Annals of Computer Science and Information Systems
Volume 33

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering

November 11–12, 2022
Hung Yen University of Technology and Education, Vietnam

Vijender Kumar Solanki, Bui Trung Thanh (eds.)

PTI

Annals of Computer Science and Information Systems, Volume 33

ISSN: 2300-5963

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering

Vijender Kumar Solanki and Bui Trung Thanh (eds.)

PTI
POLSKIE TOWARZYSTWO INFORMATYCZNE
POLISH INFORMATION PROCESSING SOCIETY

Annals of Computer Science and Information Systems, Volume 33

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering

USB: ISBN 978-83-965897-7-4
WEB: ISBN 978-83-965897-6-7
ISSN: 2300-5963
DOI: 10.15439/978-83-965897-6-7

Ul. Solec 38/103
00-394 Warsaw
Poland

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

Cover photo:
Miłosz Kulawiak,
Elbląg, Poland

Also in this series:

Volume 32: Communication Papers of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-965897-4-3, ISBN USB: 978-83-965897-5-0**
Volume 31: Position Papers of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-965897-2-9, ISBN USB: 978-83-965897-3-6**
Volume 30: Proceedings of the 17th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-962423-9-6, ISBN USB: 978-83-965897-0-5**
Volume 29: Recent Advances in Business Analytics. Selected papers of the 2021 KNOWCON-NSAIS workshop on Business Analytics**ISBN WEB: 978-83-962423-7-2, ISBN USB: 978-83-962423-6-5**
Volume 28: Proceedings of the 2021 International Conference on Research in Management & Technovation, **ISBN WEB: 978-83-962423-4-1, ISBN USB: 978-83-962423-5-8**
Volume 27: Proceedings of the Sixth International Conference on Research in Intelligent and Computing in Engineering, **ISBN WEB: 978-83-962423-2-7, ISBN USB: 978-83-962423-3-4**
Volume 26: Position and Communication Papers of the 16th Conference on Computer Science and Intelligence Systems, **ISBN WEB: 978-83-959183-9-1, ISBN USB: 978-83-962423-0-3**
Volume 25: Proceedings of the 16th Conference on Computer Science and Intelligence Systems, **ISBN Web 978-83-959183-6-0, ISBN USB 978-83-959183-7-7, ISBN ART 978-83-959183-8-4**
Volume 24: Proceedings of the International Conference on Research in Management & Technovation 2020, **ISBN WEB: 978-83-959183-5-3, ISBN USB: 978-83-959183-4-6**
Volume 23: Communication Papers of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-959183-2-2, ISBN USB: 978-83-959183-3-9**
Volume 22: Position Papers of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN WEB: 978-83-959183-0-8, ISBN USB: 978-83-959183-1-5**
Volume 21: Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, **ISBN Web 978-83-955416-7-4, ISBN USB 978-83-955416-8-1, ISBN ART 978-83-955416-9-8**

Dear Reader, it gives us immense pleasure to share the glimpses for the Seventh International Conference on Research in Intelligent and Computing in Engineering (RICE 2022). The RICE 2022 is organized by Hung Yen University of Technology and Education (UTEHY), Vietnam; Jointly Co-organised by Thu Dau Mot University (TDMU), Vinh University of Technology Education (VUTED), Ho Chi Minh City University of Transport (UTH) during November 11-12, 2022. It is great that after 3 stressful years of the Covid-19 pandemic, we were able to hold the Conference in hybrid mode.

This Conference provides an international forum which brings together the researchers as well as the industry practitioners, who are actively involved in the research in fields of intelligent computing, data science or any other emerging trends related to the theme covered by this conference. The Conference provides opportunities for scientists and industry professionals and students to meet and exchange ideas on the current state-of-the-art in the fields of intelligent systems and applications.

We are truly thankful to the Polish Information Processing Society (PTI), Poland for approving the proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering. It shall be published in the Annals of Computer Science and Information Systems series by the PIPS (ISSN-2300-5963). The books of this series have been submitted to Copernicus, DBLP, Cross Ref, Scholar, BazEkon, Open Access Library, Academic Keys, Journal Click, PBN, ARIANTE. At this stage, the effort, whole hearted support and suggestions given by Editor-in-Chief Prof. Marcin and Prof. Maria Ganzha is highly applaudable and commendable.

We would like to announce that the 7th edition of RICE attracts a good virtual footfall from various country researchers. It is an honor for us to receive valuable insights and visions regarding areas of interest to the conference from five keynote speakers from five countries. The details are given below:

- *Assoc. Prof. Marco Anisetti,* Università degli Studi di Milano, Italy
- *Prof. Manukid Parnichkun*, Industrial System Engineering Program, Asian Institute of Technology, Thailand
- *Prof. Gwanggil Jeon,* Incheon National University, Korea
- *Prof. Ashish Mishra,* Gyan Ganga Institute of Technology and Sciences, India
- *Dr. Le Anh Ngoc,* Director, Swinburne Innovation Space Swinburne University of Technology, Vietnam.

As per the guidelines issued by the Ministry of Vietnam, and keeping in view of health concerns, flexible options were shared with participants to join as Online/Offline Mode.

We are thankful to the leaders of UTEHY, TDMU, VUTED, UTH for their sincere cooperation and whole hearted support to make RICE 2022 a successful event. There are many names who play a vital role, but still we think that it is the right place to mention their support to make this conference a huge success.

- *Assoc. Prof. Bui Trung Thanh*, The Rector of Hung Yen University of Technology and Education, Vietnam
- *Dr. Nguyen Quoc Cuong,* The Rector of Thu Dau Mot University, Vietnam
- *Dr. Pham Huu Truyen,* The Rector of Vinh University of Technology Education, Vietnam
- *Dr. Nguyen Tien Thuy,* The Chairman of Ho Chi Minh City University of Transport, Vietnam
- *Dr. Vu Hong Son,* Hung Yen University of Technology and Education, Vietnam
- *Dr. Nguyen Ho Quang*, Thu Dau Mot University, Vietnam
- *Dr. Le Khac Binh*, Vinh University of Technology Education, Vietnam
- *Dr. Nguyen Quang Sang*, Ho Chi Minh City University of Transport, Vietnam

Last but not least, we would like to take this opportunity to extend our deepest gratitude to the Advisory Board, Organizing Committee, International Scientific Committee, institutions, companies, and volunteers, who have directly or indirectly supported the success of this conference.

Looking forward to welcoming you to upcoming RICE edition.

Proceeding's Editors – RICE 2022

Vijender Kumar Solanki, *Assoc. Prof.*, CMR Institute of Technology, Hyderabad, India.

Bui Trung Thanh, *Assoc. Prof.*, Hung Yen University of Technology and Education, Hung Yen, Vietnam.

Annals of Computer Science and Information Systems, Volume 33

Seventh International Conference on Research in Intelligent and Computing in Engineering

November 11–12, 2022. Hung Yen, Vietnam

TABLE OF CONTENTS

Seventh International Conference on Research in Intelligent and Computing in Engineering

Short Papers

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 1–6
DOI: 10.15439/2022R45
ISSN 2300-5963 ACSIS, Vol. 33

Analyzing Control Structures of SSSCs for Fast Compensation in the Case of Unbalanced Voltage Sag in Power System

Trinh Tran Duy
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
duytrinhktv@gmail.com

Nguyen Huu Thai
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
thainguyenktv@gmail.com

Trung Vo Tien
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
votientrung.vut@gmail.com

Dung Vo Tien
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
tdungtmv@gmail.com

Minh Vu Thi
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
Vuminhdhv@gmail.com

Tuyen Bui Trung
Faculty of electrical engineering
Vinh University of Technology Education
Vinh, Viet Nam
tuyenbt.acs@gmail.com

***Abstract*—Fast compensation of voltage sag on the three-phase electric power caused by different faults is a problem studied to ensure that loads work normally. This paper introduces two vector control structures for Series synchronous static compensators (SSSC): the PI controller on the dq reference frame and the Sequence-decoupled resonant (SDR) controller on the αβ reference frame. An analysis was performed to compare the two control structures to evaluate their qualities and advantages. Based on that conclusion, we propose a solution to apply suitably to controllers of SSSC devices. The control structures are simulated in Matlab/Simulink. The simulation results verified our analyses and evaluations. From there, this paper presents a recommended control structure for SSSC.**

***Index Terms*—Dynamic voltage restorer, Voltage sag, Power quality.**

I. Introduction

Voltage sag and the harmonics on the power grid are defined in IEEE Std. 1159-1995. These are serious problems, which are related to power quality. Furthermore, its frequency is the highest among all types of grid faults with 31% for voltage sag and 18% for harmonics [1]. Figure 1 shows a voltage sag in the power system and Fig. 2 shows its effects on electrical equipment and systems.

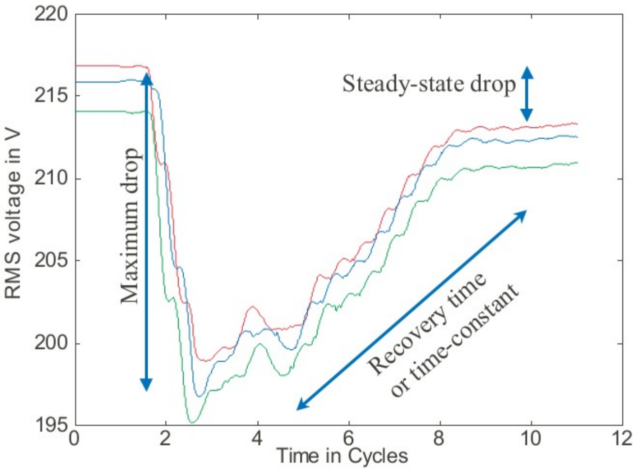


Fig. 1. Three-phase voltage sag [2].

Voltage sag is a type of disturbance that occurs unpredictably and depends on various faults occurring in the power system, for example, especially short circuit faults. Voltage sag occurs during small time and has voltage amplitude and phase angle variations that change rapidly and continuously during the faults.

The effect of voltage sag seriously affects the loads and is one of the causes of economic losses. The results of a power quality survey about the effects of voltage sag and harmonics of electricity consumers in 1,400 locations in 8 countries [1] are shown in Fig. 2.

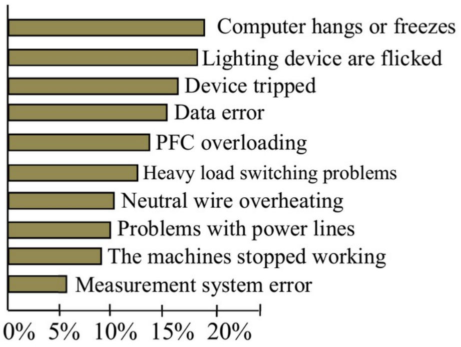


Fig. 2. Effects of voltage sag to electrical equipment and systems [2]

To solve this problem, there are many solutions to have been presented such as Generator – engines; Transformer-based regulator devices; Static switches; UPS systems; Series synchronous static compensators (SSSCs); D-Stacom. However, SSSCs are considered a highly efficient solution and the most suitable choice for consumers [2].

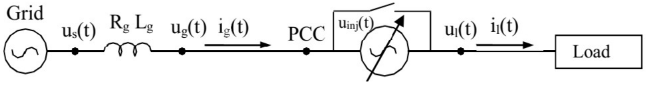


Fig. 3. Schematic of the operation principle of SSSC [3]

Basically, the SSSC is designed to be connected in series between the source and the load (Fig. 3). The SSSC is considered a voltage source whose magnitude, phase angle and frequency can be adjusted through an inverter.

The principle of regulating and stabilizing the load voltage of the SSSCs (u_L) is shown in Fig. 4. Where, i_l is the load current vector; φ is the phase difference angle between the load voltage and the load current; ψ is the phase jump angle of the load voltage, it is referenced to the grid voltage during voltage sag.

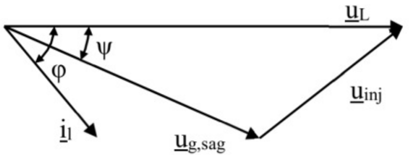


Fig. 4. Vector diagram of the sag compensation principle of SSSC [3]

When a voltage sag occurs, the SSSC will generate an injectable voltage (u_{inj}) whose magnitude and phase angle are determined by the controller. The load voltage becomes:

$$\vec{u}_L = \vec{u}_{g,sag} + \vec{u}_{inj} \tag{1}$$

The parameter of the voltage vector will depend on the rate of change of the grid voltage (u_{gsag}) and will be determined by the controller of the SSSCs.

In fact, when a voltage sag occurs, the rate of balanced voltage sag is approximately 5% and one of unbalanced voltage sag is approximately 95% [1,2]. A very important issue in control is that the balanced voltage sag just had to inject the positive sequence components, while the unbalanced voltage sag must inject positive sequence components, negative sequence components and zero sequence components. Therefore, when designing the controller, this issue should be taken into account. The solution for this problem is that we need to implement separately the sequence components to control. Figure 5 shows a diagram vector of positive sequence components and negative sequence components of the source voltage.

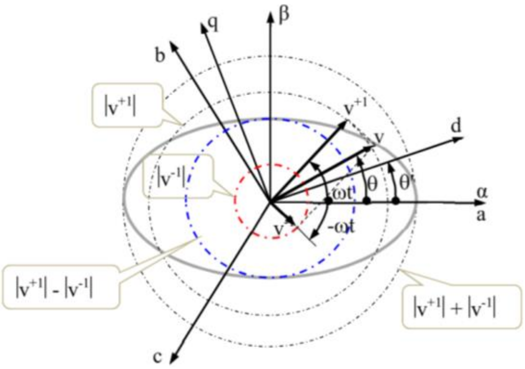


Fig. 5. Vector diagram of source voltage in the unbalanced voltage sag [4]

Where, the vector v is the voltage of the power system, which is separated by the sum of the positive sequence component v^{+1} and the negative sequence v^{-1}.

The vector v contains information for both the positive and negative sequence components. Therefore, the control of the voltage vector v will be replaced by the control of two vectors: the positive sequence component v^{+1} and the negative sequence component v^{-1}.

A control structure for an SSSC in case of unbalanced voltage sag is built as shown in Fig. 6.

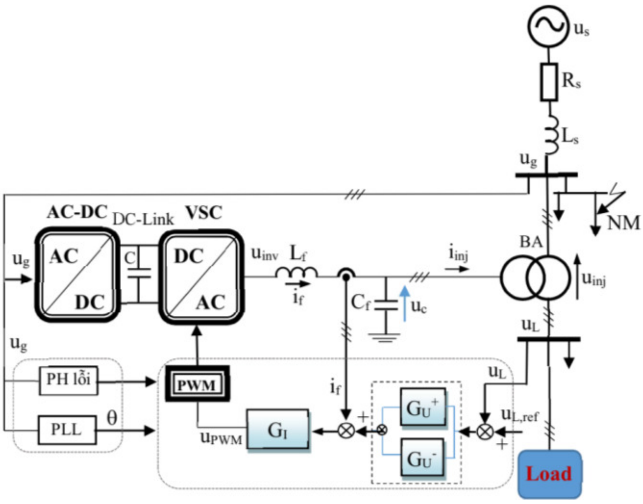


Fig. 6. SSSC control structure with current feedback from the inverter.

Where, N - three-phase electric power; R_s, L_s are source resistance, source inductance, respectively; u_g is the voltage of power grid; u_L is load voltage, BA is a series transformer; L_f, C_f is filter inductance, filter capacitor, respectively; AC/DC is rectifier converter; VSC-DC/AC is inverter; DC-link is DC bus; PH is voltage sag detector; PLL is phase-locked loop; G_{U+}, G_{U-} are positive and negative sequence components voltage controller; G_I is current controller; PWM is pulse width modulator; i_f is current feedback signal; u_L is load voltage feedback signal; $u_{L,ref}$ reference load voltage.

The SSSC control structure shown in Fig. 6 can be designed based on *abc* reference frame or *dq* reference frame or *αβ* reference frame [5].

There are many controllers that have applied this structure to SSSC [5,6,7,8]. Two prominent controllers of them are PI controller for reference frame *dq* and Sequence Decoupled Resonant (SDR) for *αβ* reference frame.

II. Three-phase SSSC control structure in case of unbalanced mains voltage sag

A. *The vector control structure cascades on the dq reference frame to the PI controller*

The control structure is built based on the vector control principle in Fig. 7.

In this structure, the current control loop of the LC filter directly affects the compensating power of the SSSC. The control loop of injection voltage controls the load voltage following the reference load voltage ($u_{L,ref}$). The current controller G_I and voltage controller G_U are PI controllers. The transformation *abc* to *αβ*, *αβ* to *dq* and a phase-locked loop (PLL) are important components of the control structure.

Projecting the vectors in equation (1) onto the *dq* reference frame, the reference value of the injection voltage is determined as follows:

$$u_{inj}^{dq} = u_C^{dq} = u_L^{dq} - u_s^{dq} \tag{2}$$

The actual insertion voltage (u_{inj}) is also the voltage across the capacitor of the LC filter.

