, 256, 128
Dense (MLP)	128, 64, 2

EdgeConv + GAT Model	
Layer Type	**Layer Shape**
EdgeConv	4, 64, 128, 256
GAT	256, 64 (4 heads)
Concat. + Pooling	-
Global MLP	512, 256, 128
Dense (MLP)	128, 64, 2

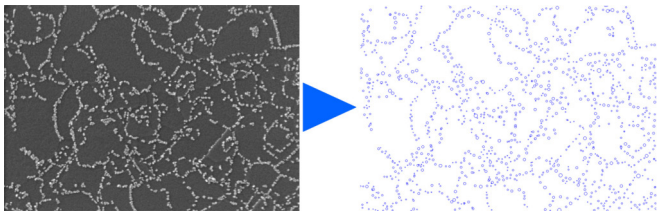


Fig. 3. Example SEM image (left) from sample S1 along with the detected particle locations (right). The nanoparticles form ordered structures.

sample. Samples S1 and S3, which contain the most images, were used for training and validation in an 80/20 split, while the remaining samples (S2, S4, and S5) were reserved for testing. This ensures a more robust assessment of model, and prevents classification based on the sample features alone.

To prepare our data for GNNs, we extracted nanoparticle coordinates from each image. While CNN-based methods have previously been used for nanoparticle segmentation [14], [16], [17], [19], classical methods have been shown to produce acceptable results in nanoparticle detection [31]. We opted for a more computationally efficient classical approach based on the Simple Blob Detection algorithm [32]. This method uses intensity thresholding and contour filtering to identify particles. Contours were filtered by size and brightness to isolate small, bright features corresponding to individual nanoparticles. An example SEM image and the corresponding extracted particle locations are shown in Figure 3.

B. Model Training

The models were trained using the cross-entropy loss function, optimized via the Adam algorithm [33]. A learning rate of 1×10^{-5} was chosen to ensure stable convergence. To prevent overfitting, we employed early stopping, halting training if no improvement was observed in the validation loss for 10 consecutive epochs. Training was performed with a batch size of 64, where each input sample consisted of 512 particle locations, randomly selected from the full set of detected particles in a given image. Batch normalization and dropout were applied throughout the network to improve generalization.

As previously noted, samples S1 and S3 were used for training and validation. The remaining samples were held out to evaluate the model on previously unseen data, ensuring robustness to variations in material and avoiding bias. Nevertheless, the training dataset introduced challenges related to both class imbalance and structural bias. Firstly, the dataset contained significantly more ordered distributions (687) than disordered ones (185), making it imbalanced. Additionally, sample S3 had a unique, globular structure, not present in the other samples. This raised the risk that the model might learn to associate specific support characteristics with disorder, rather than focusing on the actual nanoparticle arrangement. To address these problems, we applied several data augmentation techniques:

- **Disordered data generation** - additional disordered particle distributions were synthetically generated, to address the class imbalance. Particle positions were initialized on a regular grid and then perturbed by adding noise drawn from a uniform distribution.
- **Geometric transformations** - particle coordinates were flipped horizontally and vertically, rotated.
- **Jittering** - small random perturbations were added to particle positions with a fixed probability.

These augmentation strategies aimed to diversify the training data and reduce overfitting to specific samples or support types, improving the model's ability to generalize to new Pd/C samples.

C. Evaluation Metrics

To assess the performance of our models, we have used typical binary classification metrics such as accuracy, recall, precision and F1 Score (harmonic mean of recall and precision). These metrics can offer complimentary insights into the performance of the model. In our experiments, ordered distributions were treated as the positive class, and disordered distributions as the negative class. In our context, both false positives and false negatives can be problematic, therefore special attention is given to the balance between precision and recall. The results can also be summarized on a confusion matrix.

D. Model Performance

Following training, both architectures described in Table II were evaluated on the held-out test samples: S2, S4, and S5. The results presented in Table III show that both models perform well on new samples and support types, achieving strong performance in classifying both ordered and disordered distributions. The DGCNN model, utilizing only EdgeConv layers, reached an accuracy of 85.16%. Our proposed hybrid model, which combines EdgeConv and GAT layers, outperformed DGCNN with an accuracy of 89.84%. The corresponding confusion matrices are shown in Figure 4, highlighting the

TABLE III
GNN RESULTS ON THE TEST SET.

	DGCNN	EdgeConv + GAT
Accuracy	85.16%	89.84%
Precision	84.38%	90.32%
Recall	85.71%	88.89%
F1 Score	85.04%	89.60%

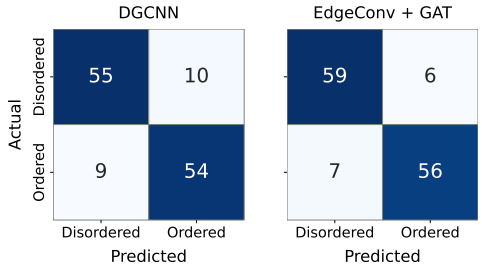


Fig. 4. Confusion matrices for the GNN results on the test set. The performance for both classes is very similar in both models. Out proposed architecture outperforms DGCNN.

models' balanced performance across both classes, as well as the improvement achieved by our hybrid model.

E. Comparison With CNNs

Recent studies have shown that CNNs can achieve high accuracy when applied to the classification of SEM images of Pd/C catalysts [20]. However, we have raised concerns that CNNs might learn visual sample and support structure characteristics, rather than nanoparticle ordering. To test this claim we have trained two CNN architectures on the dataset: ResNet34 [34] and ConvNeXt [35]. Both models were trained using a similar scheme as described for GNNs, however with starting weights pretrained on the ImageNet [36] dataset with a 224x224 input size.

The results on the training, validation and testing sets are presented in Table IV. Although both CNNs achieved excellent performance on the training and validation data, they failed to generalize to new samples and carbon support types, with test accuracy dropping to 49.22%. As such, their use in real-world scenarios may be limited. CNNs classified all testing images as containing ordered nanoparticle patters, which can be explained as being due to the visual characteristics of the carbon support in the testing images. Samples S4 and S5 use graphite bars as the support, which in SEM images is more visually similar to graphite powder (ordered samples S1, S2) than to nanoglobular carbon present in the sample S3. This conclusion is based on our previous t-SNE visualization of the dataset, presented on Figure 2.

V. CONCLUSION

In this paper, we propose a novel geometric deep learning approach for classifying dispersion patterns in palladium on carbon (Pd/C) catalysts. Our method is based on graph neural networks (GNNs) and operates directly on particle location

TABLE IV
CNN ACCURACY ON THE TRAINING, VALIDATION AND TESTING IMAGES.

	ResNet34	ConvNeXt
Train.	90.15%	94.22%
Val.	94.53%	98.44%
Test.	49.22%	49.22%

data extracted from scanning electron microscopy (SEM) images. This approach enables classification of catalysts based on the spatial arrangement and geometrical patterns of nanoparticles. As a result, it offers significant advantages over traditional image-based methods, which are prone to overfitting due to irrelevant visual features unrelated to nanoparticle distribution.

First, we present a Dynamic Graph CNN (DGCNN) architecture [8] applied to the classification of dispersion patterns in Pd/C catalysts. Second, in order to mitigate the effects of Laplacian smoothing in deeper graph architectures, we introduce a hybrid deep learning model that incorporates a Graph Attention (GAT) layer stacked on top of a EdgeConv layer. These architectures are compared with standard convolutional neural networks (CNNs), specifically ResNet34 [34] and ConvNeXt [35].

The dataset we used in our study consists of 1000 scanned electron microscopy (SEM) images collected from five differnt Pd/C catalyst samples [7]. We tested our methods on multiple Pd/C samples with distinct carbon support types to the ones used in training, demonstrating that our proposed methods can reliably detect dispersion defects under real-world variations in material composition..

To assess the performance of the proposed deep learning architectures (DGCNN [8], EdgeConv + GAT, ResNet34 [34], and ConvNeXt [35]), we used standard binary classification metrics, including accuracy, recall, precision, and F1 score. Our results show that the hybrid model combining EdgeConv and GAT layers outperforms the DGCNN, achieving an accuracy of 89.84% compared to 85.16% (see Figure 4 for details). In contrast, the image-based CNNs, ResNet34 and ConvNeXt, both achieved significantly lower accuracy scores of 49.22% (see Table IV) on testing data.

These results demonstrate that both the DGCNN and our proposed EdgeConv + GAT model outperform traditional CNN architectures, with the hybrid model achieving the highest accuracy among all tested methods. Overall, our findings highlight the potential of graph neural networks as a powerful alternative to image-based methods for structure-aware analysis and quality assessment of nanomaterial-based catalysts.

In future work, our GNN architecture could be trained and evaluated on a more diverse set of catalyst types. Depending on the characteristics of the SEM images, this may also require incorporating deep learning–based nanoparticle detection or segmentation methods. Additionally, further research could explore strategies for deepening the GNN architecture while maintaining training stability and enhancing performance.

REFERENCES

[1] Hutchings, G. J. 2009. Heterogeneous catalysts—discovery and design. *J. MaterChem.* 19(9), 1222–1235. https://doi.org/10.1039/B812300B

[2] Tao, F. (Ed.). 2014. *Metal Nanoparticles for Catalysis: Advances and Applications*. Catalysis Series. Royal Society of Chemistry. https://doi.org/10.1039/9781782621034

[3] Lunxiang, L. and Liebscher, J. 2007. Carbon–Carbon Coupling Reactions Catalyzed by Heterogeneous Palladium Catalysts. *Chem. Rev.* 107(1), 133–173. https://doi.org/10.1021/cr0505674

[4] Mao, Z., Gu, H. and Lin, X. 2021. Recent Advances of Pd/C-Catalyzed Reactions. *Catalysts* 11(9), 1078. https://doi.org/10.3390/catal11091078

[5] Felpin, F.-X. 2014. Ten Years of Adventures with Pd/C Catalysts: From Reductive Processes to Coupling Reactions. *Synlett* 25(08), 1055–1067. https://doi.org/10.1055/s-0033-1340668

[6] Suga, M. *et al.* 2014. Recent progress in scanning electron microscopy for the characterization of fine structural details of nanomaterials. *Prog. Solid State Chem.* 42(1–2), 1–21. https://doi.org/10.1016/j.progsolidstchem.2014.02.001

[7] Boiko, D. A., Pentsak, E. O., Cherepanova, V. A. *et al.* 2020. Electron microscopy dataset for the recognition of nanoscale ordering effects and location of nanoparticles. *Sci. Data* 7, 101. https://doi.org/10.1038/s41597-020-0439-1

[8] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M. and Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38(5), Article 146, 12 pages. https://doi.org/10.1145/3326362

[9] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. and Bengio, Y. 2018. Graph Attention Networks. In: *Int. Conf. on Learning Representations (ICLR)*. https://openreview.net/forum?id=rJXMpikCZ

[10] LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature* 521, 436–444. https://doi.org/10.1038/nature14539

[11] Fahim, S. F., Tasnim, N., Kibria, G., Morshed, M. S., Nishat, Z. T., Azad, S. B., Nath, S., Dey, A. L., Shuddho, M. A. and Niloy, N. T. 2024. A Proficient Convolutional Neural Network for Classification of Bone Age from X-Ray Images. In: *Proc. 9th Int. Conf. on Research in Intelligent Computing in Engineering*, *Ann. Comput. Sci. Inf. Syst.* 42, 17–21. http://dx.doi.org/10.15439/2024R60

[12] Bruno, A., Moroni, D. and Martinelli, M. 2023. Efficient Deep Learning Approach for Olive Disease Classification. In: *Proc. 18th Conf. on Computer Science and Intelligence Systems*, eds. Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D., *ACSIS* 35, 889–894. http://dx.doi.org/10.15439/2023F4794

[13] Ciecholewski, M. 2023. Urban Scene Semantic Segmentation Using the U-Net Model. In: *Proc. 18th Conf. on Computer Science and Intelligence Systems*, eds. Ganzha, M., Maciaszek, L., Paprzycki, M., Ślęzak, D., *ACSIS* 35, 907–912. http://dx.doi.org/10.15439/2023F3686

[14] Faraz, K., Grenier, T., Ducottet, C. *et al.* 2022. Deep Learning Detection of Nanoparticles and Multiple Object Tracking of Their Dynamic Evolution During In Situ ETEM Studies. *Sci. Rep.* 12, 2484. https://doi.org/10.1038/s41598-022-06308-2

[15] Ronneberger, O., Fischer, P. and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. and Frangi, A. (eds), *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015*, *Lect. Notes Comput. Sci.* 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[16] Alnaasan, M., Al Zoubi, W., Alhammadi, S., Kang, J.-H., Kim, S. and Ko, Y. G. 2024. Well-Defined High Entropy-Metal Nanoparticles: Detection of the Multi-Element Particles by Deep Learning. *J. Energy Chem.* 98, 262–273. https://doi.org/10.1016/j.jechem.2024.06.038

[17] Mohsin, A. S. M. and Choudhury, S. H. 2024. Label-Free Quantification of Gold Nanoparticles at the Single-Cell Level Using a Multi-Column Convolutional Neural Network (MC-CNN). *Analyst* 149(8), 2412–2419. https://doi.org/10.1039/D3AN01982A

[18] Kirillov, A., et al., 2023. Segment Anything. In: *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Paris, France, pp. 3992–4003. https://doi.org/10.1109/ICCV51070.2023.00371

[19] Monteiro, G. A. A., Monteiro, B. A. A., dos Santos, J. A., et al., 2025. Pre-trained artificial intelligence-aided analysis of nanoparticles using the segment anything model. *Sci. Rep.* 15, 2341. https://doi.org/10.1038/s41598-025-86327-x

[20] Boiko, D. A., Pentsak, E. O., Cherepanova, V. A., Gordeev, E. G., and Ananikov, V. P., 2021. Deep neural network analysis of nanoparticle ordering to identify defects in layered carbon materials. *Chem. Sci.* 12, 7428–7441. https://doi.org/10.1039/D0SC05696K

[21] Simonovsky, M., and Komodakis, N., 2017. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, pp. 29–38. https://doi.org/10.1109/CVPR.2017.11

[22] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M., 2021. Deep learning for 3D point clouds: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 43(12), 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

[23] Zhang, S., Tong, H., Xu, J., and Maciejewski, R., 2019. Graph convolutional networks: a comprehensive review. *Comput. Soc. Netw.* 6, 11. https://doi.org/10.1186/s40649-019-0069-y

[24] Jiao, Z., Liu, Y., and Wang, Z., 2024. Application of graph neural network in computational heterogeneous catalysis. *J. Chem. Phys.* 161(17), 171001. https://doi.org/10.1063/5.0227821

[25] Gu, G. H., Noh, J., Kim, S., Back, S., Ulissi, Z., and Jung, Y., 2020. Active learning across intermetallics to guide discovery of electrocatalysts for CO_2 reduction and H_2 evolution. *J. Phys. Chem. Lett.* 11(9), 3185–3191. https://doi.org/10.1021/acs.jpclett.0c00634

[26] Mehta, B., Kothari, K., Nambiar, R., and Shrawne, S., 2024. Toxic molecule classification using graph neural networks and few-shot learning. In: *Proc. 19th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, ACSIS, vol. 41, pp. 105–110. http://dx.doi.org/10.15439/2024F3810

[27] Hinton, G. E., and Roweis, S., 2002. Stochastic neighbor embedding. *Adv. Neural Inf. Process. Syst.* 15. https://dl.acm.org/doi/10.5555/2968618.2968725

[28] Dosovitskiy, A., et al., 2020. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*. https://doi.org/10.48550/arXiv.2010.11929

[29] Oquab, M., et al., 2023. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*. https://doi.org/10.48550/arXiv.2304.07193

[30] Li, Q., Han, Z., and Wu, X.-M., 2018. Deeper insights into graph convolutional networks for semi-supervised learning. *Proc. AAAI Conf. Artif. Intell.* 32(1). https://doi.org/10.1609/aaai.v32i1.11604

[31] Boiko, D. A., Sulimova, V. V., Kurbakov, M. Y., Kopylov, A. V., Seredin, O. S., Cherepanova, V. A., Pentsak, E. O., and Ananikov, V. P., 2022. Automated recognition of nanoparticles in electron microscopy images of nanoscale palladium catalysts. *Nanomaterials* 12(21), 3914. https://doi.org/10.3390/nano12213914

[32] OpenCV team, 2025. OpenCV: Open Source Computer Vision Library. Accessed: 2025-05-25. https://docs.opencv.org/4.x/d0/d7a/classcv_1_1SimpleBlobDetector.html

[33] Kingma, D. P., and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

[34] He, K., Zhang, X., Ren, S., and Sun, J., 2016. Deep residual learning for image recognition. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[35] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S., 2022. A ConvNet for the 2020s. In: *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 11976–11986. https://doi.org/10.1109/CVPR52688.2022.01167

[36] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., 2009. ImageNet: A large-scale hierarchical image database. In: *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Miami, FL, USA, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 61–68
DOI: 10.15439/2025F7772
ISSN 2300-5963 ACSIS, Vol. 44

Utilization of Large Language Models for conformity assessment: Chances, Threats, and Mitigations

János Litzinger, Daniel Peters, Florian Thiel
8.52 Embedded Systems
Physikalisch-Technische Bundesanstalt
Berlin, Germany
{janos.litzinger,daniel.peters,florian.thiel}@ptb.de

Florian Tschorsch
Chair of Privacy and Security
Technische Universität Dresden
Dresden, Germany
florian.tschorsch@tu-dresden.de

***Abstract*—Assessing the conformity of software in measurement instruments is a laborious process and a major bottleneck in the process of developing new devices. Large Language Models have been shown to effectively handle complex tasks and have the ability to surpass humans with regard to speed and accuracy. However, integrating them into the technology stack can bring major security and privacy risks. This position paper performs a threat modeling in this context. By addressing the discovered confidentiality risks the paper draws a way for safely implementing Large Language Models as an essential tool in the process of conformity assessment.**

I. Introduction

In the European Union, measurement instruments such as electricity meters, taximeters or automatic weighing instruments are regulated regarding their specific metrological properties. In order to be sold on the European market, these instruments need to pass a conformity assessment that verifies whether the instrument complies to regulatory requirements. Most modern measurement instruments have a software component that handles, among other things, the storage and transmission of measurement data. Thus, this software is also subject to regulation and therefore requires a conformity assessment. The assessment involves searching for relevant information in software documentation provided by the manufacturer and the decision whether it conforms with the requirements defined for the measurement instrument. This process depends on manual labor and is very time consuming.

Large Language Models (LLMs) have been applied to nearly any field in natural language processing (NLP)—from text classification, question answering or information retrieval to named entity recognition. Those models are trained on enormous data sets with trillions of tokens [1], consisting of newspaper articles, websites, books, and social media entries. However, the training data mainly holds publicly available text [1]. The Physikalisch-Technische Bundesanstalt (PTB)[1], Germany's national metrology institute, envisions to leverage its vast amount of textual data to augment existing models with metrological expertise. Especially, highly contextualized tasks such as conformity assessment of software documentation can benefit from models that are adjusted for the metrological domain.

In this position paper, we outline a path forward to streamline conformity assessment by integrating LLMs into the assessment pipeline. Through threat modeling, we identify potential risks associated with deploying LLMs in risk-sensitive environments. Our analysis highlights confidentiality and integrity as the main security objectives in this context. To better understand the current state of research, we review relevant literature on information leakage in LLMs and discuss how it may help to protect LLM-assisted conformity assessment.

The following section provides an overview of the conformity assessment process and the use of NLP methods. Section III performs threat modeling following the established PASTA method [2], while Section IV discusses related research. Section V outlines directions for future work, and Section VI concludes the paper.

II. Conformity assessment

Measurement instruments that are used in commercial or administrative contexts need to deliver reliable, deterministic measurements. Most users of measurement devices or persons affected by them are not always able to verify these measurements and therefore rely on trusting that the instruments function as intended and output correct measurements. Legal metrology ensures this trust by formulating regulations for measurement instruments in those contexts. These regulation not only dictate the hardware but also the software side of these devices. The EU Directive 2014/32/EU [3], better known as Measurement Instruments Directive (MID), harmonizes the national regulations and enables manufactures of measurement instruments to produce for the entire market of the European Union. In order to receive a MID approval, manufacturers need to prove that their product conforms with the requirement of the MID. In practice, this is achieved by providing a Notified Body with the product and the appropriate hardware and software documentation. In Germany, the PTB functions as such a Notified Body and assesses the hardware's and

[1] https://www.ptb.de/cms/en.html

software's conformity with the requirements defined in the MID, whereas for most devices an assessment is only carried out on document basis.

The software is typically assessed along the lines of the WELMEC Guide Software "7.2" [4]. The WELMEC Guide differentiates between different classes of instruments which determine the specific requirements for the software. Those are defined in blocks for separation and download of software as well as for the transmission and storage of measurement data. Furthermore, each class of measurement instrument has its own specific requirement, e.g., electricity meters or automatic weighing instruments. Assessing the software requires a search for the relevant information in the provided documentation and the comparison with the requirements in the WELMEC Guide. The difficulty of searching in the documentation lies in the diversity of those documents. Each manufacturer uses their own terminology, document structure, and composition of different documents. Thus, the assessors need to adapt their search queries to the manufacturer's unique language. Due to its manual nature, this processing step has become a major bottle-neck in conformity assessment and hinders a fast time to market.

A. LLMs for conformity assessment

To process the vast amount of documents that are generated throughout a conformity assessment, PTB developed a software, which allows to search the provided documentation in regard to the requirements defined in the WELMEC Guide. However, it fails to extract most of the information needed for assessing the software. Therefore, a lot of manual labor remains. Nevertheless, the approach showed major advantages to a purely manual procedure and stresses the need for a more automated approach to conformity assessment.

The task of conformity assessment involves two major fields of NLP, namely classification and Information Retrieval (IR). Traditional methods such as tf-idf [5], while being strong baselines for classification and IR, they fail when being exposed to out-of-distribution data. LLMs on the other hand are able to abstract from their training data since they embed words or tokens in a semantically clustered vector space. LLMs use these embeddings and efficiently model word semantics up to sentences, paragraphs and entire documents. Especially the transformer architecture [6] has been shown to deliver state-of-the-art results in various language understanding tasks [7]. Transformers can be trained in parallel on large data sets and thus have been scaled up in recent years to large language models with billions of parameters, trained on trillions of tokens [1], [8]. Due to the huge computational resources needed to create such models, training a large language model from scratch for a specific use case or domain is impracticable. Therefore, the main application of language models shifted to the paradigm of adjusting pre-trained language models to a specific task or domain, better known as *fine-tuning*. This paradigm gave rise of so-called *foundation models* that are trained without a specific task and are later further adjusted. While some models (e.g., [9], [10]) hide their models behind free or paid APIs, other models are published for local use (e.g., [8]). These models are also known as *open weight* models since their weights are freely available for researchers, developers, and the end user.

The approach of adjusting pre-trained models to downstream NLP-tasks has shown impressive results on various benchmark test sets[2] and is therefore suitable for the task of conformity assessment. For the practical usage, we propose a system that make use of an embedding model that has been fine-tuned for retrieval of relevant text chunks for a given query. This is done by fine-tuning the model for document embeddings, mapping the input text to an n-dimensional vector. By embedding the document chunks and queries into this vector space, relevant chunks can be retrieved by a neighborhood search. The retrieved chunks are then used as additional context of a generative model, that has been adapted for the task of conformity assessment. This would enable the user to query the model with respect to certain documents asking whether it is in line with the requirements defined in the WELMEC guide. This method is referred as retrieval-augmented generation (RAG) [11]. While it is possible to set up this RAG-pipeline exclusively with pre-trained models, we assume that those models can benefit from the rich training data for conformity assessment inside PTB. Not only could the envisioned system assist the assessors of software, furthermore it could help manufacturers of measurement instruments compiling the documentation and thus reduce the administrative process even more. Due to the entailed security issues of this concept, we see the need for a thorough threat analysis even in this early stage of development.

III. Threat modeling

The documentation provided by the manufactures is of sensitive nature. Not only does it consist of publicly available documentation such as user manuals, it is rather a full documentation of the internal function of the measurement device. From the overall software architecture to fine details such as start parameter for algorithms — the documentation holds enough information to rebuild the measurement instrument and its software from scratch. Due to that sensitive nature of the documents, confidentiality is of hightest concern when designing software to assist in conformity assessment.

Threat modeling is a systematic approach to identifying potential threats, assessing associated risks, and developing appropriate mitigation strategies. While widely used in software development, it applies more broadly to understanding the security and privacy implications of complex systems. Common threat modeling methods include PASTA [2], STRIDE [12], and LINDDUN [13], which have different focuses and follow different approaches. PASTA (Process for Attack Simulation and Threat Analysis) takes a risk-centric view, aligning business objectives with technical requirements to derive risks from threats and known vulnerabilities. STRIDE identifies threats based on a data flow diagram (DFD) and

[2]https://paperswithcode.com/area/natural-language-processing

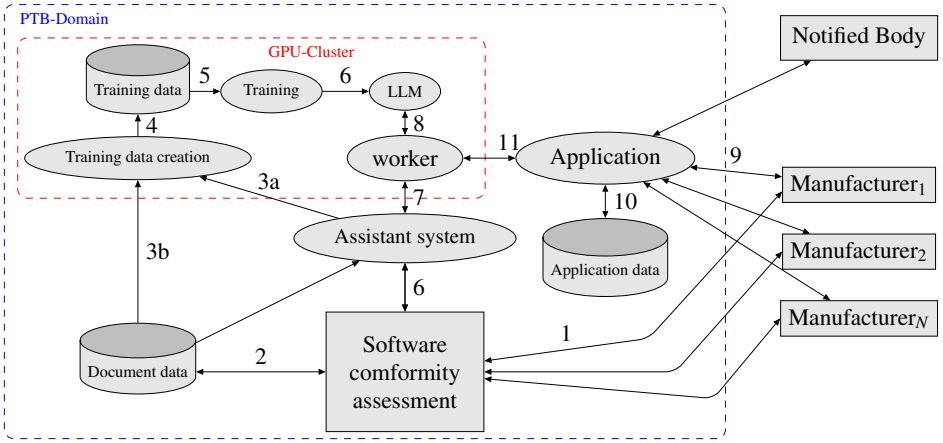


Fig. 1. Data flow diagram of the proposed system. Rectangles reference entities, ellipses symbol process, and cylinders represent data bases. The arrow indicates a flow of data, whereas the direction is indicated by the head. Arrows with heads on either end stand for a bidirectional data flow. The dashed line, here in blue and red, indicates a trust boundary.

categorizes them as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. LINDDUN, in turn, defines privacy-focused threat categories such as linkability, identifiability, and non-compliance. Risk assessment methods have also been tailored to measurement instruments [14], however we will hold on to methods known by a broader audience. While a threat-centric approach such as STRIDE might seem suitable for conformity assessment, a risk-driven method like PASTA is more appropriate due to its independence from predefined threat categories. LINDDUN may be well-suited for systems with high privacy requirements; however, practical experience shows that it can become overly detailed. In early design stages, systems often lack the specificity needed for such granular analysis and benefit more from broader, flexible approaches like PASTA.

In the following, we adopt PASTA, which involves seven stages: Stage 1 defines the context, including business objectives. Stage 2 outlines the technology stack and system scope. Stage 3 decomposes the application and identifies its actors. Stage 4 focuses on threat identification, which is linked to known vulnerabilities in Stage 5. Stage 6 simulates potential attacks, and Stage 7 maps the findings back to the original business objectives defined in the initial stage.

Context (Stage 1): The main purpose of the proposed application is to assist the software tester, while assessing the documentation provided by the manufacturer. Secondly, it should help the manufacturer of measurement instruments with compiling the appropriate software documentation needed for the conformity assessment. These documents can hold sensible information such as start parameters, technological innovations or novel solutions to known problems. They might also include typically public information such as User Manuals but since the conformity assessment is done prior to the market launch manufacturer have the interest to keep this information confidential from their competitors. Thus, the documentation documentation can not be made accessible to any 3rd-Party. Furthermore, the regulatory environment of conformity assessment makes it necessary to store these documents up two 10 years. Since the software testers should benefit from the assistant system during their everyday work, the system needs to be designed for high availability.

Technology Stack (Stage 2): The main component of the system is a GPU-Cluster running a free and open-source operating system. Due to the current dependence of the CUDA library[3] while developing LLMs the cluster is equipped with NVIDIA GPUs and thus runs proprietary drivers. The LLMs are pre-trained by 3rd parties such as Meta (Llama), Google (Gemma) or Mistral (Mistral). Usually these bigger models are published as "open-weight"-models, meaning the training algorithm and data is kept secret. Smaller models might be published as "open-source". Most of the models (big or small) are obtained via Huggingface[4] – a platform to publish machine learning models. It also develops a python library for developing and running models that will most likely be used for fine-tuning the models. Two commonly known libraries to work with language models are Ollama[5] and LangChain[6], whereas the former focuses on efficiently running the models, the latter offers an abstraction layer to build applications around language models. Ollama can be run in a docker container or manually installed whereas LangChain can be drawn into the project via the python package manager *pip*.

Application decomposition (Stage 3): The data flow diagram in Fig. 1 shows an abstracted model of the proposed system. The manufacturer provides the documentation to the conformity assessment either via E-Mail, with a link for download or uploads it to a file-sharing system (1). Those documents are usually in the PDF file format but can also consist of text files holding html or in the doc(x) file format. Those documents are stored (2) in a network drive where the software testers can access it. The data can also be accessed

[3]https://docs.nvidia.com/cuda/
[4]https://huggingface.co/
[5]https://ollama.com/
[6]https://www.langchain.com/

(3) by the process that creates the training data for the model. Creating the training data involves transforming the PDF and docx files into a machine-readable format such as Markdown. This can be done by using python libraries such as Nougat[7], Marker[8] or even using a LLM for layout detection. Most importantly the overall structure of the documents need to be kept since the software tester tend to reference the sections in their reports. Hence, the valuable information lies in the connection of assessment report (3a) and documentation (3b). The creation of the training data (4) and its storage are happening on the GPU-cluster (red dashed line), since some pre-processing needs access to the GPUs. The training data is then used to fine-tune existing LLMs for the task of conformity assessment (5). The trained model will be deployed on the same cluster for production (6). The assistant system can then query the model for document embeddings and search or for question-answering. A worker module ingest those queries (7) and feeds them into the model (8) in order to efficiently schedule the tasks. Manufactures or other Notified Bodies have also the possibility to access the model through a dedicated application (9) where they can check their documentation (10) or query the model (11). That way manufacturers are able to compile their documentation prior to handing it in for conformity assessment. Note that the entities *Manufacturer* and *Notified Body* are outside of the PTB trust zone (blue dashed line), whereas the testing persons of the conformity assessment work inside that trust zone.

Threat analysis (Stage 4): There are multiple threats that could affect the application or even the whole service of conformity assessment. Those can be found in Table I. First, there is the danger of data loss. Either the documentation provided by the manufactures or the work done by the software testers could be irretrievably lost (1). That would be costly in financial and reputational terms but would not threat the whole existing application. The same should hold for the data connected to the application used by the manufacturers. The training data (2) itself could also be lost, but since their creation is an automatic process it could be restored by running the process again. There are two major threats: corrupted integrity of data (3), meaning an undetected change of data, intended or unintended. This could affect the correctness of the result of the conformity assessment and is a major threat to the whole service. Connected in some way is the threat that the model itself outputs wrong results (4). This would lead to lower accuracy and effectiveness in the assistant system as well as in the manufacturer facing application. The other threat is violating the confidentiality of the data provided by the manufacturer (5), e.g., sharing the data with an unauthorized third party, as this information could be used to copy products from a competing manufacturer.

Vulnerability analysis (Stage 5): Identifying and addressing system vulnerabilities is essential to gain a clear understanding of the risks. In order to gather helpful insights from the vulnerability analysis, we restrict it to those that directly map to the threats identified in the previous stage. Since the software libraries used in the system are mainly from open-source providers, the system itself is vulnerable to coding and configuration flaws introduces by these libraries. Furthermore, the processes in the system could gain privilege on the server due to wrong configuration or coding imperfections. Additionally, authentication for the manufacturers could be imperfect, such that it would allow a manufacturer to inspect data that does not belong to its account.

TABLE I
THREAT OVERVIEW

	Threat	Description
1	Data loss	Deletion of data needed for conformity assessment
2	Training data loss	Deletion of data used for training the LLM
3	Corrupted data integrity	Undetected change of data needed for conformity assessment and training
4	Wrong model output	Factually incorrect output of the LLM that can lead to wrong decisions in conformity assessment
5	Information leakage	Leakage of information from conformity assessment to an unauthorized third party

However, the usage of LLMs introduces vulnerabilities to the system that are inherent to machine learning models. When used to generate text, LLMs are known to suffer from sometimes misleading or even factual incorrect answers, also known as *hallucination* [15]. They might be queried in such a way that their response is delivering unintended results. This vulnerability is called *prompt injection*. But most importantly to the current use case, multiple research has shown, that machine learning models are prone to reveal information about their training data (e.g., [16], [17], [18]).

Attack Modeling (Stage 6): In the scenario of conformity assessment, the vulnerabilities identified in the previous stage open up an attack surface for an adversary that intends to withdraw stored information. This adversary could be a manufacturer that wants to gain information about its competitors upcoming product. Since this information might be valuable to the malicious manufacturer, it might allocate appropriate resources for such an attack or even hire a contractor. The attacker can use the manufacturer-facing application as an entry point to run an extraction attack against the model. For the application, it would seem like queries to the model and therefore would be undetected. Furthermore, the adversary can use a so called Membership Inference Attack (MIA) either to strengthen the extraction attack or to verify that a certain manufacturer where handing in their documentation for conformity assessment. In a different attack scenario, a manufacturer exploit the usage of an LLM to cheat its way through the conformity assessment. It could hide instructions for the model asking it to only output positive responses. This could be achieved by using text in white color against a white background in the documentation such that a human reviewer

[7]https://github.com/facebookresearch/nougat
[8]https://github.com/VikParuchuri/marker

is unable to see it by only reading the document. Thus, the malicious instructions would be passed further to the model.

Risk and Impact Analysis (Stage 7): Using a prompt injection attack in order to pass the conformity assessment might be viable to some malicious manufacturers but also leave traces in the system. Note that due to the regulatory context the documentation used for the conformity assessment is stored for at least 10 years. Therefore, the risk of being exposed of cheating might be to high. On the other hand, attacking the model to gain information about its training data seems to be a reasonable approach by a manufacturer. The risk of being exposed is relatively low since the attack itself leaves no obvious traces. It might also not trivial to prove that querying the model was an attempt to gain information about the training data. Nevertheless, every prompt to the model and its response should be saved in order to monitor its usage as well as tracking its behavior over time. Therefore, the threat that confidential data is revealed to a third party is an existing risk that should be dealt with when using LLMs in a high risk environment such as in conformity assessment. Maintaining the integrity of the data is a common practice in the conformity assessment in PTB. For every file provided by the manufacture a checksum is calculated and stored. That way even the smallest changes in files can be detected. Therefore, we restrict the further examination of threats to confidentiality of training data in LLMs.

IV. Assessing LLM information leakage

While confidentiality and privacy are distinct concepts, they share common principles. Privacy refers to the rights of individuals or groups to control access to personal data. Confidentiality refers to restricting access to and disclosure of information, including proprietary and personal data—thus overlapping with privacy goals. As such, research on privacy in LLMs offers valuable insights into their behavior when trained on sensitive data, as in the case of conformity assessment.

While it is desired that language models memorize certain training data such as Wikipedia articles in order to return factually correct text, this is not the case for other training data. For example, models trained on clinical notes might reveal sensitive data about the patients health condition violating the patients' privacy ([19], [20]). In addition, research on Google's auto-complete system 'Smart compose' [21] trained on user e-mails showed that such a model memorizes long random numbers, e.g., social security numbers, that can be extracted by prompting the model [17]. This is not a theoretical threat as [18] were able to extract Personal Identifiable Information (PII) such as names, phone numbers and e-mail addresses from the language model GPT-2 [22].

A. Attack Scenarios

The fact that a machine learning model memorizes parts of its training data can be exploited by two major attacks. The most prominent attack is the Membership Inference Attack (MIA), where an adversary tries to infer whether some data point was a member of the training data, hence was used to train the model under attack [16]. While most of the proposed attacks assume access to the output vector of the model (grey-box scenario) [19], some attacks also work on output labels (black-box scenario) [23]. The intuition behind MIA is, that a model is expected to be more "certain" predicting the label or token for data it has seen during training than for unknown data. An adversary can use this information to train a model that predicts the membership status of a given data point. To test and train this model the adversary needs a dataset for which they can be sure that it was part of the training dataset. For proprietary models, the training dataset is usually not available, but as [16] have shown a dataset from a similar distribution is sufficient. Recalling the attack scenario in the previous section a manufacturer that has access to the model would know if its data were used, since their approval is needed when using their data. Furthermore, their dataset is from a similar distribution if they previously handed in documentation for the conformity assessment.

The authors of [16] trained so-called *shadow models* on that data and then queried with unknown data and data it has seen while training. This results in a dataset for classification where a data point consists of the output vector of the shadow models with a binary label indicating whether that data point was part of the training dataset. After being trained on this dataset the attack model is able to predict the membership for a given data point by using the output of the model under attack. This reference based approach has been applied to LLMs by [18] but remains computationally expensive for large models. Other approaches calculate the model's loss over the target sample [24] and extending approaches such as calibrating with zlib entropy [18], and a neighborhood comparison approach [25].