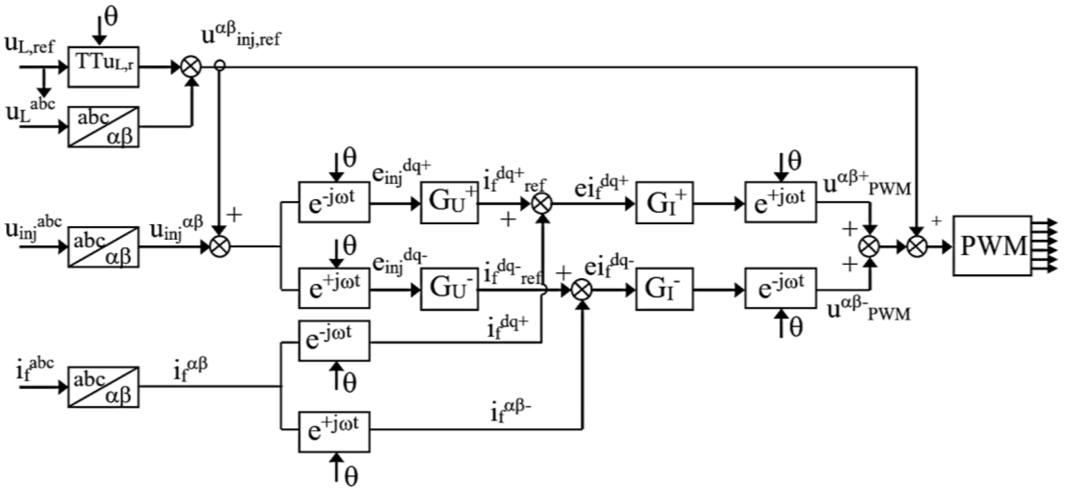


Fig. 7. SSSC control structure on rotation dq reference frame.

- ***Current control loop***

The positive and negative sequence current components are separated for control.

Positive sequence current control loop:

$$u_{inv}^{dq+}(k) = u_{inj}^{dq+}(k) + R_f i_f^{dq+}(k) \mp j\omega L_f i_f^{dq-}(k) + G_I^{+} \frac{L_f}{T_s}(i_f^{dq*+}(k+1) - i_{inj}^{dq+}(k)) \tag{3}$$

Negative sequence current control loop:

$$u_{inv}^{dq-}(k) = u_{inj}^{dq-}(k) + R_f i_f^{dq-}(k) \pm j\omega L_f i_f^{dq+}(k) + G_I^{-} \frac{L_f}{T_s}(i_f^{dq*-}(k+1) - i_{inj}^{dq-}(k)) \tag{4}$$

Where: *G* is the transfer function of the discrete controller Integral – Proportional.

- ***Voltage control loop***

The positive and negative sequence voltage components are separated for control.

Positive sequence voltage control loop:

$$i_f^{dq+}(k+1) = i_{inj}^{dq+}(k) + G_U^{+} \frac{C_f}{T_s}(u_{inj}^{dq*+}(k) - u_{inj}^{dq+}(k)) \pm j\omega C_f u_{inj}^{dq-}(k) \tag{5}$$

Negative sequence voltage control loop:

$$i_f^{dq-}(k+1) = i_{inj}^{dq-}(k) + G_U^{-} \frac{C_f}{T_s}(u_{inj}^{dq*-}(k) - u_{inj}^{dq-}(k)) \mp j\omega C_f u_{inj}^{dq+}(k) \tag{6}$$

B. SSSC control structure on αβ reference frame with SDR controller

The control structure is simpler because it uses a quadratic resonant regulator that can handle the positive and negative sequence components separately as shown in Fig. 8.

- ***Current control loop***

The equation describing the current loop by using a resonant controller is as follows:

$$u_{inv,ref}^{\alpha\beta}(k+1) = u_{inj}^{\alpha\beta}(k) + (G_{I-SDR}^{+} \frac{L_f}{T_s}(i_{f,ref}^{\alpha\beta}(k) - i_f^{\alpha\beta}(k)) + G_{I-SDR}^{-} \frac{L_f}{T_s}(i_{f,ref}^{\alpha\beta}(k) - i_f^{\alpha\beta}(k))) \tag{7}$$

- ***Voltage control loop***

The equation describing the voltage loop by using a resonant controller is as follows:

$$i_f^{\alpha\beta*}(k+1) = i_{inj}^{\alpha\beta}(k) + (G_{U-SDR}^{+} \frac{C_f}{T_s}(u_{inj,ref}^{\alpha\beta}(k) - u_{inj}^{\alpha\beta}(k)) + G_{U-SDR}^{-} \frac{C_f}{T_s}(u_{inj,ref}^{\alpha\beta}(k) - u_{inj}^{\alpha\beta}(k))) \tag{8}$$

Where,

G_{U-SDR}^{+} , G_{U-SDR}^{+} - are the voltage and current resonance controller for the positive sequence component, respectively.

G_{U-SDR}^{-} , G_{I-SDR}^{-} - are the voltage and current resonance control for the negative sequence component, respectively.

In the introduced control structure, the SDR regulator is equivalent to conventional resonant controllers but it can be built with the MIMO control structure, which is easier to implement. The SDR controller has been described detailed and designed in [3, 8].

The SDR controller with multiple inputs and multiple outputs is described as follows:

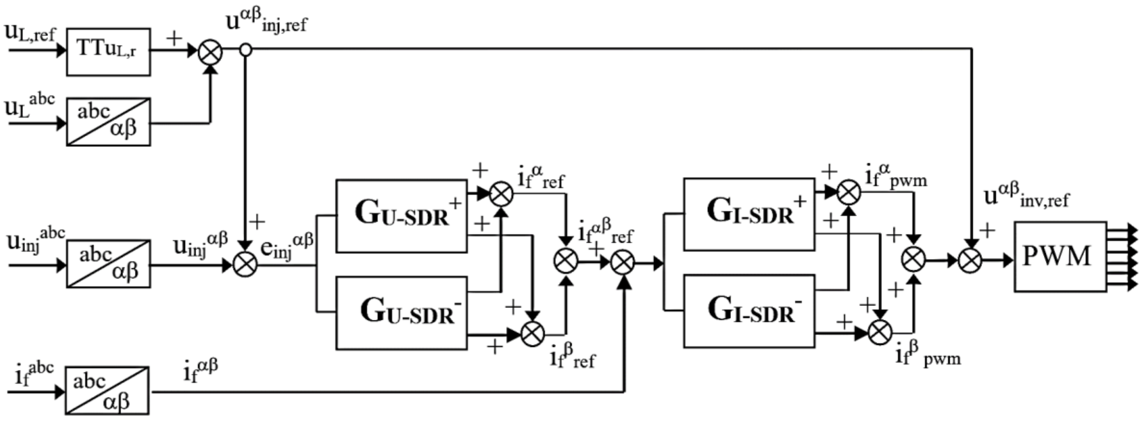


Fig. 8. SSSC control structure on αβ reference frame.

$$y^+_{\alpha 1}(s) = \frac{1}{s}\left[K_I.e_\alpha(s) - \omega_1.y^+_{\beta 1}(s)\right]$$
$$y^+_{\beta 1}(s) = \frac{1}{s}\left[K_I.e_\beta(s) + \omega_1.y^+_{\alpha 1}(s)\right] \quad (9)$$

$$y^-_{\alpha 1}(s) = \frac{1}{s}\left[K_I.e_\alpha(s) + \omega_1.y^-_{\beta 1}(s)\right]$$
$$y^-_{\beta 1}(s) = \frac{1}{s}\left[K_I.e_\beta(s) - \omega_1.y^-_{\alpha 1}(s)\right] \quad (10)$$

According to this method, the equations described in equations (9) and (10) can be easily performed on the *αβ* reference frame in the time domain by digital methods.

Figure 9 shows the implementation structure diagram of the controller to control positive or negative sequence voltage components or both.

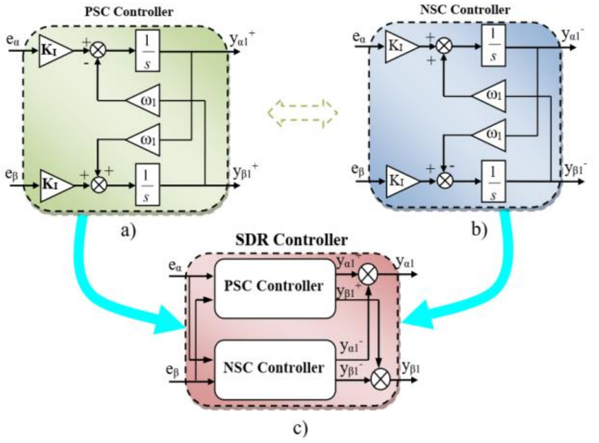


Fig. 9. Structure diagram of the SDR controller on αβ reference frame . a) Positive sequence component controller b) Negative sequence component controller c) both sequence components [4,7].

III. Simulation Results

A. *Case 1: Compensation for unbalanced voltage sag*

An unbalanced voltage sag occurs on the grid between t = 0.12s and t = 0.25s. The amplitude of voltage sag in phases is not equal: phase C 40%, phase B 10%.

Source voltage u_s

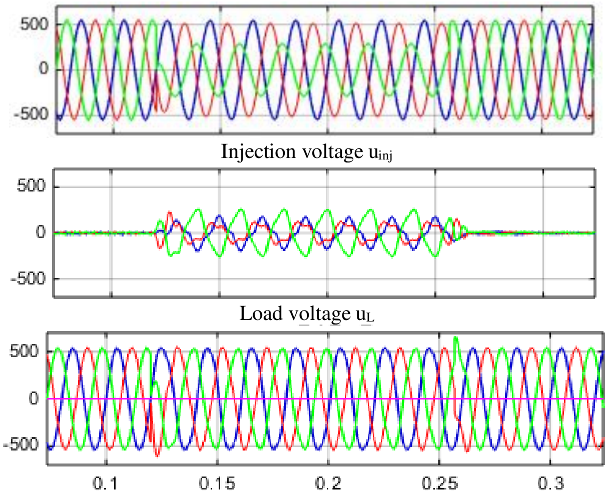


Fig. 10. Simulation results of SSSC using control structure on *dq* reference frame in case of unbalanced voltage sag.

Source voltage u_s

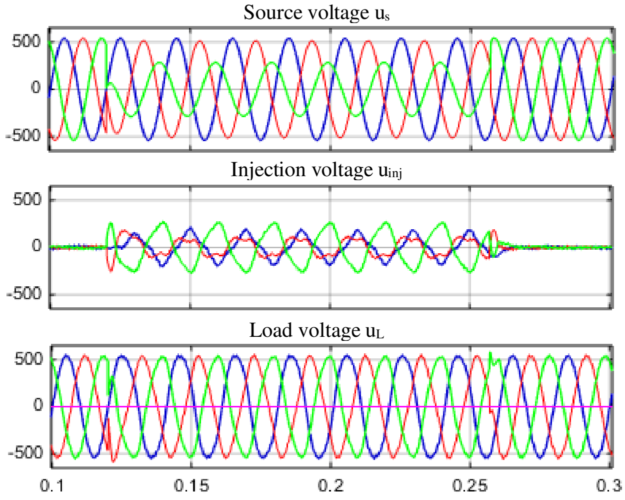


Fig. 11. Simulation results of SSSC using control structure on *dq* reference frame in case of unbalanced voltage sag.

Figure 11 shows the difference between the reference load voltage and the actual load voltage of the two control structures during the time period from t = 0.12s to 0.25s. The results show that the error of the two structures is approximately zero. The results also show that the actual load voltage follows exactly the reference value.

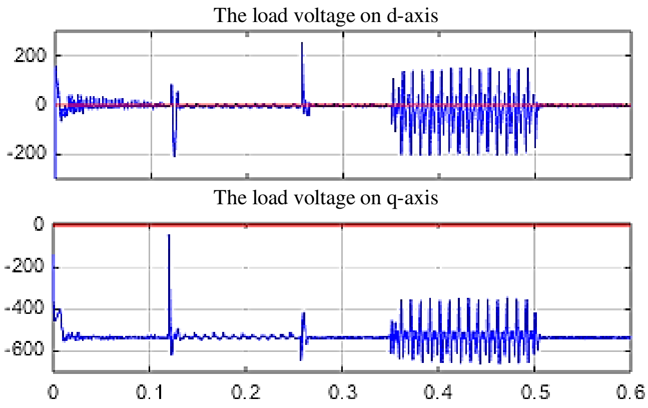


Fig. 12. Simulation results of the load voltage of the control structure on the *dq* reference frame .

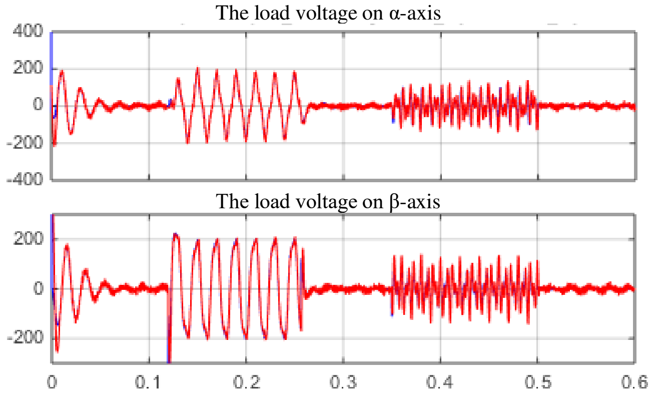


Fig. 13. Simulation results of the load voltage of the control structure on the αβ reference frame .

Fig.12 shows the result of harmonic spectrum and total harmonics (THD) of the load voltage of both control structures during the period from 0.12s to 0.25s.

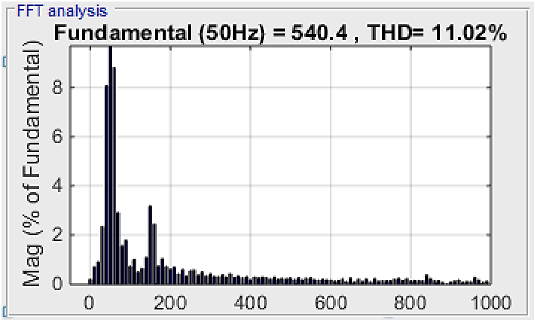


Fig. 14. Harmonic analysis results of SSSC using control structure on dq reference frame .

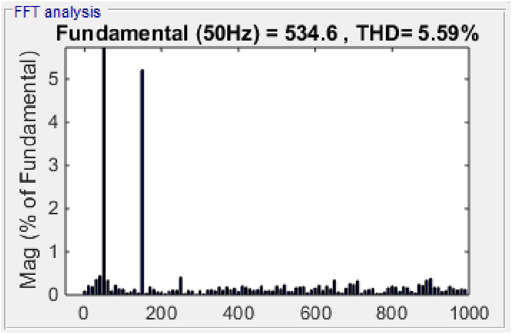


Fig. 15. Harmonic analysis results of SSSC using control structure on αβ reference frame .

While harmonic spectrum of the structure using the PI controller includes: 3th-0.5%, 5th-3%, 7th-6%, THD = 11.02%, the ones of structure using the SDR controller includes harmonic: 3th-0.5%; 5th-3%; 7th-6%. THD = 5.59%.

B. *Case 2: Compensation for unbalanced voltage sag containing harmonic components in the fault.*

A voltage sag occurs during from 0.12s to 0.25s. Addition, during 0.17s to 0.3s, harmonic components: 3th = -10%, 5th = -10% are added.

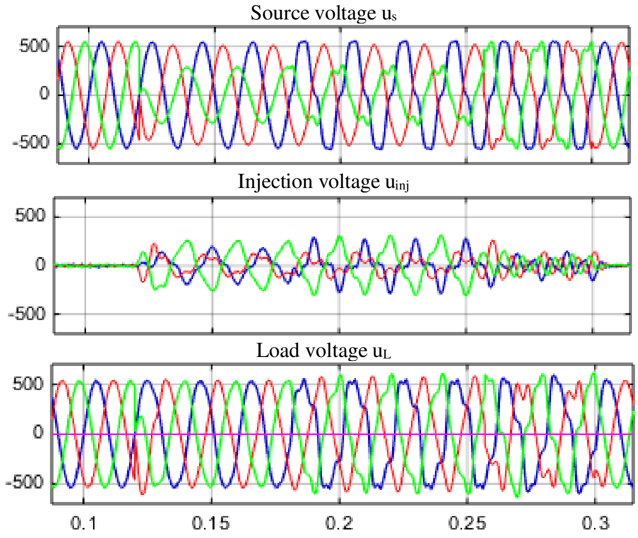


Fig. 16. Simulation results of unbalanced voltage sag compensation of SSSC using control structure on *dq* reference frame .

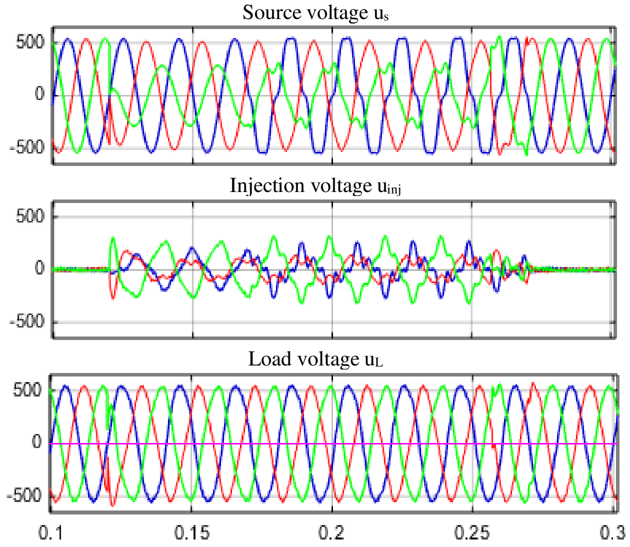


Fig. 17. Simulation results of unbalanced voltage sag compensation of SSSC using control structure on alpha-beta reference frame .

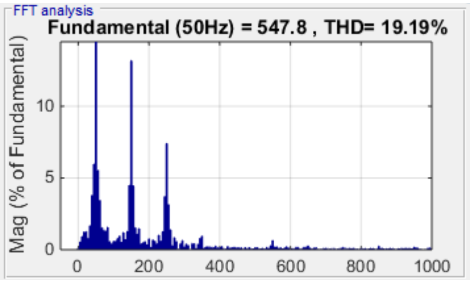


Fig. 18. Harmonic analysis results of SSSC using control structure on *dq* reference frame .

Figure 18 shows the results of the analysis of the harmonic spectrum and the total harmonics of the load voltage while the voltage sag was ocurring. The results show that the load voltage in the control structure on the *dq* reference frame includes harmonic components: 3th-10%, 5th-10%, 7th-6% and the total harmonics increased from 14.56% (pre-sag) to 19.19% (in-sag).

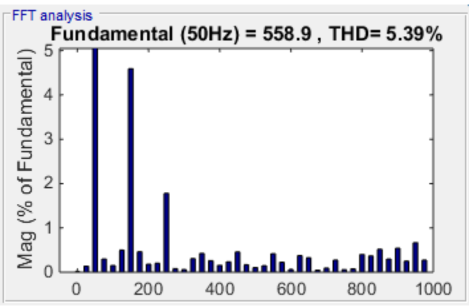


Fig. 19. Harmonic analysis results of SSSC using control structure on *αβ* reference frame .

Meanwhile, the load voltage in the control structure on the *αβ* reference frame using the SDR controller include harmonic components: 3th-10%, 5th-10% and the total harmonics are reduced from 14.56% (pre-sag) to 5.39% (in-sag).

Remark 1: The vector control structure on the reference frame *dq*, which uses PI controllers, is capable of controlling the positive and negative sequence components independently. The control signals on this structure are DC signals. Therefore, this system is capable of making tolerance to 0. However, this control structure requires 8 PI controllers for 4 control channels and has to use 6 *αβ/dq* transformations and 4 change *abc/αβ* transformations. The results show that this control structure is capable of fast compensatory response, which is approximately from 0.005s to 0.01s (Fig. 10, Fig. 16 and 17). The load voltage is balanced and follows the reference voltage. Total harmonics are 11.02% (Fig. 14). However, the ability compensation in case of including harmonic components of this structure is not high quality, and the load voltage is not pure sine (Fig. 10, Fig. 16 and 17).

Remark 2: The vector control structure on the *αβ* reference frame with the SDR controller is capable of controlling the sequence components on the *αβ* reference frame independently. The reason of this problem is that its controller is built as a MIMO structure. Adjustment of controller parameters can be done independently for each sequence component. Therefore, the SDR controller has the ability to calibrate more precisely. The results show that the load voltage follows the reference voltage, the load voltages are balanced and the total harmonics of the load voltage are small, 5.59%. In addition, it is capable of compensating both voltage drop and harmonics effectively. The total harmonics of the load voltage are reduced from 14.56% to 5.39% (Fig. 12). The control structure on the αβ reference frame with the SDR controller is simple and easy to implement.

IV. Conclusion

This paper analyzes and evaluates two control structures of SSSCs for fast compensation of balanced and unbalanced voltage sag. Vector control structure on dq reference frame with PI controller and vector control structure on *αβ* reference frame with SDR controller was applied. Following the result of analysis, simulation and evaluation, the vector control structure on the *αβ* reference frame with SDR controller which has many advantages is recommended to use for SSSCs.

References

[1] Bhattacharyya, S., Myrzik, J. M. A., & Kling, W. L. (2007, September). Consequences of poor power quality-an overview. In 2007 42nd International Universities Power Engineering Conference (pp. 651-656). IEEE.

[2] Strzelecki, R. M. (Ed.). (2008). Power electronics in smart electrical energy networks. Springer Science & Business Media.

[3] Duy, T. T., Tien, D. V., Gono, R., & Leonowicz, Z. (2016, June). Mitigating voltage sags due to short circuits using dynamic voltage restorer. In 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC) (pp. 1-6). IEEE.

[4] Ye, J., Gooi, H. B., Wang, B., Li, Y., & Liu, Y. (2019). Elliptical restoration based single-phase dynamic voltage restorer for source power factor correction. Electric Power Systems Research, 166, 199-209.

[5] Wang, F., Benhabib, M. C., Duarte, J. L., & Hendrix, M. A. (2009, February). Sequence-decoupled resonant controller for three-phase grid-connected inverters. In 2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition (pp. 121-127). IEEE.

[6] Vo Tien, D., Gono, R., Leonowicz, Z., Tran Duy, T., & Martirano, L. (2018). Advanced control of the dynamic voltage restorer for mitigating voltage sags in power systems. Advances in Electrical and Electronic Engineering, 16(1), 36-45.

[7] Duy, T. T., Trung, T. B., & Tien, D. V. (2021, August). Voltage Sag Reduction using a Dynamic Voltage Restorer under Different Types of Faults in The Power System. In 2021 International Conference on System Science and Engineering (ICSSE) (pp. 246-251). IEEE.

[8] Lee, G. M., Lee, D. C., & Seok, J. K. (2004). Control of series active power filters compensating for source voltage unbalance and current harmonics. IEEE Transactions on industrial electronics, 51(1), 132-139.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 7–10 DOI: 10.15439/2022R40 ISSN 2300-5963 ACSIS, Vol. 33

Improved DSR Routing Protocol Performance Using Intelligent Mobile Agent

Pham Quoc Hung
Faculty of Information Technology
Hung Yen University of Technology and Education
Hungyen, Vietnam
quochungskh@gmail.com

Nguyen Vinh Quy*
Faculty of Information Technology
Hung Yen University of Technology and Education
Hungyen, Vietnam
vinhquynguyen@gmail.com

Dao Manh Linh
Faculty of Information Technology
Hung Yen University of Technology and Education
Hungyen, Vietnam
daolinh1289@gmail.com

Vi Hoai Nam
Faculty of Information Technology
Hung Yen University of Technology and Education
Hungyen, Vietnam
vihoainam@gmail.com

***Abstract*—Mobile Ad hoc Network (MANET) is a special wireless network that has the advantage of operating independently of fixed network infrastructure, cheap cost, quick deployment, and high mobility. In this study, the research team controls the DSR routing protocol in the network using a mobile agent that moves from one network node to another in order to improve the performance of MANET. The research team assessed the efficacy of the use of agent technology in the DSR procedure by NS2 simulation software version 2.34. The results of the experiment demonstrate that the use of mobile agent technology enhances MANET performance.**

***Index Terms*—MANET, Agent, DSR, LSD, LB-DSR, LET, EDR.**

I. Introduction

Today, along with the explosion of information technology and the strong development of wireless networks, mobile devices, science, and technology have developed dramatically in recent years, in which mobile networks are customized. MANET is widely applied in many fields such as science and technology, engineering, economics, medical and rescue, and accident prevention. With great benefits, MANET is expected to be one of the outstanding technologies to serve connection demands because of its ability to operate independently of fixed network infrastructure, low cost, quick deployment, and high mobility.

In recent years, many studies have been conducted to improve the performance of routing protocols in MANET with the application of Mobile Agents. An agent is a physical or logical entity in which processes are autonomous and can travel from node to node for completing tasks [1][2]. The basic principle is to use mobile agents to process the data, which can reduce network load, boost performance, reduce latency, and improve compatibility across heterogeneous network environments [3][4]. Interconnected and continuously connected devices place requirements while utilizing MANET network. With typical MANET, network nodes move freely and independently from each other. The communication between MANET nodes is peer-to-peer. It means that there is no level priority and no differentiation between node roles.

The integration of mobile agents into the network protocol will increase the optimization of the transmission protocol in the network. In this particular case is the optimal choice of network operation.

In [1][3] the research team published an algorithm using a mobile agent to improve the efficiency of the AODV (Ad hoc On-Demand Distance Vector), DSR (Dynamic Source Routing) routing protocol in MANET. The goal of this research is to propose a route selection mechanism to balance the traffic across nodes in the overall network to reduce congestion by setting an evaluation function for the congestion threshold at each node. The MAR-AODV protocol (Mobile Agent Routing-AODV) [1] and MAR-DSR protocol (Mobile Agent Routing-DSR) have a lower probability of packet congestion than the original AODV and DSR protocols. Some studies propose protocols to reduce latency and save energy based on mobile agents [5] [6] while others work on load balancing using mobile agents [8]. In this paper, the research team has implemented agent technology with the inclusion of LSD and Bandwidth fields in RREQ and RREP packets to serve as factors for choosing the most effective route of the DSR protocol. The DSR protocol in the presence of an agent is called LB-DSR to compare the performance rating with the original DSR protocol.

II. On-Demand Routing Protocol in MANET

Routing in networks is very important, routing can happen before the system needs to transmit data or during data transmission. On-demand routing is one of the routing methods that only occur when the system needs to transfer data. On-demand routing is considered to be efficient and has advantages in MANET, in which DSR, and AODV protocols are significant. In this study, the research team focuses on

* Corresponding author

analyzing and applying agent technology to improve the DSR protocol.

A. Dynamic Source Routing protocol

The Dynamic Source Routing (DSR) protocol is a simple and efficient routing protocol in MANET. It is a reactive protocol that uses a source routing algorithm that allows the network to automatically organize and configure itself without the intervention of an administrator or the network's infrastructure. The header of the data packet contains the order of nodes that need to be traversed to reach the destination. Therefore, intermediary nodes simply need to communicate with their neighbors to forward packets. At each node in the network, there is always a buffer memory of packets that will receive information about the path and perform transmission on the selected path. Conversely, when there is no route in the route-cache or a route exists but is no longer valid, DRS will perform route discovery by sending RREQ (Route REQuest) broadcasts to neighboring nodes on the network. When a route is found, the RREP (Route REPly) packet contains the order of the hops to the destination and is transmitted back to the source.

The operation of the DSR protocol consists of two main mechanisms:

1. Create routing information (Route Discovery)
2. Maintain routing information (Route Maintenances)

The path discovery algorithm is as follows:

Step 1: Through the request ID field, verify whether the packet has been received or not. If it already exists, remove it and execute no further processing. Otherwise, go to step 2.

Step 2: Check in the Route Cache that the route to the destination node is valid. Then, send a response to the source node with an RREP packet containing information about the path to the destination and terminate the process. Otherwise, go to step 3.

Step 3: Check whether the destination address to find is the same as the current node's address. If there is a match, send back to the source node an RREP packet with information about the path to the destination and terminate the process. Otherwise, send broadcast RREQ packets to neighboring nodes. Neighboring nodes after receiving the RREQ packet will check the information and return to step 1. The process continues until the source node receives information about the path to the destination, or the end of the packet's lifetime (running out of Time to Live).

B. Some disadvantages of DSR protocol

Analyzing the working mechanism of the DSR protocol, there are some comments as follows:

At each node, information about the entire route is always maintained, so when there is a path error or local congestion at a certain node, there will be a problem of packet drop or unspecified transmission error.

DSR uses source routing, whereby it always replies to all routing requests. This approach allows DSR to gather more paths to the destination and better transmission capacity than AODV. However, this is only useful in case the network has few transmitters and the mobility is not high. As nodes increase and mobility is high, the possibility of nodes losing connection with each other leads to a significant decrease in network performance.

In the DSR protocol, route maintenance is not concerned with the state of the nodes in the route. Specifically, when there is an incoming request, if there is a route in the cache, then proceeds to transmit immediately, even if that intermediate node exists or is congested. This is also a basic drawback that needs to be improved to further enhance the performance of the DSR protocol.

III. Proposed Solution for MANET Based on Mobile Agent

A. Related works

The nodes in the network are maintained on battery power. With limited battery power, nodes will lose connection when they run out of power or move out of coverage. Therefore, in order to establish stability in MANET, it is important to maintain the state stability of the node and the connection between the nodes.

Other studies simply analyze particular parts of the issue such as M. K. Marina et al [11], Jiwon Park et al [9] calculated multipath but disregarded the stability factor; W. Su et al [12], Abubakar Bello Tambuwal et al [7] evaluated the TTL factor but ignored the effect of the energy problem; M. Bheemalingaiah et al [13] calculated energy costs but neglected the TTL factor; Jiwon Park et al [9] measured route quality but overall network stability is not discussed.

B. Method

To stabilize the link between and the node, it is necessary to determine two parameters: Link Expiration Time (LET) [12] and Energy Drain Rate (EDR) [10].

The mobility factor LET was proposed in [12], by using the motion parameters (velocity, direction) of the nodes. When sending data, nodes consume considerable energy. Besides, they also spend energy listening to the neighbor nodes as well. That means they still lose power even when not transmitting data. The Energy Drain Rate of a node is defined in [10] as the rate of dissipation of energy of a node.

Considering the two factors LET and EDR, the stability of the link depends directly on the time and inversely depends on the node energy. Therefore, the level of Link Stability Degree (LSD) is determined:

$$LSD = LET/EDR$$

The higher the LSD value, the more reliable the link stability.

In the operation of the DSR protocol, there is a mobile agent to determine the LSD when the source node sends the RREQ packet to the neighbors. The value of the LSD and the request Bandwidth will be initialized and added to the RREQ packet. The RREQ packet in the DSR protocol will add two additional information fields, LSD and Bandwidth as shown in table I.

TABLE I. RREQ PACKET STRUCTURE HAS LSD, BANDWITDH

S A	DA	Type	ID	TTL	Hops	LSD	Bandwidth	Path

At the nodes when receiving the RREQ packet, the mobile agent determines that packets with the same identifier traverse different paths, updating the value of the LSD. Only nodes with LSD greater than the requested LSD and Bandwidth greater than or equal to the required Bandwidth will be selected to forward the packet. If a node receives duplicate RREQ packets with the same source address and ID as received from other paths, the RREQ packet is dropped.

To monitor RREQ packets, each node will store a Neighbor Information Table (NIT). This is the node's agent, which has access to data such as Bandwidth, LSD, ID, Hops, Source Address (SA), and Destination Addresses (DA).

TABLE II. INFORMATION FIELD IN THE NIT TABLE.

SA	DA	ID	Hops	LSD	Bandwidth

The node selects the route with the highest LSD when it gets several RREQ packets. For the same LSD, the path with the fewest hops is chosen. The route with the most bandwidth is picked if the hops are also the same. If there is an equal amount of bandwidth, a first-come-first-served selection will be made. Then, the source node then receives a RREP packet that also includes the addresses of the nodes and the total bandwidth for each routing request.

IV. SIMULATION AND ANALYSIS OF TEST RESULTS

A. Simulation parameters

In this part, the research team set up simulations and evaluated the performance of the main network nodes based on the following criteria: Average delay, Throughput, and packet delivery rate on NS2 simulation software version 2.34 according to the following two scenarios:

- Performance evaluation of network nodes in the DSR protocol of conventional MANET;
- Performance evaluation of network nodes in DSR protocol with the mobile agent in LB-DSR protocol in MANET.

TABLE III. SIMULATION PARAMETERS

Parameters	Values
Simulation area	(500x500) m
Number of nodes	30
Flow-type	CBR
Bandwidth	11 Mbit /s
Size of packet	1024 byte
Time	90 s
MAC layer	802.11b
Transport layer	UDP
Speed	2(m/s)

The simulation system consists of 30 mobile nodes randomly arranged in an area of (500x500) m, using IEEE 802.11b standard with a speed of 11 Mbps and using User Datagram Protocol (UDP). Simulation is done in 90 seconds. The number of terminal connections in the simulations is 5, 10, 15, 20, 25 and 30, respectively.