Generative models such as GPT have been shown to be particular vulnerable to attacks that aim to gain PIIs ([18], [26]). For example, *PII reconstruction* aims to reconstruct PII for a given data point. In the case of language models the adversary queries the model with an incomplete sentence or masked item for the to be reconstructed PII, for example "John Doe lives in [MASK], England". The missing item can be filled by a masked language model and the resulting sentence acts as a query to the target model. The perplexity of the target model is then used to infer whether the sentence has been seen during training. Perplexity is the measure for a generative model on how "surprised" by a token the model is. A variant of PII reconstruction is *PII inference* where the adversary wants to infer which item in a set of candidates was part in the training. As for PII reconstruction, the PII candidate is inferred by the lowest perplexity. In the scenario of conformity assessment a adversarial manufacturer could prompt the model for parameters of a known product from a competitor. Both of these attacks assume that the adversary has background knowledge on what PII to extract from the model. However, [26] showed that it is also possible to extract PII by simply generating text. The intuition behind this attack is that the model tends to generate text it has seen during training. The authors generated thousands of sentences and

used a named entity recognition algorithm such as flair[9] or spaCy[10] to find text with PIIs. By cross-checking with the developers of GPT at OpenAI they found that their method was able to extract 23% of PII with a precision of 30%.

B. Quantifying LLM Confidentiality

The leakage of sensitive information through language models is closely tied with memorization of training data ([17], [27], [28]). The intuition behind this observation is that the model first generalizes over data through the first epochs of training and then starts to memorize the data in later epochs. This is due to the fact that it encounters the same text multiple times and adjusts its weights accordingly. In its extreme case, memorization leads to overfitting, where the weights of the model shifted towards the training dataset unable to generalize from it any more. Overfitting is indicated by a bigger difference between the evaluation metric for the training and validation set. While overfitting is seen as a natural marker for memorization and thus information leakage, it has been shown that training data does get memorized without any overfitting of the model ([28], [27]).

Memorization in a model and thus the likelihood to leak its training data is usually measured by the performance of an MIA. While it might be tempting to use accuracy as a metric of performance, it lacks expressiveness when it is applied in the context of confidentiality. While some data might not be extracted and thus the false negative rate rises, false positives directly influence the usability of the attack model by diluting its positive results [29]. Area Under ROC Curve is a slightly more informative measure as it takes different classification thresholds into account. Still, it is an aggregate measure that fails to give a good sense on whether an attack delivers successful results with a low false positive rate. The authors of [29] therefore suggest reporting the true positive rate at extremely low false positive rates, e.g., at $0.1\,\%$.

Another way of evaluating memorization in a model is to measure exposure or extractability. In [17], *canary* text is inserted into the training data holding a "secret" random number. To measure how much a specific canary is memorized by the model they calculate an *exposure* metric using the log-perplexity of the sequence. The authors of [17] report a positive correlation for the number of insertions of a canary and exposure, hence the degree of memorization. They tested, how exposure influences the probability of the canary sequence to be extracted and found that when exposure exceeds a certain threshold, in their case 30, the probability of extraction quickly shifts from near 0 to near 1. This hints that the more a sequence is memorized by a model, the more likely it is to be extracted, by accident or by a malicious actor.

For [18], a string is extractable from an language model (LM) if there exist a prefix or context for which the LM outputs the string. From that they give a definition of memorization where a string is "k-eidetic memorized" if it is extractable from the LM and occurs in k examples of the training data. Hence, if a string is only present in a few documents and can be extracted, it is much worse that if a string occurs all over the training data [18]. Measuring k-eidetic memorization is thus a good method to determine whether a model is vulnerable to disclose parts of its training data.

In [27], a slightly different notion of memorization is used to study the effects of memorization. It defines a string as memorized if there exist a string s and a prompt p such that the output of the LM is equal to s when prompted with p. The authors of [27] found an effect of model size on the speed of memorization. Smaller models need to encounter a training example more often that larger models to fully memorize it. Thus, when training larger models the danger of memorizing sensitive data increases. In [30] also the length of the prompt is taken into account. It reports that larger models generally memorize more of its training data. Not only in overall quantity but also for the particular string. Smaller models tend to output only fractions of a training example or only thematically similar text. In accordance with [18], it found that repetition of examples in the training data increases memorization. Furthermore, if a prompt to a model is longer, then more memorization of the model is discovered. Interestingly, [30] found that some tokens require more context to be extracted from the model.

Furthermore, recent work [31] has tried to predict memorization from smaller models to larger ones in order to give developers a hint, if the model shows unwanted behavior. Even though it was found that small models might not act as a forecast for bigger models, this direction of research is still a challenging path to follow, as different forecasting methods are still left untouched.

C. Mitigations against Privacy Leaks

Information leaks of language models can be mitigated at different levels in the development and deployment of these models. With regard to memorization of PII equivalent material, sanitation of the training data is practical method. This can be done by blacklisting sensitive strings and removing them from the training data. However, [17] notes that this approach, while being best practice, is far from being perfect and can still miss sensitive strings. Moreover, [26] show that while PII scrubbing reduces the extraction rate, it does not protect against membership inference attacks.

As [30] and [17] have discussed, the number of occurrences of a specific training examples increase their chance to be memorized. This observation is in accordance with [26] and [32]. Intuitively, removal of duplicate training examples seems as a promising starting point to reduce memorization in a language model. In [32], the authors showed that by deduplicating the training data, they were able to lower the chance of a membership inference attack. Furthermore, deduplication also benefits the model performance itself, when duplicates are removed between the training and the test set [33].

[9]https://huggingface.co/flair/ner-english-ontonotes-large
[10]https://spacy.io

In addition to methods that apply in the pre-training stage of the language model, which should reduce the overall memorization of the model, privacy preserving training methods such as differentially private stochastic gradient decent (DP-SGD) [34] can be used. In [17], it is shown that by using DP-SGD, the exposure of their inserted canaries dropped significantly. The learning algorithm bases on Differential Privacy (DP) [35] that gives a strong privacy guarantee to individual training examples. While in differential private databases the privacy guarantee is given to individual rows, the situation in huge text corpora is different. It might make sense to apply differential privacy on per document level, yet private and sensitive text might occur across multiple documents. Moreover in the domain of conformity assessment, a manufacturer might use the same components across multiple instruments and thus sensitive information about that component are distributed over multiple documents. Thus, a carefully defined usage of differential private learning algorithms is necessary to protect training data from being leaked. Despite DP-SGD presents itself as an optimal mitigation against MIAs and extraction attacks, it is far from being perfect, since it comes with a utility cost manifesting in increased compute and decrease performance of the model.

Sensitive information can also be protected in a post-training stage. Simply filtering out sensitive information during inference is a naive approach that cannot hold to its promises. The authors [36] reported that a filter-approach can prevent generating verbatim text from the training data. However, the model is still able to produce text holding sensitive information by avoiding verbatim repetition and generating alternative texts with synonyms. Applying DP to inference of the model is another approach of preserving privacy of the training data. In [37], multiple LMs were fine-tuned with disjoint private data. During inference all models are queried and if all models come to a consensus about the predicted token, the generated token is seen as not holding private data. On the other hand, if the models disagree the prediction of a public model is mixed in. While this approach achieves comparable privacy to DP-SGD, storage and computation increase.

As has been shown above, data curation methods such as deduplication can significantly reduce memorization of training data but do not fully prevent a model from leaking private data. Differential Private Learning on the other hand can give such a privacy guarantee, but suffers from increased compute and is prone to ill-defined privacy scopes. It also comes with a utility cost. Filtering a models output only prevent certain text from being generated by do not apply for non-verbatim extraction.

V. Opportunities for further research

In the following, we outline directions for future research aimed at assessing and mitigating information leakage risks in LLMs within the context of conformity assessment.

Building on the attacks outlined in Section IV-A, further research should focus on evaluating their impact on the proposed system, particularly for embedding or classification models, as these are most relevant. In order to evaluate these attacks, a notion of sensitive strings need to be developed. Analogously to PII for the privacy domain, extraction attacks are to be evaluated with regard to how many sensitive strings can be extracted. Furthermore, gradually weighting information leakage can help to grade the severity of a violation of confidentiality. Additionally, broader forms of extraction need to be examined. While parts of the training data may contain secret sequences, it remains unclear whether entire concepts, such as novel solutions developed by the manufacturer, can be extracted. In such cases, no suitable evaluation method exists to quantify the information leakage, as current approaches rely primarily on string comparison. Semantic string comparison may offer a promising starting point for assessing whether entire concepts can be extracted from a model.

Mitigation strategies against information leakage would benefit from this research, as improved semantic string comparison and a notion of sensitive strings could enable effective data sanitation similar to PII scrubbing and support deduplication. In the long term, manufacturers, notified bodies, and conformity assessment could collaboratively define a training dataset that is safe for model training, with minimal risk of memorization. While such a data set would be a desirable solution, its development and coordination are likely to be time-consuming. In the meantime, creating synthetic data sets might be a suitable interim solution.

VI. Conclusion

In this paper, we have shown that LLMs can be utilized in conformity assessment of software in measurement devices. We sketched a system that can benefit software testers, Notified Bodies as well as manufacturers. Due to the sensitive nature of the documents involved in conformity assessment, we conducted a threat modeling using the established PASTA framework. The threat modeling yielded a possible threat of violating confidentiality. A literature review on information leakage by LLMs showed that LLMs tend to memorize parts of their training data, which can be extracted via multiple attack methods. Fortunately, mitigation such as data sanitation and differential privacy in training exist but come with a certain utility cost. Nevertheless, the overall advantages of utilizing LLMs for conformity assessment persist and the path of integrating them into an assistant system for software testers, manufacturers, and other notified bodies should be consistently followed in order to streamline conformity assessment.

References

[1] L. Gao *et al.*, "The pile: An 800gb dataset of diverse text for language modeling," 2020, arXiv:2101.00027.

[2] T. Ucedavélez and M. M. Morana, *Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis*. Hobekin: John Wiley & Sons, 2015.

[3] European Parliament and C. of the European Union, "Directive 2014/32/EU of the European Parliament and of the council," 2014. [Online]. Available: https://eur-lex.europa.eu/eli/dir/2014/32/oj

[4] WELMEC Guide 7.2, "Software guide (EU measuring instruments directive 2014/32/EU," 2023. [Online]. Available: https://www.welmec.org/welmec/documents/guides/7.2/2023/WELMEC_Guide_7.2_2023.pdf

[5] K. Spärck Jones, "A statistical interpretation of termspecificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972. doi: 10.1108/eb026526

[6] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[7] A. Wang *et al.*, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf

[8] A. Dubey *et al.*, "The llama 3 herd of models," 2024, arXiv:2407.21783.

[9] OpenAI *et al.*, "Gpt-4 technical report," 2024, arXiv:2303.08774.

[10] Gemini Team *et al.*, "Gemini: A family of highly capable multimodal models," 2024, arXiv:2312.11805.

[11] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021, arXiv:2005.11401.

[12] L. Kohnfelder and P. Garg, "The threats to our products," *Microsoft Interface*, 1999. [Online]. Available: https://adam.shostack.org/microsoft/The-Threats-To-Our-Products.docx

[13] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011. doi: 10.1007/s00766-010-0115-7

[14] M. Esche and F. Thiel, "Software risk assessment for measuring instruments in legal metrology," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds., vol. 5. IEEE, 2015. doi: 10.15439/2015F127 pp. 1113–1123.

[15] Z. Ji *et al.*, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, Mar. 2023. doi: 10.1145/3571730

[16] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, " Membership Inference Attacks Against Machine Learning Models ," in *2017 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2017. doi: 10.1109/SP.2017.41. ISSN 2375-1207 pp. 3–18.

[17] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019. ISBN 978-1-939133-06-9 pp. 267–284. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/carlini

[18] N. Carlini *et al.*, "Extracting training data from large language models," 2021, arXiv:2012.07805.

[19] F. Mireshghallah, K. Goyal, A. Uniyal, T. Berg-Kirkpatrick, and R. Shokri, "Quantifying privacy risks of masked language models using membership inference attacks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022. doi: 10.18653/v1/2022.emnlp-main.570 pp. 8332–8347.

[20] E. Lehman, S. Jain, K. Pichotta, Y. Goldberg, and B. Wallace, "Does BERT pretrained on clinical notes reveal sensitive data?" in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, K. Toutanova *et al.*, Eds. Online: Association for Computational Linguistics, Jun. 2021. doi: 10.18653/v1/2021.naacl-main.73 pp. 946–959.

[21] M. X. Chen *et al.*, "Gmail smart compose: Real-time assisted writing," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019. doi: 10.1145/3292500.3330723. ISBN 9781450362016 p. 2287–2295.

[22] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[23] W. Shi *et al.*, "Detecting pretraining data from large language models," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=zWqr3MQuNs

[24] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018. doi: 10.1109/CSF.2018.00027 pp. 268–282.

[25] J. Mattern, F. Mireshghallah, Z. Jin, B. Schoelkopf, M. Sachan, and T. Berg-Kirkpatrick, "Membership inference attacks against language models via neighbourhood comparison," in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023. doi: 10.18653/v1/2023.findings-acl.719 pp. 11 330–11 343.

[26] N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Analyzing leakage of personally identifiable information in language models," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023. doi: 10.1109/SP46215.2023.10179300 pp. 346–363.

[27] K. Tirumala, A. Markosyan, L. Zettlemoyer, and A. Aghajanyan, "Memorization without overfitting: Analyzing the training dynamics of large language models," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 38 274–38 290. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/fa0509f4dab6807e2cb465715bf2d249-Paper-Conference.pdf

[28] F. Mireshghallah, A. Uniyal, T. Wang, D. Evans, and T. Berg-Kirkpatrick, "Memorization in nlp fine-tuning methods," 2022, arXiv:2205.12506.

[29] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022. doi: 10.1109/SP46214.2022.9833649 pp. 1897–1914.

[30] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural language models," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023. [Online]. Available: https://openreview.net/forum?id=TatRHT_1cK

[31] S. Biderman *et al.*, "Emergent and predictable memorization in large language models," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 28 072–28 090. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/59404fb89d6194641c69ae99ecdf8f6d-Paper-Conference.pdf

[32] N. Kandpal, E. Wallace, and C. Raffel, "Deduplicating training data mitigates privacy risks in language models," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 17–23 Jul 2022, pp. 10 697–10 707. [Online]. Available: https://proceedings.mlr.press/v162/kandpal22a.html

[33] K. Lee *et al.*, "Deduplicating training data makes language models better," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, May 2022. doi: 10.18653/v1/2022.acl-long.577 pp. 8424–8445.

[34] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: Association for Computing Machinery, 2016. doi: 10.1145/2976749.2978318 p. 308–318.

[35] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the Third Conference on Theory of Cryptography*, ser. TCC'06. Berlin, Heidelberg: Springer-Verlag, 2006. doi: 10.1007/11681878_14 p. 265–284.

[36] D. Ippolito *et al.*, "Preventing generation of verbatim memorization in language models gives a false sense of privacy," in *Proceedings of the 16th International Natural Language Generation Conference*, C. M. Keet, H.-Y. Lee, and S. Zarrieß, Eds. Prague, Czechia: Association for Computational Linguistics, Sep. 2023. doi: 10.18653/v1/2023.inlg-main.3 pp. 28–53.

[37] A. Ginart, L. van der Maaten, J. Zou, and C. Guo, "Submix: Practical private prediction for large-scale language models," 2022, arXiv:2201.00971.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 69–73
DOI: 10.15439/2025F2775
ISSN 2300-5963 ACSIS, Vol. 44

Integrating Real-ESRGAN with CNN Models for UAV Image Based Plant Disease Detection

Sravya Malladi
Independent Researcher
San Jose, California, USA
sravya.malladi@gmail.com

Pranav Kulkarni
Stanford University
Palo Alto, California, USA
pranavsk@stanford.edu

***Abstract*—The integration of deep learning models with UAV captured images for plant disease detection has been explored in many papers and has the potential to revolutionize commercial precision agriculture, by allowing for early and efficient detection and classification of crop disease stages. In order to address the limitations posed by low-resolution aerial imaging, this paper proposes the additional integration of an Enhanced Super Resolution Generative Adversarial Network (ESRGAN) with a Convolutional Neural Network model for field monitoring through UAV captured imagery. UAVs are a cost effective method of monitoring large swaths of agricultural land; however, it is difficult to capture images of a high enough quality and clarity to be adequately analyzed by a CNN. The images typically lack the necessary resolution for accurate classification, especially for diseases with smaller, less noticable symptoms. The Real-ESRGAN model is employed to generate a dataset of high-resolution images, from low-resolution inputs, allowing the disease detection CNN to more accurately and effectively identify and classify disease stages in Armillaria afflicted cherry trees. This solution offers a solution to the problem posed by traditional UAV based approaches that enhances classification accuracy even in suboptimal conditions. Through this integrated approach, the model was able to reach an increased validation accuracy, as well as significantly decreased loss values due to the ESRGAN enhanced imagery allowing for clearer detection of early stage Armillaria symptoms. This integrated system provides a practical scalable solution for commercial agriculture, allowing for more comprehensive and efficient crop disease monitoring. Future research can be explored to optimize the architecture of this model and expand its applicability to other crops and environmental conditions, allowing more efficient precision agriculture and paving the way for more sustainable farming practices.**

***Index Terms*—Crop Monitoring, Enhanced Super Resolution GAN, Deep Learning, UAV imagery, Precision Agriculture**

I. Introduction

THE advancement of deep learning has impacted commercial agriculture significantly, particularly in the classification and recognition of plant diseases. Traditional crop inspection methods are prone to human errors such as psychological and cognitive biases [1]. Furthermore, the vastness of agricultural land and the scarcity of trained plant pathologists make manual monitoring impractical [2]. Deep learning, specifically Convolutional Neural Networks (CNNs), offers a promising alternative by automating disease detection and classification tasks with high accuracy [3]. In order to be properly analyzed by a CNN, UAV captured images must have a high enough resolution and clarity so that the symptoms of a plant disease can be seen. This problem is amplified when said plant disease has very small symptoms, or when the farm area is very large. The proposed solution to this limitation is targeted sampling of a field, where images are acquired from a small section of the field's area. While this is a solution would provide farmers more information than the traditional methods of scouting for diseases, the difficulties faced by a model when analyzing UAV images image still stand. The Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) model detailed in the section below presents a method for crop disease classification on low resolution images. The model is used to generate high resolution images from low resolution crop images. This is called Image Super Resolution (SR). Though there are SR methods besides ESRGAN, the paper shows that the ERAGAN model, and its successor the Real-ESRGAN model, generate higher visual quality images than other methods used [4][5].

II. Literature Review

A. Literature Review

In commercial agriculture, identifying disease severity is crucial for making timely and effective decisions to reduce financial losses and fight plant infections[6]. Machine Learning models that classify different stages of a plant disease, are therefore, most helpful. For example, the regression model proposed in Detection and Characterization of Stressed Sweet Cherry Tissues Using Machine Learning identifies different stages of Amarilma, a devastating cherry tree disease that causes annually 8 million dollars in losses in the United States alone[7]. Commercial farmers use aerial and satellite imagery to monitor their crop fields. According to The application of small unmanned aerial systems for precision agriculture: a review, UAV captured aerial imagery is a cost effective solution that can be used for crop disease detection, reducing the need for in person monitoring [8][9]. The use of UAVs paired with detection technologies, is a transformative practice that will greatly facilitate the practicality of precision agriculture[9]. These technologies enhance crop monitoring, optimize resource use, and minimize the environmental footprint of farming by reducing the application of fertilizers and pesticides. However, despite their success, these techniques face challenges when applied in real-world agricultural conditions. In order to be properly analyzed by a model,

UAV captured images must have a high enough resolution and clarity so that the symptoms of a plant disease can be seen. An image of this quality is hard to capture with a UAV which is subject to wind, lighting conditions, and other environmental challenges [9][11]. As detailed in Millimeter-Level Plant Disease Detection From Aerial Photographs via Deep Learning and Crowdsourced Data, this problem is amplified when said plant disease has very small symptoms, or when the farm area is very large [12]. To address low-resolution imagery, several approaches can be considered. One example is hyperspectral imagery, which captures information across dozens or hundreds of narrow spectral bands. This can reveal subtle physiological and biochemical changes in plants that are not visible in standard RGB images, improving disease detection even at lower spatial resolutions. However, specialized hyperspectral cameras are expensive, and the data they produce is extremely large and complex, requiring extensive preprocessing, calibration, and storage [13]. These requirements make hyperspectral imaging difficult to implement efficiently for routine agricultural applications. Another alternative is SRCNN (Super-Resolution Convolutional Neural Network). SRCNN is of the earliest deep learning based super resolution models that employs a three-layer CNN architecture to upscale images. However, although this architecture is straightforward and computationally efficient, it has limited capacity to capture complex textures or fine details, which are critical for early detection of diseases with small and difficult to see symptoms [14]. Another proposed solution for this problem is to use GAN models for data augmentation [2][15][16]. In contrast, GAN-based super-resolution methods offer a highly practical alternative. GANs can be used to enhance the resolution of standard RGB images captured by UAVs, allowing models to detect fine disease symptoms without the need for specialized sensors, requiring no extra hardware beyond a conventional camera. Generated images and lesions are used for data augmentation, where the model is trained on an expanded dataset. GAN-based models are capable of reconstructing realistic textures and subtle visual cues. Therefore augmenting data with GAN super resolution models can greatly improve the effectiveness of plant disease classification models[15][16][17] The best GAN model for this purpose seems to be the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) model. Super-resolution models are designed to reconstruct high-resolution (HR) images based on low-resolution (LR) inputs. When given a low-resolution image $X \in \mathbb{R}^{h' \times w' \times c}$, the model generates a corresponding high-resolution image $Y \in \mathbb{R}^{h \times w \times c}$, where $h > h'$ and $w > w'$[18]. The ESRGAN model improves upon the earlier SRGAN framework by emphasizing perceptual realism. Its generator network employs deep Residual-in-Residual Dense Blocks (RRDBs) to extract important features and recreate fine details in the images. The model also uses a Relativistic average GAN (RaGAN) discriminator, which compares the generated images to real images. A crop disease detection and classification model integrated with ESRGAN has been shown to be better at detecting crop disease than the models that use other Super Resolution methods, with a higher classification accuracy due to greater visual quality [4][19]. The REAL-ESRGAN model further expands upon the ESRGAN model, generating images of an even greater visual quality. The generator network of this model is trained using a combination of content, perceptual, and adversarial losses [5]. Content loss measures pixel-wise similarity between real-world and generated images:

$$L_{\text{content}} = \frac{1}{N} \sum_{i=1}^{N} \|G(X_i) - Y_i\|_2^2$$

Perceptual loss encourages high-level similarity using feature maps from a pre-trained network, measuring how real the images appear in terms of patterns, textures, and shapes:

$$L_{\text{perceptual}} = \frac{1}{N} \sum_{i=1}^{N} \|\phi_j(G(X_i)) - \phi_j(Y_i)\|_1$$

Adversarial loss guides the generator to produce images that are difficult for the discriminator to distinguish from real HR images:

$$L_{\text{GAN}}(G, D) = \mathbb{E}_Y \Big[\log(D(Y) - \mathbb{E}_X[D(G(X))]) \Big] + \mathbb{E}_X \Big[\log(1 - (D(G(X)) - \mathbb{E}_Y[D(Y)])) \Big]$$

The overall generator loss is a weighted combination of these components:

$$L_G = L_{\text{content}} + \lambda \cdot L_{\text{perceptual}} + \eta \cdot L_{\text{GAN}}$$

where λ and η balance the contributions of perceptual and adversarial losses.

The discriminator outputs a probability map indicating the likelihood that each pixel belongs to a real image. The discriminator loss is calculated using a weighted sum over all pixels. Spectral normalization is added in Real-ESRGAN to regularize the discriminator weights:

$$\hat{W} = \frac{W}{\sigma(W)}$$

Through these enhancements, the model is better able to handle real-world image degradation, making it more reliable for practical image restoration applications and an effective approach for improving the quality of aerial images [5][19][20][21].

III. Methodology

The dataset used in this study is the Cherry Tree Disease Detection dataset from the article above, Detection and Characterization of Stressed Sweet Cherry Tissues Using Machine Learning, which contains both hyperspectral and standard JPG images of cherry trees at different stages of Armillaria infection. The stages represented are healthy, stage 1, and stage 2. To prepare the dataset for analysis, the images that were originally organized by the day of data collection were consolidated into broader categories corresponding to each disease stage. This reorganization facilitated the removal of

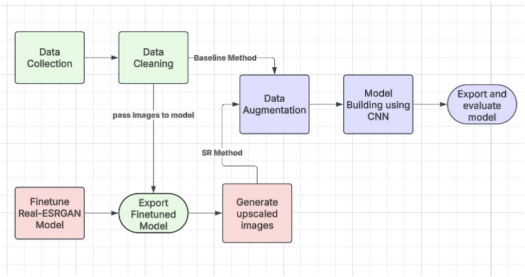


Fig. 1. Methodology

irrelevant or redundant images and ensured that the dataset was structured consistently for model training and evaluation.

The first stage of the project employed a Convolutional Neural Network (CNN) in TensorFlow to classify images of cherry trees into the three categories described above. Data augmentation techniques, such as resizing and rescaling, were applied to increase the size and diversity of the training dataset, therefore increasing the model's robustness. After data cleaning and augmentation, the images were divided into training and validation sets, which were subsequently used for model development and evaluation.

The model architecture consisted of 64 convolutional layers, including pooling and fully connected layers. The structure used allowed the model to learn the features and patterns in the images associated with each different stage of the disease.

In the second stage of the project, the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) was integrated and used to generate a new higher resolution version of the original cherry tree dataset. The ESRGAN model was fine-tuned on a super resolution dataset composed of paired high-resolution and low-resolution images, thereby enhancing image resolution and clarity.

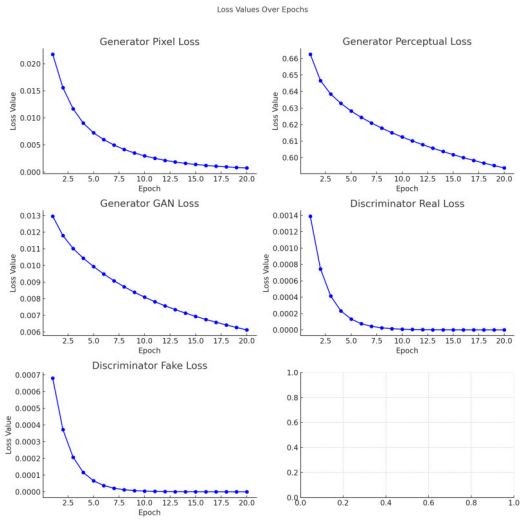


Fig. 2. Graph of various losses over epochs

The fine-tuning process began with downloading the publicly available Real-ESRGAN model from its GitHub repository. The options file was then modified so that the training and validation sections used the custom dataset of paired high-resolution and low-resolution images. Once the training script was executed, the dataset was iteratively adjusted to improve super resolution performance. After several refinements, this process yielded an optimized balance between image clarity and model runtime.

Fig. 3. Image up-scaling

By running the finetuned ESRGAN model on the original Cherry Tree Dataset, high-resolution versions of images from the dataset were generated. The generated images were used as input for classification, allowing for a direct comparison of performance and accuracy between the baseline CNN and the ESRGAN augmented CNN.

A. Results

Fig. 4. Baseline model

The integrated system achieved a 94 percent validation accuracy in classifying cherry tree disease stages, compared to 83 percent for the original CNN model. Images enhanced by ESRGAN consistently produced higher accuracy and greater confidence values during classification. Additionally, loss values decreased significantly when compared directly with the

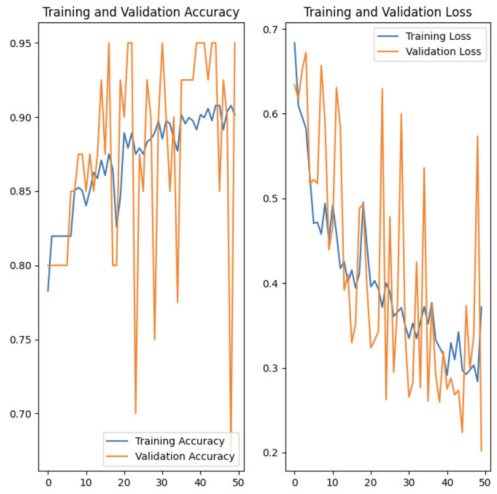


Fig. 5. New model

original CNN trained on the standard dataset. This is due to the fact that the enhanced model could detect subtle symptoms of Armillaria that were difficult to discern in lower-resolution images. Therefore, the combined use of the two models allowed the system to classify plant disease stages more effectively, even from lower-quality images, compared to using the original CNN alone.

B. Discussion and Applications

The increased validation accuracy achieved by integrating the two models demonstrates its effectiveness in discerning between healthy cherry trees and those in various stages of Armillaria. These results show the importance of high resolution imaging in plant disease detection, validating the performance of the integrated system.

This accuracy level shows that the ESRGAN enhanced images provide greater visual clarity for the CNN to make proper classifications, compared to the original images. The results show that the CNN will be able to make reliable classifications, even when the original UAV captured images lack resolution,because of the integration of the ESRGAN Model, suggesting that the system has successfully mitigated the challenges posed by the difficulty of getting high resolution crop images from UAVs.

The results reflect that the upscaled images provided by ESRGAN significantly improve the model's ability to detect Armillaria symptoms, which is crucial for timely intervention in commercial cherry tree farming. The high accuracy and confidence levels reflected by this system means that commercial farms could rely on this to detect and classify Armillaria disease stages with few errors, increasing the efficiency of crop inspection, and allowing for better monitoring of plant disease. This in turn allows for early stage detection, allowing for better disease management, minimizing yield loss.

This method solves the problems from previous papers that struggled with the limits posed by the resolution of UAV captured images. By introducing a super resolution model, however, these become much less challenging. Unlike previous models that required time consuming targeted sampling to be practical for commercial farming, this system allows for more efficient analysis of broader areas, with a stable accuracy.

So, the results prove that the system is both effective and practical for commercial farming in the real world. By enhancing image clarity, ESRGAN allows the CNN to easily identify lesions that would be too hard for models that take in lower resolution images, proving that super resolution techniques can help improve ML models in agriculture.

C. Conclusion

The research presents a large advancement in crop disease detection through the integration of ESRGAN model and CNN model to classify UAV captured images. By successful enhancement and accurate classification, the model shows near perfect accuracy in identifying the stages of Armillaria in cherry trees. The integrated hybrid approach addresses the challenge posed by low quality images captured by UAVs, providing a proper solution for real world applications in commercial agriculture.

Areas for future research would include expanding the dataset to include a wider range of diseases. Additionally, working on detection for other, more widely grown crop types would both increase the model's robustness, but also its real world applicability. Exploration of different enhancement techniques along with ESRGAN could cause improvements in classification accuracy. It would be prudent to investigate the model's performance in various environmental conditions, like harsh weather, or unclear lighting, to see if it would still perform as well, providing insights into practicality.

Another avenue for future research includes optimization of the model architecture. Exploring different configurations and techniques, like using larger pretrained models that would have to be finetuned (YOLO) could enhance the system's performance [22]. This model would not only classify different images, but also identify the specific locations of each individual lesion. This would help the practicality of the system proposed in this paper immensely, because it would allow images to be taken over a broader region, and would allow for more precise disease identification. This was not used in this paper due to lack of available data.

Using UAV data for predictive modeling has strong applications in the future as well. In conclusion, the research establishes a foundation for leveraging Super Resolution image augmentation techniques for Agricultural disease classification, paving the way for solutions that enhance productivity and sustainability in farming.

REFERENCES

[1] C. H. Bock, G. H. Poole, P. E. Parker, and T. R. Gottwald, "Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging," Crit. Rev. Plant Sci., vol. 29, no. 2, pp. 59–107, 2010. [Online]. Available: https://doi.org/10.1080/07352681003617285

[2] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," Biosyst. Eng., vol. 172, pp. 84–91, 2018. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2018.05.013

[3] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Comput. Electron. Agric., vol. 145, pp. 311–318, 2018. [Online]. Available: https://doi.org/10.1016/j.compag.2018.01.009

[4] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in Proc. Eur. Conf. Comput. Vis. Workshops, 2018, pp. 63–79. [Online]. Available: https://doi.org/10.1007/978-3-030-11021-5_5

[5] X. Wang, L. Xie, and C. Dong, "Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data," in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops, 2021, pp. 1905–1914. [Online]. Available: https://doi.org/10.1109/ICCVW54120.2021.00217

[6] X. Zeng and Y. Ma, "GANs-based data augmentation for citrus disease severity detection using deep learning," Agronomy, vol. 10, no. 12, p. 1939, 2020. [Online]. Available: https://www.mdpi.com/2073-4395/10/12/1939

[7] C. Chaschatzis et al., "Detection and characterization of stressed sweet cherry tissues using machine learning," Remote Sens., vol. 12, no. 3, p. 531, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/3/531

[8] S. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: A review," Precision Agric., vol. 13, pp. 693–712, 2012. [Online]. Available: https://link.springer.com/article/10.1007/s11119-012-9274-5

[9] H. Zhu et al., "Intelligent agriculture: Deep learning in UAV-based remote sensing imagery for crop diseases and pests detection," Front. Plant Sci., vol. 15, 2025. [Online]. Available: https://doi.org/10.3389/fpls.2024.1435016

[10] A. D. Boursianis et al., "Internet of Things (IoT) and agricultural unmanned aerial vehicles (UAVs) in smart farming: A comprehensive review," Internet Things, vol. 18, p. 100187, 2022. [Online]. Available: https://doi.org/10.1016/j.iot.2020.100187

[11] J. Agrawal and M. Y. Arafat, "Transforming farming: A review of AI-powered UAV technologies in precision agriculture," Drones, vol. 8, no. 11, p. 664, 2025. [Online]. Available: https://doi.org/10.3390/drones8110664

[12] O. Bongomin et al., "UAV image acquisition and processing for high-throughput phenotyping in agricultural research and breeding programs," Plant Phenome J., vol. 7, no. 1, e20096, 2025. [Online]. Available: https://doi.org/10.1002/ppj2.20096

[13] A. Purushothaman and R. Pannerselvam, "Hyperspectral imaging and its applications: A review," Front. Bioeng. Biotechnol., vol. 12, pp. 123456, Jun. 2024. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC11253060/

[14] S. Ramesh and S. Srinivas, "RDA-CNN: Enhanced Super Resolution Method for Rice Plant Disease Classification," Comput. Syst. Sci. Eng., vol. 42, no. 1, pp. 273–287, 2022. [Online]. Available: https://www.techscience.com/csse/v42n1/45755

[15] S. A. Wahabzada et al., "Millimeter-level plant disease detection from aerial photographs via deep learning and crowdsourced data," Front. Plant Sci., vol. 9, p. 1453, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpls.2018.01453/full

[16] K. Sathya and M. Rajalakshmi, "RDA-CNN: Enhanced Super Resolution Method for Rice Plant Disease Classification," Comput. Syst. Sci. Eng., vol. 42, no. 1, pp. 33–47, Jul. 2022. [Online]. Available: https://www.techscience.com/csse/v42n1/45755/html

[17] A. ul Haq and S. Kaur, "Super resolution image based plant disease detection and classification using deep learning techniques," Propuls. Tech. J., vol. 45, no. 1, pp. 1020–1022, 2024. [Online]. Available: https://www.propulsiontechjournal.com/index.php/journal/article/view/4108

[18] L. Bi and G. Hu, "Improving image-based plant disease classification with generative adversarial network under limited training set," Front. Plant Sci., vol. 11, 2020. [Online]. Available: https://doi.org/10.3389/fpls.2020.583438

[19] J. Wen, Y. Shi, X. Zhou, and Y. Xue, "Crop disease classification on inadequate low-resolution target images," Sensors, vol. 20, no. 16, p. 4601, 2020. [Online]. Available: https://doi.org/10.3390/s20164601

[20] Ş. B. Çetin, "Real-ESRGAN: A deep learning approach for general image restoration and its application to aerial images," Advanced Remote Sensing, vol. 3, no. 2, pp. 90–99, 2023. [Online]. Available: https://publish.mersin.edu.tr/index.php/arsej/article/view/1072

[21] F. Rezapoor Nikroo et al., "A comparative analysis of SRGAN models," arXiv preprint, arXiv:2307.09456, 2023. [Online]. Available: https://arxiv.org/abs/2307.09456

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 75–80
DOI: 10.15439/2025F8176
ISSN 2300-5963 ACSIS, Vol. 44

Interpreting NAS-Optimized Transformer Models for Remaining Useful Life Prediction Using Gradient Explainer

Messaouda Nekkaa
ORCID: 0000-0002-6472-8266
University M'hamed Bougara of Boumerdes
LIST / Electrical Systems Engineering Department
35000 Boumerdes, Algeria
Email: m.nekkaa@univ-boumerdes.dz

Mohamed Abdouni
Sonatrach Industry
Djenane El Malik, Hydra, 16111 Algiers, Algeria
Email:mohamed.abdouni@sonatrach.dz

Dalila Boughaci
ORCID: 0000-0001-5210-8951
University of Science and Technology Houari Boumediene
LRIA / Computer Science Department
BP 32 El-Alia, Bab Ezzouar, 16111 Algiers, Algeria
Email: dalila_info@yahoo.fr, dboughaci@usthb.dz

***Abstract*—Remaining Useful Life (RUL) estimation of complex machinery is critical for optimizing maintenance schedules and preventing unexpected failures in safety-critical systems. While Transformer architecture has recently achieved state-of-the-art performance on RUL benchmarks, their design often relies on expert tuning or costly Neural Architecture Search (NAS), and their predictions remain opaque to end users. In this work, we integrate a Transformer whose hyperparameters were discovered via evolutionary NAS with a gradient-based explainability method to deliver both high accuracy and transparent, per-prediction insights. Specifically, we adapt the Gradient Explainer algorithm to produce global and local importance scores for each sensor in the C-MAPSS FD001 turbofan dataset. Our analysis shows that the sensors identified as most influential, such as key temperature and pressure measurements, match domain-expert expectations. By illuminating the internal decision process of a complex, NAS-derived model, this study paves the way for trustworthy adoption of advanced deep-learning prognostics in industrial settings.**

***Index Terms*—Remaining Useful Life (RUL), Transformers, Neural Architecture Search (NAS), Explainable AI (XAI), Gradient Explainer, C-MAPSS, Interpretability.**

I. Introduction

PROGNOSTICS and Health Management (PHM) plays a critical role in modern industrial systems, enabling increased reliability, optimized maintenance, and the prevention of catastrophic failures in high-value assets such as aircraft engines and manufacturing equipment [1]. A core component of PHM is the accurate estimation of Remaining Useful Life (RUL), the time before a component or system can no longer perform its intended function.

The rise of deep learning has significantly advanced RUL prediction. Recurrent Neural Networks (RNNs) [2], and more recently Transformer-based architectures [3], have demonstrated strong performance due to their ability to model complex temporal dependencies in multivariate sensor data.

Building on these advances, Mo Hyunho et al. [4] proposed a Neural Architecture Search (NAS) framework using evolutionary algorithms to automatically discover optimal Transformer architectures for RUL prediction. Applied to the well-established C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset [6], their NAS-derived Transformers outperformed manually designed alternatives, setting a new performance benchmark [4].

Despite these gains, deep-learning complex models often operate as "black boxes" [7]. Their complex, high-dimensional structures obscure the reasoning behind predictions.

In safety-critical settings, this lack of interpretability is a major barrier to adoption, where understanding why a model predicted a specific RUL is essential for trust, verification, and regulatory acceptance.

Explainable AI (XAI) seeks to address this issue by providing human-understandable insights into model behavior. However, most existing XAI studies focus on standard or simpler architectures, leaving the interpretability of NAS-derived Transformers underexplored, especially within the PHM domain [7], [8].

To our knowledge, no prior work has applied advanced gradient-based XAI techniques to these automatically discovered architectures in the context of RUL estimation.

This paper addresses that gap by adapting SHAP's Gradient Explainer ; a theoretically grounded and computationally efficient method ; for use with the NAS-optimized Transformer developed by Mo Hyunho et al. Our goal is to enhance the

Topical area: Advanced Artificial Intelligence in Applications

transparency of this state-of-the-art model by generating global and local feature attributions for RUL predictions on the C-MAPSS FD001 subset.

Our contributions are threefold:

1) **Gradient-based Explanation for NAS-Transformer:** We adapt and apply the Gradient Explainer algorithm to a NAS-optimized Transformer architecture specifically designed for RUL prediction.
2) **Global and Local Attribution Analysis:** We perform comprehensive explanation analysis, including both global sensor rankings and per-instance local saliency maps, on the FD001 subset of C-MAPSS.
3) **Actionable Insights for PHM:** We extract interpretable, domain-relevant insights into which sensors and time points most influence the model's predictions, enhancing trust, transparency, and deployability in industrial contexts.

The rest of this paper is organized as follows:

Section II reviews related work on RUL prediction and explainable AI. Section III describes the dataset, model architecture, and the adaptation of the Gradient Explainer. Section IV presents experimental results, including global and local explanations. Section V concludes with future research directions.

II. Related Work

This section reviews literature pertinent to our research, covering Remaining Useful Life (RUL) prediction with deep learning, the role of Neural Architecture Search (NAS) in Prognostics and Health Management (PHM), existing Explainable AI (XAI) techniques for complex models, and the specific challenges and advancements in explaining Transformer and NAS-optimized architectures.

A. RUL Prediction in PHM

Remaining Useful Life (RUL) refers to the time remaining before a system fails, expressed as RUL $= T - t$, where T is the failure time and t is the current time [1]. RUL estimation methods are broadly categorized into model-based and data-driven approaches. Model-based methods rely on prior physical knowledge, which can be hard to generalize in practice and may struggle with the complexities of real-world degradation processes. In contrast, data-driven approaches, particularly those leveraging deep learning (DL), have gained prominence due to their ability to learn complex patterns directly from sensor data and enabling end-to-end modeling, eliminating the need for manual feature engineering [2].

Early DL applications in RUL prediction included Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), which showed promise in feature extraction from time-series data. Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) variants, became popular for their inherent ability to model temporal dependencies in sequential sensor readings. However, RNNs can face challenges with long range dependencies and computational efficiency for long sequences [9], [10].

B. Transformer-Based Models for Time Series

Transformer architecture, originally introduced for natural language processing in the famous paper of Vaswani et al. [3], has emerged as a powerful self-attention mechanism that allows it to capture global dependencies between input sequence elements effectively, overcoming some limitations of RNNs. Consequently, Transformers have been increasingly adapted for various time-series forecasting tasks, including RUL prediction, often demonstrating superior performance.

C. Neural Architecture Search (NAS) in Deep Learning

While DL models, including Transformers, offer significant potential, their performance is highly dependent on their architecture. Designing optimal architecture manually is a time-consuming, iterative, and expertise-driven process [4]. Neural Architecture Search (NAS) has emerged as a field that automates this design process, algorithmically searching for the best-performing neural network architecture for a given task and dataset [4].

D. Explainable AI (XAI) for Complex Models

The increasing complexity and performance of DL models, especially transformer-based models with their attention characteristics, often come at the cost of interpretability, leading to their characterization as "black boxes". In safety-critical applications like PHM, this lack of transparency is a major concern, as understanding why a model makes a certain prediction is crucial for trust, debugging, and regulatory compliance. Explainable AI (XAI) encompasses a range of techniques aimed at making the decisions of AI systems more understandable to humans [11].

Common XAI methods can be broadly categorized. Perturbation-based methods, like LIME (Local Interpretable Model-agnostic Explanations), explain individual predictions by learning a simpler, interpretable model on local perturbations of the input [11].

Surrogate models aim to approximate the complex model with a more transparent one. Gradient-based methods, such as Integrated Gradients and SmoothGrad, utilize model gradients to attribute importance to input features. SHAP (SHapley Additive exPlanations), grounded in co-operative game theory, provides a unified framework for feature attribution by calculating Shapley values, which represent the marginal contribution of each feature to the prediction [12].

III. Material and methods

In this section, we present our methodological frame-work. We first describe the C-MAPSS FD001 dataset and its preprocessing pipeline. Next, we introduce the NAS-optimized Transformer architecture used for RUL predic-tion. Finally, we detail our adaptation of the SHAP Gradient Explainer for feature-attribution analysis applied to this model.

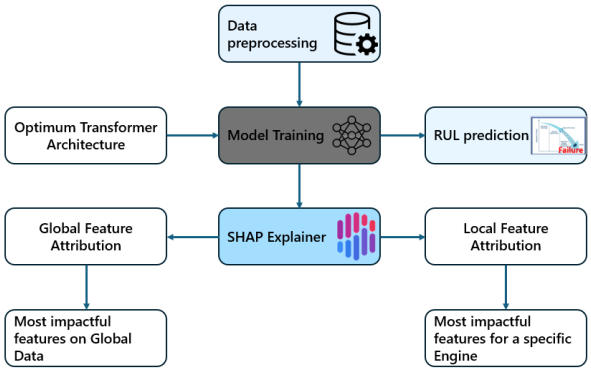


Fig. 1. Methodological Frame-Work

A. Data and Preprocessing

We base our experiments on NASA's widely used C-MAPSS (Commercial Modular Aero-Propulsion System Simulation) dataset, which simulates turbofan engine degradation under different operating conditions and fault modes. C-MAPSS comprises four subsets (FD001–FD004), each containing multivariate time-series from 21 sensors and 3 operating settings.

In this work, we focus on FD001, which models a single fault mode under one operating condition [5].

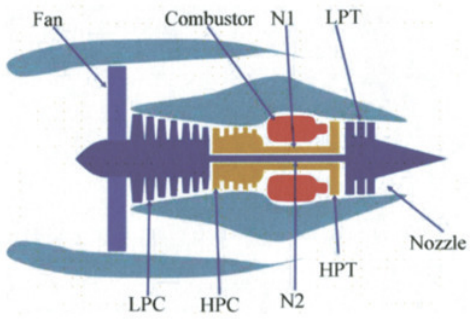


Fig. 2. Diagram of the turbofan Engine

Data preprocessing steps were aligned with those typically employed for this dataset and consistent with the foundational work [4]:

- **Sensor Selection:** From the original 21 sensor channels, we computed the 21×21 inter-sensor Pearson correlation matrix to identify constant or redundant signals. Any sensor with zero variance (constant readings) or entirely null values was removed, leaving 14 informative sensors.
- **Normalization:** All sensor and aggregate features were scaled to [0, 1] using min–max normalization, with scaling parameters fitted exclusively on the FD001 training set to avoid data leakage.
- **Windowing:** We applied a sliding window of 40 raw timesteps and appended 2 aggregate rows (slope and mean), resulting in 42-timestep sequences. The target RUL is defined as the number of cycles remaining at the final point in each window.

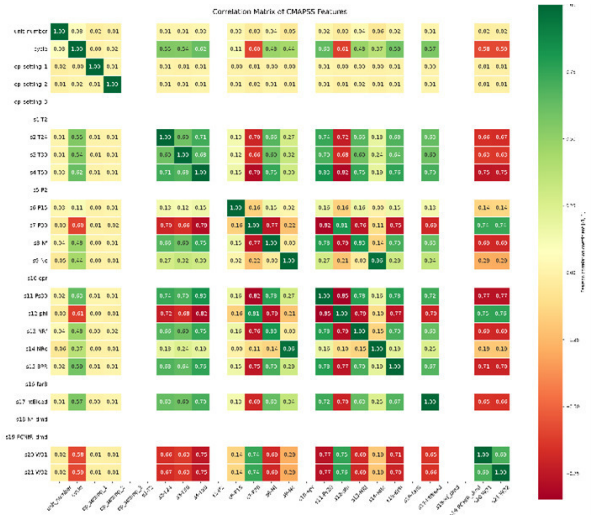


Fig. 3. Pearson's correlation matrix heat map of the Commercial Modular Aero-Propulsion System

B. Foundational NAS-Optimized Transformer Architecture

Our work builds upon the Transformer architecture developed by Mo Hyunho et al. [4], who applied Neural Architecture Search (NAS) to design high-performing models for RUL prediction. Rather than re-running their computationally intensive search process, we adopt the optimal architecture they identified as the basis for our explainability study.

This architecture was discovered using an evolutionary algorithm that explored an 11-dimensional genotype defining various hyperparameters of the Transformer, including embedding dimensions, number of attention heads, feed-forward layer dimensions, and the number of encoder/decoder layers.

The core structure of this NAS-optimized Transformer architecture, as described by Mo Hyunho et al. [4], features several key components tailored for time-series RUL prediction:

- **Input Representation:** Each input is a multivariate time-series window with 42 timesteps and 14 sensor channels, resulting in an input matrix of shape (42, 14). The 42 timesteps include 40 raw cycles and 2 aggregate features (slope and mean), as described in Section III-A.
- **Embedding and Positional Encoding:** Raw sensor readings at each timestep are first passed through an input embedding layer to project them into a higher-dimensional space (`d_model`) . To retain temporal information, sinusoidal positional encodings are added to these embeddings.
- **Dual-Encoder Mechanism:** A key feature of the architecture is its use of two parallel encoders:
  - *A Sensor Encoder:* that applies multi-head self-attention across the sensor dimension to assess inter-sensor dependencies.

– *A Timestep Encoder:* that uses self-attention across the time dimension to capture temporal patterns.
Each encoder is composed of `N_enc` layers, each containing multi-head attention and position-wise feed-forward sublayers, combined with residual connections and layer normalization.

- **Feature Fusion:**
Outputs from the sensor and timestep encoders—denoted F_s and F_t are concatenated and passed through a fusion layer:

$$\text{Fusion}(F_s, F_t) = \text{Concat}(F_s, F_t) \cdot W^F \quad (1)$$

This operation merges sensor-wise and temporal features into a unified representation.
- **Decoder:** The fused features are input to a decoder composed of N_{dec} layers, again using multi-head attention and feed-forward sublayers. The decoder processes only the final α timesteps of the encoder output $typically\alpha = 4$, focusing on recent history for prediction. Its final output is a scalar representing the estimated RUL.

We configured our model using the specific optimal genotype parameters reported by Mo Hyunho et al. [13], ensuring consistency with the NAS-discovered Transformer architecture used in their original work.

C. Gradient Explainer Algorithm

To interpret the predictions of the NAS-optimized Transformer, we adopted SHAP's Gradient Explainer [12], a member of the gradient-based attribution family introduced in Section II. Gradient Explainer estimates feature contributions by computing expected gradients relative to a background distribution, enabling both local explanations (per Engine) and global insights (across the dataset).

This method was chosen for its compatibility with non-standard architecture Transformers and structured multivariate time series, as encountered in our 42×14 input windows. While other techniques such as LIME and Integrated Gradients are valuable in broader explainability contexts [14], [15], SHAP Gradient Explainer offers theoretical consistency, computational efficiency, and additive attribution, aligning well with the goals of transparency in RUL forecasting.

1) Background Selection: : SHAP requires a background dataset to serve as a reference for calculating expected gradients. We use all 100 training windows as the background set, ensuring full coverage of operating conditions and RUL states. This choice balances computational efficiency with stability in the resulting attributions.

2) Batched SHAP Computation.: Due to memory constraints, SHAP values are computed in batches of size 10. Each test sample (of shape 42×14) is passed to the explainer, which returns a tensor of SHAP values with the same shape. These represent the contribution of each sensor at each timestep (including slope and mean rows) to the model's RUL prediction.

IV. Results and Discussion

This section presents the results of our explainability pipeline to evaluate the NAS-optimized. We report results on global feature importance, local attribution for specific predictions, and validate the reliability of the explanations through coherence checks.

1) Global Attribution.: To understand which features were most influential across all test samples, we applied SHAP's Gradient Explainer using 100 stratified background windows. The resulting SHAP values were aggregated across all test inputs, and the top features were visualized using a bar summary plot (Figure 4) and the beeswarm summary plot (Figure 5).

The most impactful feature was `BPR_t41`, the mean value of the Bypass Ratio sensor, which positively influenced RUL predictions. Other highly influential features included the slopes of `phi`, `P30`, and the mean or trend of sensors like `T24` and `W32`. These results confirm that both recent degradation trends (slope features) and operating-level signals (mean features) contribute meaningfully to the model's decisions.

The top-ranked sensors correspond to known degradation-related physical components, supporting the model's alignment with domain expectations. Less informative features were grouped into an "Other" category, highlighting the concentration of decision impact among a small subset of sensor-time features.

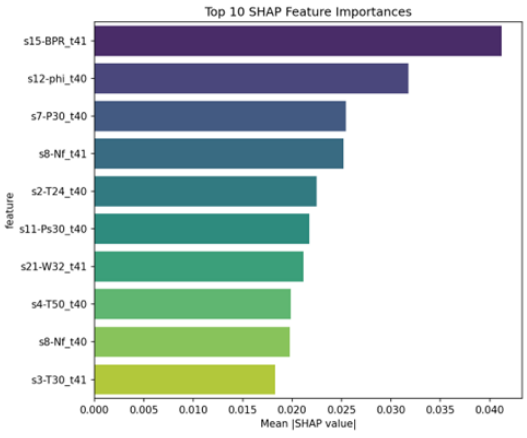


Fig. 4. Global Sensor Ranking barplot

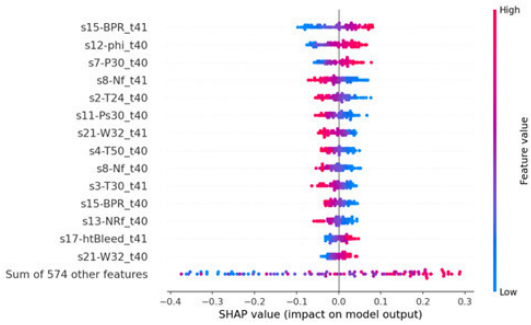


Fig. 5. Global SHAP Summary (Beeswarm) Plot

2) Local Feature Attribution: : To explore how the model forms individual predictions, we examined SHAP waterfall plots for representative test samples. Figure 6 shows a case where the predicted Remaining Useful Life (RUL) was significantly lower than average (0.072 vs. 0.709). Negative contributions came from slope features such as `phi_t40`, `NRf_t40`, and `P30_t40`, which indicate rapid degradation in pressure and rotational speed. A single feature, `BPR_t41`, contributed positively, but only marginally.

Notably, the largest reduction in prediction came from the aggregate contribution of 579 other features, which collectively pulled the estimate downward by -0.25. This highlights the model's ability to synthesize both prominent and subtle signals across the input sequence. The explanation aligns with real-world intuition: sharp declines in critical sensors indicate worsening engine health, justifying a lower RUL forecast.

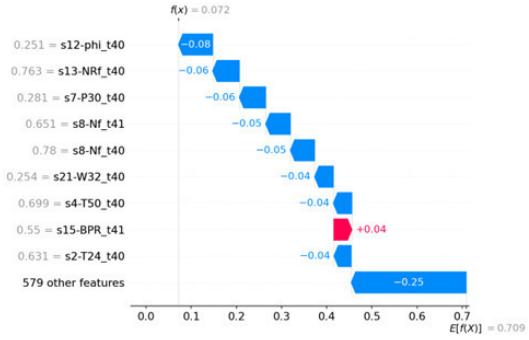


Fig. 6. Local SHAP Waterfall Plot Engine 41

A. Coherence Checks

To assess the trustworthiness of the model's explanations, we conducted a qualitative analysis of the SHAP outputs. Specifically, we reviewed whether the top-ranked features identified by the Gradient Explainer aligned with known degradation indicators in the turbofan engine domain.

Our global attribution analysis revealed that the most influential features included trends and mean values from key sensors such as Bypass Ratio (BPR), high-pressure compressor pressure (P30), rotational speeds (`Nf`, `NRf`), and temperatures (`T24`, `T30`). These are consistent with established knowledge about engine wear and failure modes. Similarly, local explanations for individual predictions showed that decreasing trends in these features often led to lower RUL estimates, reinforcing their interpretability.

Although we did not formally quantify explanation robustness (e.g., using Spearman correlation), the consistent emergence of domain-relevant features in both global and local attributions suggests that the model has learned meaningful and physically plausible relationships. This coherence is a promising indicator for the model's transparency and practical applicability in industrial settings.

V. Conclusion

This paper presents an explainability study of a NAS-optimized Transformer model for Remaining Useful Life (RUL) prediction on the C-MAPSS FD001 benchmark. We integrate SHAP's Gradient Explainer into the model pipeline to generate both global sensor importance rankings and local per-sample attribution maps. Our results show that the model's most influential features, particularly sensor trends and means in airflow, pressure, and temperature, are consistent with known degradation indicators in jet engines.

By illuminating how the model forms each prediction, our approach enhances transparency and supports trust in deep learning-based prognostics. While this study focuses on a single dataset and architecture, the method is generalizable and can be extended to other PHM tasks or architectures.

Future work will incorporate formal stability tests, expert validation, and broader dataset coverage.

References

[1] E. Zio, "Prognostics and health management (PHM): Where are we and where do we (need to) go in theory and practice," *Rel. Eng. Syst. Saf.*, vol. 218, Art. 108119, 2022. Available: https://doi.org/10.1016/j.ress.2021.108119.

[2] O. Serradilla, E. Zugasti, J. Rodriguez, *et al.*, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Appl. Intell.*, vol. 52, pp. 10934–10964, 2022. Available: https://doi.org/10.1007/s10489-021-03004-y

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, … and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. Available: https://doi.org/10.48550/arXiv.1706.03762

[4] H. Mo and G. Iacca, "Evolutionary neural architecture search on transformers for remaining useful life prediction," *Mater. Manuf. Process.*, pp. 1–18, 2023. Available: https://doi.org/10.1080/10426914.2023.2199499

[5] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," *Proc. Int. Conf. Prognostics Health Manag. (PHM)*, Denver, CO, USA, pp. 1–9, 2008. Available: https://doi.org/10.1109/PHM.2008.4711414.

[6] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, K. Huang, S. Scardapane, I. Spinelli, M. Mahmud, and A. Hussain, "Interpreting black-box models: A review on explainable artificial intelligence," *Cogn. Comput.*, vol. 16, pp. 45–74, 2024. Available: https://doi.org/10.1007/s12559-023-10179-8

[7] A. T. Keleko, B. Kamsu-Foguem, R. H. Ngouna, and A. Tongne, "Health condition monitoring of a complex hydraulic system using deep neural network and DeepSHAP explainable XAI," *Adv. Eng. Softw.*, vol. 175, Art. 103339, Jan. 2023. Available: https://doi.org/10.1016/j.advengsoft.2022.103339

[8] G. Youness and A. Aalah, "An explainable artificial intelligence approach for remaining useful life prediction," *Aerospace*, vol. 10, no. 5, pp. 1–23, 2023. Available: https://doi.org/10.3390/aerospace10050474

[9] T. Markovic, A. Dehlaghi-Ghadim, M. Leon, A. Balador, and S. Punnekkat, "Time-series anomaly detection and classification with long short-term memory network on industrial manufacturing systems," *Proc. 18th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, vol. 35, pp. 171–181, 2023. Available: https://doi.org/10.15439/2023F5263.

[10] S. Zhao, Y. Zhang, S. Wang, B. Zhou, and C. Cheng, "A recurrent neural network approach for remaining useful life prediction utilizing a novel trend features construction method," *Measurement*, vol. 146, pp. 279–288, 2019. Available: https://doi.org/10.1016/j.measurement.2019.06.004.

[11] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020. Available: https://doi.org/10.1016/j.inffus.2019.12.012

[12] SHAP (SHapley Additive exPlanations): a game theoretic approach to explain the output of any machine learning model. Available: https://github.com/shap/shap

[13] M. Ho, "NAS_transformer: Neural architecture search for transformer-based models," GitHub repository, 2023. Available: https://github.com/mohyunho/NAS_transformer

[14] S. Chakraborty *et al.*, "Interpretability of deep learning models: A survey of results," *Proc. IEEE SmartWorld*, San Francisco, CA, USA, pp. 1–6, 2017. Available: https://doi.org/10.1109/UIC-ATC.2017.8397411.

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," *Proc. Int. Conf. Mach. Learn. (ICML)*, PMLR, 2017. Available: https://doi.org/10.48550/arXiv.1703.01365.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 81–86
DOI: 10.15439/2025F4661
ISSN 2300-5963 ACSIS, Vol. 44

Adapting CycleGAN architecture for Unpaired Diachronic Text Style Transfer

Adrian Niedziółka-Domański
0009-0003-4797-7484
Maria Curie-Sklodowska University
Plac Marii Curie-Skłodowskiej 5
20-031 Lublin, Poland
Email: niedziolkadadrian@gmail.com

Jarosław Bylina
0000-0002-0319-2525
Maria Curie-Sklodowska University
Plac Marii Curie-Skłodowskiej 5
20-031 Lublin, Poland
Email: jaroslaw.bylina@mail.umcs.pl

***Abstract*—Diachronic text style transfer aims to transform text from one historical period into the style of another while preserving its meaning. However, the scarcity of parallel corpora across time periods makes supervised approaches impractical. In this work, we propose to adapt the CycleGAN architecture, originally developed for unpaired image-to-image translation, to model linguistic change over time. Our method employs a generator and discriminator, both conditioned on temporal information, and trained using a combination of adversarial and cycle-consistency losses. We propose a time-conditioned generative framework that supports both discrete and continuous temporal representations, enabling the model to interpolate between historical language styles. The model is trained on unaligned historical texts and can transform language from any period to another. This approach offers a data-efficient solution for diachronic language modeling and opens new research directions in historical linguistics, digital humanities, and unsupervised style transfer.**

I. Introduction

ONE of the main challenges in working with diachronic textual data lies in the limited availability of directly aligned texts from different historical periods. Unlike modern translation datasets, where sentence-level or even word-level correspondences are often available, historical corpora typically lack such parallel structures. For instance, there is rarely a source with one-to-one correspondence between a text written in Middle English and its equivalent in Modern English. This absence of parallel data complicates efforts to apply conventional supervised methods to historical language normalization, translation, or style transfer tasks. As a result, there is a growing need for methods capable of learning mappings between historical and modern language forms without relying on direct supervision or aligned corpora.

In this paper, we propose adapting the CycleGAN architecture, originally developed for unpaired image-to-image translation [1], to the domain of unpaired diachronic text style transfer. Our goal is to demonstrate that the CycleGAN framework, with appropriate modifications for textual data, can serve as a viable approach to style transformation across time periods.

II. Background and Related Work

CycleGAN is a type of Generative Adversarial Network (GAN) [2] designed for unpaired data translation, originally proposed for image-to-image translation tasks (Zhu et al., 2017) [1]. The model consists of two generators and two discriminators. Each generator learns to map data from one domain to another (e.g., from domain A to B, and from B to A), while each discriminator evaluates whether the generated output appears realistic within its respective domain (Fig. 1).

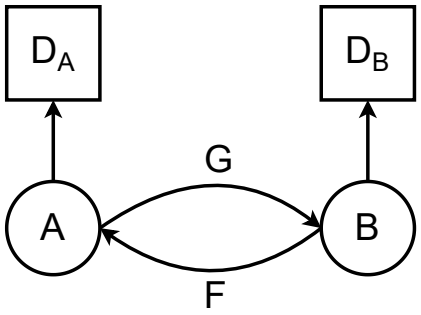


Fig. 1. CycleGan architecture, based on [1]. G and F are generators, A and B are two domains and D_A and D_B are discriminators working in these domains.

A core innovation of CycleGAN is the cycle consistency loss, which ensures that if an input sample is translated to the target domain and then back to the original domain, the result should closely resemble the initial input. This regularization term helps the model retain the core content of the source while adjusting its style to match the target domain.

In formal terms, given two domains X and Y, and two generators $G : X \to Y$ and $F : Y \to X$, the cycle consistency loss is defined as:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} \left[\|F(G(x)) - x\|_1 \right] + \mathbb{E}_{y \sim p_{data}(y)} \left[\|G(F(y)) - y\|_1 \right]$$

This encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$, thereby enforcing that the content of the input is retained after a round-trip translation.

A good example of this process involves translating images of horses to zebras and back again. Even without paired examples (i.e., no exact horse-zebra image pairs), the network

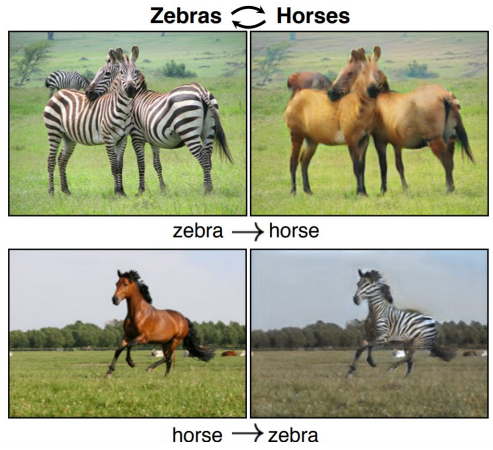


Fig. 2. Example of image to image translation using CycleGan presented in original paper [1].

learns meaningful transformations through adversarial learning combined with the cycle consistency constraint (Fig. 2).

While CycleGAN enables unpaired translation between two domains, it does not scale efficiently to scenarios involving multiple domains. Each pair of domains would require separate generator and discriminator pairs, which would make the model increasingly complex and computationally expensive. In contrast, StarGAN [3] extends the CycleGAN framework to support multi-domain translation within a single unified architecture. StarGAN achieves this by conditioning the generator and discriminator on domain labels, enabling style transfer across many categories using a shared set of parameters. The generator $G(x, c')$ takes an input sample x and a target domain label c', and produces an output in the desired style. The discriminator not only distinguishes real from fake samples but also predicts their domain label (Fig. 3).

In this work, although our primary architecture is inspired by CycleGAN, we also leverage the principles of StarGAN to investigate possible multi-era style transformation tasks. This enables flexible style transfer across multiple historical stages, effectively allowing the model to map between linguistic variants from different centuries using a unified, conditional architecture - without the need to train separate models for each specific pair of eras.

A. Adaptations of CycleGAN for Unsupervised Text Style Transfer

Although CycleGAN was originally proposed for unpaired image-to-image translation, its underlying principles have inspired a number of adaptations in the field of natural language processing and also in tasks involving unsupervised text style transfer. The goal of these adaptations is to utilize CycleGAN's ability to learn mappings between two domains without the need for aligned or parallel training data, a feature especially relevant when working with diachronic corpora or stylistically divergent text.

One of the contributions in this direction is the work by Huang et al. [4], where they proposed a Cycle-Consistent Adversarial Autoencoder model designed specifically for unsupervised text style transfer. Their method combines an autoencoder with cycle consistency loss and adversarial training, allowing the model to keep the semantic content while changing the writing style.

Lorandi et al. [5] proposed a more direct application of the CycleGAN architecture to text style transfer, where they focused on sentiment transformation between positive and negative expressions. Their model, called TextCycleGAN, works without paired data and uses cycle-consistent adversarial training to learn bidirectional mappings between different texts. Although they use a fairly basic LSTM design for both generators and discriminators, their results on the Yelp dataset achieve strong sentiment accuracy and fluency, proving that CycleGAN can work effectively with text data.

Similarly, Wang et al. [6] used CycleGAN for a more structured task: converting abstracts into conclusions in scientific papers. By considering abstracts and conclusions as different stylistic domains, they demonstrated that CycleGAN can learn style transformation patterns within specialized types of text, using only unpaired data.

These studies show the growing potential of CycleGAN-inspired architectures for text style transfer tasks. By demonstrating that effective stylistic transformations can be achieved in an unsupervised manner, without the need for aligned or parallel corpora, they lay the groundwork for extending such approaches to more complex linguistic domains. They provide a strong basis for exploring how CycleGAN-based models might perform in the context of diachronic language data, where the scarcity of parallel examples across historical periods makes supervised approaches very hard to implement.

This motivates our own research idea, in which we adapt the CycleGAN framework to perform style transfer between different variants of the language over the centuries.

III. Proposed Method

A. Task Formulation

The goal of this work is to perform unpaired diachronic text style transfer. That is, given a piece of text written in the linguistic style of a certain historical period, e.g., the 15th century, we aim to generate a version of that text that retains its original meaning but is expressed in the linguistic style of a different period, such as the 21st century.

Crucially, we assume there are no parallel corpora linking these periods. That means we do not have direct sentence-level alignments between time periods. This makes the task a fully unsupervised sequence transformation problem.

B. Formal Setting

Formally, let $\mathbb{T}$ denote the temporal domain associated with linguistic style. We consider two possible representations of time:

1) Discrete time domain:

$$\mathbb{T}_{\text{disc}} = \{t_1, t_2, \ldots, t_n\}$$

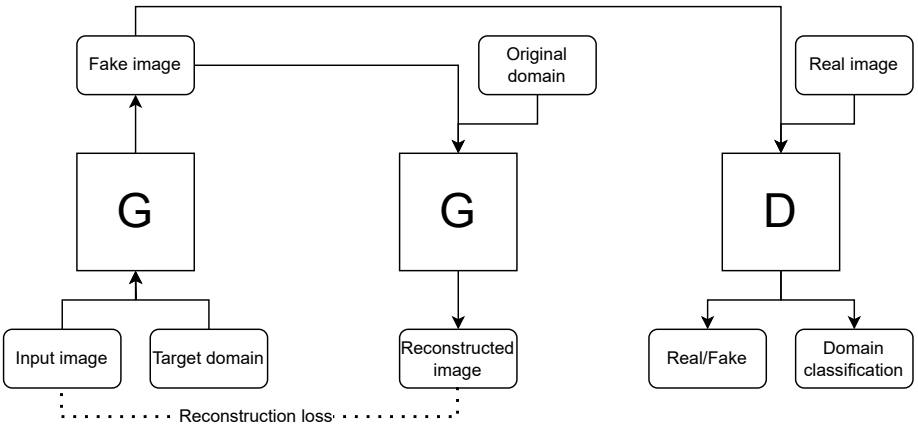


Fig. 3. Overview of StarGAN, consisting of two modules, a discriminator D and a generator G, based on [3].

where each t_i corresponds to a fixed historical period (e.g., 15th century, 16th century, etc.), or to broader linguistic eras (e.g., Old English, Middle English, Modern English).

2) Continuous time domain:

$$\mathbb{T}_{\text{cont}} \subset \mathbb{R}$$

where time is modeled as a real-valued scalar, such as the year or century of origin. This formulation allows the model to reason about intermediate or underrepresented styles and enables smooth interpolation across time.

In this work, we emphasize the continuous representation due to its potential for fine-grained modeling of historical language change. However, the proposed framework remains compatible with discrete labels, which may be more practical in cases where time annotations are coarse or categorical.

Let $x \in X_t$ denote a text sample originating from time period $t \in \mathbb{T}$. Our objective is to learn a generative function:

$$G(x, t') \to \hat{x}_{t'}$$

where t' is the target time period and $\hat{x}_{t'}$ is a text that preserves the meaning of x but adopts the linguistic characteristics of period t'.

To ensure that the model preserves the semantic content of the input, we adopt a cycle-consistency mechanism inspired by StarGAN, where a single shared generator G is used for both forward and reverse transformations. Specifically, given a source text x from time period t, we first translate it to the target style t', and then we use the same generator to map $\hat{x}_{t'}$ back to the original style t:

$$G(G(x, t'), t) \approx x$$

This should allow us to enforce that the transformation is approximately invertible, encouraging the generator to preserve content while altering only the stylistic features associated with time.

C. Model Losses

Our model is suppose to be trained using a combination of adversarial and cycle-consistency objectives, adapted for the temporal style transfer task.

1) Adversarial Loss: Rather than using separate discriminators for each time domain (as in CycleGAN), we employ a single shared discriminator D that is conditioned on the target time period t'. Its objective is to perform **real/fake classification**–that is, to determine whether a given sentence is a genuine example from time t' or a synthetic sample generated by the model. By conditioning on t', the discriminator learns to judge the temporal authenticity of the input relative to the specified style period.

The generator $G(x, t')$ attempts to transform a text sample x from its original time period t into the style of target time t'. The discriminator then assesses whether the result is:

1) **Authentic**, and
2) **Temporally consistent** with t'

To train this system adversarially, we define the adversarial loss as follows:

$$\mathcal{L}_{adv} = \underbrace{\mathbb{E}_{x' \sim p_{\text{data}}(x'|t')}[\log D(x', t')]}_{\text{real samples from target time } t'} + \underbrace{\mathbb{E}_{x \sim p_{\text{data}}(x|t),\ t \neq t'}[\log(1 - D(G(x, t'), t'))]}_{\text{generated samples styled for } t'}$$

This formulation encourages the discriminator to correctly distinguish real samples from generated ones. Specifically, it rewards the discriminator for identifying genuine examples from time t', and penalizes it when it fails to detect synthetic ones. The generator, conversely, is optimized to fool the discriminator into classifying its outputs as authentic. So it ensures that G learns to generate text indistinguishable from true samples belonging to the target time period t'.

2) Cycle-Consistency Loss: To ensure that the semantic content of the text is preserved during style transfer across dif-

ferent time periods, we impose a cycle-consistency constraint using the same generator G. Formally, this loss is defined as:

$$\mathcal{L}_{cyc} = \mathbb{E}_{x,t,t'} \left[\|G(G(x, t'), t) - x\|_1 \right]$$

This term encourages the model to reconstruct the original input text x after sequentially transforming it to a different temporal style t' and then back to its original style t. By minimizing this reconstruction error, the model is guided to produce style-transferred outputs that maintain the original meaning and content, rather than simply generating stylistically plausible but semantically unrelated text. In essence, cycle-consistency enforces that the transformations are invertible and that the core semantic information remains stable across diachronic style mappings.

3) Full Objective: The overall training objective combines the adversarial loss and cycle-consistency loss to jointly optimize generator G and discriminator D. Formally, the full loss function is given by:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{cyc}\mathcal{L}_{cyc}$$

where λ_{cyc} is a hyperparameter that balances the importance of cycle-consistency relative to the adversarial loss.

The generator G aims to minimize this combined loss, learning to produce temporally consistent and semantically faithful style transfers, while the discriminator D is trained to maximize the adversarial loss, improving its ability to distinguish real from generated samples conditioned on the target time period.

Thus, the model should achieve effective unpaired diachronic text style transfer by encouraging realistic temporal style generation and content preservation simultaneously.

D. Proposed Model

Our proposed model adopts a transformer-based architecture, drawing inspiration from CycleGAN and StarGAN, specifically designed for diachronic text style transfer. The model consists of two main components:

- Generator $G(x, t')$
- Discriminator $D(x, t')$

The generator will probably be a conditional transformer encoder-decoder model [7] that transforms a given input text x from its original time period t into the style of a target time t'. The temporal condition t' will be injected into the model in the decoder part.

The discriminator will most likely be a decoder-only transformer that evaluates whether the input x is a real sample drawn from the target time period t' or a generated one. It receives the time condition t' as additional input and is trained to perform binary classification (real/fake) with respect to this condition.

Fig. 4 illustrates the overall model architecture, including the generator and discriminator modules, temporal conditioning flow, and the cycle path used during training.

Algorithm 1 Training Procedure

Require: Training corpus $\mathcal{D} = \{(x_i, t_i)\}$ with time labels
for each minibatch $\{(x_i, t_i)\}_{i=1}^{N}$ sampled from $\mathcal{D}$ **do**
  for each x_i in minibatch **do**
    Select a target time $t'_i \in \mathbb{T} \setminus \{t_i\}$ uniformly at random
    Generate transformed sentence: $\hat{x}_{t'_i} \leftarrow G(x_i, t'_i)$
    Reconstruct original: $\hat{x}_{t_i} \leftarrow G(\hat{x}_{t'_i}, t_i)$
  end for
  Compute adversarial loss $\mathcal{L}_{adv}$
  Compute cycle-consistency loss $\mathcal{L}_{cyc}$
  Update discriminator D to maximize $\mathcal{L}_{adv}$
  Update generator G to minimize $\mathcal{L}_{adv} + \lambda_{cyc}\mathcal{L}_{cyc}$
end for

E. Training and Evaluation

The proposed model is trained end-to-end using a combination of adversarial and cycle-consistency losses. As shown in algorithm 1, training proceeds by iterating over mini-batches of text samples drawn from the training corpus $\mathcal{D} = \{(x_i, t_i)\}$, where each sample is annotated with its corresponding time period t_i.