B. Simulation results and analysis

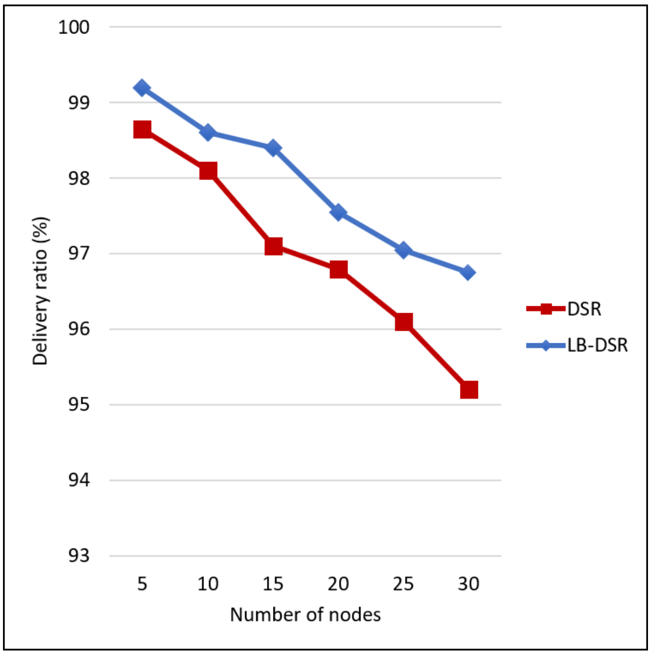


Fig. 1. Packet delivery rate

The team assesses the nodes' stability in the first experiment, figure 1, using the following criterion: packet delivery rate. The simulation results use the packet delivery ratio to compare the two protocols' performance. The results demonstrate that the LB-DSR protocol has a higher packet delivery rate than the original DSR system. With 5 nodes, the success percentage is more than 99%. The distribution ratio of both decreases as transmitted traffic volume rises. With the DSR protocol, however, the packet delivery rate dramatically drops as the amount of sent traffic rises from 20 to 30.

In figure 2, simulation results show that the transmission throughput of the LB-DSR protocol is always larger than that of the DSR protocol. When the number of terminal connections increases, the throughput decreases. Especially when the number of connections increases, the throughput of the network decreases due to congestion or a rapid decrease in energy at the nodes.

Figure 3 shows the performance of two protocols DSR and LB-DSR. The results show that the LB-DSR protocol has a lower delay time than the DSR protocol because the chosen route is the one with the highest stability and the best bandwidth. However, as the number of transmitted traffic increases, the delay time also increases but is still lower than the original DSR protocol.

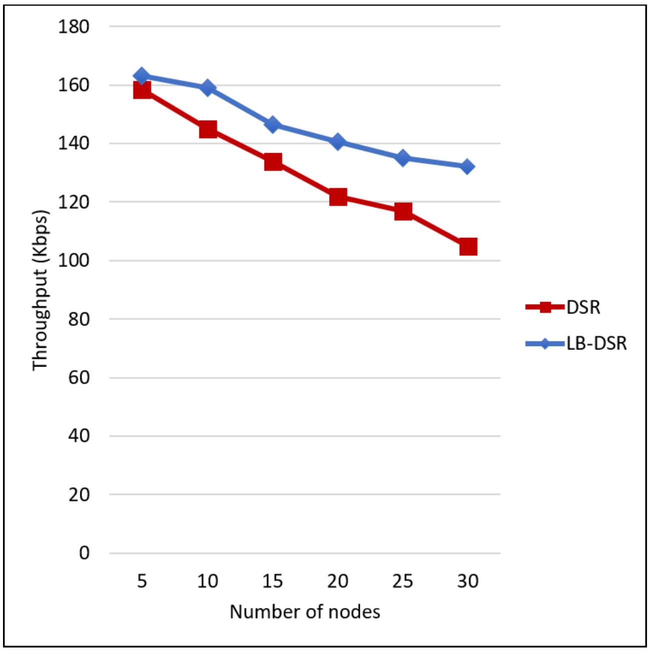


Fig. 2 Average throughput

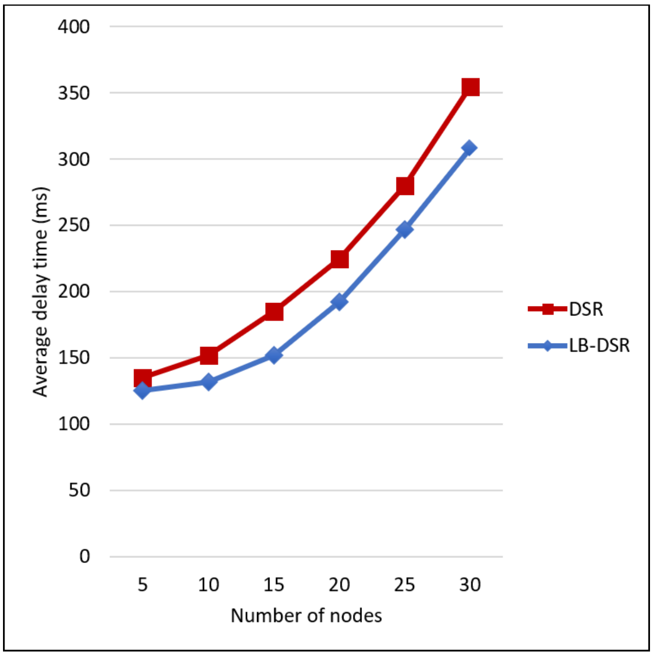


Fig. 3 Average delay time

V. Conclusions

The team assesses the nodes' stability in the first experiment, figure 1, using the packet delivery rate criterion. The simulation results use the packet delivery ratio to compare the two protocols' performance. The results demonstrate that the LB-DSR protocol has a higher packet delivery rate than the original DSR system. With 5 nodes, the success percentage is more than 99%. The distribution ratio of both decreases as transmitted traffic volume rises. With the DSR protocol, however, the packet delivery rate dramatically drops as the amount of sent traffic rises from 20 to 30.

References

[1] Cung Trong Cuong, Nguyen Thuc Hai, Vo Thanh Tu, “MAR-AODV: Innovative Routing Algorithm in MANET Based on Mobile Agent,” Proc. IEEE WAINA, Spain, 2013, pp. 62-66.

[2] A. Korichi and Y. Zafoune, "MAAC Protocol: Mobile Agents Based Address Auto-Configuration Protocol for MANET," International Conference on Smart Communications in Network Technologies (SaCoNeT),2018, pp. 194-199.

[3] Binternational, “Performance Analysis of AODV and QAODV Protocol”, Journal of Emerging Engineering ResBArch and Technology, 2014, pp. 54 – 57.

[4] Dr. Shuchita Upadhayaya and Charu Gandhi, ”Quality of Service Routing in Mobile Ad hoc Networks Using Location and Energy Parameters”, International Journal of Wireless & Mobile Networks (IJWMN), Vol 1, No 2, 2009, pp. 138-147.

[5] M. N. Anjum et al., "Implementing Mobile Agent Based Secure Load Balancing Scheme for MANET," 2019 International Conference on Opto-Electronics and Applied Optics (Optronix), 2019, pp. 1-6, DOI: 10.1109/OPTRONIX.2019.8862375.

[6] Qing Liang et al., “A dynamic source routing protocol based on path reliability and link monitoring repair”, 2021, DOI: 10.1371/journal.pone.0251548.

[7] Abubakar Bello Tambuwal et al., “Improvement of DSR routing protocol using TTL-based scheme in Mobile Ad Hoc Networks”, IEEE International Conference on RFID-Technology and Applications (RFID-TA), 2013, DOI:10.1109/RFIDTA.2013.6694513.

[8] Ankit Kumar , Madhavi Sinha , “Design and Analysis of Improved Dynamic Source Routing Protocol in Vehicular Ad-Hoc Network”, International Journal of Sensors, Wireless Communications and Control, Volume 12, Issue 2, 2022, DOI:10.2174/2210327910999201230105923.

[9] Jiwon Park et al, ”Multipath AODV Routing Protocol in Mobile Ad Hoc Networks with SINR-Based Route Selection”, International Symposium on Wireless Communication Systems, IEEE, 20089, pp:682-688.

[10] D.Kim et al, "Routing mechanisms for Mobile Ad hoc Networks Based on Energy Drain Rate", Mobile Computing, Vo12, 2003, pp: 161-173.

[11] M. K. Marina & S. R. Das, "On-Demand MultiPath Distance Vector Routing in Ad hoc Networks",Proceedings of the Ninth International Conference on Network Protocols (ICNP}, IEEE Computer Society Press, 2011, pp. 14-23.

[12] W.Su, Sung-Ju Lee & M.Gerla,”Mobility Prediction in Wireless Networks”.,MILCOM , 2000, Vol l, pp.491-495.

[13] M. Bheemalingaiah et al.,"Energy Aware Node-Disjoint Routing in Mobile Ad Hoc Networks", Journal of Theoretical and Applied Information Technology, 2009, pp 416-431.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 11–14 DOI: 10.15439/2022R49 ISSN 2300-5963 ACSIS, Vol. 33

Adaptive PID-Type Iterative Learning Control for DC Motor Position

Tri-Quang Le
Dept. Automobile Engineering
Hung Yen University of Technology and Education
Hung Yen, Vietnam
quangletri@gmail.com

Nguyen Ngoc Tu
Dept. Automobile Engineering
Hung Yen University of Technology and Education
Hung Yen, Vietnam
nguyenngoctuuct@gmail.com

Le Ngoc Truc
Dept. Automobile Engineering
Hung Yen University of Technology and Education
Hung Yen, Vietnam
lengoctruc@gmail.com

***Abstract*—The paper aims to control the DC motor position. The proposed method is adaptive PID-type iterative learning control based on fuzzy logic. The Developed processor-in-the-loop simulation based on Simulink and Arduino Mega 2560 demonstrated the high performance of the proposed solution.**

***Index Terms*—PID, ILC, DC motor, Adaptive control**

Introduction

DC motor has many advantages, such as being inexpensive, reliable, and giving high torque at a low speed [1]. Moreover, it is robust control for both speed and position. Hence, it widely applies, for example, the electric vehicles, sheets, mirrors, etc., enhancing the control accuracy of the position and speed of the DC motor are two aspects that attract many researchers worldwide. Chotai [2] applied the linear control methods (PID and LQR) to control the speed of the DC motor. The conventional PID instances three terms: proportional, integral, and derivation. The controller output is the sum of these terms based on the error information. PID has many advantages as simply an algorithm and guarantees satisfactory performance. The limitation of PID, the value of control gain (Kp, Ki, Kd), constantly leads to reduced control quality in the vast range of operations, especially in nonlinear under-actuated systems [3]. LQR is a type of linear optimal control; its quality depends on the accuracy of the system model and the value of the weight matrices Q and R.

The linear control method cannot be well handled since DC motors operate in disturbances and uncertain conditions. So the nonlinear control method, such as backstepping, and sliding mode control, is used against the influence of external disturbance. An excellent control method is the robust adaptive backstepping proposed by Roy, T. K., et al. [4] for controlling the speed of a series DC motor. Finally, it is intelligent methods that overcome uncertainty problems well compared with others. Primarily, it can maintain high performance at a wide range of operations, for example, fuzzy control algorithm [6-8]; RBF neural network and Genetic algorithm [9]; and iterative learning control [10,11].

This paper is the first step in developing an autonomous steering system for a small self-develop electric car using a DC motor. The adaptive PID-type iterative learning control is a proposed method. In which fuzzy logic was used for adjusting the value of control factors (Kp, Kd) corresponding to the amount of the error and change of error. In combination with repeated work of the controller, the static error was minimally reduced, leading to the total system quality improvement. Simulation results were verified on the developed processor-in-the-loop (PIL) simulation based on Arduino Mega 2560 and MATLAB Simulink.

The rest of the paper includes sections: section 2 presents the modeling of the DC motor; section 3 represents the proposed method; section 4 is experiment results and discussions; Finally, conclusions are presented in section 5.

Modeling of DC Motor

Figure 1 presents the principle circuit of the DC motor. Some physical parameters can be defined as follows. J denotes the moment of inertia of the rotor; b stands for viscous motor friction constant. Kb, Kt, and R mean electromotive force constant, motor torque constant, and electric resistance, respectively. Finally, L is the electric inductance.

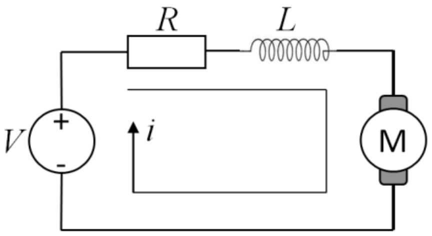


Fig. 1. Principle diagram of DC motor

The electric torque

$$T = K_t * i(\tau) \quad (1)$$

$$e(t) = K_b * \dot{\theta} \quad (2)$$

Assume the torque constant is equal back emf constant

$$K_t = K_e = K \quad (3)$$

Applying Kirchoff’s and Newton’s 2nd law

$$J\ddot{\theta} + b\dot{\theta} = Ki(t) \quad (4)$$

$$U(t) - K\dot{\theta} = i(t) * R + L\frac{di(t)}{dt} \quad (5)$$

Convert equations (4)(5) to Laplace domain

$$s(Js + b) = KI(s) \quad (6)$$

$$(Ls + R)I(s) = V(s) - Ks\Phi(s) \quad (7)$$

From equations (6)(7), the transfer function of speed and voltage

$$\frac{\dot{\Phi}(s)}{V(s)} = \frac{\omega(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (8)$$

Because the position is speed integration, so transfer function of position and voltage has formed as equation (9)

$$\frac{\Phi(s)}{V(s)} = \frac{K}{s[(Js + b)(Ls + R) + K^2]} \quad (9)$$

$$\frac{\Phi(s)}{V(s)} = \frac{0.04612}{0.0004448s^2 + 0.002078s + 0.002142} \quad (10)$$

The parameter estimation toolbox was used to predict parameters J, b, L, R, and K, estimation values as shown in Table 1. The final transfer function was presented in equation 10. And figure 2 describes the processing determination parameter of the DC motor.

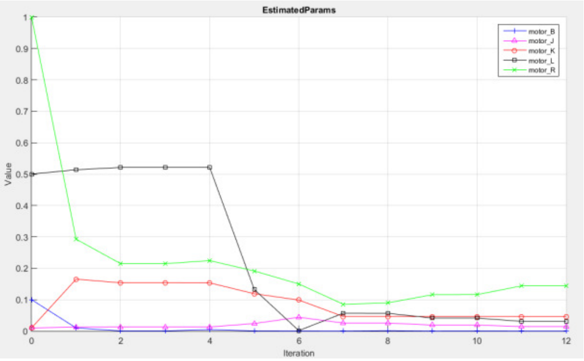


Fig. 2. Results of the estimation of the DC motor parameters

TABLE I. ESTIMATION PARAMETERS OF THE DC MOTOR

Estimation	The symbols				
	J	*b*	*K*	*R*	*L*
Value	0.014393	0.00010516	0.046115	0.14412	0.030905

CONTROL APPROACH

A. *PIL structure*

Figure 3,4 presents the PIL structure and diagram of the connected circuit: the Laptop installed MATLAB&Simulink, Arduino Mega 2560, power bridge circuit L298N, and the DC motor attached encoder.

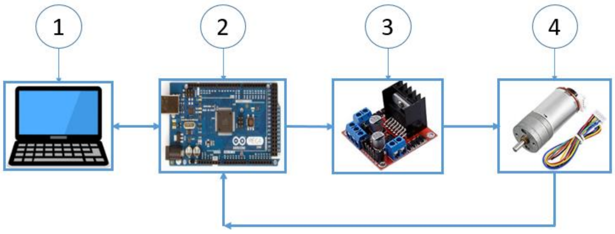


Fig. 3. Block diagram of PIL simulation

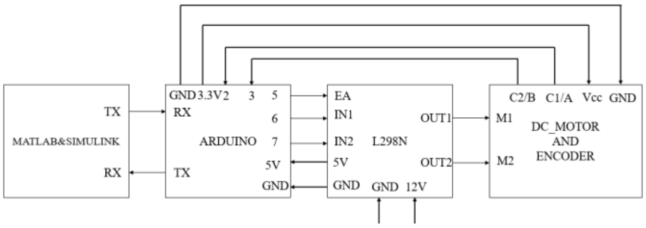


Fig. 4. Diagram of wire connection

The control program was developed in MATLAB&Simulink installed on Laptop. Arduino Mega 2560 is considered a data acquisition device. Communication between Arduino and Simulink is via a USB port. Three control signals sent from pins 5,6,7 to three-pin EA, IN1, and IN2 control motor speed and rotation direction. The encoder sends the motor position signal to Arduino, then compares it with the desired value to determine the error.

B. *Control approaches*

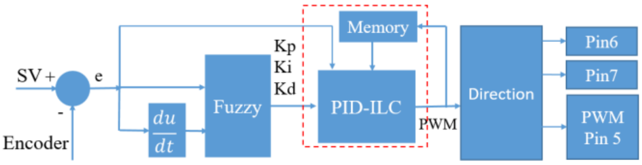


Fig. 5. Block diagram of the proposed control approach

Figure 5 illustrates the block diagram of the developed controller, called adaptive PID-type iterative learning control. Fuzzy logic adjusts PID control gain (Kp, Ki, Kd) corresponding to the amount of the error and change of error. And the block of direction controls DC motor rotation direction via Pin 6, 7. And Pin 5 controls motor speed.

1. *Fuzzy logic controller*

Used fuzzy logic is Sugeno type, which has two input and two output signals, showed figure 6. K1, K2, and K3 are scale factors. In order to fuzzify the input variables, fuzzy sets are used as follows:

e = {NB, NS, Z, PS,PB}

de = { NB, NS, Z, PS,PB}

Kd = { NB, NS, Z, PS,PB }

Kd = { Z, PS,PB}

Linguistic variables have the meaning as follows: negative (N), positive (P), zero (Z), small (S), and big (B). And the membership function and fuzzy rules are presented in figure 7-10, respectively. And figure 11 and 12 present adaptive PID control factor (Kd, Kp), respectively.