For each input sentence x_i in a mini-batch, a target time t'_i is randomly sampled from the set of available time labels, excluding the original time t_i. The generator G then transforms the sentence into the style of the target time, producing $\hat{x}_{t'} = G(x, t')$. To enforce semantic preservation, this generated sample is passed again through the generator, that is now conditioned on the original time period to reconstruct the source sentence: $\hat{x}_t = G(\hat{x}_{t'}, t)$, where $\hat{x}_t \approx x$.

After all forward and backward transformations are completed for the mini-batch, two loss functions are computed:

- The adversarial loss $\mathcal{L}_{adv}$ encourages the discriminator D to distinguish real samples from generated ones, while guiding the generator to produce temporally consistent and realistic outputs.
- The cycle-consistency loss $\mathcal{L}_{cyc}$ enforces that the content of the original sentence is preserved across the round-trip transformation between time styles.

The discriminator is updated to maximize the adversarial loss, while the generator is updated to minimize a weighted combination of both losses: $\mathcal{L}_{adv} + \lambda_{cyc}\mathcal{L}_{cyc}$.

Due to the lack of parallel diachronic corpora, automatic evaluation is challenging. We propose the following evaluation strategies:

- **Temporal Classification Accuracy**: A pretrained time classifier can be used to assess whether generated samples are stylistically consistent with the target time period.
- **Cycle Reconstruction Error**: Content preservation can be approximated, for example, by measuring the L_1 distance between the input sentence and its reconstruction after a cycle pass.
- **Human Evaluation**: Expert evaluations by historians or linguists can offer valuable insights into the fluency,

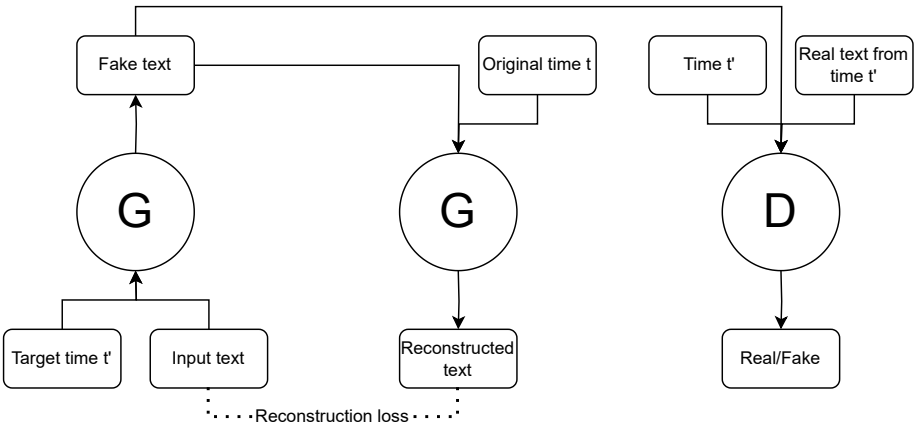


Fig. 4. Proposed model architecture

semantic accuracy, and historical authenticity of the generated text.

This training strategy should help the model learn a smooth, time-aware text style transformation function that can generalize across different historical periods, even without parallel supervision. By conditioning both the generator and discriminator on a continuous temporal index, the model may learn to recognize subtle patterns in the evolution of linguistic features over time. Instead of memorizing fixed mappings between specific time periods, the generator will learn to interpolate and extrapolate stylistic attributes across the temporal space. This should allow flexible text generation at arbitrary points in the historical timeline, including periods for which little or no direct training data exists.

IV. Expected contribution

This paper introduces a new framework for diachronic text style transfer, allowing sentences written in the style of one historical era to be transformed into stylistically consistent versions from another period. Unlike conventional style transfer approaches that often depend on aligned or parallel corpora, our method operates in a fully unsupervised manner, enabling training on naturally occurring, unaligned historical texts. This marks a substantial advancement in tackling the issues of the scarcity of alligned data in diachronic text corpora.

A core innovation of our model lies in its use of continuous time representations to condition both the generator and the discriminator. Rather than assigning fixed domain labels (e.g., “15th century” or “modern English”), time is treated as an input variable, allowing the model to learn smooth, temporally-aware transitions between language styles. This enables more granular control over the generated outputs and allows the model to capture linguistic change over time as a continuous process based on continuous data, rather than relying on discrete class divisions. Moreover, by viewing time as a continuous factor, the model could potentially predict extrapolate how language might evolve and even create believable future versions of it.

We adapt adversarial and cycle-consistency learning techniques, originally developed for images (CycleGAN, StarGAN), to the domain of natural language. Our proposed architecture uses a single shared generator trained with a combination of adversarial and cycle-consistency objectives, ensuring that generated sentences not only match the target time’s style but also preserve the original semantic content. This allows the model to strike a balance between stylistic transformation and content fidelity, which is critical for meaningful diachronic translation.

This work advances the field of diachronic NLP by introducing a general, data-efficient method for modeling language over time. It creates new opportunities for research in historical language translation and computational philology. By combining ideas from image style transfer and natural language processing, this study provides a foundation for future models that better understand, generate, and adapt text across various historical periods.

Potential applications of our approach include the modernization of historical documents, stylistic harmonization of corpora for linguistic research and speculative modeling of future language evolutions.

V. Possible limitations and future work

Although the proposed model is looking very promising, it can also have several possible limitations. One of the big issues is the lack of large, high-quality diachronic corpora that cover long historical periods. This can limit the variety and reliability of the transformations the proposed model can learn. In addition, the current architecture may not be able to completely capture the complexity of language changes, such as shifts in grammar, meaning, or vocabulary. As a result, the model may focus on surface-level characteristics while overlooking some deeper linguistic structures.

Another limitation occurs when the training data only covers distant time periods, for example, the 15th and 21st centuries. In such cases, the model may produce intermediate linguistic forms that did not exist in the past. The challenge of learning smooth transformations across large temporal gaps becomes evident in this interpolation task. To overcome this issue, temporal conditioning must be carefully designed, potentially incorporating constraints informed by historical linguistic knowledge.

Future research could focus on utilizing outside linguistic knowledge, such as syntactic parsers or etymological databases, to improve semantic preservation and style accuracy. The model could also be adjusted for finer temporal resolutions, like working with decades instead of entire centuries, or expanded to manage multilingual historical corpora.

REFERENCES

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017. doi: 10.1109/ICCV.2017.244. ISBN 978-1-5386-1032-9 pp. 2242–2251. [Online]. Available: http://ieeexplore.ieee.org/document/8237506/

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," Jun. 2014, arXiv:1406.2661 [stat]. [Online]. Available: http://arxiv.org/abs/1406.2661

[3] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun. 2018. doi: 10.1109/CVPR.2018.00916. ISBN 978-1-5386-6420-9 pp. 8789–8797. [Online]. Available: https://ieeexplore.ieee.org/document/8579014/

[4] Y. Huang, W. Zhu, D. Xiong, Y. Zhang, C. Hu, and F. Xu, "Cycle-Consistent Adversarial Autoencoders for Unsupervised Text Style Transfer," in *Proceedings of the 28th International Conference on Computational Linguistics*, D. Scott, N. Bel, and C. Zong, Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020. doi: 10.18653/v1/2020.coling-main.201 pp. 2213–2223. [Online]. Available: https://aclanthology.org/2020.coling-main.201/

[5] M. Lorandi, A. Mohamed, and K. McGuinness, "Adapting the CycleGAN architecture for text style transfer." Galway, Ireland: Zenodo, Aug. 2023. doi: 10.5281/zenodo.8268838. [Online]. Available: https://doi.org/10.5281/zenodo.8268838

[6] H. Wang, Y. Lepage, and C. L. Goh, "Unpaired Abstract-to-Conclusion Text Style Transfer using CycleGANs," in *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. Depok, Indonesia: IEEE, Oct. 2020. doi: 10.1109/ICACSIS51025.2020.9263246. ISBN 978-1-7281-9279-6 pp. 435–440. [Online]. Available: https://ieeexplore.ieee.org/document/9263246/

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 87–92
DOI: 10.15439/2025F0373
ISSN 2300-5963 ACSIS, Vol. 44

Towards Human-Robot Interaction in Agriculture Using Large Language Models

Lavanyan Rathy, Håvard Pedersen Brandal, Weria Khaksar
Norwegian University of Life Sciences, Ås, Norway
Email: lavanyan.rathy@nmbu.no; havard.pedersen.brandal@nmbu.no, weria.khaksar@nmbu.no

***Abstract*—Labor shortages and usability challenges limit the adoption of robotics in agriculture. This work explores how Large Language Models (LLMs) and Vision-Language Models (VLMs) can bridge this gap by enabling non-expert users to command robots using natural language. A modular system was developed to interpret instructions, execute tasks, and generate visual field reports. Evaluations in a simulated field showed that hybrid prompting strategies yielded reliable plans, while VLMs supported effective object detection and contextual reporting. This approach reduces entry barriers to robotics and promotes accessible, intelligent agricultural automation.**

Keywords: Large Language Models, AI, HRI, NLP, Precision farming, digital agriculture

I. Introduction

A. Motivation and Background

ROBOTICS is a rapidly evolving field with the potential to address pressing global challenges, particularly in sectors like agriculture [7]. However, deploying robotic systems in practice often demands high technical expertise, limiting accessibility for non-experts.

Norwegian agriculture, for example, faces critical challenges such as labor shortages, food waste, and reduced productivity [6], [4], [2]. Robotic solutions could address these issues by automating labor-intensive tasks. However, the complexity of current systems often discourages adoption, especially among farmers unfamiliar with robotics or programming [5].

Recent advances in artificial intelligence, particularly large language models (LLMs), present an opportunity to close this usability gap. LLMs can interpret and respond to natural language instructions, enabling intuitive, conversational interfaces. This could significantly lower barriers to adoption, allowing farmers to operate advanced robotic systems through simple, everyday language [8].

B. Problem Statement and Objectives

Despite the potential of robotics to transform agriculture, usability remains a core barrier. Most current systems are not designed for non-technical users, limiting their impact on productivity and sustainability [6].

This work addresses that challenge by exploring how LLMs and vision-language models (VLMs) can make human-robot interaction (HRI) more natural and accessible. Specifically, the system interprets written instructions, plans and executes robotic actions, and processes visual data to generate human-readable field reports.

The main objectives of this study are to:

- **Develop a multimodal LLM/VLM system** that translates natural language and visual input into ROS2-compatible robot actions.
- **Evaluate the accuracy and reliability of LLM-generated action plans**, including the impact of robotic hardware limitations.
- **Analyze how different prompt engineering strategies** affect command quality and consistency.
- **Assess VLM capabilities for object detection and spatial reasoning** in agricultural environments.
- **Demonstrate VLM-based visual reporting**, including structured outputs that enhance transparency and oversight.

C. Research Questions

To evaluate the proposed approach, this research is guided by the following questions:

- How accurately can an LLM generate executable ROS2 action plans from natural language instructions, and how do hardware limitations affect execution?
- How do different prompt engineering strategies influence output quality and consistency?
- How effectively can a VLM identify and localize agricultural objects, and what are its spatial limitations?
- Can VLMs produce interpretable, natural-language field reports from visual input that support human-robot collaboration?

II. Background and Related Work

Recent advances in LLMs and VLMs have enabled more intuitive human-robot interaction, particularly in contexts requiring high-level reasoning and accessibility for non-experts. LLMs such as GPT-4 exhibit strong generalization capabilities across tasks like planning, summarization, and code generation without retraining. Their ability to interpret natural language and produce structured outputs makes them a compelling option for high-level robotic control [8].

Prompt engineering has emerged as a key factor in improving the consistency and accuracy of LLM outputs. Direct prompting involves single-shot commands but often lacks reliability. Chain-of-thought (CoT) prompting helps by introducing intermediate reasoning steps, while few-shot prompting provides examples to anchor the model's behavior. Hybrid strategies, combining CoT and few-shot, can further enhance both interpretability and execution success in planning tasks [1].

VLMs extend this capability by jointly processing image and text inputs. Trained on large-scale image-caption datasets, models like CLIP and BLIP can identify and describe visual content, perform spatial reasoning, and generate contextual reports. This is particularly valuable in agriculture, where visual cues, such as detecting obstacles or crop conditions, play a vital role in robot operation [3].

Integrating LLMs and VLMs in robotic applications introduces a multimodal reasoning layer, enabling systems to move beyond hard-coded control toward flexible, adaptive interaction. Although prior work has demonstrated the potential of these models in lab settings, their deployment in field robotics, especially under agricultural constraints, remains underexplored. This research addresses that gap by combining LLM and VLM modules in a ROS2-based system that translates natural language commands and visual input into executable robot actions and structured field reports.

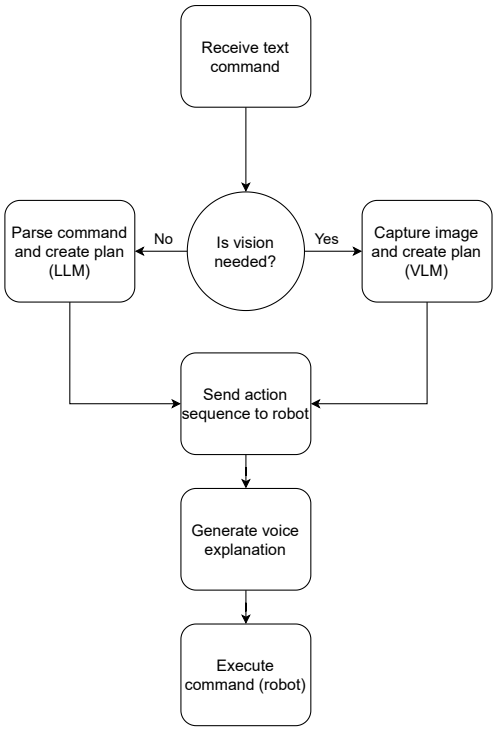


Fig. 1. High-Level Architecture for LLM-Based Robotic System

III. Methodology

A. System Architecture

The system follows a modular architecture combining language and vision models for robotic control. As shown in Figure 1, it processes natural language commands through an LLM to generate ROS2-compatible action plans. If visual input is required, a VLM interprets camera images to support perception and reporting. The robot then receives executable commands and a spoken summary of intent for transparent interaction.

The natural language command is processed through a Langchain pipeline using a `FewShotPromptTemplate`, which embeds dynamic user input and curated examples to shape the model's interpretation. The prompt structure includes a task description, spatial constraints, and example command formats. The LLM response contains a natural-language summary and a structured plan expressed in pseudo-code or action-like instructions. These are then parsed and verified using a YAML schema to ensure semantic and syntactic validity.
An example output may resemble:

```
Plan:
  - drive(2)
  - turn(90)
  - drive(2)
```

This intermediate representation allows modular validation and easier debugging. Internally, each action string is mapped to a corresponding ROS2-compatible function. For instance, `drive(2)` translates to a call to the navigation stack or a custom publisher on the `/cmd_vel` topic with linear velocity commands for a specified duration. Angle commands like `turn(90)` trigger a PID-regulated angular velocity loop with quaternion goals defined in radians. All interpreted commands are time-stamped and executed via a ROS2 executor, ensuring synchronization and feedback integration. For planning errors or misinterpretation, fallback handlers can re-query the LLM using augmented prompts that include failure context.

B. Simulation Environment

Development and validation were conducted in Gazebo Classic using the Peik robot, modeled in URDF/Xacro to replicate real-world geometry and sensor layout (Figure 2). ROS2 middleware facilitated communication across components. Peik's simulated sensors include a front-mounted RGB-D camera and an IMU. The robot was simulated in a maize field using Gazebo, with onboard RGB-D sensing and inertial measurement to support planning, perception, and trajectory tracking. A modular ROS2 architecture handled action execution and data flow between the LLM, VLM, and navigation stack.

The robot base is configured with a `'base_link'` and `'camera_link'` transform, aligned using static TF publishers. The URDF includes a ZED-like camera plugin with near-true RGB-D behavior. Odometry is simulated using differential drive parameters in Gazebo, allowing accurate benchmarking of LLM trajectory plans versus actual ground truth paths. The robot's rotational behavior is tuned with angular velocity limits of ± 1.5 rad/s and a max forward speed of 0.5 m/s, constrained for safety in narrow-field crop paths.

Fig. 2. Peik operating in a simulated maize field

C. LLM-Based Command Interpretation

User instructions are sent via a ROS2 topic and processed by an OpenAI-powered LLM using the Langchain framework. Prompts are dynamically constructed to include reasoning and explicit robot actions. Responses are parsed into a human-readable explanation (spoken aloud) and a command list (e.g., `drive(2)`, `turn(90)`), which is executed by the robot. The system triggers visual processing if the response contains the keyword `[CAMERA_REQUIRED]`.

D. VLM Integration for Perception

For visual reasoning, the system captures a JPEG image from the robot's camera, encodes it in base64, and sends it with a text prompt (e.g., `"What's in this image?"`) to a GPT-4-based VLM. The model returns a natural-language description of the scene, including obstacle presence or task-relevant objects. This output is both published and spoken by the robot for transparency.

IV. Results

A. Trajectory Execution

The system was evaluated using a square-pattern navigation task, where the LLM generated a plan from the command: *"Move in a square pattern, each side one meter long."*. The robot successfully executed the plan with minor trajectory drift. A PID controller improved tracking accuracy compared to open-loop control. Figure 3 shows the odometry trace before and after adjustment.

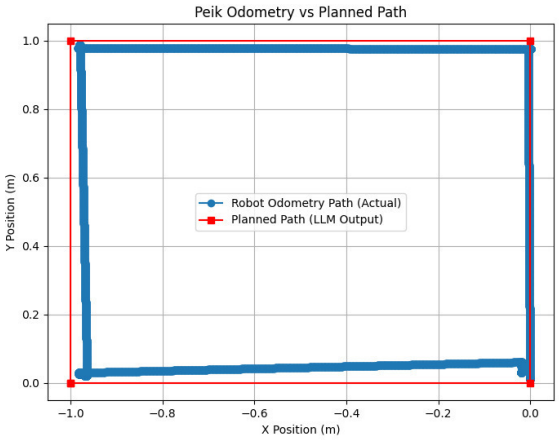


Fig. 3. Robot trajectory: Open-loop vs PID control

B. Prompting Strategy Comparison

Four prompting strategies were compared: Direct, Chain-of-Thought (CoT), Few-Shot, and Hybrid. Each strategy was tested using the same navigation task in the simulation. Figure 4, 5, 6 and 7 shows one example of each run.

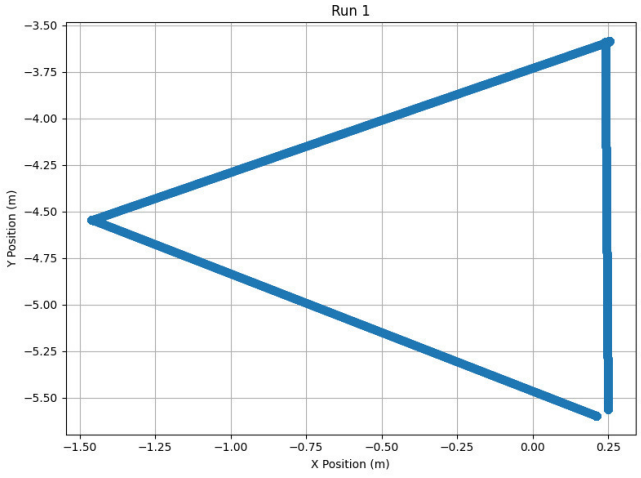


Fig. 4. Example of prompt strategy (CoT) run

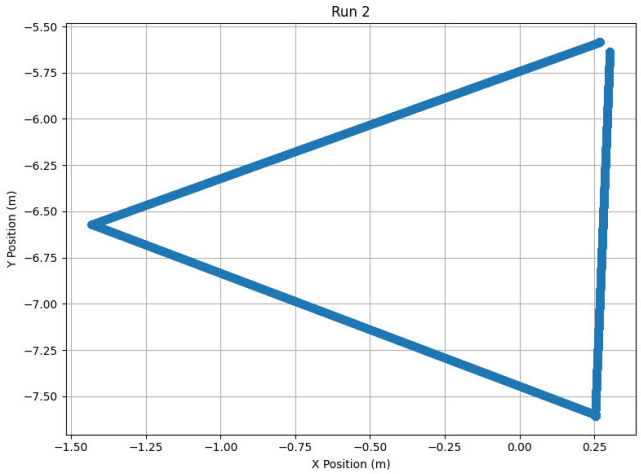


Fig. 5. Example of prompt strategy (Direct) run

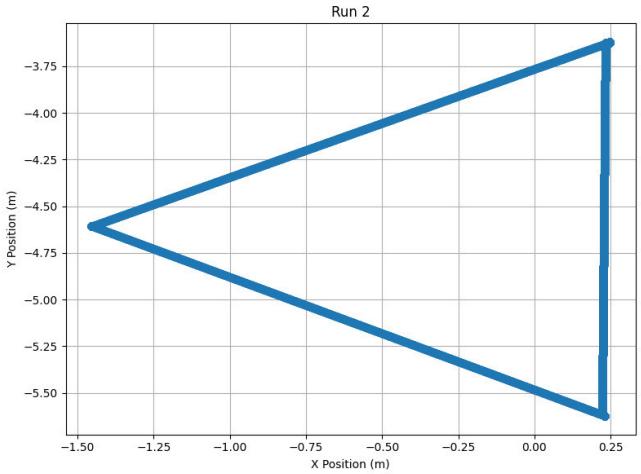


Fig. 7. Example of prompt strategy (Hybrid) run

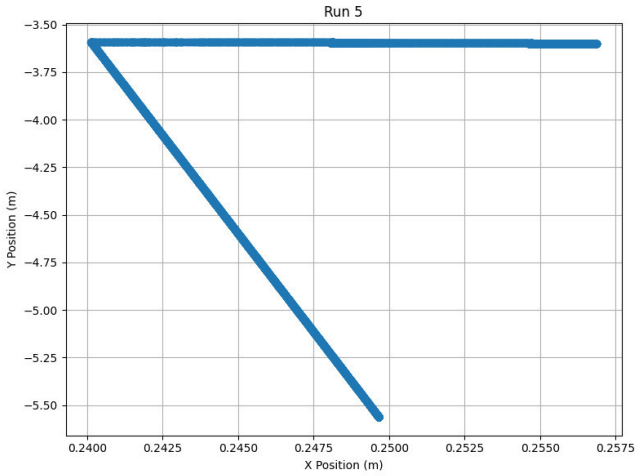


Fig. 6. Example of prompt strategy (Few-Shot) run

A quantitative comparison assessed each strategy's performance over five repetitions of a trajectory planning task. Table I summarizes the average task success rate and angular deviation across strategies.

TABLE I
PROMPT STRATEGY EVALUATION

Strategy	Success Rate
Direct Prompt	5/5
Chain-of-Thought	5/5
Few-Shot	1/5
Hybrid (CoT + FS)	3/5

C. Object Detection via VLM

The robot captured field images and passed them to GPT-4 with prompts like *"Describe what's in this image"*. The VLM consistently identified crops, tools, and obstacles like bottles or weeds.

D. Visual Field Reporting

In extended prompts (e.g., *"Generate a report of what you see in this field"*), the VLM produced coherent natural-language summaries highlighting plant health, potential obstructions, and environmental conditions. These reports were structured and human-readable, supporting autonomous decisions and remote operator review.

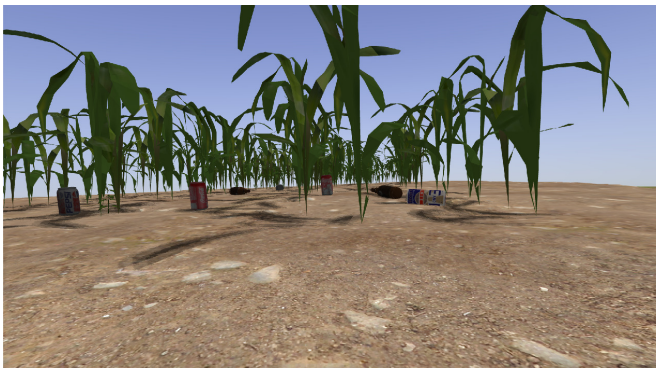


Fig. 8. Example of robot pov for GPT-4-generated field report

V. DISCUSSION

A. LLMs as Planners, Not Controllers

The findings validate the role of LLMs as high-level planners capable of translating abstract natural language instructions into executable robot behaviors. However, while LLMs can produce coherent and logically sound plans, their real-time execution fidelity is limited by hardware-level dynamics and environmental variance. As shown in the square-pattern task, deviations from expected paths were frequent in

open-loop mode, highlighting the importance of integrating traditional low-level control mechanisms like PID regulators. This reinforces the necessity of hybrid architectures, where symbolic reasoning from LLMs is grounded by deterministic feedback control.

B. Prompt Engineering Trade-offs

The prompt design significantly influenced output quality, with hybrid prompting (Few-shot + CoT) achieving the best balance of reliability and generalization. Direct prompts were quick to generate but tended to fail under ambiguity or complex task structures. Chain-of-thought prompting improved transparency by encouraging intermediate reasoning, sometimes resulting in verbose or over-engineered plans. Few-shot prompting offered stability by anchoring the model's output style with curated examples, but in practice, it did not generalize well to tasks requiring geometric adaptation. Hybrid prompting combined examples with reasoning, improving robustness in some cases but introducing inconsistency in others. This aligns with observations from the thesis, which showed that prompt selection directly affects the syntactic structure, interpretability, and trajectory adherence, especially in angle-sensitive instructions like turning 120° versus 90°.

The results of the triangle movement experiment further highlight the impact of system prompt design on LLM-driven control. Despite using the same user prompt ("Move in a triangle pattern"), the system's output and robot behavior varied significantly across prompting strategies.

1) Direct Prompting: Direct prompting achieved excellent performance, with 5 out of 5 successful runs and high consistency. This approach benefited from a system prompt instructing the LLM to generate concise, minimal step-by-step outputs without explicit reasoning. However, direct prompting is highly dependent on a well-phrased initial instruction. If user input is vague or lacks geometric precision, the model lacks mechanisms to infer missing context, potentially reducing robustness.

2) Chain-of-Thought (CoT) Prompting: CoT prompting also yielded strong performance, matching direct prompting with 5 out of 5 successful runs. In this case, the model was guided to reason that a triangle requires three sides of equal length and external angles of 120°. This explicit explanation helped the LLM generalize to the correct geometry.

3) Few-shot Prompting: Few-shot prompting demonstrated poor generalization, with only 1 out of 5 successful executions. Although the model was provided with examples (e.g., moving in a square), it frequently overfitted to these patterns and failed to extrapolate to triangles. Common errors included using 90° turns instead of 120° or stopping prematurely after one or two sides.

4) Hybrid Prompting (Few-shot + CoT): Hybrid prompting, which combines examples with structured reasoning, achieved 3 out of 5 successful runs. This method produced promising results when the examples and reasoning segments were well-aligned. While hybrid prompting offers strong potential, its effectiveness depends on carefully crafted prompt design to avoid interference between modes.

5) Overall Observations: Direct and chain-of-thought prompting emerged as the most reliable methods for producing executable, ROS2-compatible plans in geometric movement tasks. Few-shot prompting alone lacked adaptability, and hybrid prompting, while promising, introduced occasional inconsistencies. These findings underscore that prompting strategy plays a central role in shaping language output and real-world robot behavior.

For robotics applications, prompt clarity, structure, and internal logic are critical to minimize ambiguity and execution failure. Future research should explore combining prompt-based control with parameterized templates, explicit reasoning paths, or constrained decoding to improve interpretability and task repeatability.

C. VLM-Based Perception and Reporting

The VLM component effectively grounded visual input into human-readable outputs, such as object labels and structured reports. Agricultural scenes were typically parsed with high accuracy, though occlusions and low-contrast conditions introduced occasional misclassifications, especially in cluttered environments. This confirms the thesis's insight that VLMs can enhance field awareness but are sensitive to camera placement, field layout, and scene quality. Additionally, the ability to produce spoken reports supports explainability, which is crucial for human trust in robot decision-making.

The object detection experiments revealed that GPT-4-based VLMs consistently identified foreign objects such as bottles, soda cans, and weeds, and provided type-correct descriptions. Crucially, when no objects were present, the model did not hallucinate, correctly reporting empty scenes. This ability to maintain grounded, reality-consistent outputs suggests strong baseline reliability under normal field conditions. However, spatial localization, particularly left/right/center descriptions, showed inconsistencies, with subjective or frame-dependent language used to describe object position. More structured prompting (e.g., referencing rows or distance bands) could improve spatial clarity.

Contextual understanding was also demonstrated: the model inferred partial occlusion when overlapping objects were present and improved classification when similar items appeared at varying distances. For example, in one run, a far object was generically labeled as "debris," while a closer object in a similar class was correctly described as a "glass

bottle." This reflects a degree of contextual refinement, where object interpretation improves with better visual cues.

Field reporting experiments further validated the model's ability to assess environmental risks and suggest mitigation strategies. Detected objects were categorized by potential hazard (e.g., "the bottle might shatter and harm equipment"), with risk ratings inferred from visible features like size and material. In empty field scenarios, the VLM demonstrated conservative behavior, noting small rocks as minor concerns rather than hallucinating threats, showing an ability to scale its judgment based on visual evidence. However, it sometimes underestimated cumulative risks (e.g., multiple soda cans described without reference to quantity), highlighting a limitation in quantitative reasoning.

Finally, the model showed early signs of predictive reasoning: in occluded scenes, it inferred the likely presence of a second object based on partial shape overlap. Such capabilities could be valuable for hazard anticipation and proactive avoidance. Nonetheless, challenges remain in depth estimation, localization precision, and interpretability across varying field conditions. Structured prompts, confidence scoring, and hybrid visual reasoning modules could help mitigate these issues for real-world deployments.

D. Human-Robot Interaction Implications

The system enables a shift in human-robot interaction (HRI) toward natural-language-based collaboration. This reduces the cognitive and technical burden on end users, making robotics more accessible for domains like agriculture, where operators are often domain experts but not programmers. This positions conversational robotics as a tool for automation and augmenting field intelligence.

E. Limitations and Future Directions

While the results obtained in the simulation were promising, several limitations remain that must be addressed to enable real-world deployment:

- **Latency**: API-based model queries, especially those involving VLMs, introduced non-deterministic delays, which hinder real-time performance.
- **Robustness**: LLM behavior became less predictable during long or complex task sequences. Inconsistent internet connectivity in field environments further reduces system reliability.
- **Scalability**: The current modular architecture supports isolated tasks but lacks mechanisms for multi-step workflows, memory across sessions, and coordination between multiple agents.

To address these limitations, future work should explore deploying LLM and VLM inference directly on edge devices to reduce latency and improve autonomy. Visual capabilities could also include crop growth monitoring, disease detection, and environmental stress assessment. Additionally, integrating adaptive feedback loops, where the robot asks for clarification when uncertain, could significantly enhance task reliability and user trust in ambiguous situations.

VI. Conclusion

This work demonstrates a modular system integrating LLMs and VLMs to enable intuitive, explainable robot control for agricultural tasks. By translating natural language instructions into executable ROS2 actions and combining this with visual perception and reporting, the system allows non-expert users to interact with robots in accessible ways. Experimental results show that LLMs can generate high-level plans reliably when supported by classical control and that VLMs can effectively interpret agricultural scenes to produce structured field reports. This approach reduces the barrier to robotics adoption in farming and opens new opportunities for human-robot collaboration in semi-structured environments.

Future work will improve real-time robustness, deploy models locally for field use, and extend visual understanding to support crop-specific tasks such as growth analysis and anomaly detection.

Acknowledgment

This work is a part of DigiFoods SFI funded by the research council of Norway under the agreement 309259.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. arXiv:2005.14165 [cs].

[2] Magnus Skatvedt Iversen. Norges Bondelag vil gjøre det lettere å få tak i sesongarbeidere, June 2024. Section: dk.

[3] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From CLIP to DINO: Visual Encoders Shout in Multi-modal Large Language Models, March 2024. arXiv:2310.08825 [cs].

[4] OECD. *Policies for the Future of Farming and Food in Norway*. OECD Agriculture and Food Policy Reviews. OECD, March 2021.

[5] David Christian Rose and Jason Chilvers. Agriculture 4.0: Broadening Responsible Innovation in an Era of Smart Farming. *Frontiers in Sustainable Food Systems*, 2, December 2018. Publisher: Frontiers.

[6] Michael Ryan. Labour and skills shortages in the agro-food sector. *OECD*, January 2023.

[7] Bruno Siciliano and Oussama Khatib, editors. *Springer Handbook of Robotics*. Springer Handbooks. Springer International Publishing, Cham, 2016.

[8] Minghe Wang, Alexandra Kapp, Trever Schirmer, Tobias Pfandzelter, and David Bermbach. Exploring Influence Factors on LLM Suitability for No-Code Development of End User IoT Applications, May 2025. arXiv:2505.04710 [cs].

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 93–97
DOI: 10.15439/2025F1171
ISSN 2300-5963 ACSIS, Vol. 44

Multitask Learning for Six-Pack Toxicity Prediction

Chun-Wei Tung*, Chia-Chi Wang†, Run-Hsin Lin*, Shan-Shan Wang*
0000-0003-3011-8440, 0000-0002-1272-1513, 0000-0002-1205-2874, 0000-0001-6163-120X
*Institute of Biotechnology and Pharmaceutical Research, National Health Research Institutes,
35 Keyan Road, Zhunan, Miaoli, Taiwan
Email: {cwtung, rhlin, shanwang}@nhri.edu.tw
†Department and Graduate Institute of Veterinary Medicine, School of Veterinary Medicine,
National Taiwan University, No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan
Email: ccwang@ntu.edu.tw

***Abstract*—The assessment of the six-pack toxicity, the crucial six systems and organ toxicities, is vital for ensuring the safe use of chemicals. Computational models capable of providing reliable predictions are acceptable for regulatory use to replace animal testing. However, data scarcity issues hindered the development of prediction models. This study proposed the first application of multitask learning to the six-pack toxicity for addressing data scarcity issues. Five algorithms were implemented and compared. Results showed that the distinct chemical space of tasks impedes the learning of shared representation of conventional algorithms, with performance worse than baseline models. In contrast, the MTForestNet algorithm built on a biological readacross concept performed best, with 3.1% and 3.3% improvement on AUC and accuracy, respectively. These findings demonstrate that biologically informed multitask learning can effectively overcome data scarcity and enhance toxicity prediction.**

***Index Terms*—multitask learning, biological readacross, six-pack toxicity, distinct chemical space, MTForestNet.**

I. Introduction

TOXICITY prediction plays a pivotal role in the early stages of drug discovery and chemical safety assessment. Among the large number of toxicity endpoints for testing, there is a suite of six key toxicity endpoints, commonly known as the 'six-pack': acute oral toxicity, acute dermal toxicity, acute inhalation toxicity, skin irritation, eye irritation, and skin sensitization. These endpoints provide important information about the system and organ toxicity of testing chemicals and are crucial for regulatory decision-making and risk assessment of industrial chemicals, pharmaceuticals, and consumer products.

The assessment of the six-pack toxicity is traditionally based on animal testing. However, the traditional experimental approaches to assess these toxicities are time-consuming and costly, and ethical concerns are raised due to extensive animal testing. In recent years, computational methods, particularly machine learning, have emerged as powerful alternatives for toxicity prediction. Several studies have developed machine learning models for predicting the six-pack toxicity [1], [2], [3].

Despite the efforts made by the scientific community, dataset size poses a major limitation on advancing the prediction performance of six-pack toxicity. It is unlikely to have a huge increase in the testing data due to the high cost and labor-intensive experiments. Compared to the conventional single-task models developed by previous studies, multitask learning algorithms capable of leveraging the shared knowledge among relevant learning tasks can be promising solutions to the prediction of six-pack toxicity.

Several multitask learning algorithms have been proposed and implemented with success for toxicity prediction. For example, three deep learning-based multitask learning algorithms, including conventional, bypass, and progressive multitask learning algorithms, were shown to outperform single-task models for several drug development-relevant datasets [4]. The three algorithms were implemented as an open-sourced library, DeepChem [4]. In addition, AutoGluon-Tabular [5], a powerful automated machine learning algorithm, implemented a multilabel learning algorithm that can be potentially useful for multitask learning. By leveraging shared knowledge, multitask learning can improve prediction accuracy, especially when training data for individual tasks is limited or imbalanced.

While the abovementioned algorithms performed well on the benchmark datasets, each dataset contains a large portion of shared training samples among tasks in the dataset [6], [7], [8], [9], [10], and therefore ensures the successful transfer of knowledge among tasks. However, the majority of learning tasks of toxicity datasets are with distinct chemical spaces containing little or no shared samples, which hinders the application of the DeepChem-based methods. To solve the issue of distinct chemical space, MTForestNet was proposed with a progressive multitask learning strategy concatenating chemical features and outputs of individual classifiers of tasks from

Topical area: Advanced Artificial Intelligence in Applications

the previous layer for accuracy improvement [11]. The algorithm showed superior performance compared to other algorithms on the zebrafish toxicity dataset, consisting of 48 tasks, and is expected to be useful for other toxicity datasets with distinct chemical space.

This study explores the application of multitask learning models to predict all six toxicity endpoints concurrently. A total of five algorithms were implemented and compared for their application to the prediction of six-pack toxicity. Results showed that the model based on MTForestNet performed best on predicting the independent test dataset with the highest average area under the receiver operating characteristic curve (AUC) value of 0.825, showing a 3.1% improvement over single-task models. The other models showed no improvement or much worse performance. The low percentage of shared samples among the six tasks further supports the usefulness of MTForestNet on predicting chemical toxicity.

II. Materials and Methods

A. Dataset

The six-pack toxicity dataset was obtained from a previous study [3] collecting the largest dataset of toxicity data from the U.S. National Toxicology Program and OECD eCHem-Portal. The dataset was randomly divided into 70% training, 10% validation, and 20% test sets for model training, tuning, and independent test, respectively. A summary of the dataset is shown in Table I. In this study, the widely used extended connectivity fingerprint (ECFP) with a diameter of 6 was utilized to encode the chemical feature vector. Specifically, a 1024-dimensional vector representing the binary occurrence of specific substructures was utilized for machine learning.

B. Single-task learning algorithm

In this study, random forest [12] was utilized as the baseline algorithm for evaluating the performance improvement based on multitask learning algorithms. Random forest was extensively used and proved to have robust and high performance in a large number of cheminformatics tasks [13], [14], [15], [16], [17]. The parameters utilized to implement random forest classifiers were set as follows: mtry=log2(total feature number) and n_estimators=500. With the parameters, a single-task random forest classifier with 500 trees and log2(total feature number) features sampled from all features was developed for each task.

C. Multitask learning algorithms

Five algorithms were implemented and compared in this study. Accuracy was utilized as the objective function to tune or select models based on the validation sets for all algorithms. DeepChem package [4] was utilized to implement three multitask learning algorithms of multitask network (DC_MTN), progressive network (DC_Progressive), and bypass network (DC_Bypass). DC_MTN incorporates shared layers for learning a joint representation of all tasks with six separate output layers, each corresponding to a specific task. DC_Progressive prevents catastrophic forgetting by adding a new column for each task and using lateral connections to transfer knowledge from previously learned tasks. DC_Bypass combines the learnable shared representation and a column of weights that bypass the shared representation for each task. The hyperparameters of the three networks were set as follows: learning_rate=0.001; dropouts=[0.20, 0.10, 0.05]; layer_sizes=[400, 200, 100]; penalty=0.001; weight_decay_penalty_type='l2'.

The multilabel learning algorithm of AutoGluon-Tabular trained an individual model for each label, with the inclusion of previous labels as features. In this way, the dependence of labels can be modeled. The default setting of AutoGluon-Tabular was applied in this study with eight classifiers, including two neural networks based on Torch and FastAI, LightGBM boosted trees, CatBoost boosted trees, XGBoost, random forest, extremely randomized trees, and k-nearest neighbors were automatically trained and stacked to achieve the highest performance on the validation set. The parameter of auto_stack was set to true for automatic model stacking in the model development. Medium (AG_Medium) and best (AG_Best) quality models were built for performance comparison using the quality parameter.

MTForestNet was proposed to deal with the distinct chemical space of tasks with little or no shared samples. The idea is based on the biological data-based read-across, where the label (target endpoint) of chemicals tends to be similar if the bioactivity profile of chemicals is similar [18], [19], [20]. MTForestNet utilized random forest as a base learner for building models, each for a task. The predicted outputs of single-task models were then fed into the next layer, where the feature vector was refined to concatenate both the chemical

TABLE I.
OVERVIEW OF DATASET SAMPLE SIZES

Task	Toxic/Nontoxic	Training	Validation	Test
Acute Dermal Toxicity	870/939	1266	181	362
Acute Inhalation Toxicity	436/428	604	87	173
Acute Oral Toxicity	6391/4723	7779	1112	2223
Eye Irritation	1824/1841	2565	367	733
Skin Irritation	1315/1311	1837	263	526
Skin Sensitization	1510/1256	1935	277	554

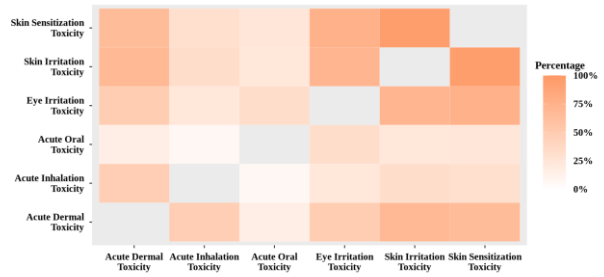


Fig. 1 The percentage of shared chemicals for each pair of tasks

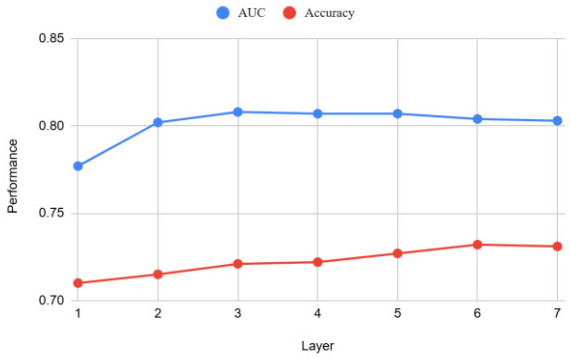


Fig. 2 The validation performance of MTForestNet

fingerprint and the six outputs from the models of the previous layer. The validation set was utilized to determine the size of the model giving the highest validation performance.

D. Hardware

The experiments were conducted in a computer equipped with two Intel® Xeon® Gold 6330, one NVIDIA RTX A6000, and 2 TiB RAM. The operating system is Ubuntu 22.04.

III. Results

A. Tasks with low percentages of shared chemicals

The percentages of shared chemicals among tasks were first analyzed to give an overview of the similarity of the six tasks. As shown in Fig.1, overall medium to low percentages of shared chemicals among tasks indicated that the six datasets lack sufficient information for learning a shared representation. The two skin-relevant tasks of skin sensitization and skin irritation shared the highest percentages of samples, where 94.94% of chemicals have both labels. The task of acute oral is associated with the lowest percentage of shared chemicals of 7.77% and 16.28% for acute inhalation and acute dermal, respectively. Among the 15 pairs of tasks, 5 pairs of tasks are associated with a percentage of shared samples less than or equal to 30%. Only 3 pairs of tasks are associated with a percentage of shared chemicals greater than or equal to 70%. The average percentage of samples shared in all pairs of tasks is 44.45%. In summary, the low percentages of shared chemicals may hinder the learning of shared representation for conventional multitask algorithms.

B. Validation performance

The application of multitask learning algorithms for predicting six-pack toxicity includes three steps of model training based on the training sets, model tuning/validation based on the validation set, and model testing using the test sets. This section provides the validation results of the implemented models. The detailed performance comparison is shown in Table II. The baseline models based on random forest provide reasonably good performance for all tasks, with an average AUC and accuracy of 0.777 and 0.711, respectively.

The validation performance of the three DeepChem-based models is much worse than that of the baseline models, with at least a 10% decrease in the average AUC. The average AUC and accuracy values are 0.673 and 0.545 for DC_MTN, 0.659 and 0.556 for DC_Bypass, and 0.600 and 0.395 for DC_Progressive, respectively. As the chemical spaces are distinct for each task, the low performance of DeepChem-based models is expected.

The AutoML models based on AutoGluon-Tabular provide slightly worse performance compared to the random forest.

TABLE II.
VALIDATION PERFORMANCE

Model	Acute Dermal Toxicity	Acute Inhalation Toxicity	Acute Oral Toxicity	Eye Irritation	Skin Irritation	Skin Sensitization
Random forest	0.773/0.680	0.794/**0.770**	0.840/0.761	0.729/0.665	0.803/0.730	0.724/0.657
MTForestNet	**0.813**/0.713	**0.833**/**0.770**	0.829/**0.772**	**0.758**/**0.689**	**0.847**/**0.768**	**0.746**/**0.679**
AG_Medium	0.773/0.707	0.723/0.690	0.841/0.763	0.708/0.649	0.804/0.722	0.724/0.671
AG_Best	0.791/**0.718**	0.777/0.701	**0.842**/0.761	0.738/0.678	0.728/0.668	0.728/0.668
DC_MTN	0.676/0.595	0.713/0.464	0.742/0.621	0.609/0.515	0.714/0.617	0.586/0.455
DC_Bypass	0.687/0.565	0.689/0.582	0.734/0.592	0.603/0.531	0.684/0.581	0.556/0.487
DC_Progressive	0.702/0.585	0.500/0.019	0.500/0.279	0.622/0.521	0.698/0.560	0.577/0.407

Performance is expressed as AUC/Accuracy. Bold numbers show the best performance in the specific task.

TABLE III.
INDEPENDENT TEST

Model	Acute Dermal Toxicity	Acute Inhalation Toxicity	Acute Oral Toxicity	Eye Irritation	Skin Irritation	Skin Sensitization
Random forest	0.836/0.732	0.758/0.676	0.832/0.745	0.767/0.703	0.822/0.751	0.751/0.679
MTForestNet	**0.865**/0.765	**0.842**/**0.740**	0.819/0.752	**0.795**/**0.719**	**0.851**/**0.795**	**0.779**/**0.708**
AG_Medium	0.826/0.729	0.765/0.711	**0.838**/0.757	0.746/0.689	0.804/0.743	0.749/0.671
AG_Best	0.729/**0.826**	0.760/0.728	**0.838**/**0.760**	0.771/0.700	0.815/0.743	0.762/0.702
DC_MTN	0.747/0.622	0.654/0.483	0.730/0.628	0.596/0.521	0.683/0.615	0.589/0.473
DC_Bypass	0.748/0.569	0.642/0.591	0.745/0.624	0.600/0.546	0.672/0.592	0.593/0.487
DC_Progressive	0.729/0.602	0.500/0.019	0.500/0.280	0.617/0.530	0.687/0.558	0.584/0.429

Performance is expressed as AUC/Accuracy. Bold numbers show the best performance in the specific task.

The average AUC and accuracy values of AG_Medium are 0.762 and 0.700, respectively. AG_Best delivers a slightly better AUC of 0.767 and slightly worse accuracy of 0.699.

The MTForestNet, designed for dealing with the distinct chemical space of tasks, performed best. Fig. 2 shows the training process with accuracy and AUC performance for each layer. The optimal number of layers of MTForestNet was determined to be six according to the accuracy of the validation set. Its average AUC is 0.804, which is 3.3% better than the baseline models. With an average accuracy of 0.732, MTForestNet provides 2.1% performance improvement over the baseline models.

Table II showed that MTForestNet performed best in 5 out of the 6 tasks in terms of AUC and accuracy. AG_Best is the best model for acute dermal toxicity and acute oral toxicity in terms of accuracy and AUC, respectively. However, AG_Best is worse than the baseline models for the other tasks, resulting in a worse average AUC and accuracy compared to the baseline model.

C. Independent test

The independent test showed similar results that MTForestNet is the only algorithm providing a superior performance over the baseline model, with an average AUC and accuracy of 0.825 and 0.747, respectively. A 3.1% and 3.3% improvement on the average AUC and accuracy was achieved compared to the random forest models. The average AUC and accuracy of random forest models are 0.794 and 0.714, respectively.

Table III showed that MTForestNet performed best in 5 and 4 tasks in terms of AUC and accuracy, respectively. While with a slightly worse mean AUC of 0.779, AG_Best models provide good accuracy of 0.743, which is close to MTForestNet models and better than the baseline models. AG_Best is the best model in 1 and 2 tasks in terms of AUC and accuracy, respectively, as shown in Table III. As for the DeepChem-based models, their performance is the worst among the evaluated algorithms and is much worse than the baseline models. The average AUC and accuracy are 0.667 and 0.557 for DC_MTN, 0.667 and 0.568 for DC_Bypass, and 0.603 and 0.403 for DC_Progressive, respectively.

D. Comparison to existing methods

There are three recently published methods aiming to predict six-pack toxicity [1], [2], [3]. However, a careful evaluation found that the three studies divide the whole dataset into training and validation sets without an independent test. All three studies applied multiple machine learning algorithms and picked the best results from validation results. In this case, the prediction performance may be overestimated. Nevertheless, a comparison to existing methods can still provide some information on the current status of prediction models for six-pack toxicity.

We first compare our results with the study [3] using the same dataset. Only accuracies rounded to two decimal places were fully disclosed in their paper, with an average value of 0.75 based on the validation set. Their average accuracy value is the same as that of the developed MTForestNet model based on the test set, indicating that MTForestNet performed very well without the need to exhaustively train and select models.

The other two studies used a smaller dataset [1], [2] for model development. There is no accuracy information reported by StopTox [1]. Instead, a balanced accuracy representing a mean of sensitivity and specificity was given based on their validation set with an average value of 0.735. Please note that the results were based on a selection of chemicals suitable for the StopTox models. There are 5.4% deemed to be not suitable for the StopTox models. Without a selection of chemicals, the MTForestNet model with an average value of balanced accuracy of 0.7445 based on the test set provides better performance. The latest study [2] exhaustively trained all models by using the combination of three algorithms and four representations of chemicals. The selection of the best models based on their validation set yields average AUC values of 0.832 and 0.802 for models based on fingerprint and descriptor, and physicochemical properties, respectively. MTForestNet with an average AUC of 0.825 based on the test set is better than the models based on physicochemical properties and comparable to the models based on fingerprint and descriptor. While they proposed to combine the best-performing models to vote for the final prediction with a higher AUC

of 0.838 based on their validation set, the iterative use of samples from the validation set is prone to overfit the validation set without generalization ability to unseen samples.

Overall, MTForestNet provides an easy-to-use and robust method for predicting six-pack toxicity. The models developed in this study were rigorously validated and independently tested, and performed better than existing methods.

E. Comparison of training times

While good performance was achieved by the MTForestNet, it would be interesting to know the efficiency of the algorithms. We therefore compare the training time of the models. The baseline model requires 58 seconds for training six models. The DeepChem algorithms with early stop enabled are efficient, although with the worst performance. The training times are 40 seconds, 1 minute and 46 seconds, and 7 minutes and 19 seconds for DC_MTN, DC_Bypass, and DC_Progressive. The AG_Medium and AG_Best took the longest training time of 8 minutes and 28 seconds and 5 hours, 48 minutes and 34 seconds, respectively. MTForestNet maintains a well-balanced training time of 7 minutes and 26 seconds and the best prediction performance. Please note that only DeepChem-based models were trained using a GPU. CPU-based training was conducted for the other algorithms, and the model training may be further accelerated by using a GPU.

IV. Conclusion

Distinct chemical space is a unique attribute of biochemical datasets with little or no common chemicals shared among the tasks. Conventional multitask learning algorithms relying on learning a shared representation obtained from the common chemicals may not provide beneficial effects on the prediction performance. This study implemented and compared three types of multitask learning algorithms. Based on the validation and independent test results, we found that the biological readacross-based MTForestNet performed best. Overall, this work represents a significant step toward a biologically grounded and performance-enhancing solution suitable for computational toxicology tasks.

Acknowledgment

This work was supported by the National Science and Technology Council of Taiwan (NSTC-113-2628-E-400-001-MY3).

References

[1] J. V. B. Borba *et al.*, "STopTox: An in Silico Alternative to Animal Testing for Acute Systemic and Topical Toxicity," *Environ. Health Perspect.*, vol. 130, no. 2, p. 27012, Feb. 2022, doi: 10.1289/EHP9341.

[2] Y. N. Fuadah, M. A. Pramudito, L. Firdaus, F. J. Vanheusden, and K. M. Lim, "QSAR Classification Modeling Using Machine Learning with a Consensus-Based Approach for Multivariate Chemical Hazard End Points," *ACS Omega*, vol. 9, no. 51, pp. 50796–50808, Dec. 2024, doi: 10.1021/acsomega.4c09356.

[3] Y. Chushak, J. M. Gearhart, and R. A. Clewell, "Structural alerts and Machine learning modeling of 'Six-pack' toxicity as alternative to animal testing," *Comput. Toxicol.*, vol. 27, p. 100280, Aug. 2023, doi: 10.1016/j.comtox.2023.100280.

[4] B. Ramsundar *et al.*, "Is Multitask Deep Learning Practical for Pharma?," *J. Chem. Inf. Model.*, vol. 57, no. 8, pp. 2068–2076, Aug. 2017, doi: 10.1021/acs.jcim.7b00146.

[5] N. Erickson *et al.*, "AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data," Mar. 13, 2020, *arXiv*: arXiv:2003.06505. doi: 10.48550/arXiv.2003.06505.

[6] Z. Tan, Y. Li, W. Shi, and S. Yang, "A Multitask Approach to Learn Molecular Properties," *J. Chem. Inf. Model.*, vol. 61, no. 8, pp. 3824–3834, Aug. 2021, doi: 10.1021/acs.jcim.1c00646.

[7] X. Qian *et al.*, "An Interpretable Multitask Framework BiLAT Enables Accurate Prediction of Cyclin-Dependent Protein Kinase Inhibitors," *J. Chem. Inf. Model.*, vol. 63, no. 11, pp. 3350–3368, Jun. 2023, doi: 10.1021/acs.jcim.3c00473.

[8] Y. Yuan Li *et al.*, "Co-model for chemical toxicity prediction based on multi-task deep learning," *Mol. Inform.*, vol. 42, no. 5, p. e2200257, May 2023, doi: 10.1002/minf.202200257.

[9] X. Lin, Z. Quan, Z.-J. Wang, H. Huang, and X. Zeng, "A novel molecular representation with BiGRU neural networks for learning atom," *Brief. Bioinform.*, vol. 21, no. 6, pp. 2099–2111, Dec. 2020, doi: 10.1093/bib/bbz125.

[10] Y. Wang *et al.*, "Multitask CapsNet: An Imbalanced Data Deep Learning Method for Predicting Toxicants," *ACS Omega*, vol. 6, no. 40, pp. 26545–26555, Oct. 2021, doi: 10.1021/acsomega.1c03842.

[11] R.-H. Lin, P. Lin, C.-C. Wang, and C.-W. Tung, "A novel multitask learning algorithm for tasks with distinct chemical space: zebrafish toxicity prediction as an example," *J. Cheminformatics*, vol. 16, no. 1, p. 91, Aug. 2024, doi: 10.1186/s13321-024-00891-4.

[12] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[13] C.-C. Wang *et al.*, "Using random forest to predict antimicrobial minimum inhibitory concentrations of nontyphoidal Salmonella in Taiwan," *Vet. Res.*, vol. 54, no. 1, p. 11, Feb. 2023, doi: 10.1186/s13567-023-01141-5.

[14] C.-Y. Chou, P. Lin, J. Kim, S.-S. Wang, C.-C. Wang, and C.-W. Tung, "Ensemble learning for predicting ex vivo human placental barrier permeability," *BMC Bioinformatics*, vol. 22, no. Suppl 10, p. 629, Sep. 2022, doi: 10.1186/s12859-022-04937-y.

[15] C.-C. Wang, Y.-C. Liang, S.-S. Wang, P. Lin, and C.-W. Tung, "A machine learning-driven approach for prioritizing food contact chemicals of carcinogenic concern based on complementary in silico methods," *Food Chem. Toxicol. Int. J. Publ. Br. Ind. Biol. Res. Assoc.*, vol. 160, p. 112802, Feb. 2022, doi: 10.1016/j.fct.2021.112802.

[16] H.-L. Lin, Y.-W. Chiu, C.-C. Wang, and C.-W. Tung, "Computational prediction of Calu-3-based in vitro pulmonary permeability of chemicals," *Regul. Toxicol. Pharmacol. RTP*, vol. 135, p. 105265, Nov. 2022, doi: 10.1016/j.yrtph.2022.105265.

[17] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003, doi: 10.1021/ci034160g.

[18] C.-C. Wang, Y.-C. Lin, Y.-C. Lin, S.-R. Jhang, and C.-W. Tung, "Identification of informative features for predicting proinflammatory potentials of engine exhausts," *Biomed. Eng. Online*, vol. 16, no. Suppl 1, p. 66, Aug. 2017, doi: 10.1186/s12938-017-0355-6.

[19] Y. Low *et al.*, "Integrative chemical-biological read-across approach for chemical hazard classification," *Chem. Res. Toxicol.*, vol. 26, no. 8, pp. 1199–1208, Aug. 2013, doi: 10.1021/tx400110f.

[20] Y. Guo, L. Zhao, X. Zhang, and H. Zhu, "Using a hybrid read-across method to evaluate chemical toxicity based on chemical structure and biological data," *Ecotoxicol. Environ. Saf.*, vol. 178, pp. 178–187, Aug. 2019, doi: 10.1016/j.ecoenv.2019.04.019.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 99–106
DOI: 10.15439/2025F4886
ISSN 2300-5963 ACSIS, Vol. 44

Exploring Multi-Agent Reinforcement Learning for Cell Mechanics

Muhammad Waris
Department of Electronics,
Quaid-e-Azam University
Islamabad, Pakistan
mwaris.22411012@ele.qau.edu.pk

Mustafa Shah
Department of Electronics,
Quaid-e-Azam University
Islamabad, Pakistan
mustafamohmand59@gmail.com

Arsenio Cutolo
Department of Structures for
Engineering and Architecture
University of Napoli Federico II, Italy
arsenio.cutolo@unina.it

Musarat Abbas
Department of Electronics,
Quaid-e-Azam University
Islamabad, Pakistan
mabbas@qau.edu.pk

***Abstract*—Cell aggregation, where cells stick together, is a key process in many biological events like how embryos form, how tissues heal, and how microbes create communities. Studying this involves looking at different types of data, from detailed molecular information to images and patient data. With new technologies, we have access to large amounts of this data in public databases. Analyzing and combining this complex information requires advanced computer methods. While there are challenges in handling and integrating these diverse datasets, exploring them helps us understand basic biology, develop models for diseases, find new drugs, and advance regenerative medicine. This report reviews these data types, sources, and analysis methods to guide research in this important field.**

***Index Terms*—Reinforcement Learning, MARL, Cell Mechanics, Cell aggregation**

I. Introduction

CELL aggregation [3], the process by which individual cells adhere to one another to form multicellular structures [4], represents a fundamental biological phenomenon observed across the tree of life. This self-assembly is not merely a passive physical process but is frequently governed by intricate molecular mechanisms and dynamic cellular behaviors. Aggregation plays critical roles in diverse contexts, ranging from the formation of complex organisms during embryonic development to the establishment of resilient microbial communities known as biofilms [14]. It is also central to physiological processes such as hemostasis, where platelets aggregate to form blood clots, and immune responses, involving the clustering of lymphocytes and other immune cells at sites of infection or within specialized lymphoid tissues. Furthermore, in vitro cell aggregation is the foundational principle behind the generation of three-dimensional (3D) cell culture models, including spheroids and organoids [24], which serve as powerful tools for studying tissue development, disease modeling, and drug screening.

Understanding the intricacies of cell aggregation across these varied biological systems requires the collection and analysis of diverse types of data. Modern high-throughput technologies, such as next-generation sequencing, advanced microscopy, and automated functional assays, are generating vast amounts of quantitative data related to cellular composition, molecular profiles, spatial organization, and dynamic behaviors within aggregating cell populations. Navigating and leveraging these extensive datasets, often stored in public repositories, presents both opportunities and challenges for researchers.

Artificial Intelligence (AI) and Machine Learning (ML) has significant uses in many areas including healthcare [9], vehicular communication [10], e-learning [2], rehabilitation [12] and risk management [7]. Reinforcement Learning (RL) is one the most promising type of ML [17] that has brought revolution in different areas and cell mechanics can also be benefited with this technology. Multi Agent RL (MARL) is the extension of RL where multiple agents are being used for multiple task within a bigger task.

This paper explores the landscape of data relevant to cell aggregation by examining key biological scenarios where it plays a critical role. The types of data generated by various experimental techniques are categorized, and prominent public data repositories where these data are stored and can be accessed are identified. The aim is to provide a structured overview for researchers seeking to utilize existing datasets to study cell aggregation phenomena.

II. Background

Cell aggregation [18] is a fundamental process that underpins the formation, function, and maintenance of biological structures at multiple scales. Its significance spans numerous fields of biological and medical research [19].

In embryonic development, cell aggregation is a primary mechanism driving morphogenesis from the zygote. Following initial cell divisions [5], blastomeres aggregate to form the morula, a compact ball of cells. This compaction is critical

for establishing cell polarity and initiating the first lineage segregation, leading to the formation of the blastocyst with its distinct inner cell mass and trophectoderm. Subsequent aggregation and rearrangement of cells within the developing embryo give rise to the three germ layers—ectoderm, mesoderm, and endoderm—which then differentiate and organize into the precursors of all tissues and organs. The precise timing and spatial control of these aggregation and differentiation events are governed by complex genetic programs and cell-cell communication mediated by signaling pathways.

Organoid formation in the laboratory directly leverages the inherent ability of cells, particularly stem cells, to aggregate and self-organize into 3D structures resembling native tissues. By providing specific biochemical cues, such as growth factors and signaling molecules, and appropriate physical environments, researchers can guide the aggregation and differentiation of pluripotent or adult stem cells to generate organoids mimicking various organs like the brain, intestine, kidney, or liver. These 3D models offer significant advantages over traditional two-dimensional cell cultures by better recapitulating the complex cell-cell interactions, tissue architecture, and physiological functions of their in vivo counterparts.

In the microbial world, biofilm formation is a widespread lifestyle characterized by the aggregation of bacteria and other microorganisms on surfaces, encased within a self-produced extracellular matrix (ECM). This aggregated lifestyle provides significant advantages, including enhanced resistance to environmental stresses, disinfectants, and antibiotics, as well as protection from host immune responses. Biofilms are implicated in numerous industrial issues and persistent infections, making the study of their formation and dispersal critical for developing effective control strategies.

Blood clotting, or platelet aggregation [20], plays a critical role in hemostasis [13], the body's natural process for stopping bleeding after a blood vessel is injured. Platelets quickly gather and stick together at the injury site, forming a plug that's strengthened by fibrin to seal the damaged vessel. While essential for survival, if platelet aggregation becomes uncontrolled, it can lead to dangerous thrombosis—the formation of clots within healthy blood vessels. This can result in serious conditions like deep vein thrombosis, pulmonary embolism, stroke, and heart attack. Understanding the mechanisms of platelet aggregation is therefore vital for diagnosing bleeding disorders and developing treatments to prevent clots.

Immune cell aggregation is a critical aspect of the adaptive immune response. Following recognition of foreign antigens, lymphocytes and other immune cells proliferate and aggregate in secondary lymphoid organs, forming structures like germinal centers within B cell follicles. These aggregates provide specialized microenvironments for processes such as B cell affinity maturation and the generation of memory cells and antibody-secreting plasma cells, which are essential for long-lived immunity and effective vaccination. Immune cell aggregation also occurs at sites of infection or inflammation, facilitating coordinated cellular interactions to clear pathogens or resolve tissue damage.

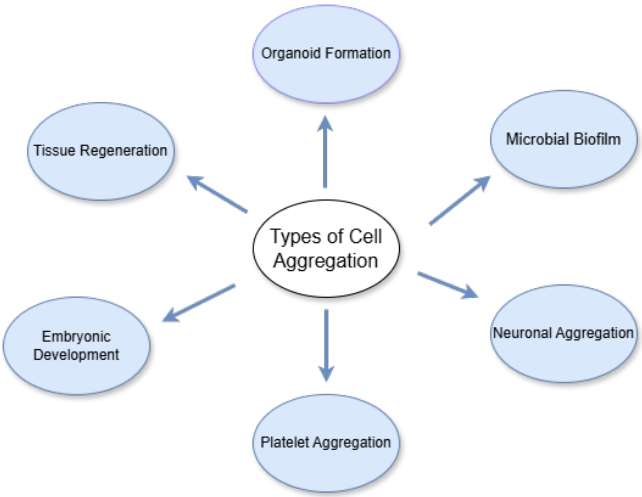


Fig. 1. Types of Cell Aggregation

In the context of tissue regeneration [15] and engineering, cell aggregation techniques are employed to create multicellular building blocks, such as spheroids or organoids, which can be used to repair or replace damaged tissues. Understanding how cells aggregate, maintain viability, and differentiate within these 3D structures is vital for developing effective regenerative therapies [23]. Mechanical forces and cell-cell interactions within these aggregates play a significant role in directing cell fate and tissue organization.

Finally, neuronal aggregation [21] is a key stage in the development of the nervous system. As newly generated neurons migrate to their final destinations in the brain, they aggregate with similar cell types to form distinct brain regions and layers. This process is guided by cell-cell recognition and adhesion molecules and is crucial for establishing the complex circuitry of the brain. Dysregulation [11] of neuronal aggregation and migration is implicated in various neurological disorders.

The study of these diverse cell aggregation phenomena is fundamentally important for unraveling basic biological principles, creating accurate models of human health and disease, and developing innovative therapeutic and biotechnological applications. Multi-Agent Reinforcement Learning (MARL) emerges as a particularly promising computational paradigm. MARL, a specialized subfield of artificial intelligence, is designed to model complex systems where multiple autonomous agents interact and learn in a shared environment. This report explores the application of MARL to the intricate domain of cell mechanics, aiming to address the inherent limitations of traditional computational approaches in fully capturing the multi-agent nature and emergent properties of cellular systems. The subsequent sections detail the necessary revisions to enhance the paper's technical depth and highlight the unique contributions of MARL to this vital field.

III. Data Types Relevant to Cell Aggregation Studies

Investigating the multifaceted nature of cell aggregation necessitates the acquisition and analysis of data across various scales, from the molecular interactions governing cell adhesion to the macroscopic morphology and dynamics of the resulting aggregates.

Molecular Data [16] provides insights into the genetic programs, protein machinery, and signaling networks that regulate cell aggregation and the subsequent behavior of aggregated cells. Transcriptomics, encompassing techniques like bulk RNA sequencing, single-cell RNA sequencing (scRNA-seq), and spatial transcriptomics, provides insight into gene expression patterns. These patterns, in turn, determine a cell's identity, its stage of differentiation, and how it responds to its surroundings within cellular groupings. Genomics data [1], including DNA sequence variations, copy number changes, and epigenetic modifications, provide the foundational genetic and regulatory landscape influencing aggregation potential and associated disease states. Proteomics data [8] identify the proteins present, their abundance, and post-translational modifications, detailing the molecular machinery of cell adhesion, ECM production, and signal transduction within aggregates. Data on signaling pathways, including the activity of receptors, kinases, and transcription factors, illuminate how cells perceive and respond to their environment and coordinate collective behaviors like aggregation, differentiation, and migration.

Cellular Data captures the physical characteristics and activities of individual cells and cell populations within aggregates. Imaging data, acquired through various microscopy techniques (light, confocal, electron, time-lapse, spatial), provides visual information on cell morphology, spatial arrangement, and the dynamic process of aggregation and structural development. Functional assay data quantifies cellular activities such as electrophysiological signaling in neuronal aggregates or organoids, transport function in epithelial structures, or responses to external stimuli like drugs or pathogens. Flow cytometry provides high-throughput, single-cell analysis of protein expression, enabling the identification and quantification of distinct cell types and their activation states within heterogeneous populations, particularly relevant for immune cells and platelets.

Clinical Data [6] provides essential context for studying cell aggregation in disease. This includes patient demographics, medical history, lifestyle factors, treatment regimens, disease severity, and clinical outcomes. Such data are critical for correlating in vitro findings with in vivo conditions and assessing the translational relevance of research, particularly in areas like thrombosis and immune disorders.

IV. Exploring Cell Aggregation Data Across Biological Contexts

The application of these diverse data types varies depending on the specific biological context of cell aggregation bein studied. Each scenario presents unique challenges and opportunities for data exploration .

In the study of organoid formation, a key aspect is understanding how these in vitro aggregates recapitulate the complexity of native organs. Single-cell RNA sequencing (scRNA-seq) is indispensable for dissecting the cellular heterogeneity within organoids, identifying the different cell types that emerge during differentiation, mapping their developmental trajectories, and comparing their molecular profiles to those of cells in primary tissues. Dedicated databases like OrganoidDB serve as valuable resources for exploring organoid transcriptomes, including extensive collections of scRNA-seq data. The inherent variability observed between individual organoids, even within the same culture, underscores the need for high-throughput quantitative data collection and analysis. This variability can be assessed through large-scale scRNA-seq studies of many organoids or through automated imaging analysis. Imaging data, particularly from brightfield, phase contrast, and confocal microscopy, provides crucial information on organoid morphology, size, growth kinetics, and the formation of complex structures like lumens. The large volume of images generated in high-throughput organoid screens necessitates automated image analysis tools, often employing machine learning, to segment, quantify, and track individual organoids. Datasets like MultiOrg specifically provide microscopy images of organoids with annotations for training such tools. Beyond structural and compositional analysis, functional assay data are critical for validating whether organoids truly mimic the physiological activities of their corresponding organs. This includes assessing barrier function, transport activity (e.g., in kidney or intestinal organoids), or electrophysiological signaling (e.g., in brain organoids). The combination of multi-omics (genomics, transcriptomics, proteomics, metabolomics) and functional data is essential for a thorough assessment of organoid authenticity, stability, and translational potential, particularly for applications in disease modeling and drug screening.

Investigating embryonic development requires unraveling precisely controlled spatiotemporal events, including cell aggregation, migration, and differentiation. Gene expression data, from bulk and single-cell transcriptomics, provides a molecular narrative of these processes, revealing which genes are active at different developmental stages and in different cell lineages. However, understanding development requires knowing where genes are expressed within the developing tissue. Spatial transcriptomics addresses this need by mapping gene expression profiles while preserving spatial information, providing molecular maps of embryonic structures and cellular organization. The four-dimensional nature of development (3D space over time) makes the integration of spatial and temporal data particularly crucial for linking molecular events to dynamic cellular behaviors and structural changes. Time-lapse microscopy captures the dynamic morphological aspects of embryonic development, including cell division timings, migration patterns, and the process of aggregation and morphogenesis in living embryos over extended periods. This generates massive datasets, particularly in applications like IVF, which necessitate advanced computational meth-

ods, such as machine learning, to automate analysis, extract morphokinetic parameters, and identify predictive patterns for embryo viability. Public repositories like GEO , TEDD , and the Allen Brain Atlas (Developing Mouse/Human) and BrainSpan provide access to vast amounts of gene expression and anatomical data from developing organisms.

The study of microbial biofilm formation relies heavily on understanding the transition from planktonic single cells to aggregated communities and the molecular mechanisms underlying this process. Genomic and transcriptomic data reveal the genes involved in surface attachment, cell-cell adhesion, ECM production, quorum sensing, and stress responses that are upregulated or downregulated during biofilm development. Multi-omics approaches, integrating genomics, transcriptomics, and proteomics, provide a more comprehensive view of the molecular changes and functional pathways involved in biofilm formation and resistance. Imaging data, particularly from confocal laser scanning microscopy (CLSM), is essential for visualizing the 3D structure of biofilms, including microcolonies, water channels, and the distribution of cells and ECM components. Time-lapse imaging allows tracking the dynamics of biofilm growth and dispersal. Public repositories like GEO and specialized biofilm databases (e.g., aBiofilm, BiofOmics, Biofilms Structural Database, BRaID) serve as sources for genomic, transcriptomic, and sometimes image data related to biofilms. Understanding the molecular mechanisms driving phenotypic shifts during biofilm formation is significantly enhanced by integrating multi-omics data, while imaging captures the essential 3D structure and dynamic processes of aggregation.

Research on platelet aggregation and thrombus formation involves characterizing the rapid cellular and molecular events occurring at sites of vascular injury.Data from aggregometry, especially light transmission aggregometry (LTA) and impedance aggregometry, quantifies how platelets clump together and the degree to which they do so when exposed to different agonists. These assays provide quantitative parameters such as maximum aggregation, slope, and lag phase. Microscopy images, especially time-lapse fluorescence and DIC microscopy of thrombus formation under flow conditions, visualize the process of platelet adhesion, shape change, aggregation, and the incorporation of fibrin and other blood cells into the growing thrombus. These images allow for quantitative analysis of thrombus size, morphology, and dynamics. Flow cytometry is used to analyze platelet activation markers and identify distinct platelet subpopulations within blood samples. Clinical data from patients with thrombotic disorders [22] or bleeding tendencies are essential for identifying risk factors, correlating laboratory findings with clinical outcomes, and evaluating the effectiveness of antiplatelet and anticoagulant therapies. Public resources like clinical trial databases (e.g., ClinicalTrials.gov), disease-specific registries (e.g., ISTH registries), and genomic databases (e.g., NIH GTR) provide access to relevant clinical and genetic data.

The study of immune cell aggregation, such as in germinal centers or at infection sites, involves characterizing the cellular composition, spatial organization, and functional interactions of immune cells. Flow cytometry, including high-dimensional techniques like CyTOF, is widely used to identify and quantify different immune cell subsets based on surface protein expression and analyze their activation states within heterogeneous populations . Repositories like ImmPort house extensive flow cytometry data from immunology studies and clinical trials . Imaging data, such as intravital microscopy, allows visualization and tracking of immune cell migration and interactions in real-time within tissues, providing spatial and dynamic context to flow cytometry findings. Databases like IDR and those linked through the Human Cell Atlas initiatives may contain relevant imaging data. Data on cytokines and chemokines are critical for understanding the molecular signals that mediate immune cell recruitment, activation, and communication within aggregates. Databases like ImmPort and specialized cytokine/chemokine resources (e.g., CYTOCON DB, Cell Interaction Knowledgebase) provide access to these data .

For tissue regeneration and engineering, data focuses on the behavior of cells within aggregates used as building blocks. Cellular data, including viability, proliferation, differentiation status (often assessed via markers), and the impact of mechanical forces or environmental cues, are critical. Imaging data captures the formation, growth, and structural organization of these cellular aggregates, as well as the integration of different cell types in co-cultures. Data on the composition and properties of the extracellular matrix within aggregates or surrounding them (e.g., hydrogels) is also important, as the ECM provides structural support and signaling cues influencing cell behavior.

Understanding neuronal aggregation during brain development involves characterizing the types of neurons, their migratory paths, and how they organize into specific brain structures. Imaging data, including light microscopy, electron microscopy, and various brain imaging modalities (MRI, fMRI), provides visual information on neuronal morphology, connectivity, and the large-scale structure of the brain formed by aggregated neurons . Specific data types include neuronal morphology reconstructions (neuronal tracing data) , electrophysiological recordings of neuronal activity , and gene expression profiles (transcriptomics) related to neuronal development and cell type specification. Large public databases like NeuroMorpho.Org and the Allen Brain Atlas suite provide access to extensive datasets on neuronal morphology, gene expression, and connectivity.