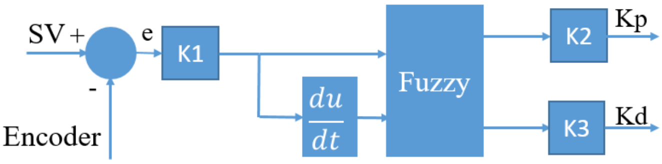


Fig. 6. Block diagram of fuzzy logic controller

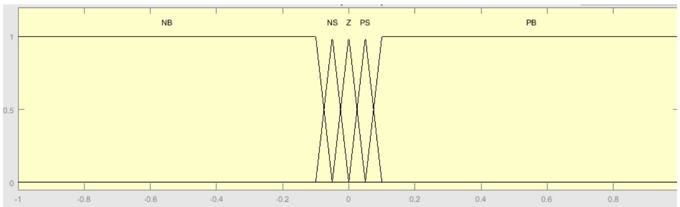


Fig. 7. Membership function of the error

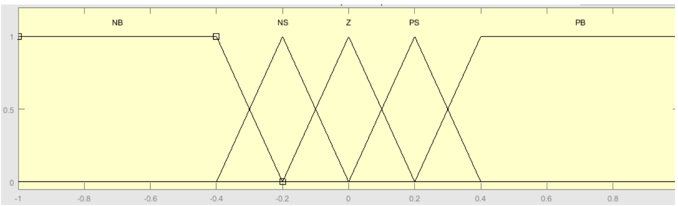


Fig. 8. Membership function of the change of error

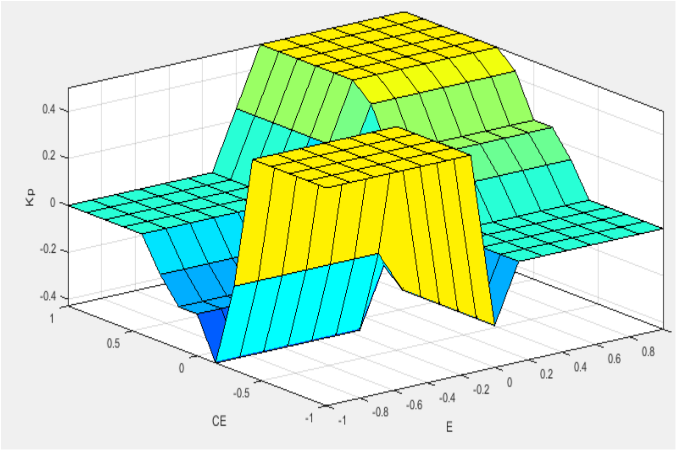


Fig. 9. Surface rules of Kp gain

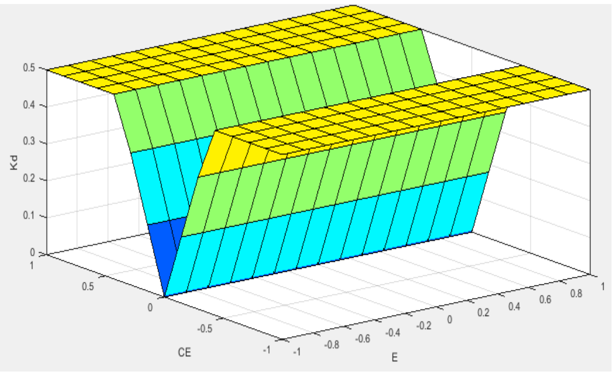


Fig. 10. Surface rules of Kd gain

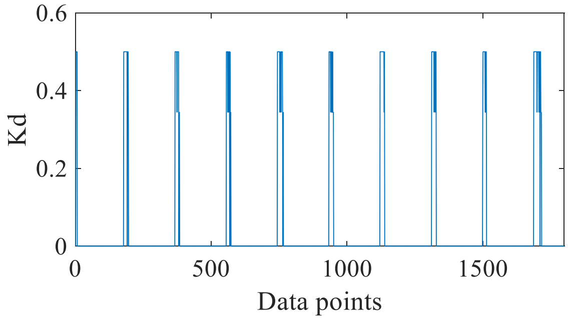


Fig. 11. Change of Kd

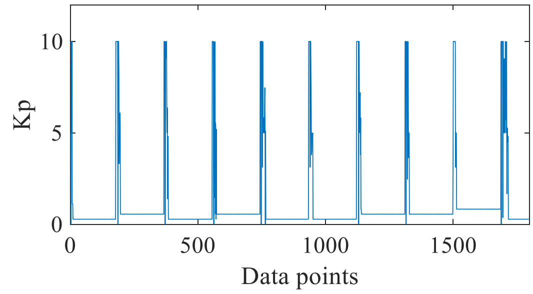


Fig. 12. Change of Kp

2. *PD-type iterative learning control*

Iterative learning control (ILC) operates based on repeated tasks, which was proposed by Arimoto et al. [12], updated law given in equation (11). Where Φ, Γ, ψ are learning gain matrices. ILC structure is presented in figure 13

$$u_{k+1} = u_k + \Phi e_k + \Gamma \dot{e}_k + \Psi \int e_k dt \qquad (11)$$

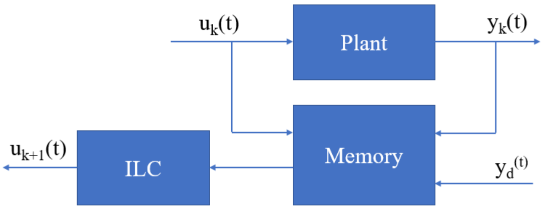


Fig. 13. Structure of the iterative learning control

Processor-In-The-Loop Simulation Results and Discussions

Two scenarios, sinusoidal and square signal, were used for verification of the proposed method., figure 14, 15. PID and the proposed controller gave excellent results, with a well-tracking reality signal and a minor error. In the sinusoidal, the current method is more oscillation than PID. It is caused by the reference signal constantly changing. So control signals generated by ILC, the sum of the current and previous control signals, also constantly change to keep the real value fitting the target value. Whereas, in case the target signal has a square form. The developed controller gave better results than PID. The static error was eliminated. It also demonstrates that ILC is more suitable for eliminating static error at a stable state for tracking control tasks.

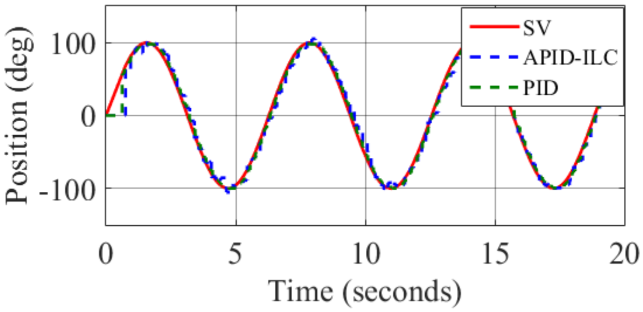


Fig. 14. PIL simulation results for the sinusoidal signal

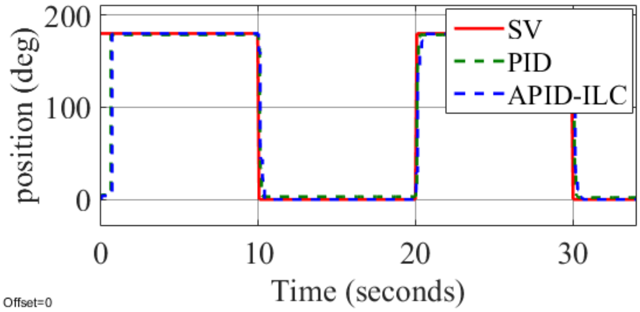


Fig. 15. PIL simulation results for the square signal

Conclusions

This paper proposed adaptive PID-type iterative learning to control DC motor position. The PIL simulation results demonstrated the high quality of the developed controller when compared with the traditional PID. In the future, we will apply the current method for the steering system on the self-developed electric car.

References

[1] De Santiago, Juan, et al. "Electrical motor drivelines in commercial all-electric vehicles: A review." *IEEE Transactions on vehicular technology* 61.2 (2011): 475-484.

[2] Chotai, Janki, and Krupa Narwekar. "Modelling and position control of brushed DC motor." *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*. IEEE, 2017.

[3] Nguyen, Hoa T., et al. "Control algorithms for UAVs: A comprehensive survey." *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems* 7.23 (2020).

[4] Roy, T. K., et al. "Robust adaptive backstepping speed controller design for a series DC motor." *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 2015.

[5] Performance Evaluation of Speed Controller Permanent DC Motor in Electric Bike Using Fuzzy Logic Control System

[6] Zhu, Hongjuan. "Research on PLC DC Motor Speed Control System Based on Quantum Fuzzy Control Algorithm." *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, 2022.

[7] Tir, Zoheir, et al. "Implementation of a fuzzy logic speed controller for a permanent magnet dc motor using a low-cost Arduino platform." *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*. IEEE, 2017.

[8] Rakhmawati, Renny, Farid Dwi Murdianto, and Gamal Tabrani Ilman Syah. "Performance Evaluation of Speed Controller Permanent DC Motor in Electric bike Using Fuzzy Logic Control System." *2018 International Seminar on Application for Technology of Information and Communication*. IEEE, 2018.

[9] Wang, Yingfa, et al. "Adaptive speed control for brushless DC motors based on genetic algorithm and RBF neural network." *2007 IEEE International Conference on Control and Automation*. IEEE, 2007.

[10] Chien, Chiang-Ju, and Kuo-Yung Ma. "Feedback control based sampled-data ilc for repetitive position tracking control of dc motors." *2013 CACS International Automatic Control Conference (CACS)*. IEEE, 2013.

[11] Ibrir, Salim, and Craig Ramlal. "Iterative learning control schemes for a class of nonlinear systems: Theory and real-time implementation." *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE, 2014.

[12] Arimoto, Suguru, Sadao Kawamura, and Fumio Miyazaki. "Bettering operation of robots by learning." *Journal of Robotic systems* 1.2 (1984): 123-140.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 15–20
DOI: 10.15439/2022R51
ISSN 2300-5963 ACSIS, Vol. 33

A Platoon Control Method based on Cooperative Adaptive Cruise Control Vehicles in Traffic Flow

Duc Lich Luu
Faculty of Mechanical Dynamics
Vinh University of Technology Education
Vinh City, Nghe an, Viet Nam.
lanlich@gmail.com

Huu Truyen Pham
Vinh University of Technology Education
Vinh City, Nghe an, Viet Nam.
Phtruyen2007@yahoo.com.vn

Thanh Binh Nguyen
Faculty of Mechanical Dynamics
Vinh University of Technology Education
Vinh City, Nghe an, Viet Nam.
ntbinhspktvinh@gmail.com

Sang Truong Ha
Faculty of Mechanical Engineering
Le Quy Don Technical University
Hanoi, Vietnam.
sanght.st@lqdtu.edu.vn

Ciprian Lupu
Department of Automatic Control and Systems Engineering
University Politehnica of Bucharest
Bucharest, Romania.
ciprian.lupu@upb.ro

***Abstract*—The increasing number of vehicles are one of the causes of accidents, exhaust pollution and traffic congestion in urban areas. These pressing problems force human to looking for the approach for a higher car flow on highways in less time and with fewer accidents. In that context, one of the approach growing roadway capacity is to make a car/truck grouping moving in a strings in order to follow short distances between the members of the platoon. This approach ensures the simultaneous deceleration or acceleration of all cars in the string. This work gives a control strategy for Cooperative adaptive cruise control (Cooperative ACC) in a platoon through a performance evaluation by simulations. The main goal of platoon control is to stay the desired spacing from front cars of the string using the constant time headway (CTH) policy while keeping the same velocity with the other cars. The results of the numerical example have demonstrated the effectiveness and applicability of the proposed approach for cars platoon.**

***Index Terms*—Cooperative Adaptive Cruise Control, Vehicle Following, Spacing Strategies, vehicle platoon, intelligent transportation systems.**

I. Introduction

In fact, the dramatic increase in the global population joined together the continued increase in the number of car/truck on the roads, it has caused a growing problem in the transportation field, related to traffic congestion and increasing pollution. Traffic congestion is often due poor decisions by drivers or the human driver can not look accurately the behavior of vehicles on the road. Traffic jams not only reduces transportation productivity and fuel consumption, but also increases environmental pollution and human health effects. In that context, these pressing problems force human to looking for the solutions. Traffic congestion may be minimized by increasing traffic, which may be done by decreasing the spacing between cars [1], [2].

However, the driver's reaction time requires a large distance between cars to avoid a collision. A way to decrease the distance between cars is employing driving automation in the longitudinal direction,i.e, in order to reach that, the use of vehicle platooning can stay the vehicle distance, there is no accident risk and velocity is not lost, which then lead to be understood such as an increasing in density of vehicles on the road.

The first idea of the vehicle in the platoon, which is given by [3]: A platoon contains multiple vehicles traveling in close proximity, actively coordinating in formation. In this perspective, this concept is very general, it doesn't give too much communication about how cars/trucks in a platoon are connected to each other and how they must move together in a formation. A more specific and suitable definition, which mentioned the Safe road trains (SRT) for the environment platooning [4]: The formation of a car/truck group in a highway situation is called a vehicle platooning, which includes the leader car/truck and all other cars/trucks that follow the leader by keeping a fixed or variable distance between the member vehicles. The follower cars/trucks in the platoon are controlled autonomously, while the car/truck leading in the platoon is controlled manually to stay a safe environment for the human driver và passengers.

Due to the aforementioned benefits of vehicle group, domestic and foreign scholars from the car manufacturer and research institute is activated conducting on developing platooning strategies. A brief introduction to several of projects about vehicle formation is discussed in [5]. Several projects address the vehicle group concept, the most well known of which are: SRT for the environment of Europe [6], California Partners for Advanced Transit and Highways project in United States [7]. The development of platooning strategies depends on many different factors, for example string stability, distance policy, control target, communication systems.

In this paper, vehicle platooning can be thought as an application based on the CACC technology, which is an advance of the ACC system. More details, it is referring to making a group of vehicles to move in the same way (same velocity, acceleration, spacing etc). Vehicles in a platoon use sensors such as radar and V2V communications.

Cooperative ACC vehicle in a platoon is developed based on the adaptive cruise control (ACC) system by adding information data between multiple vehicles to improve reactivity. This information is get from the V2V or I2V communication. DSRCs is employed to communicate information between vehicles. The benefit of this Cooperative ACC system is that data un-measurable by sensors, e.g., the preferred velocity of preceding cars/trucks may be taken into account. The added information of the predecessor car/truck or back of the platoon to be informed about the traffic situation ahead of the platoon leader, instead of only acknowledging directly neighbored cars, i.e When the preceding vehicle is about to reduce speed or increase speed, the information can be immediately passed on to vehicles following in the platoon, then the vehicles following can react more quickly to the changes, so it does not need to wait for sensor data to confirm pressing the brake or accelerator and can thus respond earlier.

Several papers have been published in the topic of ACC system and more recently with Cooperative ACC system. For example, the Adaptive Cruise control system for the string of cars is one of the most important applications in the ADASs which decreases the time allowed on the road, as well as stays small inter-car spacing between cars using only sensors as radar, laser or cameras [8], [9]. Spacing policies used the most commonly to adjust the spacing between cars/trucks, consisting of the CS policy (see [10], [11], etc) and the CTH policy (see [12], citewu2020spacing, etc). A platoon with the CS policy imply that the desired spacing between members in a platoon is fixed and independent of the speed of preceding. While in the CTH policy, the desired spacing is proportional to its speed, and that is called the time headway h_i. The distance in both policies is determined by measuring device mounted on its front bumper to the rear bumper of the preceding car/truck. Some papers show that the authors mainly considered the constant time headway policy using different approaches such as the LQIR method in [13], SMC in [11], MPC in [14]. The operation of the CACC system affects on V2V communication through technologies: the vehicular adhoc networks. In the presence of V2V communication, the information transiting/receiving describes the connectivity of the Cooperative ACC system of cars [15].

Unlike the previous studies [16], the primary contribution of this work gives the design of a Cooperative ACC system for a string of cars that is employed the V2V wireless communication. Only the information with the directly preceding car is considered, with the advantage that if information does not work, the function of standard Adaptive Cruise control will be active. A platoon is to follow the preceding car at a desired spacing that depends on the car speed. The spacing policy relates the desired inter-car spacing via the constant time headway policy. The leader car is presented by the CC system with the desired of the CL system is the desire speed, employing a using a PID controller. The followers car in a platoon gives a spacing control algorithm for the Cooperative ACC system using the constant time headway policy with two controllers, consisting of a feed-back Adaptive Cruise control controller and a feed-forward Cooperative ACC controller. The followers car employ onboard radar mounted on the front bumper of each car to measure the spacing between two consecutive pairs of cars, which are employed in feedback ACC controllers. The preceding car velocity is available through V2V and is employed by a feed-forward Cooperative ACC controller. The delay in the signal being communicated is taken into account in simulation.

The next part of this study is as follows. In Section II establish the vehicle model. In Section III, we describe the CC controller design for the leader car. In Section IV design the control algorithm for the followers car. And then, simulations to confirm the theoretical results is presented in Section V. Finally, Section VI we get the conclusion.