V. Public Data Sources

Access to publicly available data is crucial for advancing research in cell aggregation. Numerous repositories host relevant datasets, often specialized by data type or biological domain.

The Gene Expression Omnibus (GEO) serves as a prominent international public repository for a wide range of high-throughput functional genomics data, including genomic, transcriptomic, and epigenomic datasets, including microarray and next-generation sequencing data. GEO supports various

organisms and experimental conditions, making it a valuable resource for studying gene expression changes during aggregation processes in diverse contexts, including embryonic development, organoid formation, and biofilm development. Data can be searched and downloaded via the GEO DataSets and GEO Profiles interfaces, FTP, or programmatic access.

For organoid-specific transcriptomic data, OrganoidDB provides a comprehensive resource for bulk and single-cell RNA-seq profiles of human and mouse organoids, integrating data from GEO and ArrayExpress. It allows searching and browsing based on organoid type, source, protocol, and developmental stage.

For neuronal morphology and related data, NeuroMorpho.Org is a centrally curated inventory of digitally reconstructed neurons from various species, providing 3D morphological data and associated metadata. The Allen Brain Atlas suite provides extensive resources for neuronal data, including gene expression atlases for adult and developing mouse and human brains, connectivity maps, and single-cell characterization data (morphological, electrophysiological, transcriptomic). Data can be accessed via web portals, APIs, and SDKs . The BRAIN Initiative Cell Census Network (BICCN) also provides access to multimodal brain cell atlas data through various archives like NeMO, BIL, and DANDI.

For immune cell and cytokine/chemokine data, the Immunology Database and Analysis Portal (ImmPort) is a major repository for immunology research data, including clinical trial data, flow cytometry, and multiplex cytokine/chemokine data. ImmPort provides tools for searching, downloading, and analyzing shared data.

The Image Data Resource (IDR) serves as a public repository for imaging data, specifically microscopy images of cells and tissues. This resource archives image datasets from published scientific research, accommodating diverse imaging techniques and organisms. Users can search for and access high-quality biological image data through this platform.

Several resources are available for clinical data concerning thrombotic disorders. These include established clinical trial databases (such as ClinicalTrials.gov), registries specific to diseases (for example, those maintained by the ISTH for rare bleeding disorders or VTE), and certain extensive claims or electronic health record databases, though access to the latter might be limited. Additionally, the NIH Genetic Testing Registry (GTR) offers details on genetic tests relevant to thrombotic conditions.

For histological images, resources like the GTEx Tissue Image Library and specialized datasets like TissueNet or those linked through initiatives like TCGA or Human Protein Atlas provide access to tissue histology images, sometimes with annotations .

For biofilm genomic and transcriptomic data, in addition to GEO, specialized databases like BBSdb and the Biofilms Structural Database (BSD) exist, though access methods vary. Some data may also be available in generalist repositories like Dryad or institutional repositories.

VI. Multi Agent Reinforcement Learning and Machine Learning to Cell Aggregation

A. Cancer: Histopathology and scRNA-seq Data Analysis

1) Image Analysis (Histopathology): Deep learning models (e.g., Convolutional Neural Networks - CNNs) can be trained on histopathology images to identify cancerous aggregation patterns, tumor boundaries, and predict malignancy. Multi-RL can then be used to optimize image segmentation and classification by learning from different expert annotations or even guiding the sampling of new image regions for analysis.

2) scRNA-seq for cell state and interaction: ML algorithms such as clustering (e.g., t-SNE, UMAP, K-means) can identify distinct cell populations and their aggregation tendencies from scRNA-seq data. Multi-RL can be employed to model the dynamic interactions between different cell types (e.g., cancer cells, immune cells, stromal cells) within the tumor microenvironment. Each cell type could be considered an agent, learning optimal strategies for proliferation, migration, or interaction based on the transcriptional states of neighboring cells, allowing for prediction of tumor growth or response to therapy.

B. Wound Healing: Microscopy and scRNA-seq Data Analysis

1) Time Lapse Microscopy for Cell Dynamics: ML algorithms can track individual cell movements and aggregation dynamics from time-lapse microscopy images. Multi-RL can model the collective behavior of cells (e.g. fibroblasts, immune cells, keratinocytes) during wound closure. Each cell or a group of cells can act as an agent, learning policies for migration, proliferation, and extracellular matrix remodeling to optimize healing efficiency, potentially identifying bottlenecks or aberrant healing processes.

2) Spatial Transcriptomics for Cellular Coordination: Integrating scRNA-seq with spatial information allows us to understand how different cell types spatially interact during wound healing. ML can identify spatial gene expression patterns that indicate successful healing. Multi-RL can then simulate the ”decision-making” of cells based on their local environment and gene expression, learning how to coordinate their actions (e.g., secreting growth factors, migrating towards specific cues) to achieve optimal tissue regeneration.

C. Embryogenesis: Live Imaging and Spatial RNA-seq Data Analysis

1) Modeling Morphogenesis: Live imaging data provides dynamic information on cell shape changes and movements. ML models can be trained to predict developmental outcomes based on initial cell configurations. Multi-RL is highly suitable for modeling complex, self-organizing processes of embryogenesis. Each cell or group of cells can be an agent, learning from its neighbors and environmental cues to make ”decisions” regarding division, differentiation, migration, and adhesion, ultimately forming complex tissues and organs. The ”reward” signal could be the successful formation of a specific tissue structure or stage of development.

2) Spatial Transcriptomics for Developmental Programs: Spatial RNA-seq data reveals gene expression patterns across developing tissues. ML can identify gene regulatory networks driving cell aggregation and differentiation. Multi-RL agents, representing different cell lineages, can learn optimal strategies for gene expression changes and physical interactions to achieve proper tissue patterning and organogenesis. This could involve simulating how cells interpret and respond to morphogen gradients and mechanical forces to reach their correct positions and fates.

D. Immune Swarming: Immune Imaging and scRNA-seq Data Analysis

1) Tracking Immune Cell Dynamics: Immune imaging data allows for tracking the movement and interactions of immune cells. ML can identify different immune cell subsets and their migration paths. Multi-RL can simulate immune swarming by treating individual immune cells or groups as agents. These agents can learn to chemotax (move along chemical gradients), interact with pathogens, and coordinate with other immune cells to effectively clear infections or respond to inflammation. The ”reward” could be the successful containment of a pathogen or resolution of inflammation.

2) Predicting Immune Response Outcomes: scRNA-seq provides insights into the transcriptional states of immune cells during aggregation. ML can correlate these states with disease outcomes. Multi-RL can be used to model the adaptive strategies of immune cells in response to evolving threats, optimizing their aggregation and effector functions. For example, agents could learn to upregulate specific receptors, secrete cytokines, or initiate cell-to-cell contact based on the presence of pathogens or signals from other immune cells, leading to a more efficient and coordinated immune response.

E. Neural Aggregation: Brain Organoids and scRNA-seq Data Analysis

1) Predicting Neuronal Migration and Circuit Formation: ML models can analyze time-lapse imaging, gene expression data, and spatial transcriptomics data from brain organoids to predict the trajectories of migrating neurons and the formation of neural circuits. Multi-RL can simulate the intricate dance of neuronal migration and circuit assembly. Individual neurons or neuronal clusters can be agents that learn to navigate complex environments, form connections with appropriate partners (synaptogenesis), and integrate into functional networks. The ”reward” signal could be the successful formation of a mature neural circuit with specific functional properties, as assessed by electrophysiological recordings or imaging data.

2) Modeling Neuroplasticity and Disease Progression: Multi-RL can be used to model neuroplasticity, where neurons learn to adapt their connections and firing patterns in response to stimuli. In the context of neurodevelopmental or neurodegenerative diseases, Multi-RL could simulate how aberrant aggregation or connectivity leads to dysfunction. Agents (neurons) could learn to compensate for damage or disease-related changes, or conversely, models could identify tipping points where the system transitions to a diseased state. This could inform strategies for intervention or rehabilitation.

F. Cardiac Cell Repair: Heart Tissue Imaging Data Analysis

1) Modeling Myocardial: Regeneration Multi-RL can simulate the complex interplay of various cell types involved in cardiac repair, including cardiomyocytes, fibroblasts, and immune cells. Each cell type could be an agent, learning to respond to signals from the damaged microenvironment (e.g., inflammatory cues, growth factors) to contribute to tissue regeneration. This could involve learning optimal strategies for proliferation, differentiation, and secretion of extracellular matrix components to promote functional tissue repair and prevent maladaptive remodeling. This type of modeling could lead to the identification of novel therapeutic targets to enhance cardiac repair.

2) Optimizing Cell Delivery and Engraftment: ML algorithms can analyze heart tissue imaging (e.g., histology, gene expression profiles) to assess the survival, integration, and functional impact of transplanted cells (e.g., stem cells, cardiomyocytes) in damaged heart tissue. Multi-RL can then be employed to optimize cell delivery strategies. Agents (e.g., individual transplanted cells or surrounding host cells) could learn to interact optimally to promote engraftment, vascularization, and functional integration into the host myocardium. The ”reward” could be measured by improvements in cardiac function, reduced scar tissue formation, or successful electrical coupling.

VII. Formulating a Cell Mechanics Problem Into MARL

The process of formulating a cell mechanics problem into a Multi-Agent Reinforcement Learning (MARL) framework requires translating biological phenomena into computational elements while preserving the complex, emergent nature of multicellular systems.

In this formulation:

A. Agents

Agents correspond to autonomous biological cells (e.g., blastomeres, epithelial cells, immune responders), each acting based on local perceptions and internal states.

B. States

States encapsulate multidimensional cell features such as spatial coordinates, polarity vectors, cell cycle phase, gene expression profiles, mechanical tension, and adhesion strength. These may be derived from real-time imaging, transcriptomics, and biomechanical simulations.

C. Actions

Actions include discrete and continuous choices like migration, division, differentiation, polarity realignment, ECM remodeling, and intercellular signaling.

D. Reward Functions

Reward Functions are formulated to capture biologically meaningful objectives—such as optimizing tissue cohesion, minimizing energy expenditure, achieving correct positional fate, or synchronizing morphogenetic movements. These may include sparse or dense feedback and require multi-objective optimization.

E. Environment

Environment refers to the spatial-temporal tissue context, characterized by dynamic morphogen gradients, extracellular matrix properties, boundary conditions, and interactions with neighboring agents.

VIII. Discussion

A. Advantages of MARL Over Single-Agent RL in Cell Mechanics

1) Decentralized Coordination: Biological cells function as autonomous entities, responding to local signals and engaging in self-organized behavior. MARL mirrors this natural decentralization, enabling accurate modeling of emergent developmental processes.

2) Modeling Emergent Properties: Complex multicellular phenomena such as morphogenesis and spatial patterning arise from local interactions. MARL is inherently suited to discover and simulate these emergent properties through distributed policy learning.

3) Robustness to Perturbations : In fluctuating and noisy biological environments, MARL provides resilience by allowing agents to adapt locally. This makes the system robust against disruptions, mimicking biological fault tolerance.

B. Challenges and Future Directions

Despite its transformative potential, applying Multi-Agent Reinforcement Learning (MARL) to cell mechanics is constrained by three core challenges. First, the vast spatial, temporal, and molecular complexity of multicellular systems creates high-dimensional environments that challenge MARL scalability. Second, designing biologically valid and multi-objective reward functions is non-trivial, requiring precise alignment with physiological outcomes. Third, integrating diverse data types like imaging, transcriptomics, and spatial omics into unified agent frameworks demands advanced modeling strategies. Addressing these challenges will require interdisciplinary advances in AI, systems biology, and data integration to fully leverage MARL for biological discovery.

Future research in MARL for cell mechanics should prioritize the development of biologically constrained multi-agent architectures, capable of encoding known intercellular signaling networks and mechanotransduction rules. Hybrid learning models that integrate reinforcement learning with supervised or self-supervised modules will be essential to leverage annotated biological datasets. Simultaneously, scalable data assimilation frameworks must be established to incorporate real-time spatial transcriptomics, live-cell imaging, and dynamic tissue properties. Integrating these MARL systems with in vitro experimental platforms via co-simulation or closed-loop control could enable predictive modeling of morphogenesis and regeneration. Collectively, these efforts will transform MARL into a practical and predictive toolset for mechanobiology, synthetic development, and regenerative engineering.

IX. Conclusions

Cell aggregation is a fundamental biological process occurring across diverse scales and contexts, from the formation of multicellular organisms to the organization of microbial communities and the coordination of cellular responses in health and disease. Studying these phenomena requires integrating data from a wide array of experimental technologies, including genomics, transcriptomics, proteomics, advanced microscopy, functional assays, flow cytometry, and clinical data [12].

The exploration of cell aggregation data is significantly enhanced by the availability of public repositories. Databases like GEO, OrganoidDB, NeuroMorpho.Org, the Allen Brain Atlas suite, ImmPort, IDR, and specialized biofilm and clinical databases provide access to vast amounts of data, enabling researchers to investigate molecular mechanisms, cellular behaviors, and clinical correlations related to aggregation.

The inherent complexity and often high-throughput nature of data generated in cell aggregation studies, such as the large volumes of images from time-lapse microscopy of developing embryos or the high-dimensional data from single-cell transcriptomics and flow cytometry of organoids or immune cells, necessitate the use of advanced computational analysis methods, including machine learning and sophisticated visualization tools.

Future efforts in cell aggregation data exploration should focus on improving data integration across different modalities and repositories, developing standardized metadata and data formats to facilitate data sharing and reuse, and creating user-friendly computational tools that enable researchers from diverse backgrounds to effectively analyze and interpret these complex datasets. By leveraging the wealth of available data and developing innovative analytical approaches, the scientific community can gain deeper insights into the fundamental principles of cell aggregation and translate this knowledge into advancements in regenerative medicine, disease understanding, and therapeutic development.

Acknowledgment

This research has been partially supported by the European Union - Next Generation EU through the Project of National Relevance ”Innovative mathematical modeling for cell mechanics: global approach from micro-scale models to experimental validation integrated by reinforcement learning”, financed by European Union-Next-GenerationEU-National Recovery and Resilience Plan-NRRP-M4C1-I 1.1, CALL PRIN 2022 PNRR D.D. 1409 14-09-2022—(Project code P2022MXCJ2, CUP F53D23010080001) granted by the Italian MUR.

REFERENCES

[1] Database resources of the national genomics data center, china national center for bioinformation in 2025. *Nucleic Acids Research*, 53(D1):D30–D44, 2025. https://doi.org/10.1093/nar/gkae978.

[2] Fares Abomelha and Paul Newbury. A vark learning style-based recommendation system for adaptive e-learning. *Annals of Computer Science and Information Systems*, 41:1–8, 2024.

[3] Anika Alexandrova-Watanabe, Emilia Abadjieva, Lidia Gartcheva, Ariana Langari, Miroslava Ivanova, Margarita Guenova, Tihomir Tiankov, Velichka Strijkova, Sashka Krumova, and Svetla Todinova. The impact of targeted therapies on red blood cell aggregation in patients with chronic lymphocytic leukemia evaluated using software image flow analysis. *Micromachines*, 16(1):95, 2025. https://doi.org/10.3390/mi16010095.

[4] Mario Argenziano, Massimiliano Zingales, Arsenio Cutolo, Emanuela Bologna, and Massimiliano Fraldi. Competition between elasticity and adhesion in caterpillar locomotion. *Journal of the Royal Society Interface*, 22(225):20240703, 2025. https://doi.org/10.1098/rsif.2024.0703.

[5] Hugo Cano-Fernández, Tazzio Tissot, Miguel Brun-Usan, and Isaac Salazar-Ciudad. A mathematical model of development shows that cell division, short-range signaling and self-activating gene networks increase developmental noise while long-range signaling and epithelial stiffness reduce it. *Developmental Biology*, 518:85–97, 2025. https://doi.org/10.1016/j.ydbio.2024.11.014.

[6] Oluwadamilola M Fayanju, Elliott R Haut, and Kamal Itani. Practical guide to clinical big data sources. *JAMA surgery*, 2025. https://jamanetwork.com/journals/jamasurgery/article-abstract/2828666.

[7] Mario Fiorino, Muddasar Naeem, Mario Ciampi, and Antonio Coronato. Defining a metric-driven approach for learning hazardous situations. *Technologies*, 12(7):103, 2024.

[8] Tiannan Guo, Judith A Steen, and Matthias Mann. Mass-spectrometry-based proteomics: from single cells to clinical applications. *Nature*, 638(8052):901–911, 2025. https://www.nature.com/articles/s41586-025-08584-0.

[9] Ahsan Ismail, Muddasar Naeem, Madiha Syed, Musarat Abbas, and Antonio Coronato. Advancing patient care with an intelligent and personalized medication engagement system. *Information*, 15:609, 10 2024.

[10] Mansoor Jamal, Zaib Ullah, Muddasar Naeem, Musarat Abbas, and Antonio Coronato. A hybrid multi-agent reinforcement learning approach for spectrum sharing in vehicular networks. *Future Internet*, 16(5):152, 2024.

[11] Kevin G Johnston, Bereket T Berackey, Kristine M Tran, Alon Gelber, Zhaoxia Yu, Grant R MacGregor, Eran A Mukamel, Zhiqun Tan, Kim N Green, and Xiangmin Xu. Single-cell spatial transcriptomics reveals distinct patterns of dysregulation in non-neuronal and neuronal cells induced by the trem2 r47h alzheimer's risk gene mutation. *Molecular Psychiatry*, 30(2):461–477, 2025. https://www.nature.com/articles/s41380-024-02651-0.

[12] Umamah Khalid, Muddasar Naeem, Fabrizio Stasolla, Madiha Syed, Musarat Abbas, and Antonio Coronato. Impact of ai-powered solutions in rehabilitation process: Recent improvements and future trends. *International Journal of General Medicine*, 17:943–969, 03 2024. https://doi.org/10.2147/IJGM.S453903.

[13] Aziz Kubaev, Fadhil Faez Sead, Mohammad Pirouzbakht, Mobina Nazari, Hani Riyahi, Omolbanin Sargazi Aval, Alireza Hasanvand, Forough Mousavi, and Hamed Soleimani Samarkhazan. Platelet-derived extracellular vesicles: emerging players in hemostasis and thrombosis. *Journal of Liposome Research*, pages 1–11, 2025. https://doi.org/10.1080/08982104.2025.2495261.

[14] Jiaming Lan, Jingyu Zou, He Xin, Jin Sun, Tao Han, Mengchi Sun, and Meng Niu. Nanomedicines as disruptors or inhibitors of biofilms: Opportunities in addressing antimicrobial resistance. *Journal of Controlled Release*, page 113589, 2025. https://doi.org/10.1016/j.jconrel.2025.113589.

[15] Sang Jin Lee, Zhenzhen Wu, Mengyu Huang, Chao Liang, Ziqi Huang, Siyuan Chen, Vidhyashree Rajasekar, Mohamed Mahmoud Abdalla, Haram Nah, Dong Nyoung Heo, et al. Crosslinker-free in situ hydrogel induces self-aggregation of human dental pulp stem cells with enhanced antibacterial activity. *Materials Today Bio*, 31:101451, 2025. https://doi.org/10.1016/j.mtbio.2025.101451.

[16] Tiqing Liu, Linda Hwang, Stephen K Burley, Carmen I Nitsche, Christopher Southan, W Patrick Walters, and Michael K Gilson. Bindingdb in 2024: a fair knowledgebase of protein-small molecule binding data. *Nucleic acids research*, 53(D1):D1633–D1644, 2025. https://doi.org/10.1093/nar/gkae1075.

[17] Cu Kim Long, Vijender Kumar Solanki, Nguyen Viet Anh, Luu Hoang Bach, Cu Ngoc Son, et al. Machine learning-based prediction models for sentiment analysis on online customer reviews: A case study on airbnb. *Annals of Computer Science and Information Systems*, 42:103–116, 2024.

[18] Muddasar Naeem, Mario Fiorino, Pia Addabbo, Antonio Coronato, et al. Integrating artificial intelligence techniques in cell mechanics. *ANNALS OF COMPUTER SCIENCE AND INFORMATION SYSTEMS*, 41:111–116, 2024.

[19] Hafza Qayyum, Syed Rizvi, Muddasar Naeem, Umamah Khalid, Musarat Abbas, and Antonio Coronato. Enhancing diagnostic accuracy for skin cancer and covid-19 detection: A comparative study using a stacked ensemble method. *Technologies*, 12:142, 08 2024. https://doi.org/10.3390/technologies12090142.

[20] Swathy Krishna Reghukumar and Iwona Inkielewicz-Stepniak. Tumour cell-induced platelet aggregation in breast cancer: Scope of metal nanoparticles. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, page 189276, 2025. https://doi.org/10.1016/j.bbcan.2025.189276.

[21] S Berlin Shaheema, Naresh Babu Muppalaneni, et al. An explainable liquid neural network combined with path aggregation residual network for an accurate brain tumor diagnosis. *Computers and Electrical Engineering*, 122:109999, 2025. https://doi.org/10.1016/j.compeleceng.2024.109999.

[22] Christine Van Laer, Renaud Lavend'homme, Sarissa Baert, Koenraad De Wispelaere, Chantal Thys, Cyrielle Kint, Sam Noppen, Kathelijne Peerlinck, Chris Van Geet, Dominique Schols, et al. Functional assessment of genetic variants in thrombomodulin detected in patients with bleeding and thrombosis. *Blood*, 145(17):1929–1942, 2025. https://doi.org/10.1182/blood.2024026454.

[23] Hilal Yilmaz, Israa F Abdulazez, Sevda Gursoy, Yagmur Kazancioglu, and Cem Bulent Ustundag. Cartilage tissue engineering in multilayer tissue regeneration. *Annals of Biomedical Engineering*, 53(2):284–317, 2025. https://link.springer.com/article/10.1007/s10439-024-03626-6.

[24] Mengru Zhu, Hao Zhang, Qirong Zhou, Shihao Sheng, Qianmin Gao, Zhen Geng, Xiao Chen, Yuxiao Lai, Yingying Jing, Ke Xu, et al. Dynamic gelma/dna dual-network hydrogels promote woven bone organoid formation and enhance bone regeneration. *Advanced Materials*, page 2501254, 2025. https://doi.org/10.1002/adma.202501254.

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 107–107 DOI: 10.15439/2025F9634 ISSN 2300-5963 ACSIS, Vol. 44

Paths to Zero Emission Computing—Reducing Energy Consumption, and carbon emissions in HPC and AI environments

Tikiri Wanduragala
Lenovo, UK
twanduragala@lenovo.com

***Abstract*—In this position note, core issues involved in creation of zero emission data centers are summarized.**

***Index Terms*—data center, zero emission, HPC.**

DIGITAL transformation projects combined with rebalancing workloads between public and private clouds for reasons of sovereignty have given rise to an increase in demand for compute capacity globally. Conventional systems and data center designs have been able to accommodate the projected growth. However, the rise of AI and more demanding HPC environments using large numbers of GPU's and associated networking, high bandwidth memory and storage systems have radically changed systems design and energy requirements. In recent years power consumption has grown by at factor of 3 up to 500W for CPU's and 1000W+ for GPU's. This has resulted rack power consumption increasing from about 15KW to100KW+.

In essence HPC and AI environments consume significantly more energy which can result in increased CO2 emissions and water usage at the data center level. The International Energy Agency (IEA) Energy and AI report (April 2025) projects global Data Centre (DC) electricity consumption to double to 945TWh globally by 2030. A Mckinsey article on "The Cost of Compute – a $7 trillion race to scale data centers" (April 2025) put the "global demand for data center capacity could almost triple by 2030, with about 70 percent of that demand coming from AI workloads".

Conventional data centres and systems are struggling to cope with the demands being placed on them and place limits on the capabilities of HPC and AI unless these issues are addressed. With traditional air-cooled systems as much as 40% of the energy provided for compute can be lost by the cooling systems.

Lenovo is a leader in providing HPC and AI factory solutions globally, experience in designing energy efficient systems. From SW to gather data from the underlying infrastructure to optimise for performance or energy usage to a range of HW solutions branded as Lenovo Neptune offer both air- and water-cooled solutions to ensure energy is utilised as efficiently as possible.

As data centers take several forms and are specific to the workloads that are designed to run on the systems inside them. Classic air-cooled systems offer the most flexibility and in Hot / Cold Aisle configuration can achieve a PUE of 1.5 to 1.6. Air cooled systems combined with rear door heat exchangers using chilled water can improve the PUE to about 1.2. Direct warm water-cooled system can improve the PUE to 1.1 to 1.06 range.

Moving from air to direct warm water-cooled systems can result in several significant benefits:

- Density more compute power in a compact footprint
- Optimal performance by keeping components within thermal design power envelope
- Possibility of reducing the carbon footprint of the installation
- Possibility of re-using waste heat in other campus location
- Higher performance per watt

In the recent past the use of water-cooled technologies was in the realm of HPC and AI installations. As HPC and AI workloads are being deployed within enterprise computing environments a hybrid approach is sometimes taken with less demanding computational workloads running on air-cooled systems and the more demanding running on direct water-cooled systems.

To realise the full potential of HPC and AI in a cost-effective sustainable manner efficient use of energy is essential. Tried and tested technologies like Lenovo Neptune™ direct warm water cooling not only offers compute efficiencies but capturing the waste energy in water offers the possibility of heat re-use but heating campus buildings making carbon neutral computing a possibility.

Topical area: Computer Science & Systems

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 109–117
DOI: 10.15439/2025F7442
ISSN 2300-5963 ACSIS, Vol. 44

DBRow: A Density-Based algorithm for autonomous navigation within crop rows

Peder Ø. Bukaasen, Weria Khaksar
Norwegian University of Life Sciences, Ås, Norway
Email: peder.ormen.bukaasen@nmbu.no; weria.khaksar@nmbu.no

***Abstract*—This paper introduces DBRow, a density-based algorithm designed to improve autonomous navigation within crop rows, addressing the growing need for efficient agricultural robotics to boost productivity and tackle labour shortages. DBRow integrates Simultaneous Localisation and Mapping (SLAM) with Density-Based Spatial Clustering of Applications with Noise (DBSCAN), overcoming the limitations of previous navigation systems that relied solely on LIDAR data for NMBU's FRE participation. Experiments conducted in simulated and controlled indoor environments evaluated DBRow using A* path planning algorithm. The results show some weaknesses in the simulated environment, but it performs well in the controlled indoor environment. The paper calls for further testing for statistically significant results and suggests future enhancements, including LIDAR preprocessing improvements and machine learning integration, to optimise navigation accuracy and automate tasks like pesticide application.**

Keywords: Robotics, Navigation, Agriculture, Farming, crop, autonomous

I. Introduction

AGRICULTURAL robotics is a broad field that involves various robots performing tasks in agricultural environments, replacing or aiding humans. Such robots are often divided into self-propelled mobile robots and robotic sensors or actuators carried by a vehicle [24]. This paper focuses on the first one, self-propelled mobile robots.

Navigating crop environments seems like a straightforward task for humans: go in the middle of the row and do not destroy any plants. Enabling navigation for a mobile robot requires more work. First, the robot needs to have some representation of its environment so that it can plan when and where it should go. The representation of the environment in this paper is a map created by a SLAM algorithm. To navigate its environment, the robot needs a planner and a controller to move the robot; the Robot Operating System 2 (ROS2) Navigation stack solved this. To autonomously navigate, an algorithm was needed to set goal points. Here, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was used to extract rows and goal positions were extracted from these rows.

II. Methodology

A. DBSCAN

DBSCAN is a clustering algorithm that can extract clusters of varying sizes, assuming they have roughly the same density. This algorithm was first proposed in [7]. As the name implies, DBSCAN uses the densities of points to assign cluster labels. Density in DBSCAN is defined as the number of points within a specified radius eps. In the literature, this radius is also denoted by ϵ. Compared to other clustering algorithms, one advantage of this algorithm is that it does not require a number of clusters to find as input. Two other advantages are that it does not make assumptions about spherical clusters as k-means clustering does, and it does not partition the dataset into hierarchies that require some manual cut-off. The DBSCAN algorithm uses three-point labels: core, border, and noise points. These points are defined in this way:

- Core points have a minimum number of points ($MinPts$) within the radius eps.
- Border points fall within the eps of another core point but do not satisfy the $MinPts$ within the radius eps.
- Noise points neither satisfy the condition for border nor core points.

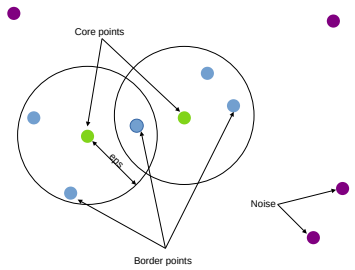


Fig. 1. Shows the different points and the eps variable used in the DBSCAN algorithm. All the purple points are noise, all the green points are core points, and all the blue points are border points.

Figure 1 shows how DBSCAN can label these points. The $MinPts$ in this example is three, and the circle shows the radius around the green points. One can see that the green points are labelled as core points since they contain three or more points within eps. The blue points are labelled border points since they fall within the radius of the core points. The purple points are labelled noise since they do not satisfy the conditions for core or border points. The algorithm can be simplified to:

1) Label all points into the three different point labels.
2) Create separate clusters for all core points or groups of core points. Two core points are considered to be in the

same cluster if they fall within the radius eps of each other.
3) Assign all border points to their respective core points.

Using these simple steps, DBSCAN can detect clusters of any shape or form as long as they are separated and have similar densities [19].

B. RANSAC

Random Sample Consensus (RANSAC) is an algorithm for fitting models to experimental data. Fisher and Bolles proposed the algorithm in 1981 [8]. The RANSAC algorithm can be seen as a trial-and-error approach to fitting data to a model where the dataset is contaminated with noise. RANSAC can be explained in four easy steps:

1) Sample the number of data points needed to fit the model.
2) Calculate the model parameters from the collected data points.
3) Score the model by the number of inlines with a predefined threshold.
4) Repeat the above steps until the best possible model is found.

Using these simple steps, RANSAC is able to fit models to the given data [5].

C. A*

The A* search algorithm was first introduced in 1968 in [10]. A* is a widely used pathfinding and graph traversal technique that utilised the strengths of both Dijkstra's algorithm and Greedy Best-first search. It is designed to efficiently compute the shortest path from a starting point to a goal node in a graph, making it particularly useful in robotics and game development. The algorithm is graph-based, and the conversion described in the path planning section is necessary for this algorithm. A* integrates the methodical search of the Dijkstra algorithm with the heuristic-driven guidance of the Greedy Best-First search. This guidance is implemented as two metrics:

- **g(n)**: The exact cost from the start node to the current node n.
- **h(n)**: The heuristic estimate of the cost from node n to the goal node.

A* evaluates paths by minimising this function:

$$f(n) = g(n) + h(n) \tag{1}$$

This function ensures a balance between the actual cost from the start and the estimated cost to reach the goal; this leads to efficient and optimal path planning [17]. The algorithm, when stripped down to its basics, is quite simple; it uses two sets: Open and Closed. The Open contains nodes that are candidates to explore. Initially, the Open set contains only the starting position. The Closed set contains nodes that have already been examined and begin empty. Each node contains a pointer to its parent to help create the optimal path at the end. The algorithm runs through a main loop that repeatedly selects the best node n from the Open set, which is the node with the lowest $f(n)$ score, and examines it. If n is the goal, the process ends; otherwise, n is moved from the Open set to the Closed set. Then, the neighbours of n that are already in the closed set are ignored, and the neighbours in the open set are scheduled to be examined. If a neighbour is not in the Open or the Closed set, it is added to the Open set with parent n [16].

D. Hardware

The robot platform used in the simulation is Peik, which Bård Tollef Pedersen and I built. In Figure 2(a), one can see an image of Peik without any sensors mounted. The specs of this robot platform are:

- **Weight:** 19.56 kg
- **Onboard computer:** Nvidia Jetson Nano ORIN
- **Operating system:** Jetpack 6.0
- **Steering type:** Skid-steer
- **Driven motors:** 4 x 350W motors, two on each side
- **Battery:** 36v 4.4Ah
- **Footprint (L x W x H):** 42 cm x 32 cm x 25 cm
- **Max speed:** 5.55 m/s
- **LIDAR:** Ouster OS1-64 (in the simulation)

For testing at the robotics lab, A Turtlebot3 Burger[22] was used. In Figure 2(b), one can see the Turtlebot3 Burger. The Turtlebot3 Burger has the following specs:

- **Weight:** 1 kg
- **Onboard computer:** Raspberry Pi 4
- **Operating system:** Ubuntu Server 22.04.5 LTS (64-bit)
- **Steering type:** Differential drive
- **Driven motors:** 2 x
- **Battery:** 11.1v 1800 mAh
- **Footprint (L x W x H):** 13.8 cm x 17.8 cm x 19.2 cm
- **Max speed:** 0.22 m/s
- **LIDAR:** LDS-02
- **IMU:** Gyroscope 3 Axis, Accelerometer 3 Axis

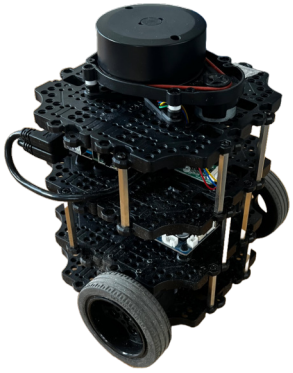
The simulations of Peik were run on a computer with a dedicated GPU. The PC used for these simulations had these specs:

- **Processor (CPU):** Intel Core I7-8700 6-Core 12-Thread 3.2/4.6 GHz
- **Graphics Processing Unit (GPU):** Nvidia GeForce GTX 1080
- **Memory:** 16 GB DDR4 2666 MHz
- **Storage:** 1000 GB M.2 SSD
- **Operating System:** Ubuntu 22.04 Jammy Jellyfish

Since the Turtlebot3 only has a Raspberry Pi 4, the code was run on a PC connected to the Turtlebot3. The computer had these specs:

- **Processor (CPU):** Intel Core Ultra 5 125H 14 Core 18 Thread 1.2/4.5 GHz
- **Memory:** 16 GB LPDDR5X
- **Storage:** 1000 GB M.2 SSD
- **Operating System:** Ubuntu 22.04 Jammy Jellyfish

(a) Peik



(b) Turtlebot3 Burger

Fig. 2. Shows the robots used in this paper, Peik (a) without any sensors mounted and the Turtlebot3 Burger (b) with the LDS-02 LIDAR.

E. Setting up the simulation environment

Setting up the simulation environment in Gazebo consists of a few steps: creating a world for the robot to move in, creating a robot model, and adding a control system and sensors to the robot. The world used for simulation in this project is generated with code from this GitHub repository [3]. This code has several options that change how the created field looks; these options can change how straight the rows are if there are holes in the crop rows and the size of the plants. These values can be specified in a YAML file. This paper uses three different simulated worlds with the main difference being the roughness of the terrain.

The robot used in this paper is a simulated version of Peik. Peik is the robot created for NMBU's participation in the Field Robot Event (FRE). A simulated version of this robot was created using Unified Robot Description Format (URDF) and Gazebo plugins for sensors and controlling the robot. The simulated robot was simplified to a box with four wheels. Utilising Xacro, the URDF files were further simplified with macros for values used several times, such as the wheel's mass or the offset of the different wheels. The base of the robot was created as a box with tags for the visual, collision and inertial. The wheels of the robot were connected using continuous joints, and the wheels also had tags for visual collision and inertia. Controlling the robot in Gazebo was done with this plugin *libgazebo_ros_diff_drive.so* [9]. This plugin leverages the wheel joints, wheel size, and wheel separation to facilitate robot control via the /cmd_vel topic. Additionally, it publishes odometry data to track the robot's movement. Simulating the Ouster OS1-64 was done by using the *libgazebo_ros_velodyne_laser.so* plugin [23] and changing the values for horizontal and vertical scans and ranges to match those of the Ouster OS1-64 [15]. This plugin simulates the LIDAR in Gazebo and publishes the point cloud to a topic. In Figure 3, one can see the simulated version of Peik in the virtual maize environment.

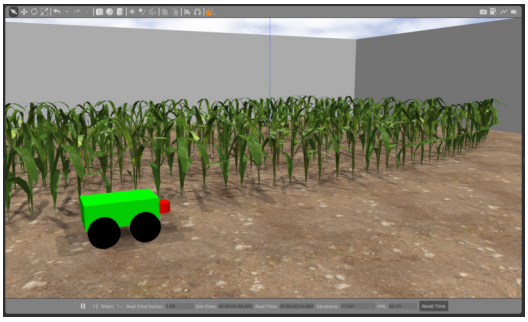
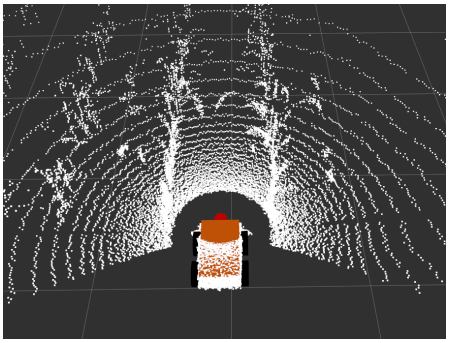


Fig. 3. Shows the robot with a LIDAR sensor in the simulated maize field environment. Here visualized in Gazebo.

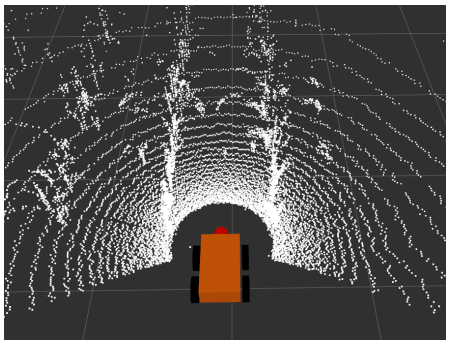
F. LIDAR preprocessing

This part is only used for the simulated robot since the Turtlebot3 had a 2D lidar and navigated in an indoor environment. The simulated lidar is a 3D sensor, and since Nav2 is mainly used for 2D data, this point cloud was projected into two dimensions. Before it was projected, some points had been removed. Firstly, the points that fell on the robot's chassis needed to be removed. This was accomplished by using a pcl::CropBox filter from the Point Cloud Library (PCL) [18]. This was a straightforward process of creating a box representing the robot, and the filter removed all points inside the box. The unprocessed point cloud can be seen in Figure 4(a), and the point cloud with the points from the robot filtered away can be seen in Figure 4(b).

Additionally, the ground plane needed to be filtered out since it was not used for navigation. The ground plane was filtered from the point cloud using RANSAC to fit the point cloud to a plane model, utilising the RANSAC filter from PCL. In this filter, restrictions were put such that the plane's normal must be within an angle threshold of the z-axis. This filter also had a maximum number of iterations to make sure that it did not run forever. Removing points was done using a threshold, and all points that fell within a threshold of the fitted plane were removed. Trial and error were used to find

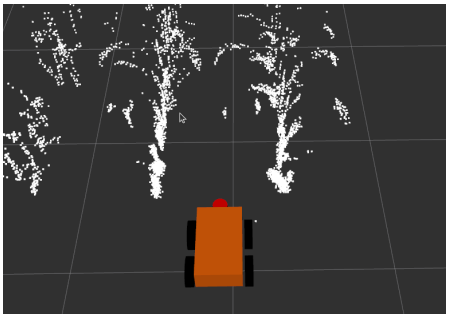


(a) Point cloud

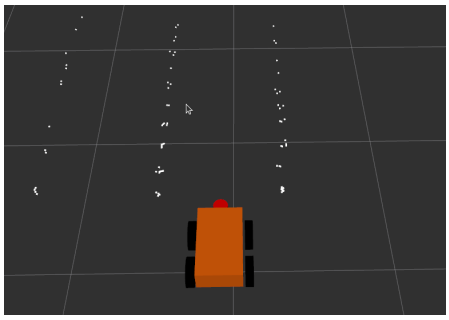


(b) Filtered point cloud

Fig. 4. (a) shows the point cloud from the simulated ouster and (b) shows the point cloud with the robot footprint filtered.



(a) Ground removed point cloud



(b) Laser scan

Fig. 5. (a) shows the point cloud where RANSAC has removed the ground, and (b) shows the projected laser scan.

the best parameters for this algorithm. The point cloud with the ground removed can be seen in Figure 5(a).

Finally, projecting the 3D point cloud into a laser scan was done utilising the pointcloud_to_laserscan package [20]. This package has several options for converting the point cloud to a laser scan; among these are min_height and max_height, which are the minimum and maximum heights to sample from the point cloud. These two parameters were tuned such that the points that remained in the laser scan mainly consisted of plant stem points. This laserscan can be seen in Figure 5(b).

1) Cartographer: Cartographer is a project developed by Google that provides a real-time solution for indoor 2D mapping using a sensor-equipped backpack [11]. The algorithm is also integrated with ROS with the cartographer ROS project. This implementation also offers real-time SLAM for 2D and 3D environments [4]. The SLAM algorithm used by Cartographers combines local and global optimisation strategies to maintain accurate mapping. Both of these approaches aim to optimise the pose of LIDAR scans [11]. The two different optimisation strategies are implemented as two related subsystems: the local and the global SLAM. Local SLAM constructs submaps that are locally consistent, accepting that they may drift over time. It handles immediate data from sensors to build submaps that are small enough to ensure local accuracy but large enough to be distinct for effective loop closure. Global SLAM runs in the background, focusing on loop closure by scan-matching scans against submaps and incorporating additional sensor data for the most consistent global map.

2) Nav2: Nav2 is a toolbox for ROS2 that allows autonomous navigation of mobile and surface robots. It is a successor to the ROS Navigation Stack and provides packages for perception, planning, control, localisation, and visualisation. Nav2 uses behaviour trees to enable autonomous navigation, which is achieved using several independent modular servers. A server can be used to localise the robot on the map or plan a path from point A to point B. These servers communicate with the behaviour tree using the different ROS2 interfaces: services, actions and topics [13]. The core of the navigation problem can be seen as planning and controlling a robot. Four of the servers in the Nav2 stack provide a robust solution for planning and control: Planner, Controller, Smoother and recovery servers [14].

G. Configuring Cartographer and Nav2

Configuring Cartographer is done by creating a *.lua file with all the parameters needed to launch the Cartographer package. This file was created by consulting the tuning guide [2], and the Lua configuration reference documentation [12].

Configuring Nav2 can be quite demanding due to its multiple components that require careful configuration and tuning. For the initial setup of the planners and controllers, the guide referenced in [21] was utilised, which outlines when to use different planners and controllers, as well as their suitability for various types of robots.

H. Navigation algorithm

The navigation algorithm can be divided into the in-row and switch-row algorithms. This subsection explains the two algorithms and the implementation of them. Both of these navigation algorithms use the information from the global costmap to set navigation goals. The data from the laser scan mainly contains plant stem values, which means that the data in the global costmap also contains plant stems. The in-row navigation is best described in some simple steps:

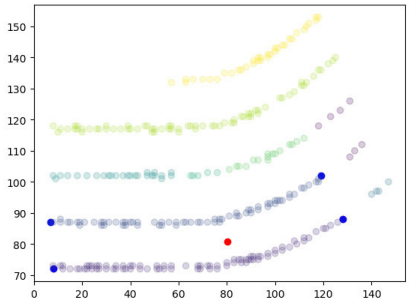
1) Get the robot's position and costmap data.
2) Cluster the costmap data using DBSCAN.
3) Find the two closest clusters to the robot, then for each of these clusters, find the furthest points from each other within its cluster.
4) From these four points, find the two closest pairs.
5) For these two pairs, calculate the mean, which should be two points in the middle of the crop rows, one in front of the robot and one behind the robot.
6) Transform the coordinates of these points into the coordinate system of the robot to easily calculate which goal is behind the robot.
7) Check if one of these goals is in front of the robot.
8) If one of the goals is in front of the robot, navigate to this goal. If not, the robot is at the end of the row.
9) When the goal is reached, repeat from step 1.

To obtain the map's position and the robot's pose, subscribers were created to track the global cost map from the global cost map topic and the robot's pose from the tracked pose topic. Extracting clusters from the cost map data involved several steps. First, lethal obstacles, defined as all values greater than 100, were extracted from the cost map. These values were then converted into a NumPy array.

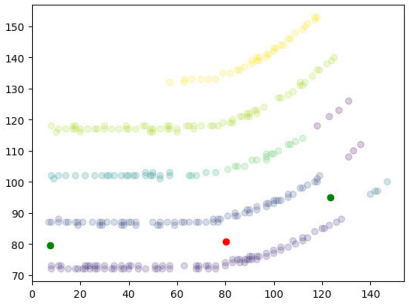
Using this array, the DBSCAN function from scikit-learn [6] was applied to cluster the data into crop rows. Next, the two closest clusters were identified by iterating through all clusters and calculating the distances between them, retaining only the two with the smallest distances. Simultaneously, the two furthest points within each cluster were determined by employing a nested loop to calculate the maximum distances. The two points that were furthest apart were saved, along with the corresponding distance for each cluster.

The goal was defined using the four points that represented the furthest distances from each other within the two closest clusters, visualised by the blue points in Figure 6(a). The four distances between these points were then compared to identify the two pairs closest to each other. From these pairs, two potential goal positions were calculated by finding the mean of the two closest pairs, represented by the green points in Figure 6(b).

Since the goals were referenced in the map's coordinate frame, they were transformed into the robot's coordinate system to determine which goals were in front of the robot. In this transformed frame, the goals behind the robot had negative values, while those in front had positive values. The goal with the largest x value was selected. If this goal was too close to the robot, it was also considered behind the robot. If both goals were determined to be behind the robot, the process would terminate, indicating that the robot had reached the end of the row. To navigate to these goals, Nav2's simple commander was used. The algorithm for switching between the rows has



(a) Four furthest points



(b) Two possible goals

Fig. 6. Shows the clustered points, the red points represent the robot's position, the blue points represent the further four points in the two closest clusters, and the green points show the two possible goals. The transparent points show the different clusters. (a) shows the points used to calculate the goals, and (b) shows the two possible goals.

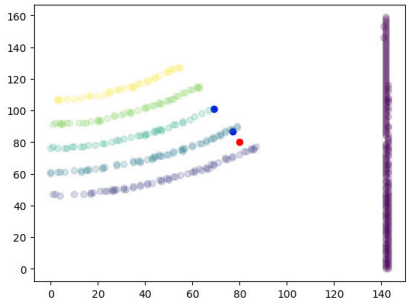
the same first two steps as the navigate row algorithm: getting the robot's position, map data, and clustering the rows. This algorithm can be described in these steps:

1) Get the robot's position and costmap data.
2) Cluster the costmap data using DBSCAN.
3) Find the minimum distance and the closest point in each of the clusters.
4) Transform the coordinates of these clusters into the robot's coordinate frame.
5) If turning left, keep all closest cluster points with a positive y value; if turning right, keep all closest cluster points with negative y values.
6) Select the two closest points from these clusters and find the mean of them, this mean is then the goal point.
7) Navigate to this goal.

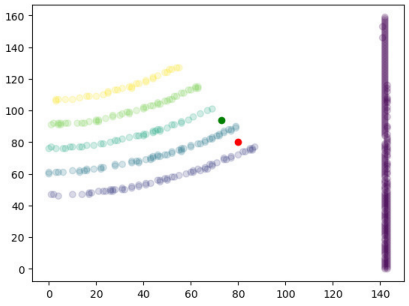
The first two steps of the switch row algorithm are the same as the navigate row algorithm. Therefore, they will not be explained further. This algorithm was implemented as an

action server in ROS2 Humble, and it also had a custom action.

The third step of the algorithm was completed by looping through all the clusters and calculating the closest distance from each cluster to the robot. The closest point was then saved together with the distance for each cluster. These points were then transformed into the coordinate frame of the robot. This transformation was done to easily separate the clusters to the left and right of the robot using the y-axis of the robot's coordinate frame. The cluster points with positive y values are to the left of the robot, and all cluster points with negative y values are to the right of the robot. Depending on the turning direction, a list of interesting clusters was created, containing only the cluster points and distances to the left or the right. From this list, the two closest clusters were selected, and the goal position was calculated as the mean of the closest points in these two clusters. These two points can be seen by the blue points in Figure 7(a), and the goal can be seen in Figure 7(b) by the green point. The heading of this goal was set to the inverted heading the robot had when standing at the end of the row, which was inverted by adding 180 degrees to it. The goal was then sent to the Nav2's simple commander. This implementation returned true if it was able to calculate and navigate to the goal; otherwise, it returned false. These



(a) Closest clusters points



(b) Goal in next row

Fig. 7. Shows the clustered points, the red points represent the robot's position, the two closest clusters closest points, and the green points show the goal position. The transparent points show the different clusters. Figure (a) shows the two closest, and Figure (b) shows the goal.

two action servers were used to navigate the entire field using two action clients implemented in one ROS2 node. For this to work, the number of rows to navigate and the first turning direction need to be specified. This node starts by initiating the action clients and the required variables for tracking the navigation, like row number and initial turning direction. Then, a goal is sent to the navigation row server. This node then waits for the node to finish while receiving and printing the feedback. When the navigate row server finishes, the switch row server is called. The robot then switches to the next row and the direction of the switch alternates between switching to the left and right. The node waited for the execution of this action server while receiving and monitoring the feedback. This node alternated between calling the navigate row action and calling the switch row action until the specified number of rows were navigated or the navigation failed, and the node shut down.

I. Experiment setup

The experiments conducted in this paper can be divided into two types: those conducted in the simulator with virtual maize plants and those conducted in the robotics lab with Turtlebot3 and thuja plants. Here, five runs in each were completed, following the design from here [1]. The simulated environment was created with five plant rows, with approximately 70 centimetres between them. In Figure 8, one can see the layout of the field used in the simulated run.

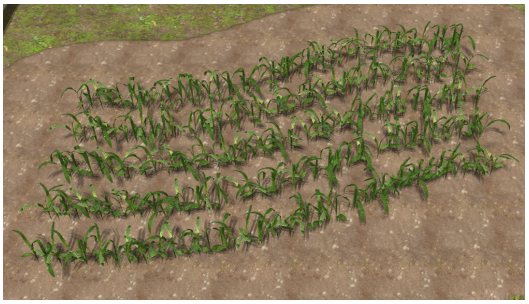


Fig. 8. Shows the layout for the simulated maize field.

In Figure 9, one can see the terrain from the simulated environment.

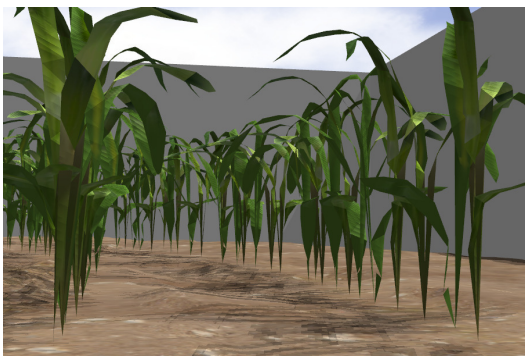


Fig. 9. Shows the terrain for the simulated maize environment

The testing environment in the robotics lab at NMBU was created using thuja plants. These plants were used because they were the only available plants in the robot lab. This environment was similar to the even simulated environment in the sense that they both have a very even ground. In the testing with the Turtlebot3, the LIDAR preprocessing steps were skipped since this robot has a 2D LIDAR. Four rows of plants were created, which meant three rows for the Turtlebot3 to navigate. These rows were approximately 3 meters long and had 0.7 meters between each row. In Figure 10, a square at the start of the row is visible. This square was used as a starting position for the robot to ensure similar conditions for all the runs.

Fig. 10. Shows the testing environment in the robotics lab at NMBU.

Measuring the performance of the algorithm was done using parameters similar to those used in the FRE competition. The performance was measured by the time used to navigate, the distance travelled, and how many plants were damaged. Calculating the distance and time was done by creating a node that subscribed to the position of the robot. This node was started by using a topic to publish, starting and stopping from the navigate field node. This node then used the positions over time to calculate the distance traversed by the robot. Due to an uncertainty in the position of the robot, a threshold was used to eliminate noise in the position data. A plant that the robot touched was not considered damaged; for a plant to be considered damaged, it had to be lying on the ground or visibly damaged. Detecting damaged plants was done by observing the simulation and the Turtlebot3 in the robotics lab.

III. Results and Discussion

A. Simulated environment

In this subsection, the results from the simulation terrain runs are presented. In Table I, the results are presented. Here, the average number of completed rows was 3.0, and the average number of damaged plants was 2.0. All of the plant damage occurred in run three. The average distance the robot managed to travel was 24.08 meters, and the average time was 234.0 seconds.

In Figure 11, the paths taken by the robot using A* are shown in the red line, and the green points show the plant's ground truth positions. In run three, ten plants were damaged, and where the plants were damaged can be seen by the overlap between the red line and the green plants. Runs two and five did not complete the field. Runs three and four had some assistance at the last row. The overlap of the red path and the green points in Figure 11. In runs two and five, one can see where the robot failed in the middle of row two. The robots in these runs were able to navigate 24.08 meters on average, which is good since the entire field is about 30 meters. This is promising for using DBSCAN to navigate crop rows. In the third run, one can see that the robot struggled a lot; this could be due to the rough terrain, causing the RANSAC not to be able to fit the plane and remove the ground points. This would add noise to the input data of the algorithm and could be the cause of the plant damage in this field, And also, what caused the robot to fail in the second and fifth runs. The robot damaged plants can be seen by the overlap of the red line and the green points in Figure 11. Another possible explanation for the poor performance in runs two, three and five could be that the rough terrain makes the point cloud laser scan pick up leaf points in the middle of the rows due to the robot being tilted.

TABLE I
Shows the results using A* in the simulated environment

Run Number	Plants damaged	Distance [m]	Time [s]
1	0.0	30.64	215.0
2	0.0	11.80	216.0
3	10.0	36.30	466.0
4	0.0	30.53	199.0
5	0.0	11.11	74.0
Average	2.0	24.08	234.0

B. Turtlebot3

This subsection presents the results from the testing in the robotics lab using the Turtlebot3 Burger. Table II shows the results for the A* planner with the Turtlebot3. Here, the Turtlebot3 managed to complete the three rows in all runs without damaging any plants. The average distance used was 9.29 meters with an average time of 99.2 seconds.

TABLE II
Shows the results for the runs with the Turtlebot

Run Number	Plants damaged	Distance [m]	Time [s]
1	0.0	9.23	95.0
2	0.0	8.72	96.0
3	0.0	9.17	94.0
4	0.0	9.51	95.0
5	0.0	9.82	116.0
Average	0.0	9.29	99.2

In Figure 12, the paths taken by the Turtlebot3 can be seen for the five runs using A*. The path in red is plotted here on the map generated by cartographer slam. Here, one can see that the robot mainly navigated to the middle of the rows and kept a distance when switching between the rows using A*. The first row here is the lowest row of plants in Figure 12, and the last row is the top row. At the beginning of the second row

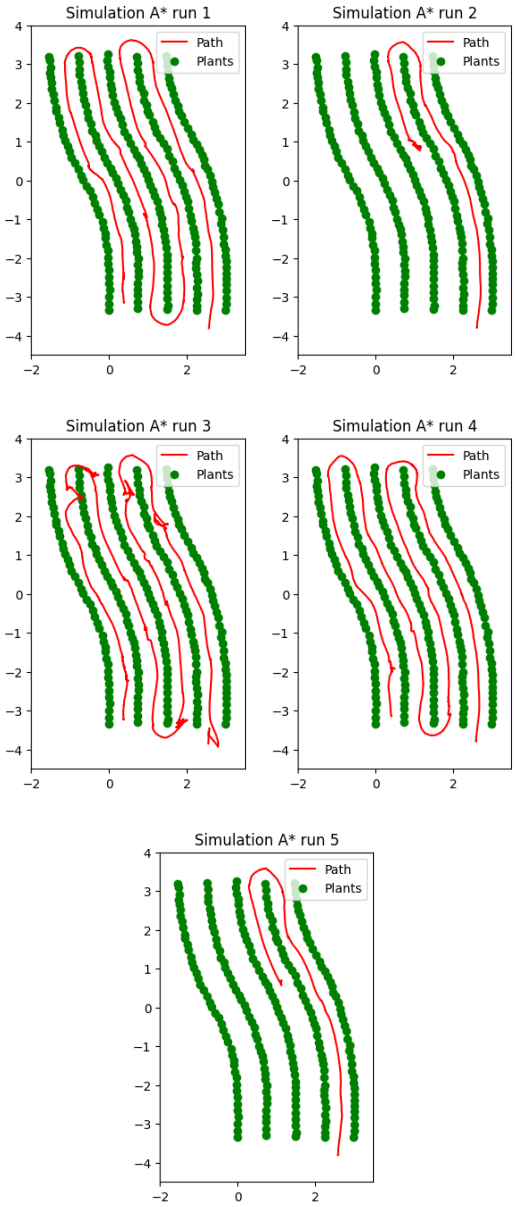


Fig. 11. Shows the five runs conducted in the rough terrain with A*. The green points visualise the ground truth position of the plants, and the red line visualises the path taken by the robot.

in run two, one can see that the robot navigated a bit closer to the plants.

The runs with the Turtlebot3 show good promise for this algorithm. One explanation of this performance could be that this environment is much simpler than the simulated one. The Turtlebot3 also did not need the preprocessing steps used in the simulated environment to remove the ground and extract stem points, since it used a 2D LIDAR and the ground in this environment was flat.

For both the Simulated and runs, one could not draw any definite conclusions since only five runs were conducted. These runs can only give an indication of the algorithm's performance.

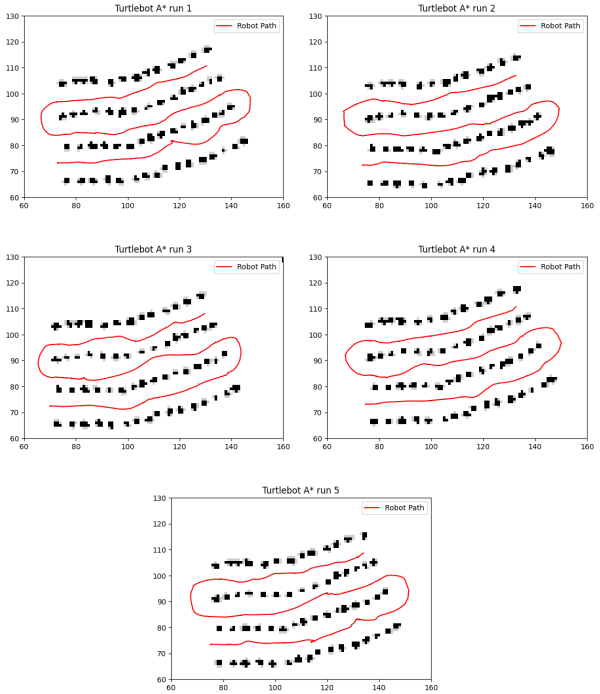


Fig. 12. Shows the five runs conducted with the Turtlebot. Here, the path is plotted in red on the map created by cartographer.

IV. Conclusion

To conclude, this paper introduces the DBRow navigation algorithm for autonomous navigation within crop rows. This algorithm addresses the limitation of the algorithm used in NMBU's last participation in FRE, which relied solely on LIDAR data. Through experiments conducted across different terrains and setups, this algorithm shows potential for being a more robust solution. This algorithm struggled a bit in the simulated terrain, but performed well in the robotics lab. A key weakness of these results is the limited number of experiments that restrict definitive conclusions. This limited number of experiments highlights the need for more expensive testing to achieve statistically significant results. The focus of further work should be on improving the lidar preprocessing and adding some object detection models for stem detection could also enhance the navigation algorithm. Conducting more extensive testing is crucial to validate the preliminary findings and refine the algorithm for practical deployment in actual agricultural environments. Additionally, adapting the navigation task to automate the manual task could enhance the algorithms' use case for agricultural operations.

Acknowledgment

This work is a part of the DLT-Farming project funded by the research council of Norway with the agreement number

344288.

REFERENCES

[1] Francisco Affonso et al. "CROW: A Self-Supervised Crop Row Navigation Algorithm for Agricultural Fields". en. In: *Journal of Intelligent & Robotic Systems* 111.1 (Feb. 2025), p. 28. ISSN: 1573-0409. DOI: 10.1007/s10846-025-02219-2. URL: https://link.springer.com/10.1007/s10846-025-02219-2 (visited on 03/10/2025).

[2] *Algorithm walkthrough for tuning — Cartographer ROS documentation*. URL: https://google-cartographer-ros.readthedocs.io/en/latest/algo_walkthrough.html (visited on 03/17/2025).

[3] Johannes Barthel et al. *Virtual Maize Field*. URL: https://github.com/FieldRobotEvent/virtual_maize_field.

[4] *Cartographer ROS Integration — Cartographer ROS documentation*. URL: https://google-cartographer-ros.readthedocs.io/en/latest/ (visited on 03/17/2025).

[5] Stanchniss Cyrill. *RANSAC – Random Sample Consensus*. Photogrammetry & Robotics Lab. University of Bonn. URL: https://www.ipb.uni-bonn.de/html/teaching/photo12-2021/2021-pho2-06-ransac.pptx.pdf (visited on 04/04/2025).

[6] *DBSCAN — scikit-learn 1.6.1 documentation*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html (visited on 04/13/2025).

[7] Martin Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". en. In: (1996). (Visited on 02/17/2025).

[8] Martin A. Fischler and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". en. In: *Communications of the ACM* 24.6 (June 1981), pp. 381–395. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/358669.358692. URL: https://dl.acm.org/doi/10.1145/358669.358692 (visited on 03/24/2025).

[9] *gazebo_ros_pkgs/gazebo_plugins/worlds/gazebo_ros_skid_steer_drive_demo.world at ros2 · ros-simulation/gazebo_ros_pkgs*. URL: https://github.com/ros-simulation/gazebo_ros_pkgs/blob/ros2/gazebo_plugins/worlds/gazebo_ros_skid_steer_drive_demo.world (visited on 04/09/2025).

[10] Peter Hart, Nils Nilsson, and Bertram Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107. ISSN: 0536-1567. DOI: 10.1109/TSSC.1968.300136. URL: http://ieeexplore.ieee.org/document/4082128/ (visited on 03/31/2025).

[11] Wolfgang Hess et al. "Real-time loop closure in 2D LIDAR SLAM". en. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, May 2016, pp. 1271–1278. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487258. URL: http://ieeexplore.ieee.org/document/7487258/ (visited on 03/17/2025).

[12] *Lua configuration reference documentation — Cartographer ROS documentation*. URL: https://google-cartographer-ros.readthedocs.io/en/latest/configuration.html (visited on 04/09/2025).

[13] *Nav2 — Nav2 1.0.0 documentation*. URL: https://docs.nav2.org/ (visited on 02/27/2025).

[14] *Navigation Concepts — Nav2 1.0.0 documentation*. URL: https://docs.nav2.org/concepts/index.html#concepts (visited on 02/27/2025).

[15] *OS1: High-Res Mid-Range Lidar Sensor for Automation & Security | Ouster*. URL: https://ouster.com/products/hardware/os1-lidar-sensor (visited on 04/09/2025).

[16] Patel. *Implementation notes*. URL: https://theory.stanford.edu/~amitp/GameProgramming/ImplementationNotes.html (visited on 04/01/2025).

[17] Patel. *Introduction to A**. URL: https://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html (visited on 04/01/2025).

[18] *Point Cloud Library (PCL): pcl::CropBox< pcl::PCLPointCloud2 > Class Reference*. URL: https://pointclouds.org/documentation/classpcl_1_1_crop_box_3_01pcl_1_1_p_c_l_point_cloud2_01_4.html (visited on 04/09/2025).

[19] Sebastian Raschka and Vahid Mirjalili. "Locating regions of high density via DBSCAN". eng. In: *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. 3rd ed. Birmingham: Packt Publishing, 2020, pp. 376–383. ISBN: 978-1-78995-575-0 978-1-78995-829-4.

[20] *ros-perception/pointcloud_to_laserscan: Converts a 3D Point Cloud into a 2D laser scan.* URL: https://github.com/ros-perception/pointcloud_to_laserscan (visited on 04/09/2025).

[21] *Setting Up Navigation Plugins — Nav2 1.0.0 documentation*. URL: https://docs.nav2.org/setup_guides/algorithm/select_algorithm.html#select-algorithm (visited on 03/05/2025).

[22] *TurtleBot3*. URL: https://emanual.robotis.com/docs/en/platform/turtlebot3/features/#features (visited on 04/09/2025).

[23] *velodyne_simulator/velodyne_description/urdf/VLP-16.urdf.xacro at master · lmark1/velodyne_simulator*. URL: https://github.com/lmark1/velodyne_simulator/blob/master/velodyne_description/urdf/VLP-16.urdf.xacro (visited on 04/09/2025).

[24] Stavros G. Vougioukas. "Agricultural Robotics". en. In: *Annual Review of Control, Robotics, and Autonomous Systems* 2.1 (May 2019), pp. 365–392. ISSN: 2573-5144, 2573-5144. DOI: 10.1146/annurev-control-053018-023617. URL: https://www.annualreviews.org/doi/10.1146/annurev-control-053018-023617 (visited on 04/25/2025).

Position Papers of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 119–129
DOI: 10.15439/2025F9629
ISSN 2300-5963 ACSIS, Vol. 44

Detection and Classification of Rumex Weeds in Grasslands Using YOLOv11

Jorid Holmen, Weria Khaksar
Norwegian University of Life Sciences, Ås, Norway
Email: jorid.holmen@nmbu.no; weria.khaksar@nmbu.no

***Abstract*—This paper explores the use of YOLOv11 and BoT-SORT for detecting and tracking *Rumex obtusifolius* and *Rumex crispus* in grasslands. Two models were developed: Model A trained on the RumexWeeds dataset, and Model B, trained using transfer learning with additional datasets. While Model A performed well on its training data, it struggled in unseen environments. Model B showed improved generalisation, achieving higher performance across diverse conditions and successfully detecting *Rumex longifolius* in Norwegian grasslands.**

Both models were integrated with BoT-SORT and achieved high tracking metrics, supporting GPS-based mapping. Real-time field testing confirmed feasibility, although detection was affected by shadows, terrain, and camera placement.

The results highlight the importance of diverse training data for robust weed detection. Future work should focus on expanding datasets, tuning hyperparameters, and improving hardware for reliable real-world deployment.

Keywords: Weed detection, AI, YOLO, Precision farming, digital agriculture

I. Introduction

THE NEED for sustainable agricultural practices has become increasingly urgent due to environmental challenges, rising input costs, and labour shortages. Traditional weed control methods, especially herbicide use, pose significant ecological risks such as biodiversity loss and water contamination [1], [2], and reducing chemical input is a central objective in EU-wide sustainability strategies [3].

Two very problematic weeds in European grasslands are *Rumex obtusifolius* and *Rumex crispus*, which degrade pasture quality and can negatively affect livestock health [4]. In Norway and other Northern regions, *Rumex longifolius* is also widespread, but remains understudied and absent from open-access datasets [5].

The introduction of deep learning has significantly advanced the field of automated weed detection in agriculture [6]. Several studies have demonstrated promising results using CNNs and YOLO-based models, with applications ranging from UAV mapping to close-range robotic systems [7], [8], [9], [10], [11]. However, these systems often face challenges in generalising across environments, due to variation in lighting, scale, background conditions, and the high cost of collecting annotated training data [6]. Despite these limitations, UAVs and ground robots are becoming increasingly relevant for precision weed control, with successful demonstrations of real-time detection, herbicide application, and object tracking in field settings [12], [13], [14].

Machine learning has enabled progress in automatic dock detection using UAVs and ground robots. For example, Anken et al. [15] used CNNs to detect 90% of *R. obtusifolius*, while Valente et al. [16] achieved reliable UAV-based detection. Güldenring et al. [17] demonstrated successful detection of *R. obtusifolius* and *R. crispus* using YOLOvX. However, models trained on limited datasets often fail to generalise across varying environments, lighting, and species [15], [17].

This paper, part of the SUSDOCK project [18], addresses the lack of data from Northern environments and aims to improve species-specific weed control. The work focuses on detecting dock weeds using deep learning and evaluates generalisation to *R. longifolius* and unseen field conditions.

Main contributions

- Developed a convolutional neural network to detect *R. obtusifolius* and *R. crispus* using open-access datasets.
- Assessed model generalisation to *R. longifolius* and new environments, with and without additional labelled data.
- Explored the use of object tracking and GPS-based mapping to localise dock occurrences.
- Tested the model on a robotic platform to demonstrate feasibility for real-time weed detection.

II. Methodology

This paper follows a structured workflow to ensure a systematic and reproducible approach from data acquisition to analysis. This section outlines the key stages of the process.

The project began by randomly splitting the dataset into training, validation, and test sets. The YOLOv11 object detection model was trained on the training set and validated on the validation set. After training, the model was evaluated on the test set using standard object detection metrics. This model is referred to as *Model A* throughout the remainder of this paper.

To assess generalisation, Model A was also tested on three previously unseen datasets, two of which were annotated. These two labelled datasets were then merged with the original training data and used to retrain the model using the best[1] weights from the initial training. This model will be referred to as *Model B*. This step aimed to explore whether performance could be improved with additional diverse data.

The next stage of the workflow involved tracking and spatial analysis using BoT-SORT, which was applied to dataset

[1]The best weights defined by the Ultralytics implementation during model training.

sequences. This enabled the counting of dock species and mapping of their GPS locations. The tracking performance was then evaluated using established metrics for multi-object tracking. Lastly, Model B was tested on a real-time robotic platform.

A. Software and Hardware

The primary software used in this paper was Python (version 3.9.21) [19], with all scripts written in standard `.py` files.

Due to the computational demands of object detection, local hardware was deemed insufficient. Instead, remote access to the High-Performance Computing (HPC) cluster Orion, provided by NMBU, was used. Orion consists of 1,680 processor cores, 12 terabytes of RAM, and 1 petabyte of storage, accessible via a $10\,\mathrm{Gbit/s}$ network. The operating system is CentOS Linux 7.9. Jobs on Orion were submitted using SLURM (Simple Linux Utility for Resource Management) by creating batch scripts with the `sbatch` command. These scripts define the resource allocation for each job.

B. The Datasets

The primary dataset used to train Model A was the RumexWeeds dataset. Three additional external datasets were used to evaluate the model's ability to generalise to unseen environments. Two of these, the Open Plant Phenotyping Database and the UAV High-Resolution images, were also used for training Model B, to assess if this improved generalisation to new data. An overview of the datasets and their usage is shown in Table I.

RumexWeeds Dataset: The RumexWeeds dataset [17] contains images of *Rumex obtusifolius* and *Rumex crispus*. It consists of 5,510 RGB images with 15,519 manually annotated bounding boxes — 81% for *R. obtusifolius* and 19% for *R. crispus*. Data was collected at three locations in Denmark, with two of them undergoing two recording sessions, resulting in five distinct sessions under varying environmental conditions. The recording sessions took place during August, September, and October. Notably, this dataset does not contain *Rumex longifolius*, Norway's most common dock species.

Images were captured using a robotic platform equipped with an RGB camera mounted $1\,\mathrm{m}$ above the ground at a 75° angle. Each image has a resolution of 1920×1200 pixels. The robot also carried GNSS, IMU, and odometry sensors, enabling accurate georeferencing and motion tracking.

Open Plant Phenotyping Database: The Open Plant Phenotyping Database [20] was used to evaluate Model A and for training and evaluation of Model B. This public dataset includes 7,590 RGB images representing 47 plant species, all recorded in Denmark during September and October. Of these, 140 images contain *Rumex crispus*, with 6,672 bounding boxes. The plants were grown in containers designed to mimic natural growth conditions. The *Rumex* samples were photographed 1–3 times daily from seedling emergence to full leaf stage. The camera was positioned directly above the boxes at a height of $1.7\,\mathrm{m}$.

UAV High-Resolution Images: The Unmanned Aerial Vehicle (UAV) High-Resolution Images dataset [16] consists of three images captured in Germany in April using a drone at altitudes of $10\,\mathrm{m}$, $15\,\mathrm{m}$, and $30\,\mathrm{m}$. The image captured at $30\,\mathrm{m}$ was excluded due to insufficient resolution for reliably detecting weeds. The images taken at $10\,\mathrm{m}$ and $15\,\mathrm{m}$ were divided into tiles with a resolution of 640×640 pixels. This process resulted in 316 images, with 610 annotated bounding boxes containing *R. Obtusifolius*. As the Open Plant Phenotyping Dataset, this dataset was used to evaluate Model A and to train and evaluate Model B.

Rumex in Norwegian Grasslands: The last dataset consists of 217 unannotated images captured in Norway's various environments, lighting conditions, and camera angles. This is not an open-access dataset, but is provided for this paper through the SUSDOCK project [18]. The images contain mostly *Rumex longifolius*, the most common dock species in Norway. Although the model was trained on other *Rumex* species, *R. longifolius* shares similar characteristics in natural grassland settings. This dataset was used to visually assess whether the model could detect docks in Norwegian environments. Four images will be focused on that both contain *R. longifolius*.

1) Data Preprocessing: YOLOv11 requires input data in the YOLO format, thus the original formats of the RumexWeeds, Open Plant Phenotyping, and UAV High-Resolution datasets were converted accordingly. A `.yaml` configuration file is also required, defining the paths to the images, label files, and a dictionary of class names.

YOLOv11 expects one label file corresponding to each image in the dataset, containing information about the bounding boxes. One bounding box is represented with the class ID, x- and y-coordinates for the centre of the box, and the width and height. There can be several bounding boxes in each annotation file.

The RumexWeeds dataset was randomly split into training, validation, and testing subsets, with 70%, 10%, and 20% allocated to each, respectively. The class distribution was stratified to ensure it was balanced across all splits. For training Model B with new datasets, the training and validation were combined with 80% of the Open Plant Phenotyping data and 80% of the UAV High-Resolution Images into the training set, and the rest of the RumexWeeds dataset was combined for the test set. This resulted in 80% training data and 20% test data. The reason for this change is the limited data on the Phenotype and UAV datasets.

For Multi-Object Tracking, randomly selected images are not suitable; instead, continuous video sequences are required. Therefore, all the sequences from one recording session of the RumexWeeds dataset were turned into one video for each sequence. The videos were annotated with tracking IDs necessary for the MOT metrics, in MOT16 format [21]. Due to the task's time-consuming nature and limited available time, only one recording session was annotated. A total of 580 annotated images were chronologically sorted, with bounding boxes visually matched to their corresponding objects and

TABLE I: Overview of datasets used, with the number of images, bounding boxes, and their intended usage.

Dataset	Images	Bounding Boxes	Annotated	Usage
RumexWeeds [17]	5,510	15,519	Yes	Train Model A and Model B
Open Plant Phenotyping Database [20]	140	6,672	Yes	Validate Model A, train Model B
UAV High-Resolution Images [16]	323	610	Yes	Validate Model A, train Model B
Rumex in Norwegian Grasslands	217	0	No	Validate Model A and Model B

TABLE II: Modified hyperparameters for YOLOv11 training.

Hyperparameter	Default	Modified Value	Reason
`epochs`	100	150	Allows the model more time to converge and potentially improve performance
`batch`	16	8	Smaller batch size can enhance generalisation and reduce overfitting, especially with limited data
`dfl`	1.5	2	Increases the impact of Focal Loss to better address class imbalance

manually assigned tracking IDs.

C. YOLOv11

For object detection, the YOLOv11 was selected. This YOLO version comes in sizes *nano, small, medium, large* and *x large*. Small was chosen for this paper due to its balance between speed and accuracy. The model was utilised through the Ultralytics implementation, which offers a high-level Python API for training, validation, and inference [22].