II. Vehicle Model

A platoon of cars is considered as described in Fig.1, and assumption that the operation of each followers installed with the Cooperative ACC system look at only one preceding vehicle with a leader and N followers, indexed $0, 1, ..., N$ with main types of information flow topology is the predecessor following. The Cooperative ACC structure was discussed and is represented in Fig.2.

The car dynamics was calculated starting from the following longitudinal motion equation [17]:

$$m\frac{d\nu(t)}{dt} = F - F_{br} - F_a - F_r - F_g \tag{1}$$

Where, the car body is subjected to longitudinal forces acting on the car, such as Section force for tyre-road F, braking forces at wheels F_{br}, Aerodynamic drag in the longitudinal direction F_a, the rolling resistance coefficient F_r, gradient resistance F_g.

Resistance forces are defined as:

$$\begin{aligned} F_a &= 0.5\rho_\nu A_\nu C_\nu (\nu + \nu_w)^2 \\ F_r &= C_r m g_\nu \cos\gamma_\nu \\ F_g &= m g_\nu \sin\gamma_\nu \end{aligned} \tag{2}$$

$$F_0 = 0.5\rho_\nu A_\nu C_\nu (\nu_0 + \nu_w)^2 + C_r m g_\nu \cos\gamma_0 + m g_\nu \sin\gamma_0 \tag{3}$$

where m is the mass of the car. C_ν, C_r are the aerodynamic drag coefficient, the rolling resistance coefficient, respectively. A_ν is the windward area of the car, ρ_ν is the air density, g_ν is the gravitational acceleration, γ_ν is the road inclination angle, ν is the forward velocity, ν_w is the wind velocity.

The above equation is nonlinear in the forward speed ν, linearization may be made by applying first-order Taylor approximation around the equilibrium point, when $\frac{d\nu(t)}{dt} = 0$. At equilibrium, Eq.1 can be solved for [17]:

where F_0 may be found using assume reasonable values for ν_0, γ_0.

Then the linearized model becomes as follows:

$$\varsigma\dot{\bar{\nu}} + \bar{\nu} = K(u + \chi) \tag{4}$$

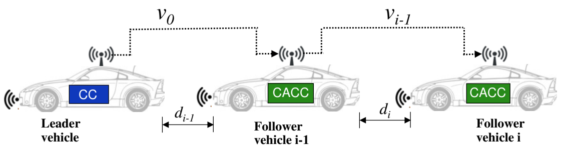


Fig. 1. Structure of vehicle platoon with the CACC system

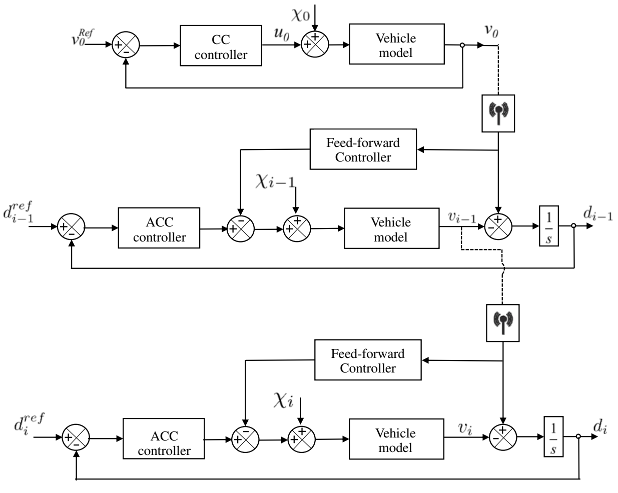


Fig. 2. The control structure of the Cooperative ACC system

where the perturbed variables are determined as, $\bar{\nu} = \nu - \nu_0$, $u = F - F_0$, $u = F - F_0$, $\gamma = \gamma_\nu - \gamma_0$, the parameters ς, K, and χ are determined as follows,

$$\begin{aligned}\varsigma &= m/(\rho_\nu A_\nu C_\nu(\nu + \nu_w)^2)\\ K &= 1/(\rho_\nu A_\nu C_\nu(\nu + \nu_w)^2)\\ \chi &= m g_\nu(C_r \sin\gamma_0 - C_r \cos\gamma_0)\gamma\end{aligned} \tag{5}$$

Using the Laplace transform for Eq.4, The transfer function of the closed-loop system for spacing error between two successive cars pairs is man be represented by:

$$\Xi(s) = \frac{K}{\varsigma s + 1} \tag{6}$$

III. Controller design for leader vehicle

For the CC systems, it builds a more realistic car model by considering the throttle dynamics, i.e adjusts the input of throttle based on a DC motor introduced by Tsujii et al. [18]. Thus, a motor throttle actuator model is presented as:

$$\Xi_a(s) = \frac{K_a}{s(\varsigma_a s + 1)} \tag{7}$$

where the function of the motor-drive duty cycle is the input and the function of the tractive force is the output.

The leader ($i = 0$) is to regulate the car speed so that it follows and stays the desired speed by the driver's command. The reference for the CL system is the desired speed. The controller of the CC system is used a PID controller (as shown in Fig.3) with the transfer function can be written in the form:

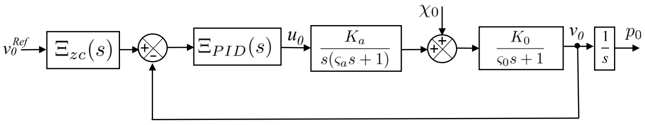


Fig. 3. Car platoon leader control structure with the proportional–integral–derivative controller

$$\Xi_{PID}(s) = K_P + \frac{K_I}{s} + K_D s \tag{8}$$

In this controller, we present an alternative design approach via the pole placement. First, the CL transfer function considering the PID expression Eq.8, the longitudinal car model Eq.6, and actuator model Eq.7 is the following:

$$\Xi_0(s) = \frac{\frac{K_D K_a K_0}{\varsigma_a \varsigma_0} s^2 + \frac{K_P K_a K_0}{\varsigma_a \varsigma_0} s + \frac{K_I K_a K_0}{\varsigma_a \varsigma_0}}{s^4 + \frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} s^3 + \frac{1 + K_D K_a K_0}{\varsigma_a \varsigma_0} s^2 + \frac{K_P K_a K_0}{\varsigma_a \varsigma_0} s + \frac{K_I K_a K_0}{\varsigma_a \varsigma_0}} \tag{9}$$

The controller only have three design variables (K_P, K_I and K_D) while the CL characteristic polynomial has four poles, we may place only three of the four CL poles. Then, the forth pole is defined by the fixed coefficient ($\frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0}$) of the CL characteristic equation.

The remaining three poles are chosen through the overshoot and the time response. The CL characteristic equation then become:

$$\begin{aligned}(s^2 + 0.6s + 0.1)(s + 0.4)(s + \frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} - 1) &=\\ s^4 + \frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} s^3 + (\frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} - 0.66)s^2 &+\\ (0.34\frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} - 0.3)s + 0.04(\frac{\varsigma_a + \varsigma_0}{\varsigma_a \varsigma_0} - 1)&\end{aligned} \tag{10}$$

The PID parameters to achieve these pole assignments determined from Eq.10 are:

$$\begin{aligned}K_P &= \frac{0.34(\varsigma_a + \varsigma_0) - 0.3\varsigma_a \varsigma_0}{K_a K_0}\\ K_I &= \frac{0.04(\varsigma_a + \varsigma_0) - 0.04\varsigma_a \varsigma_0}{K_a K_0}\\ K_D &= \frac{\varsigma_a + \varsigma_0 - 0.66\varsigma_a \varsigma_0 - 1}{K_a K_0}\end{aligned} \tag{11}$$

The CC controller proposes two zeros to the C-L transfer function that growing up the control system overshoot. Two zeros proposed by the PID controller may be compensated by giving two zero-cancellation block as in Fig.3 in the feed-forward path. A transfer function takes the form as below:

$$\Xi_{zc}(s) = \frac{\frac{K_I K_a K_0}{\varsigma_a \varsigma_0}}{\frac{K_D K_a K_0}{\varsigma_a \varsigma_0} s^2 + \frac{K_P K_a K_0}{\varsigma_a \varsigma_0} s + \frac{K_I K_a K_0}{\varsigma_a \varsigma_0}} \tag{12}$$

IV. Cooperative ACC system design for follower vehicle

The block diagram of follower car employing the Cooperative ACC system is illustrated as Fig.4. The main control objective of followers equipped with the Cooperative ACC system is to guarantee maintain the desired spacing to the corresponding preceding car defined by the spacing policy, while maintaining the same velocity as the leader, i.e.

$$\begin{cases} \lim\limits_{t\to\infty} \|\delta_i(t)\| \to 0 \\ \lim\limits_{t\to\infty} \|\nu_i(t)\| \to \nu_{i,ref}(t) \end{cases} \tag{13}$$

Using the CTH policy is getting increasingly popular in a platoon, which is one of the spacing policies used most commonly and it has been mentioned in [2], [19], [20]. Through the CTH policy, the desired inter-car distance between the preceding car and the following car:

$$d_i^{ref}(t) = c_0 + T_d \nu_i \tag{14}$$

where the constant c_0 and T_d are the minimal safe inter-car distance (at zero velocity) and the time gap, respectively, l_{i-1} is the car length.

Define the measured inter-car distance between two consecutive pairs is follows as:

$$d_i(t) = p_{i-1} - (p_i + l_{length}) \tag{15}$$

The spacing errors for the CL system between two consecutive pairs is defined in the form:

$$\delta_i(t) = p_{i-1} - p_i - l_{length} - c_0 - T_d \nu_i \tag{16}$$

The system of the follower car has two controllers, consising of a a feed-back Adaptive Cruise control controller for staying the desired spacing between two consecutive pairs and a a feed-forward Cooperative ACC controller to compensate the effect of the measurable disturbance. The desired spacing is kept by the Adaptive Cruise Control controller. Considering the possibility of V2V failure, the Adaptive Cruise Control controller are used the two integrals of the error since it is assumed that the speed of the lead car that acts as a disturbance to be rejected by the control law can be modeled as as a ramp-type signal input. Thus, the double integrator is integrated in the control algorithm, the followers car may follow the lead vehicle even when it accelerates/decelerates. The need for dual integrator made the employ of a proportional–integral–derivative controller impossible. Considering this aspect, a state-based control algorithm with a double integrator was employed for the Adaptive Cruise Control controller.

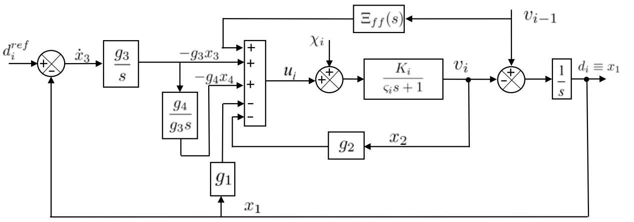


Fig. 4. The control system structure for a follower car

A feedback controller for this system is obtained through the following state equations:

$$\begin{aligned} \dot{x}_1 &= -x_2 + \nu_{i-1}, \\ \dot{x}_2 &= -(1/\varsigma_i)x_2 + (K_i/\varsigma_i)(u_i + \chi_i), \\ \dot{x}_3 &= d_i^{ref} - d_i, \\ \dot{x}_4 &= x_3 \end{aligned} \tag{17}$$

In which, the measured values $x_1 = d_i$ and $x_2 = v_i$; x_3, the integral of the error, $\delta_i = d_i^{ref} - d_i$; and x_4, the double integral of the error. The state based controller yields:

$$u_i = -g_1x_1 - g_2x_2 - g_3x_3 - g_4x_4 = g^T x \tag{18}$$

where, $g^T = -[g_1 \quad g_2 \quad g_3 \quad g_4]$ and $x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T$. The controller gains, g^T may be obtained by a variety of approachs, and a pole-placement design technique is utilized in the MATLAB software.

The feed-back Adaptive Cruise control controller is used to decrease the effect of the measurable disturbances or rejecting the measurable disturbance, and is obtained by taking the inverse of the longitudinal vehicle model Eq.6:

$$\Xi_{ff}(s) = \Xi_i^{-1}(s) \tag{19}$$

the inverse of the transfer function Eq.6 may be approximated due this inverse is not realizable:

$$\Xi_i^{-1}(s) \approx \frac{1 + s\varsigma_i}{K_i(1 + s\varsigma_i/\bar{\chi})} \tag{20}$$

where $\bar{\chi}$ is the frequency range for which the inversion is valid.

The state model Eq.17 of the followers car i is rewritten in the form:

$$\begin{aligned} \dot{X} &= AX + B^T U + GW \\ Y &= CX \end{aligned} \tag{21}$$

where the matrices, vectors, respectively:

$$A = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & -1/\varsigma_i & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}; B = \begin{bmatrix} 0 & K_i/\varsigma_i & 0 & 0 \end{bmatrix}$$

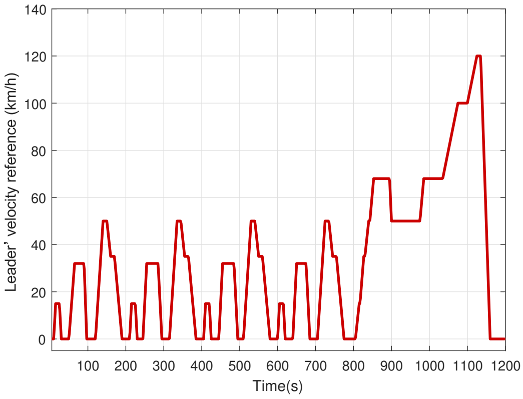


Fig. 5. NEDC (velocity reference)

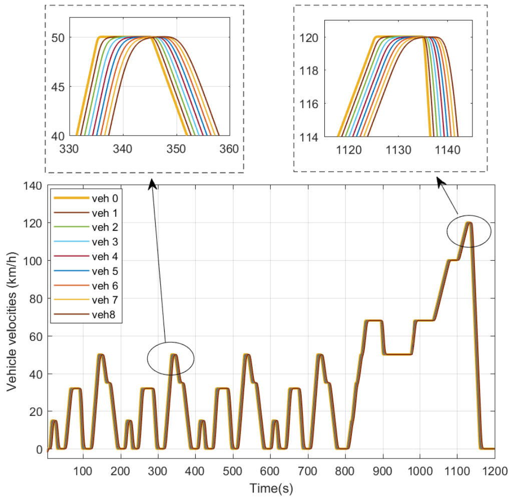


Fig. 6. The velocities of all cars in the string

$$G = \begin{bmatrix} 1 & 0 \\ 0 & K_i/\varsigma_i \\ 0 & 0 \\ 0 & 0 \end{bmatrix}; C = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}; W = \begin{bmatrix} \nu_{i-1} \\ \chi_i \end{bmatrix}$$

Applying Ackermann's formula, the following state feed-back gain matrix can be determined:

$$g^T = [0 \quad 0 \quad 0 \quad 0]R^{-1}P_z(A) \tag{22}$$

Here, $R = [B \quad AB \quad A^2B \quad A^3B]$, $P_z(s) = (s^2{+}2\zeta\omega_n s{+}\omega_n^2)(s+4\omega_n)^2$ are respectively the controllability matrix, the characteristic polynomial of the CL system determine through the desired performances.

For the feed-forward controller, which employes V2V technology in order to have the data about the speed of the preceding car. Normally, the communications between members in a platoon is done at every 100 milli-seconds . From this reason, in the controllers where the delay occurs, its maximum value will be 100 milliseconds.

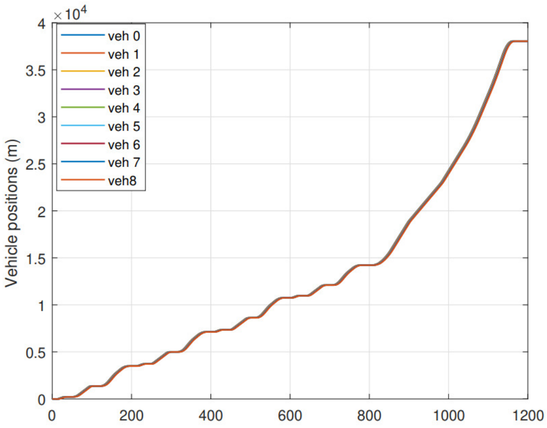


Fig. 7. Vehicle positions of all cars in the string

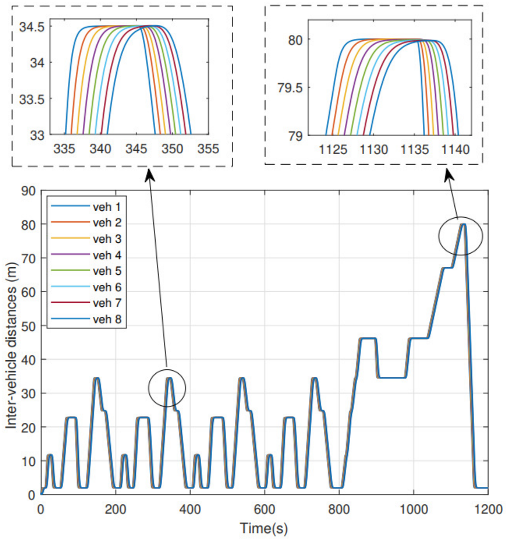


Fig. 8. Distance tracking curve under NEDC condition

V. Numerical Example

In order to achieve the control effect of the Cooperative ACC system for the cars in traffic flow more intuitively, by employing software Matlab/Simulink to evaluate with the last 8 cars followed the first car.

The basic parameters of passenger car model are given as follows (see in [17]):

$$m = 1000kg, C_\nu = 0.5, A_\nu = 1m^2,\ \rho_\nu = 1.202kg/m^3,$$
$$C_r = 0.015, \nu_w = 2m/s, \nu_0 = 20m/sm/s,$$
$$g_\nu = 9.81m/s^2, \gamma_0 = 0, l_{length} = 5m$$

For the leader, the actuator parameters used here are $K_a = 10$ and $\varsigma_a = 0.05s$ [17]. The PID controller's design is done, resulting the tuning parameters $K_P = 28.123$, $K_I = 3.2145$, $K_D = 85.3515$. For the followers, the desired distance is determined according to Eq.14 depending on its vehicle velocity with $c_0 = 2m, T_d = 0.65s$. The ACC controller includes a state feed-back controller with four states, the structure of the ACC controller is employing the pole placement design technique, resulting $g^T = [-3.01 * 10^6 \quad 9 * 10^4 \quad 38.68 * 10^6 \quad 18.439 * 10^7]$. Also, the structure of the feed-forward Cooperative ACC controller was employing Eq.19, Eq.20.

The velocity reference v^{ref} used in this simulation study is the New European Driving Cycle (NEDC) (mentioned in [21]) presented in Fig.5. NEDC cycle consists four urban cycles and one suburban cycle. The average velocity of urban is 19km/h. The average velocity of suburban is 62km/h. The average velocity ofthe whole cycle is 32.12km/ h. The total time are 1200s.

The simulation results of the platoon with proposed controller are indicated as follows:

- In Fig.6, the vehicle velocities together with v^{ref} are illustrated.
- In Fig.7, it is possible to observe the path of the vehicles according to their position along with the total distance traveled.
- Fig.8 depictes the inter-car distances with zooms in two specific areas.

For more details, the velocity signals of each car are described in Fig.6. It may be observed that every car is accelerating and decelerating in order to follow the imposed speed profile.

In Fig.7, the positions of the nine vehicles are at the steady-state, at the same time the collision was avoided.

In Fig.8, the spacing between two consecutive pairs, after a velocity dependent reference is used the input for the control systems are depicted. The desired car spacing determined by fixed time headway will be larger. It may be seen that in this work the spacing wave forms follow the velocity profile applies as reference for the leader.

Specifically, when the vehicle's speed increases, the spacing between two consecutive pairs also increases. This proportionality property is also true when reducing the velocity. It stays excellent spacing tracking and makes the actual car spacing near the expected value at high velocity for better economic performance.

VI. Conclusions

In this study, the platoon control method based on cooperative adaptive cruise control vehicles in traffic flow was presented. The vehilce platoon includes one leader and the followers, in which the leader that is controlled through the Cruise control system since in this study the velocity reference is set up by the driver, the followers in a platoon integrate the feed-back ACC controller and the feed-forward Cooperative ACC controller. This method allows to stay the desired spacing from front cars of the platoon though the CTH policy while keeping the same velocity with the other vehicles.

From the result of the numerical example, the obtained responses depicte that the behavior of the cars in a string is the desired one, i.e. the velocities and the inter-car spacing follow to keep at the reference values.

Acknowledgment

This paper was partially supported by Vinh University of Technology Education. The authors would like to thank the support of this institution.

References

[1] C. Wu, Z. Xu, Y. Liu, C. Fu, K. Li, and M. Hu, "Spacing policies for adaptive cruise control: A survey," *IEEE Access*, vol. 8, pp. 50 149–50 162, 2020.

[2] D. L. Luu and C. Lupu, "Vehicle string using spacing strategies for cooperative adaptive cruise control system," *UPB Sci. Bull., Series C*, vol. 83, no. 1, pp. 91–106, 2021.

[3] S. E. Shladover, "Operation of automated guideway transit vehicles in dynamically reconfigured trains and platoons (extended summary)," Tech. Rep., 1979.

[4] C. Bergenhem, Q. Huang, A. Benmimoun, and T. Robinson, "Challenges of platooning on public motorways," in *17th world congress on intelligent transport systems*, 2010, pp. 1–12.

[5] B. Kuhn, M. Lukuc, M. Poorsartep, J. Wagner, K. N. Balke, D. Middleton, P. Songchitruksa, N. Wood, M. Moran *et al.*, "Commercial truck platooning demonstration in texas–level 2 automation." Texas. Dept. of Transportation. Research and Technology Implementation Office, Tech. Rep., 2017.

[6] A. Davila, E. del Pozo, E. Aramburu, and A. Freixas, "Environmental benefits of vehicle platooning," SAE Technical Paper, Tech. Rep., 2013.

[7] S. E. Shladover, "Path at 20—history and major milestones," *IEEE Transactions on intelligent transportation systems*, vol. 8, no. 4, pp. 584–592, 2007.

[8] G. N. Bifulco, L. Pariota, F. Simonelli, and R. Di Pace, "Development and testing of a fully adaptive cruise control system," *Transportation Research Part C: Emerging Technologies*, vol. 29, pp. 156–170, 2013.

[9] A. Tigadi, R. Gujanatti, A. Gonchi, and B. Klemsscet, "Advanced driver assistance systems," *International Journal of Engineering Research and General Science*, vol. 4, no. 3, pp. 151–158, 2016.

[10] S. Darbha, S. Konduri, and P. R. Pagilla, "Vehicle platooning with constant spacing strategies and multiple vehicle look ahead information," *IET Intelligent Transport Systems*, vol. 14, no. 6, pp. 589–600, 2020.

[11] G. Guo and D. Li, "Adaptive sliding mode control of vehicular platoons with prescribed tracking performance," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7511–7520, 2019.

[12] P. A. Ioannou and C.-C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular technology*, vol. 42, no. 4, pp. 657–672, 1993.

[13] Y. Jiang, "Modeling and simulation of adaptive cruise control system," *arXiv preprint arXiv:2008.02103*, 2020.

[14] T. Takahama and D. Akasaka, "Model predictive control approach to design practical adaptive cruise control for traffic jam," *International Journal of Automotive Engineering*, vol. 9, no. 3, pp. 99–104, 2018.

[15] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.

[16] D. L. Luu, C. Lupu, and T. Van Nguyen, "Design and simulation implementation for adaptive cruise control systems of vehicles," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*. IEEE, 2019, pp. 1–6.

[17] A. G. Ulsoy, H. Peng, and M. Çakmakci, *Automotive control systems*. Cambridge University Press, 2012.

[18] M. Tsujii, H. Takeuchi, K. Oda, and M. Ohba, "Application of self-tuning to automotive cruise control," in *1990 American Control Conference*. IEEE, 1990, pp. 1843–1848.

[19] G. Ma, B. Wang, and S. S. Ge, "Robust optimal control of connected and automated vehicle platoons through improved particle swarm optimization," *Transportation Research Part C: Emerging Technologies*, vol. 135, p. 103488, 2022.

[20] V. K. Vegamoor, S. Darbha, and K. R. Rajagopal, "A review of automatic vehicle following systems," *Journal of the Indian Institute of Science*, vol. 99, no. 4, pp. 567–587, 2019.

[21] C. Pan, A. Huang, J. Wang, L. Chen, J. Liang, W. Zhou, L. Wang, and J. Yang, "Energy-optimal adaptive cruise control strategy for electric vehicles based on model predictive control," *Energy*, vol. 241, p. 122793, 2022.

DOI: 10.15439/2022R42
ISSN 2300-5963 ACSIS, Vol. 33

A DNA Steganography Algorithm Based on The DNA-XOR Technique

Vinh-Quy Nguyen
Hung Yen University
of Technology and Education
Hungyen, Vietnam
vinhquynguyen@gmail.com

Dinh-Chien Nguyen*
Hung Yen University
of Technology and Education
Hungyen, Vietnam
https://orcid.org/0000-0002-4939-6355

Viet-Hung Dang
Hung Yen University
of Technology and Education
Hungyen, Vietnam
dangviethung1107@gmail.com

Thanh-Hue Nguyen Thi
Hung Yen University
of Technology and Education
Hungyen, Vietnam
huentt1509@gmail.com

Thu-Hang Phan Thi
Tan Lap junior high school
Hungyen, Vietnam
phanthuhang.tk36@gmail.com

Dinh-Thinh Luu
Doan Thi Diem junior high school
Hungyen, Vietnam
luuthinh@yenmy.edu.vn

***Abstract*—Steganography is the technique of concealing secret data in a physical object, such as a database, video, image, audio, QR code, and DNA sequence…. The DNA sequence is also considered for data hiding problems with the strong development of Bioinformatics. Many proposed algorithms to conceal data in DNA and RNA sequences, but they could not increase the amount of hidden data. In this study, we propose a DNA steganography algorithm for improving the embedded capacity in the DNA sequence. The algorithm uses the DNA-XOR technique based on the XOR operation. The secret data are encoded to a DNA sequence and then matched with the original DNA sequence by DNA-XOR operation to make a new DNA sequence. By this algorithm, we can embed two bits for each nucleotide. Moreover, with many kinds of combinations of nucleotides, the algorithm shows that the proposed method also improves the security of hidden data.**

***Index Terms*—Steganography, DNA sequence, XOR operation, DNA-XOR.**

I. Introduction

Steganography is the technique of concealing a secret data within a message or physical object such as database, image, audio, video, etc. [1-6]. Nowadays, many researchers studied data hiding methods for digital images [1-3].

Biological computing, including Bioinformatics, has supported researchers exploit techniques, which hide secret data in DNA (Deoxyribose nucleic acid) sequences, RNA (Ribonucleic acid) sequences, and protein structures.

In 1953, two famous genetic scientists, Watson and Crick found the DNA sequence structure [7]. In the research, DNA is the genetic material in each organism, consisting of humans, animals, and plants. DNA data is stored in a computer system with four nucleotide bases consist of Adenine (A), Guanine (G), Cytosine (C) và Thymine (T). Nucleotides combined into a DNA sequence by rule A pair with T and C pairs with G.

In 2012, Church et al. [8] proposed a study by using DNA sequence to store data, based on DNA synthesis and sequencing in Next Generation DNA Sequencing (NGS) by using DNA microchips. With these discoveries, concealing data in a DNA microchip will be fairly rewarding for cryptographers. However, with current technologies, DNA microchips are still quite expensive, the scientists have implemented hiding data in DNA sequences, which are taken from the website of the Information Technology Center in National Center for Biotechnology Information (NCBI) [9] or the website of the European Bioinformatics Institute in Ensembl (EMBL-EBI) [10]. In 2002, Shimanovsky et al. [11] presented two methods for hiding data in DNA and RNA. The first technique allows embedding the secret data in a non-coding DNA sequence, which is a DNA sequence that has not been transcripted and transferred to the genetic sequence. The second technique is used by scientists to organize data in active coding without converting it to the Amino Acid sequence. The techniques in this study can be used in the protection of intellectual property rights in biotechnology. Based on the analysis of Shimanovsky, many researchers have studied hiding secret data in DNA sequences [12-18], especially in non-coding regions.

In 2019, AI-Harbi et al. [19] presented some techniques for DNA-based steganography in security analysis. The study showed the advantages and disadvantages of a few methods, then suggested ways to improve the techniques for the DNA hiding field. Singh and Sharam [20] provided a review of nineteen existing algorithms of DNA-based cryptography.

To improve the embedded capacity, this study proposes an algorithm for data hiding in DNA sequences by using DNA-XOR operation. The rest of our work deals with the related studies, the proposed algorithm, and the implementation results. Section II introduces the methods of hiding information in the study of Shiu et al. [12]. Improving the data hiding algorithm with a higher capacity of embedding data by substitution method will be presented in the third section. Section IV shows the implementation results of the study. Conclusions and future studies will be shown in Section V.

* — corresponding author

II. Related work

In 2010, Shiu et al. [12] proposed three methods for concealing data based on the properties of DNA sequences.

The first method, the Insertion method, was performed by coding a DNA sequence to a binary string by the rule A - 00, C - 01, G -10, T - 11, then dividing the binary string into segments with size k bits. One secret data bit is inserted at the beginning of each segment, then merged all segments into a new binary string, and decoded into a pseudo-DNA sequence. On the receiver's side, the hiding data will be extracted at the beginning segments by the same rule, and then merge into a binary string of hiding data.

The second method, Complementary Pair, uses dynamic programming to find the longest complimentary substring, and then uses complementary rule, ((AC)(CG)(GT)(TA)). Each complementary substring is placed between the same nucleotide, for example, TACGT. The secret data will be embedded before the complementary substring, by the rules 00-A, 01-C, 10-G, and 11-T. For instance, with the DNA sequence S=ACGTAGCTGTTCTGTTCTCCTTCATGGAT and the secret data m='10110100'='GTCA', we can find a complementary substring (bold characters), and then insert a nucleotide before each complementary substring. The embedded DNA sequence is S'=ACGGTAGCTGTTTCT-GTTCCTCCTTCAATGGAT. The hidden data will be extracted by the same algorithm with the embedding process, finding a complementary substring and then extracting the nucleotide before the substring. Finally, the binary string will be decoded from extracted nucleotides.

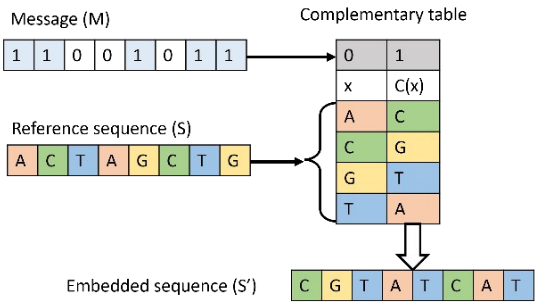


Fig. 1. Embedding process in the study [12]

The third method, the Substitution Method, employed the Complementary rule (x, C(x)). If the secret bit is 0, skip nucleotide, otherwise, alter the nucleotide x to nucleotide C(x), as in "Fig. 1".

On the receiver side, the DNA sequence, which brings hidden data (fake), is compared with the original sequence (Orig). If a nucleotide in fake is equal to a nucleotide in Orig, we get hidden bit 0, otherwise, the hidden bit is 1 "Fig. 2".

The methods did not improve the embedded capacity. So, in this study, we introduce a data hiding algorithm to improve the capacity, but still against massive attacks, in the following section.

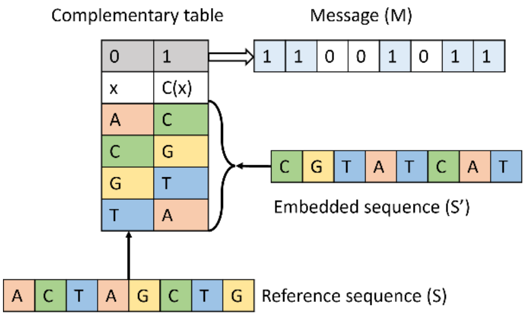


Fig. 2. Extraction process in the study [12]

III. Proposed Steganography algorithm

A. Embedding process

This study improves the embedding capacity by using DNA-XOR technique. We consider a nucleotide is presented by two bits, for example A - 00, C - 01, G – 10, and T -11. Based on XOR operation between two binary bits, we can build a table of DNA-XOR (Table 1). With this idea, we can embed two secret bits for each nucleotide, so that, the proposed algorithm can take more capacity than Shiu's [12] study.

Table 1. DNA-XOR Table

	A	C	G	T
A	A	C	G	T
C	C	A	T	G
G	G	T	A	C
T	T	G	C	A

To embed message into a DNA Sequence, we can perform by following process,

Step 1: Encrypt Message to DNAseq by the rules, A - 00, C - 01, G – 10, and T -11

Step 2: Embed DNAseq into RefSeq using DNA-XOR table in Table 1

Fig. 3 shows the Embedding phase of the proposed algorithm.

Message may be a text or other format, so it must be encrypt to DNA sequence (Fake). To easy known, we assume that the message is in binary string.

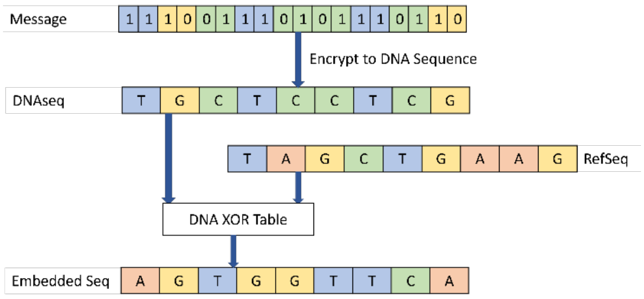


Fig. 3. The embedding process

B. Extraction process

To extract the hidden data,

- Firstly, we match the Embedded Seq with the original DNA sequence (RefSeg) by using the DNA-XOR table (Table 1), and get the Extracted Seg.
- Then, decrypt the Extracted Seg to bynary seqence (Message)

Fig. 4 presents Extraction process of this algorithm.