By default, the Ultralytics implementation uses pre-trained weights from the COCO (Common Objects in Context) dataset, which contains 80 object classes. These weights help improve training efficiency and accuracy when working with custom data. Another default setting is data augmentation. In addition to regular data augmentation, YOLO implements mosaic augmentation.

The default hyperparameters provided by Ultralytics include preprocessing steps such as image resizing and pixel value scaling. Given that hyperparameter tuning is time-consuming and the YOLOv11 creators have already invested significant effort in optimising the defaults, this paper primarily relied on those standard settings. However, some key parameters were adjusted to better align with the dataset's characteristics, as shown in Table II.

Ultralytics also simplifies evaluation by providing built-in support for standard object detection metrics. For this project, the evaluation metrics were inference speed, precision, recall, mAP50, and mAP50-95.

D. BoT-SORT

BoT-SORT was used for object tracking, as it is the default multi-object tracker in the Ultralytics pipeline. BoT-SORT,

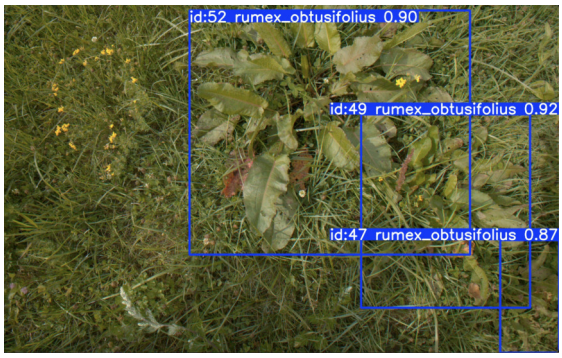


Fig. 1: Extraction from the tracking video, showing a frame with three detected *Rumex obtusifolius* plants, annotated with tracking IDs 47, 49, and 52. The boxes also display class names and detection confidence scores.

with the trained YOLOv11 model as the detection algorithm, was applied to the videos, one for each sequence in the recording session. The output included bounding boxes with unique tracking IDs across frames, forming annotations in MOT16 format and a video visualising the tracked detections. A frame from the tracking video is shown in Figure 1, highlighting how detected objects are assigned consistent tracking IDs.

Tracking IDs were used to associate detected objects with their corresponding GPS coordinates from the RumexWeeds dataset. These locations were visualised using *matplotlib* for static plots and *folium* for interactive maps. The ground truth distribution in the interactive map is shown in Figure 2. When pressing the points in the interactive map, information about what *Rumex* type it is will appear: red points for *R. crispus* and green points for *R. obtusifolius*. In addition, the total number of tracked instances was used to estimate the number of *R. obtusifolius* and *R. crispus* plants.

BoT-SORT was applied to shorter annotated video sequences to evaluate the tracking performance. The output detections in the MOT16 format were compared to the ground truth using the *py-motmetrics* library [23]. A challenge in evaluating tracking is that the tracker may assign different object IDs than those in the ground truth. The evaluation addresses this challenge by mapping the tracking IDs based on Intersection over Union (IoU), requiring a threshold of 0.5 or higher. This ID alignment ensures that metrics such as IDF1 and MOTA accurately reflect tracking performance,

Fig. 2: Ground truth GPS coordinates of dock plants in the RumexWeeds dataset. Each green point represents *R. obtusifolius* and each red point represents *R. crispus*.

Fig. 3: A picture of the robot whilst driving in the field.

rather than being skewed by identity mismatches. The tracking performance was assessed using the three key metrics MOTA, MOTP and IDF1.

E. Real-Time Robotic Platform – A Proof of Concept

To test the feasibility of applying the model in a robotic setting, Model B was selected for deployment. The test was conducted in a field located in Askim, Norway, which contains a high density of *R. longifolius* plants.

The robot was equipped with a Logitech C920s Pro HD webcam, positioned approximately $30\,\mathrm{cm}$ above the ground at an angle of $30°$. The camera has a resolution of 1920×1080 pixels and was connected to a MacBook for simplicity and mobility. A picture of the robot while driving in the field is shown in Figure 3.

The output from the test consisted of a video showing the predicted bounding boxes, along with their confidence scores and assigned tracking IDs. An example of a frame from the video, without any bounding boxes, is shown in Figure 4. Additionally, a text file was generated containing frame-by-frame information, including tracking IDs, bounding box coordinates, and confidence values.

Limitations: Due to limited time and resources, several constraints affected the proof-of-concept test. First, the webcam used was not optimal for field robotics applications, but was selected for its immediate compatibility with the MacBook. Second, the real-time detection code was not fully optimised

Fig. 4: An example frame from the robot during recording.

for the camera settings, leading to performance limitations. Furthermore, the vision system was not physically integrated into the robot's control system, as full hardware integration would have required more development time than the project timeframe allowed. Finally, no GPS module was connected to either the MacBook or the robot, meaning that no spatial localisation data was recorded during the test.

The terrain in the field was bumpy, resulting in the robot's inconsistent driving speed. Due to the camera's mounting position, large portions of the surrounding landscape, including the sky and nearby objects, were captured in many frames. Furthermore, shadows from the robot, the operators, and the low sun position affected the image quality.

However, this is only a proof-of-concept, which means the conditions does not need to be ideal. In spite of these limitations, the tests will still be able to tell the feasibility of the model in a robotic setting.

III. Results and Discussion

A. Object Detection Performance: Model A

Table III shows the evaluation metrics for Model A, trained solely on the RumexWeeds dataset. The model performed well on the training domain, with a high precision of 0.922, a recall of 0.887, and mAP values of 0.949 for mAP50 and 0.703 for mAP50-95. The lower mAP50-95 reflects the model's reduced localisation precision across varying IoU thresholds. On the external Phenotype and UAV datasets, performance declined substantially. While precision remained relatively moderate on the Phenotype data with a value of 0.714, recall dropped significantly to 0.001. The UAV dataset showed poor performance across all metrics. This demonstrates a considerable drop in performance when external data is evaluated. The inference speed is consistent for all three datasets, ranging from $2.761\,\mathrm{ms}$ for the RumexWeeds dataset, $3.450\,\mathrm{ms}$ for the Phenotype dataset and $5.025\,\mathrm{ms}$ for the UAV dataset. The inference was measured on an HPC, which is significantly faster than typical robotic platforms. The Phenotype and UAV datasets showed slower speeds, likely due to more complex images or larger input sizes. These differences should be considered when deploying the model on resource-constrained platforms.

TABLE III: Detection performance of Model A on the validation sets. Results are reported for three datasets: RumexWeeds, Phenotype, and UAV. Metrics include inference speed (ms per image), precision, recall, and mAP50 and mAP50-95.

Dataset	Inference Speed (ms)	Precision	Recall	mAP50	mAP50-95
RumexWeeds	2.761	0.922	0.887	0.949	0.703
Phenotype	3.450	0.714	0.001	0.359	0.198
UAV	5.025	0.015	0.051	0.015	0.009

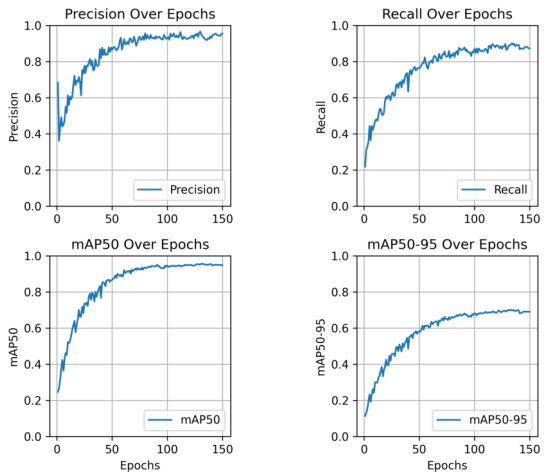
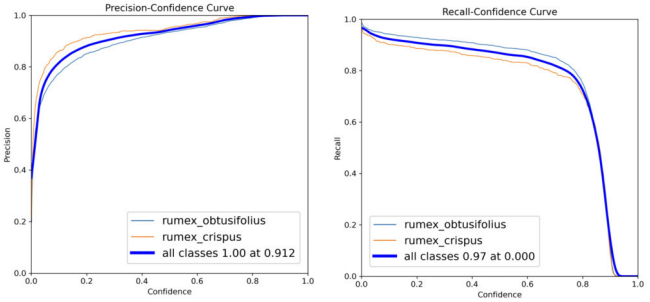


Fig. 5: Training curves for Model A. The plots show the progression of precision, recall, mAP50, and mAP50-95 over 150 epochs on the RumexWeeds dataset. The model converged steadily across all metrics.

Figure 5 presents the training curves for Model A over 150 epochs. The plots illustrate the progression of precision, recall, mAP50, and mAP50-95 throughout training on the RumexWeeds dataset. All four metrics showed a rapid increase during the initial epochs, particularly up to around epoch 50, followed by a more gradual improvement and eventual stabilisation near epoch 100. The curves began at moderate values, with precision, recall, and mAP50 starting between 0.2 and 0.4 suggesting that COCO pretraining provided a strong foundation, while mAP50-95 starts lower, around 0.1. Some fluctuations are observed, likely due to the small batch size, but overall, the trends indicate convergence.

Figure 6a and Figure 6b present the precision– and recall–confidence curves for Model A. The model demonstrated consistently high precision across a broad range of confidence thresholds for both classes, though slightly higher for *R. crispus*. In contrast, recall values were initially high but declined more sharply as confidence increased. This matches the observation of a lower mAp50-95 score, meaning the model prioritised accurate predictions over broader detection coverage, leading to missed detections or less precise bounding boxes at stricter thresholds. Fine-tuning the confidence threshold could improve the balance between recall and precision. The curves followed similar trends for both *R. obtusifolius* and *R. crispus*, with slightly lower recall observed for *R. crispus*, unlike precision.



(a) Precision-confidence curves for Model A.

(b) Recall-confidence curves for Model A.

Fig. 6: Precision and recall confidence curves for Model A, showing performance for *R. obtusifolius*, *R. crispus*, and combined class scores.

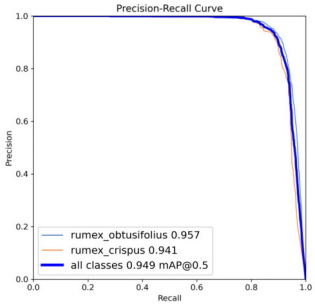


Fig. 7: Precision–recall curve for Model A. The model achieved a high average precision for both *R. obtusifolius* (0.957) and *R. crispus* (0.941), with a combined mAP50 of 0.949.

The corresponding precision–recall curve is shown in Figure 7. The model achieved a combined mAP50 of 0.949 across both target classes. The curve demonstrates that precision remains high as recall increases, particularly for *R. obtusifolius*, which achieved a slightly higher average precision than *R. crispus*, at 0.957 and 0.941 respectively. The curves for both classes followed a similar shape, with minimal divergence across most recall values. The different performances on *R. obtusifolius* and *R. crispus*, is likely due to dataset imbalance, where 81% of annotations were *R. obtusifolius* vs. 19% *R. crispus*. Although focal loss was used to mitigate this, it did not fully offset the imbalance. To improve this, more *R. crispus* images should be annotated, and targeted data augmentation may also help.

While performance on RumexWeeds was strong, Model A's performance dropped significantly on the Phenotype and

UAV datasets. This is likely due to visual domain differences: RumexWeeds images were collected under consistent, ground-based conditions, whereas the external datasets varied in angle, scale, lighting, background, and plant stage. These unfamiliar conditions reduced generalisation. The limited number of *R. crispus* examples further hindered generalisation to new conditions. These results reflect a common deep learning issue: strong performance on training data does not guarantee robustness in new settings. Fine-tuning the model with images better matching target deployment conditions could improve generalisation.

B. Object Detection Performance: Model B

Table IV presents the detection performance of Model B, which was trained using transfer learning on a combination of three datasets. On the combined validation set, the model achieved an inference speed of 2.335 ms, a precision of 0.932, a recall of 0.873, an mAP50 of 0.930, and an mAP50-95 of 0.688. Performance on the RumexWeeds dataset remained strong, with precision, recall, and mAP values comparable to those of Model A. Notably, Model B showed substantial improvements on the external datasets. For example, the Phenotype dataset reached a precision of 0.934, a marked increase compared to Model A. However, when compared to the RumexWeeds dataset, the two new datasets exhibited slightly lower values for recall, mAP50, and mAP50-95, and a higher inference speed.

Model B achieved slightly better mAP50 and mAP50–95 on RumexWeeds than Model A, suggesting that base performance was maintained or improved, partly due to extended training. Still, mAP50–95 scores lagged behind mAP50 across all datasets, indicating that precise localisation remains a challenge.

The inference speed of the combined dataset were similar to the RumexWeeds, likely due to the large proportion of RumexWeeds images. The Phenotype and UAV datasets ran slower at $4.444\,\mathrm{ms}$ and $3.500\,\mathrm{ms}$, respectively. As with Model A, slower speeds may be due to increased image complexity or resolution. Interestingly, UAV was faster than Phenotype in Model B, reversing the pattern from Model A, possibly due to retraining effects or dataset changes.

Figure 8 shows the training curves for Model B over 150 epochs. As with Model A, the plots display the progression of precision, recall, mAP50, and mAP50-95. The values increased rapidly during the early stages of training and stabilised after approximately 50 epochs. The curves started at relatively high values, with precision, recall, and mAP50 beginning between 0.75 and 0.9, while mAP50-95 starts lower, around 0.6. This is typical in transfer learning, where early CNN layers retain useful low-level features. The consistent structure of dock weeds across datasets helped the model learn new features efficiently.

Figures 9a and 9b show the confidence-based precision and recall curves for Model B. In the precision curve, precision remained consistently high across the entire confidence range for both *R. obtusifolius* and *R. crispus*. The recall curve

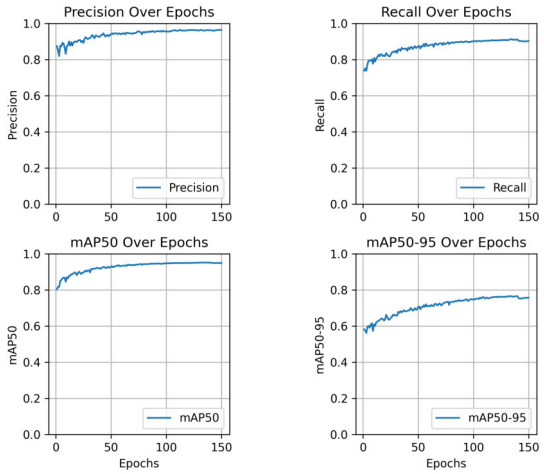
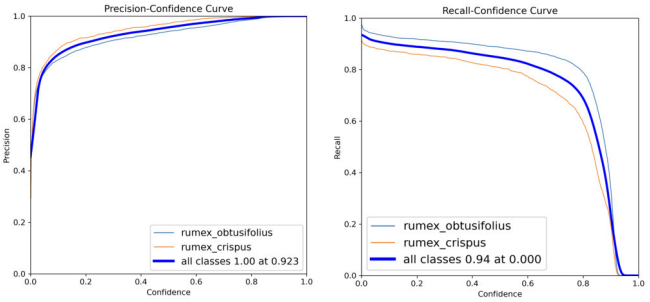


Fig. 8: Training curves for Model B, which was trained using transfer learning on a combined dataset (RumexWeeds, Phenotype, and UAV). The plots show the evolution of precision, recall, mAP50, and mAP50-95 over 150 epochs.



(a) Precision-confidence curves for Model B.

(b) Recall-confidence curves for Model B.

Fig. 9: Precision and recall confidence curves for Model B, showing performance for *R. obtusifolius*, *R. crispus*, and combined class scores.

showed that recall is highest at lower confidence thresholds and decreases steadily as the confidence increases. Recall for *R. crispus* drops more rapidly than for *R. obtusifolius*.

The precision–recall curve in Figure 10 shows that Model B achieves an mAP50 of 0.930. Average precision for *R. obtusifolius* is 0.953, while *R. crispus* reaches 0.907. The class-wise curves follow a similar shape, with high precision across most recall levels. These patterns closely mirror those observed for Model A.

Model B was trained using transfer learning from Model A, with additional labelled data from the Phenotype and UAV datasets. It showed strong detection performance on the combined dataset and improved results on the external datasets compared to Model A. As shown in Table IV, precision, recall, and mAP scores increased significantly on both external datasets, reflecting greater robustness to varied image conditions. This improvement stems from the added data diversity

TABLE IV: Detection performance of Model B on the validation sets. Model B was trained using transfer learning with data from RumexWeeds, Phenotype, and UAV datasets. Metrics include inference speed (ms per image), precision, recall, and mAP50 and mAP50-95.

Dataset	Inference Speed (ms)	Precision	Recall	mAP50	mAP50-95
Combined Data	2.335	0.932	0.873	0.930	0.688
RumexWeeds	2.259	0.946	0.888	0.959	0.733
Phenotype	4.444	0.934	0.765	0.836	0.607
UAV	3.500	0.879	0.775	0.836	0.560

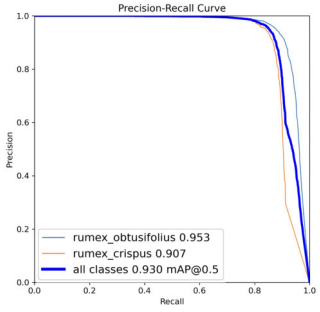


Fig. 10: Precision–recall curve for Model B, trained using transfer learning. The model achieved strong performance on *R. obtusifolius* (0.953) and slightly lower average precision on *R. crispus* (0.907), with a combined mAP50 of 0.930.

and the benefits of transfer learning, where Model A's weights provided a solid starting point.

C. Generalisation to Norwegian Grasslands

Detection results were visualised on images collected from Norwegian grasslands containing mostly *Rumex longifolius* to evaluate how well the models generalise to unseen environments and species. Four images were selected, each shown with predicted bounding boxes from both Model A and Model B.

In the examples, both models identified dock plants in varied settings, including dense vegetation and challenging lighting conditions. Some variation in the number and classification of detections can be observed between the two models. Predictions included both *R. obtusifolius* and *R. crispus* labels.

Figures 11 and 12 show detection results in scenes with visual complexity. These images contain background distractions such as shoes, camera equipment, and uneven lighting, making the detection task more difficult. The *Rumex longifolius* plants are not immediately noticeable even to the human eye. In Figure 11, Model A produced a single prediction in a bright area near a camera leg. In contrast, Model B identified a *R. crispus* leaf, though the prediction has low confidence and is accompanied by a duplicated bounding box. In figure 12 Model A detected a central plant as *R. crispus* with a confidence of 0.69. Model B also identified this plant, but with slightly lower confidence. Additionally, Model B predicted two extra detections with low confidence in areas where no dock plants are visible. It also detected a plant in the upper left with 0.54 confidence, which Model A missed entirely.

TABLE V: BoT-SORT tracking metrics for Model A and Model B. Metrics include Multiple Object Tracking Accuracy (MOTA), Precision (MOTP), and IDF1.

Model	MOTA	MOTP	IDF1
Model A	0.894	0.893	0.883
Model B	0.898	0.89	0.883

The qualitative results from the Norwegian grasslands dataset provide insight into how well the models generalise to completely unseen environments and species. Neither Model A nor Model B was trained on images of *Rumex longifolius*, yet both produced detections on the unlabelled Norwegian images. Model B showed a better overall result. However, both models also displayed false positives, including misclassifications of sunlit areas, plant residues, and patches of grass. This suggests that although the models are capable of transferring some learned features to unfamiliar conditions, their ability to distinguish *R. longifolius* from the background remains limited. The improved responsiveness of Model B indicates that additional training data from varied domains contributes to broader generalisation, but the presence of misclassifications also highlights the need for further adaptation or fine-tuning when deploying such models in new and different environments.

D. Tracking Evaluation Using BoT-SORT

Tracking performance for Model A and Model B was evaluated using the BoT-SORT tracking algorithm. Table V presents the resulting scores across three standard multi-object tracking metrics: MOTA, MOTP and IDF1. The results showed that both models achieved similar performance, with only minor differences observed in MOTA and MOTP, with values between 0.89 and 0.90. The IDF1 score remained identical at 0.883 for both.

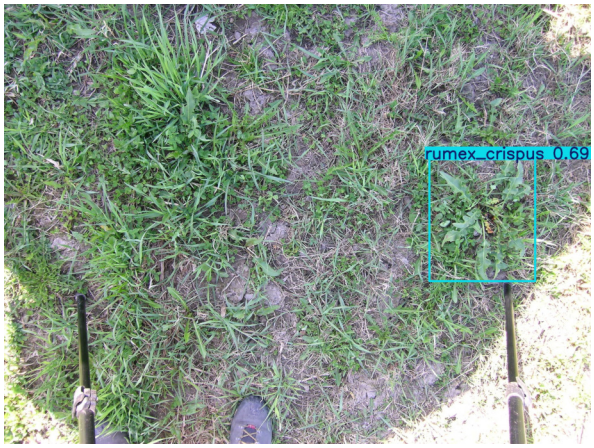
To further assess how well the models perform in tracking dock plants over time, the predicted number of detections was compared to the manually annotated ground truth. As shown in Table VI, both models correctly detect five instances of *R. crispus*, while both overestimate the number of *R. obtusifolius* plants by ten. In addition, both models produced a distribution that closely matched the expected locations. Most predictions were concentrated along a path.

The tracking results using BoT-SORT show that both Model A and Model B maintained high tracking performance across video sequences. This suggests that as long as the object detector provides consistent and confident detections, the tracking algorithm is able to assign and maintain identities effectively.

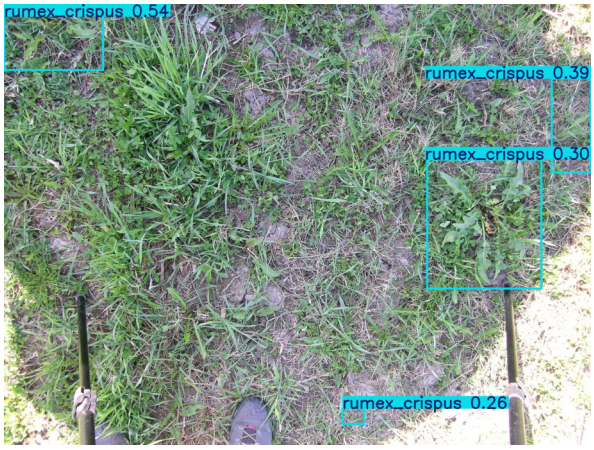
(a) Model A

(b) Model B

Fig. 11: Detection results in a sparse vegetation scene with visual distractions such as camera equipment and bright lighting. Model A (top) produced one detection near the camera leg, while Model B (bottom) detected a dock leaf with low confidence and overlapping boxes.



(a) Model A



(b) Model B

Fig. 12: Detection results in a visually cluttered scene. Model A (top) identified one dock plant with high confidence. Model B (bottom) detected the same plant and additional low-confidence detections, some of which appear to be false positives.

TABLE VI: Number of dock plants detected by Model A and Model B compared to the manually annotated ground truth.

	Rumex Obtusifolius	Rumex Crispus
Ground truth	41	5
Model A	51	5
Model B	51	5

When comparing the number of tracked detections with the ground truth, both models correctly identified all instances of *R. crispus*, but overestimated the number of *R. obtusifolius*. This overcount likely results from multiple detections on the same plant across frames or slightly offset bounding boxes being treated as separate objects. This observation coincides with the low recall and mAP50-95 values of both Model A and Model B on high confidence thresholds. Since bounding box offset is a contributing factor, this points to potential improvements in the tracking pipeline. Despite these minor inaccuracies, the spatial distribution of tracked detections closely matched the expected GPS coordinates, indicating that the pipeline is suitable for mapping dock presence in the field.

This demonstrates the potential of the combined detection and tracking pipeline for supporting automated weed monitoring and management in real-world farming environments.

E. Real-Time Robotic Platform Performance

Model B was tested in a real-world field environment, resulting in six different video sequences with corresponding text files containing detection information. Table VII summarises the results from each sequence, including the number of frames, the number of unique tracking IDs, the number of unique tracking IDs with average confidence above 0.50 and the number of actual *R. longifolius* plants present in the sequences. The tracking ID number does not correspond well with the ground truth number, due to several false positives.

TABLE VII: Statistics from the six sequences showing information about the number of frames, tracking IDs, and ground truth counts.

Sequence	Frames	Tracking IDs	Average confidence > 0.50	Ground Truth Docks
1	433	28	13	2
2	715	60	34	2
3	607	41	19	3
4	762	17	11	2
5	637	81	53	3
6	708	31	16	8

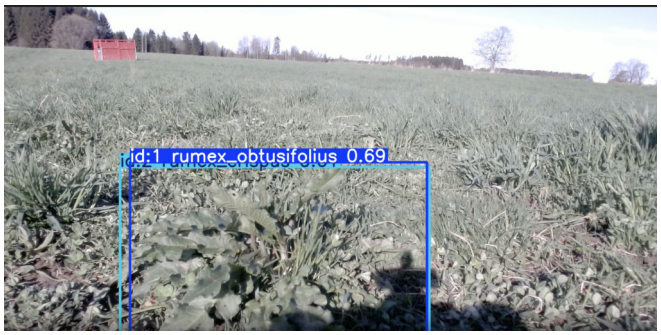


Fig. 13: Frame from sequence 1 showing multiple bounding boxes around a dock and background elements.

In addition to this information, the text files contained a line saying “Coordinates: Location not available" for each detection, meaning it tried to collect the GPS information, but was not able to since there was no GPS module connected.

As illustrated in Figure 13, sequence 1 shows a *R. longifolius* plant with two relatively confident bounding boxes. As the robot moved, an additional bounding box appeared around the same dock. Significant background content, such as the sky, trees, and red farming equipment, is also visible, likely leading to false positives where background objects were misclassified as *Rumex* in later frames. Figure 14 provides an example where a non-*Rumex* object was confidently classified as a dock.

Figure 15 shows a cropped frame from sequence 3, where a *R. longifolius* appears very close to the camera. Only part of the dock is detected, with relatively low confidence. A similar situation is visible in Figure 16 from sequence 5, where the same dock is divided into multiple bounding boxes across different leaves, each with varying confidence levels.

Figure 17 shows two frames from sequence 4. In this situation, the sun is shining directly into the camera, causing strong image diffusion. As a result, the two visible *R. longifolius* plants were not detected at all. A similar issue occurred in sequence 6, where sunlight again affected the camera’s visibility. According to Table VII, sequence 6 generated 31 unique tracking IDs, but only two out of eight actual docks were detected. This pattern, where most docks were missed, is unique to sequences 4 and 6. In contrast, in the other sequences, all docks were detected in some form, although

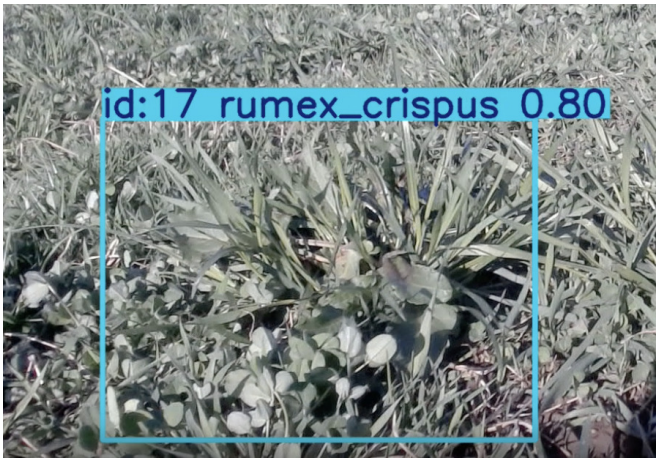


Fig. 14: Frame from sequence 1 showing a non-*Rumex* object incorrectly classified as *Rumex*.

Fig. 15: Frame from sequence 3 showing a close-up of *R. longifolius* with partial and low-confidence detection.

with too many, too few, or poorly placed bounding boxes.

The robotic platform test served as a proof-of-concept to assess whether it would be possible to detect *R. longifolius* in the field using a robot. The overall results were not ideal. However, in cases where the conditions were favourable, such as in Figure 13, the platform successfully detected the dock plants, though with several bounding boxes. This suggests that under improved conditions, the system has the potential to perform significantly better.

Several factors could have contributed to the false positives observed during the field test. The camera on the robot was positioned relatively low, and a higher mounting position would likely have captured a more complete view of the scene. Additionally, tilting the camera further towards the ground could reduce background noise, such as trees and the sky.

Fig. 16: Frame from sequence 5 showing a *R. longifolius* with multiple overlapping bounding boxes.

Fig. 17: Both images show a *R. longifolius* that has not been detected. The frames have become diffused due to the sun.

Güldenring et al. [17] used a camera height of approximately $1\,\mathrm{m}$ and an angle of $75°$, which appeared to be more effective. Another challenge was that the test was conducted when the sun was relatively low in the sky, causing strong shadows and uneven lighting. Capturing images closer to midday would likely improve lighting conditions. Finally, the use of a non-specialised camera and detection code that was not fully optimised for the hardware may also have contributed to the reduced detection performance.

In addition to false positives in detection, a high number of unique tracking IDs were observed. The ground surface was uneven and textured, causing the robot to move unpredictably across the grassland. The BoT-SORT algorithm predicts object movement to maintain consistent tracking IDs. However, the irregular movement of the camera likely made it difficult for the tracking algorithm to generate stable and meaningful tracking results. In future applications, using a larger or wider robot platform could help reduce camera instability and improve tracking accuracy.

Lastly, the absence of GPS coordinates meant that mapping dock occurrences in the field was not possible. However, the proof-of-concept demonstrated that it would be feasible to collect GPS data alongside detection results if such data were available. This indicates a promising potential for mapping dock occurrences in future applications.

IV. Conclusion and Further Work

This paper explored the use of YOLOv11 and BoT-SORT for detecting and tracking dock weeds in grasslands, focusing on improving generalisation across different environments and species. Two models were trained and tested: Model A, trained only on the RumexWeeds dataset, and Model B, which used transfer learning with additional datasets to improve robustness.

The results showed that Model A performed very well on the RumexWeeds dataset but struggled to generalise to new environments, such as the Open Plant Phenotyping Database and UAV High-Resolution Images. Model B, trained with additional data, improved performance on these external datasets while maintaining high accuracy on the original RumexWeeds data. Both models detected *R. longifolius* in images from Norwegian grasslands, with Model B performing slightly better. These findings demonstrate that adding more diverse training data is an effective way to improve the generalisation of deep learning models for weed detection.

The tracking results showed that both Model A and Model B achieved high scores across all evaluated tracking metrics. This indicates that the combined detection and tracking system worked reliably for counting and mapping dock weeds. However, challenges such as slightly inaccurate bounding boxes and overcounting suggest that further improvements to detection precision and tracking stability are needed.

Testing the system in real-time using a robotic platform showed that it is possible to detect *R. longifolius* plants under field conditions, although the results were not ideal. Factors such as strong shadows, a low camera angle, background distractions, and an uneven ground surface likely affected detection accuracy. These results highlight that hardware setup and environmental conditions are critical factors when applying the model outside controlled environments. Despite these challenges, the proof-of-concept showed promising potential for real-time robotic weed detection in future applications.

Further Work: The promising results of this paper show that the system has strong potential and should be developed further. Based on the findings discussed above, several specific areas for improvement have been identified that could further strengthen the system.

First, the project would benefit greatly from expanding the training datasets. If the focus remains on *R. obtusifolius* and *R. crispus*, it would be essential to collect additional images of *R. crispus* to better balance the class distribution. In addition, given the emphasis on Norwegian grasslands, creating a large, open-access dataset specifically for *R. longifolius* would be highly valuable. Another idea worth exploring is training *R. crispus* and *R. obtusifolius* as a single class, as done by Güldenring et al. [17]. Since both species are targeted for removal in the same way, merging them into one detection class could simplify the classification task and possibly improve the model's ability to detect *R. longifolius* as well.

Model B was trained using the same hyperparameters as Model A for simplicity. Future work could investigate tuning the hyperparameters specifically for Model B, as this may further improve performance, particularly when training on more varied data.

For the robotic platform, it would be beneficial to implement the improvements suggested in the discussion, such as optimising camera position and movement stability. In addition, designing a camera flash solution that provides consistent lighting, similar to the one used by Kilter [13], could help reduce issues caused by varying weather and lighting conditions during field operations.

Acknowledgment

This work is a part of the SUSDOCK project funded by the research council of Norway.

REFERENCES

[1] A. Van Bruggen, M. He, K. Shin, V. Mai, K. Jeong, M. Finckh, and J. Morris, "Environmental and health effects of the herbicide glyphosate," *Science of The Total Environment*, vol. 616-617, pp. 255–268, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0048969717330279

[2] A. Klik and J. Rosner, "Long-term experience with conservation tillage practices in austria: Impacts on soil erosion processes," *Soil and Tillage Research*, vol. 203, p. 104669, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167198720304517

[3] J. Wesseler, "The eu's farm-to-fork strategy: An assessment from the perspective of agricultural economics," *Applied Economic Perspectives and Policy*, vol. 44, no. 4, pp. 1826–1843, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aepp.13239

[4] S. Hejduk and P. Dolezal, "Nutritive value of broad-leaved dock (rumex obtusifolius l.) and its effect on the quality of grass silages," *Czech Journal of Animal Science*, vol. 49, no. 4, pp. 144–150, 2004. [Online]. Available: https://cjas.agriculturejournals.cz/artkey/cjs-200404-0003.php

[5] P. E. Hatcher, L. O. Brandsaeter, G. Davies, A. Lüscher, H. L. Hinz, R. Eschen, and U. Schaffner, "Biological control of rumex species in europe: opportunities and constraints." *CABI*, p. 470–475, 2008. [Online]. Available: https://doi.org/10.1079/9781845935061.0470

[6] J. Zhang, F. Yu, Q. Zhang, M. Wang, J. Yu, and Y. Tan, "Advancements of uav and deep learning technologies for weed management in farmland," *Agronomy*, vol. 14, no. 3, 2024. [Online]. Available: https://www.mdpi.com/2073-4395/14/3/494

[7] J. Zhao, T. W. Berge, and J. Geipel, "Transformer in uav image-based weed mapping," *Remote Sensing*, vol. 15, no. 21, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/21/5165

[8] E. C. Tetila, B. L. Moro, G. Astolfi, A. B. da Costa, W. P. Amorim, N. A. de Souza Belete, H. Pistori, and J. G. A. Barbedo, "Real-time detection of weeds by species in soybean using uav images," *Crop Protection*, vol. 184, p. 106846, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0261219424002746

[9] P. Wang, Y. Tang, F. Luo, L. Wang, C. Li, Q. Niu, and H. Li, "Weed25: A deep learning dataset for weed identification," *Frontiers in Plant Science*, vol. Volume 13 - 2022, 2022. [Online]. Available: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.1053329

[10] Y. Mu, R. Feng, R. Ni, J. Li, T. Luo, T. Liu, X. Li, H. Gong, Y. Guo, Y. Sun, Y. Bao, S. Li, Y. Wang, and T. Hu, "A faster r-cnn-based model for the identification of weed seedling," *Agronomy*, vol. 12, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2073-4395/12/11/2867

[11] T. W. Berge, T. Torp, F. Urdal, and M. Vallestad, "Sensor technology for precision weeding in cereals: Evaluation of a novel convolutional neural network to estimate weed cover, crop cover and soil cover in near-ground red-green-blue images," Norwegian Institute of Bioeconomy Research (NIBIO), Ås, Norway, NIBIO Report 8(134), 2022. [Online]. Available: https://nibio.brage.unit.no/nibio-xmlui/handle/11250/3031834

[12] T. Jin, K. Liang, M. Lu, Y. Zhao, and Y. Xu, "Weedssort: A weed tracking-by-detection framework for laser weeding applications within precision agriculture," *Smart Agricultural Technology*, vol. 11, p. 100883, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772375525001169

[13] T. Utstumo, F. Urdal, A. Brevik, J. Dørum, J. Netland, Overskeid, T. Berge, and J. Gravdahl, "Robotic in-row weed control in vegetables," *Computers and Electronics in Agriculture*, vol. 154, pp. 36–45, 11 2018.

[14] Kilter Systems, "Kilter systems – ai-powered agricultural robotics," 2024, accessed: 14 April 2025. [Online]. Available: https://www.kiltersystems.com

[15] T. Anken and A. Latsch, "Characteristics of a spot sprayer for the treatment of rumex obtusifolius in meadows," *agricultural engineering.eu*, vol. 78, no. 3, 2023. [Online]. Available: https://www.agricultural-engineering.eu/landtechnik/article/view/3295

[16] J. Valente, S. Hiremath, M. Ariza-Sentís, M. Doldersum, and L. Kooistra, "Mapping of rumex obtusifolius in nature conservation areas using very high resolution uav imagery and deep learning," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, p. 102864, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843222000668

[17] R. Güldenring, F. K. van Evert, and L. Nalpantidis, "Rumexweeds: A grassland dataset for agricultural robotics," *Journal of Field Robotics*, vol. 40, no. 6, pp. 1639–1656, 2023.

[18] "Susdock: Sustainable control and mapping of dock plants," https://www.ri.se/en/susdock, accessed: 2025-04-22.

[19] Python Software Foundation, *Python Language Reference, version 3.9.21*, 2023. [Online]. Available: https://docs.python.org/3.9/

[20] S. L. Madsen, S. K. Mathiassen, M. Dyrmann, M. S. Laursen, L.-C. Paz, and R. N. Jørgensen, "Open Plant Phenotype Database of Common Weeds in Denmark," *Remote Sensing*, vol. 12, no. 8, p. 1246, Apr. 2020. [Online]. Available: https://www.mdpi.com/691100

[21] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," 2016. [Online]. Available: https://arxiv.org/abs/1603.00831

[22] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," 2024. [Online]. Available: https://arxiv.org/abs/2410.17725

[23] C. Heindl, Toka, and J. Valmadre, "py-motmetrics: Python implementation of metrics for multiple object tracking," https://github.com/cheind/py-motmetrics, 2024, accessed: 2025-03-21.

Author Index