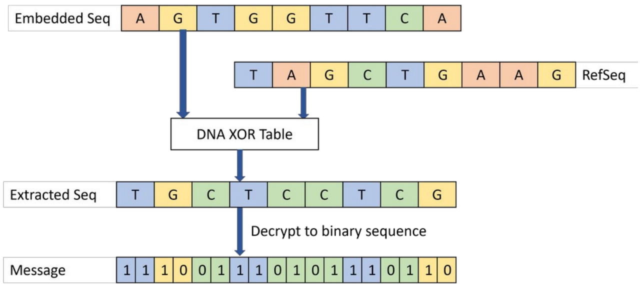


Fig. 4. Extraction process

IV. Experimental results

In this study, we implemented the data hiding algorithm by Matlab programming language on Core i5 Computer with 8GB RAM. DNA sequences database was downloaded from the NCBI website [9].

Compared with Shiu et al.'s algorithm, the proposed algorithm improved the embedded capacity of the secret data hidden in DNA sequences. (Table 2)

Table 2. Comparing the embedded capacity of the proposed algorithm with an algorithm in [12]

DNA Sequence	**Number of nucleotide s**	**Shiu [12] algorithm**	**Proposed algorithm**
AC153526	200117	200117	400234
AC166252	149884	149884	299768
AC167221	204841	204841	409682
AC168874	206488	206488	412976
AC168901	191456	191456	382912
AC168907	194226	194226	388452
AC168908	218028	218028	436056
Average	***195005***	***195005***	***390010***

Fig. 5 shows the difference of nucleotide between the proposed method, presents by the yellow bars, and Shiu's method [12], shows in the green bars, with the Refence seqence. Although hiding data of the proposed method significantly changes the value of each nucleotide in the DNA sequence, the difference in nucleotides of both methods is negligible compared to the reference DNA sequence. For example, after embedding data in the DNA sequence with code AC166252, the number of Nucleotides A, C, G, and T are 39000, 34700, 35200, and 40100, respectively. The results show that the proposed algorithm can balance with the original DNA sequence, therefore, detecting data hiding in the reference DNA sequence will be more difficult.

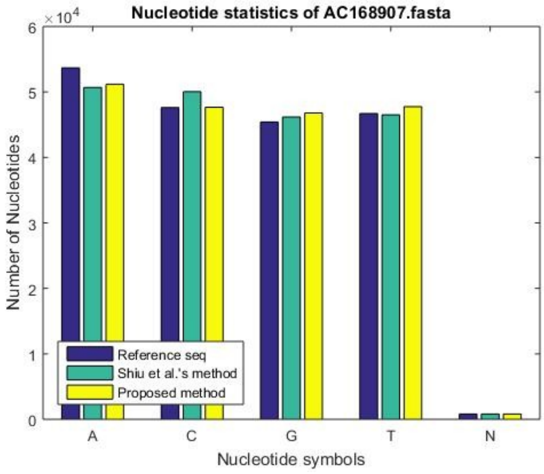


Fig. 5. The difference of nucleotide

Because we use the DNA-XOR table, the data extraction will be more difficult for the people who want to obtain the secret data. So, the proposed algorithm also improves the security of the hidden data in the DNA sequence. This characteristic is an important condition in data hiding problems

V. Conclusion

There are many data hiding algorithms in DNA sequences, but most of these algorithms either increase the size of the reference DNA sequence or can not improve the embedded capacity. The proposed algorithm can improve the capacity of embedding data, which hide in DNA sequences. For each nucleotide, the algorithm can embed two data bits. With a character (8 bits) in the text file, we use four nucleotides, so we can embed a 1MB text file into a DNA sequence of 4MB size. Moreover, the algorithm also improves the security for anyone who wants to get secrete data.

In 2020, Tabalabaei et al. proposed a new technique for recording information in DNA backbone [21]. This techniqe can change the storage devices technology, and helps researchers more improve DNA steganography methods.

References

[1] T. S. Nguyen, C. C. Chang, M. C. Lin, "Adaptive lossless data-hiding and compression scheme for SMVQ indices using SOC," Smart Comput. Review, vol. 4, nno. 3, pp. 230-245, 2014.

[2] J. Mielikainen, "LSB matching revisited," IEEE Signal Process. Letts., vol. 13, pp. 285–287, 2006.

[3] C. C. Chang, T. S. Nguyen, "A reversible data hiding scheme for SMVQ indices," Informatica, vol. 25, no. 4, pp. 523-540, 2014.

[4] C. V. Nguyen, D. Tay, and G. Deng, "A fast watermarking system for H.264/AVC video," in Proc. IEEE APCCAS, Dec. 2006, pp. 81–84.

[5] M. Fallahpour, M. David, "Reversible data hiding based on H. 264/AVC Intra prediction." Digital Watermarking. Springer Berlin Heidelberg, pp. 52-60, 2008.

[6] Chien. N. D, Son N. T, & Hsu F. R, “An algorithm for DNA sequence hiding in H. 264/AVC video.” In Proceedings of the Seventh Symposium on Information and Communication Technology ACM, pp. 229-234, December 2016.

[7] J.D. Watson, F.H.C. Crick, “Molecular structure of Nucleic acids: A structure for deoxyribose nucleic acid,” Nature 171 (1953), pp. 737, 738.

[8] Church, G. M., Gao, Y., & Kosuri, S., “Next-generation digital information storage in DNA,” Science, 337(6102), 1628-1628, 2012.

[9] National Center for Biotechnology Information, https://www.ncbi.nlm.nih.gov/

[10] Ensembl, http://www.ensembl.org/downloads.html

[11] Shimanovsky, B., Feng, J., & Potkonjak, M., “Hiding data in DNA” International Workshop on Information Hiding (pp. 373-386). Springer Berlin Heidelberg, 2002.

[12] Shiu, H. J., Ng, K. L., Fang, J. F., Lee, R. C., & Huang, C. H., “Data hiding methods based upon DNA sequences,” Information Sciences, 180(11), 2196-2208, 2010.

[13] Haughton, D., & Balado, F., “BioCode: Two biologically compatible Algorithms for embedding data in non-coding and coding regions of DNA,” BMC bioinformatics, 14(1), 121, 2013.

[14] Wang, Z., Zhao, X., Wang, H., & Cui, G., “Information hiding based on DNA steganography,” Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on (pp. 946-949). IEEE, 2013.

[15] Najaftorkaman, M., & Kazazi, N. S., “A method to encrypt information with DNA-based cryptography,” International Journal of Cyber-Security and Digital Forensics, 4(3), 417-427, 2015.

[16] UbaidurRahman, N. H., Balamurugan, C., & Mariappan, R., “A novel string matrix data structure for DNA encoding algorithm,” Procedia Computer Science, 46, 820-832, 2015.

[17] Huang, Y. H., Chang, C. C., & Wu, C. Y., “A DNA-based data hiding technique with low modification rates,” Multimedia tools and applications, 70(3), 1439-1451, 2014.

[18] Liu, H., Lin, D., & Kadir, A., “A novel data hiding method based on deoxyribonucleic acid coding,” Computers & Electrical Engineering, 39(4), 1164-1173, 2013.

[19] Al-Harbi, O. A., Alahmadi, W. E., & Aljahdali, A. O. “Security analysis of DNA based steganography techniques.” SN Applied Sciences, 2(2), 1-10, 2020.

[20] S. Singh and Y. Sharma, "A Review on DNA-based Cryptography for Data hiding," 2019 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, Tamilnadu, India, pp. 282-285, 2019.

[21] Tabatabaei, S. K., Wang, B., Athreya, N. B. M., Enghiad, B., Hernandez, A. G., Fields, C. J., ... & Milenkovic, O. (2020). DNA Punch Cards: Storing Data on Native DNA Sequences via Nicking. BioRxiv, 672394.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 25–27
DOI: 10.15439/2022R46
ISSN 2300-5963 ACSIS, Vol. 33

Navigation for drones in GPS-Denied environments based on Vision processing

Nguyen Duy Ninh
Faculty of Electrical and Electronic
Institute of Military science and Technology
Hanoi, Vietnam
duyninh.vdt@gmail.com

Tran Thi Van
General education University
of Labour and Social Affairs
Hanoi, Vietnam
tranvantk4@gmail.com

Khoa Nguyen Dang*
International School,
Vietnam National University
Hanoi, Vietnam
khoand@vnuis.edu.vn

Bui Thanh Tung
International School,
Vietnam National University
Hanoi, Vietnam
buithanhtung@vnu.edu.vn

***Abstract*—This paper presents a method for the navigation of drones in GPS-denied Environments based on vision processing. A camera is attached to the drone to fly in full conditions like GPS, video streaming, control waypoint, etc. Then, some information such as the image features and location of the drone is extracted and stored, which is used for the next flight with GPS-denied Environments. The results are shown that the drone positions could estimate with high accuracy.**

***Index Terms*—UAV, GPS-Denied, vision**

I. Introduction

Unmanned aerial vehicles (UAVs) have known as useful robotics and have many applications such as in the military [1], agriculture [2], transportation [3], etc. To effectuate these tasks, UAVs must be developed and tested many times under different conditions. For example, the structure designing process needs to be evaluated by the NX software [4], and the Heritage Building Information Modeling [5]. Besides, the flight control system (FCS) is also considered in detail with the hardware layer and software (algorithm) layer.

Some hardware was suggested as DJI [6], pixhawk [7], etc. They brought useful values to the UAV development community. For testing software, some researchers used Matlab Simulink to present their UAV model as fixed-wing [8], quadrotor [9], and hybrid [10, 11] with the control algorithms as PID [12], Fuzzy-PID [13], Neural network [14], and sliding mode control [15]. Recently, software in the loop simulation (SILS) is developed to present more details in simulation [16]. All demonstrations showed well results and high performance.

Although UAVs have spent many years of development and application, the approach is to control UAVs following the trajectory and waypoint of the Global Positioning System (GPS). The sensors such as Inertial Measurement Unit (IMU) and GPS are very important for all operations of flight which measures the attitude and location feedback of all controllers.

Normally, the IMU used the estimations and computed the attitude information in local devices. While GPS needs the satellite which is dependent on the service provider. Therefore, GPS could no stable, lost, and denied. This paper presents to detect and computed the location of UAVs in the GPS- Denied conditions using vision processing. Phantom 4 DJI Quad-rotor is selected for demonstrating our algorithms.

II. Experiment setup

In this experiment, the Phantom 4 DJI is used with the parameters in Table. 1.

TABLE I. Phantom 4 DJI specifications

Name	**Specifications**
Weight	1.38kg
Max speed	20 m/s
Max flight time	28 mins
Satellite Positioning Systems	GPS/GLONASS
Max Service Ceiling Above Sea Level	19685 feet (6000 m)
Camera	1/2.3” CMOS, Effective pixels:12.4 M, FOV 94° 20 mm (35 mm format equivalent) f/2.8

For making the dataset, the UAV is controlled in manual mode. The video capture and mapping are performed in Hoa Binh Park, Bac Tu Liem, Ha Noi, Vietnam. Herein, the mapping is formatted in the tif and geotagged tif file which are tagged locations via the Drone Deploy on PC [17]. The parameter setup of all flights are defined in Table. 2.

TABLE II. Parameter setup for all flights

Devices/Steps	**Parameters name**	**Details**
CPU	Operating system	Ubuntu 20.04
	CPU	Intel i5-7500
	RAM	8 GB
Video capture	Resolution	720x1280
	FPS	29
	Times	2m00s (120 s)
	GPS frequency	10 Hz
Mapping	Resolution	9844x9265

III. Detecting GPS location algorithms

Based on the mapping containing the image infomation and GPS-tagged locations, this paper uses the scale-invariant feature transform (SIFT) [18, 19] to match the current vision image with the dataset which is recorded in previous flights. Therefore, the current location of the drone could be computed and given to the drone for safetying flight with no GPS conditions.

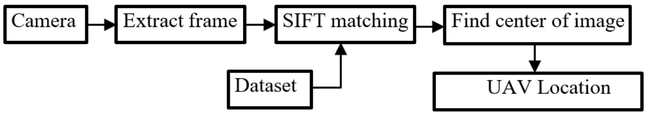


Fig. 1. Find UAV location algorithms

Fig. 1 shows details for finding location of UAV. Firstly, the frame is extracted from camera. They are used to match with all images from dataset which includes location of UAV in last flight by using SIFT algorithm. If the number of feature matching is larger than a threshold values (15 features), the algorithm will select this image for finding the center of image and it could present to UAV location because all images are capture from UAV and UAV is a center of viewing as our configurations.

IV. Results and discussion

Using the algorithm in Sect. 3, the UAV flight is performed and the results are shown in Fig. 2 and Table. 3 and 4.

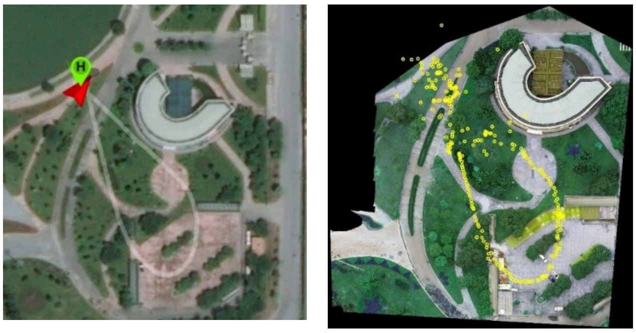


Fig. 2. Comparation between previous flight and the current flight

As the description in Sec. 2, a PC is used to detect the location of the UAV. The specification is CPU Intel i5-7500, 8 GB RAM, and OS Ubuntu 20.04. To improve the speed of algorithms, the map in the dataset will be re-scale to create a small size file and it is also applied to the images in the current flights. Let's define these parameters as map_scale and uav_scale.

In this experiment, the map_scale and uav_scale are adjusted to check the performance of the algorithms. Table. 3 and 4 present the different output in each case.

TABLE III. Result with diffrent map_scale

No.	map_scale	uav_scale	Matching %	Error locations (m)	Time processing (s)
1	5%	25%	70.27%	3.72 m	49.45
2	10%	25%	74.16%	3.04 m	112.54
3	20%	25%	69.16%	2.73 m	353.16
4	33.33%	25%	61.11%	2.91 m	925.7

As the results in Table. 3, the adjusted map_scale is not bearing to ratio matching, but the accuracy of location estimation could be improved. The large scaling needs a long time for processing. In Table. 4, the uav_scale is really affected to the matching. It leads to the performance of the finding the location UAV. To integrate on the UAV, the Nvidia Xavier board will be used and all scales are considered again in the future works.

TABLE IV. Result with diffrent map_scale

No.	map_scale	uav_scale	Matching %	Error locations (m)	Time processing (s)
1	10%	12.5%	22.22%	3.07 m	86.73
2	10%	25%	74.16%	3.04 m	112.54
3	10%	50%	96.38%	2.26 m	190.84
4	10%	75%	98.33%	2.72 m	324.65

V. Conclusion

This paper presents an experiment related to estimating the location of UAV in GPS-Denied environment using vision processing which could help navigation for the UAV. Based on Phantom 4 DJI, the mapping and location are tagged and store in a dataset which is used to match with the current frame from camera on UAV for next flight in no GPS condition. The matching process is applied by SIFT algorithm. All results are shown that the proposal algorithm could compute the location and help navigation for UAV.

References

[1] A. Konert and T. Balcerzak, "Military autonomous drones (UAVs) - from fantasy to reality. Legal and Ethical implications," Transportation Research Procedia, vol. 59, pp. 292-299, 2021/01/01/ 2021.

[2] M. Yaqot and B. C. Menezes, "Unmanned Aerial Vehicle (UAV) in Precision Agriculture: Business Information Technology Towards Farming as a Service," in 2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA), 2021, pp. 1-7.

[3] A. Gupta, T. Afrin, E. Scully, and N. Yodo, "Advances of UAVs toward Future Transportation: The State-of-the-Art, Challenges, and Opportunities," Future Transportation, vol. 1, no. 2, pp. 326-350. doi: 10.3390/futuretransp1020019

[4] A. Kurnyta, W. Zielinski, P. Reymer, K. Dragan, and M. Dziendzikowski, "Numerical and Experimental UAV Structure Investigation by Pre-Flight Load Test," (in eng), Sensors (Basel), vol. 20, no. 11, May 26 2020.

[5] [5] T. S. Rachmawati and S. Kim, "Unmanned Aerial Vehicles (UAV) Integration with Digital Technologies toward Construction 4.0: A Systematic Literature Review," Sustainability, vol. 14, no. 9. doi: 10.3390/su14095708

[6] M. Stanković, M. M. Mirza, and U. Karabiyik, "UAV Forensics: DJI Mini 2 Case Study," Drones, vol. 5, no. 2. doi: 10.3390/drones5020049

[7] N. Fariñas-Álvarez, F. Navarro-Medina, and H. González-Jorge, "Metrological Validation of Pixhawk Autopilot Magnetometers in Helmholtz Cage," World Electric Vehicle Journal, vol. 13, no. 5. doi: 10.3390/wevj13050085

[8] J. Moore, A. Fein, and W. Setzler, "Design and Analysis of a Fixed-Wing Unmanned Aerial-Aquatic Vehicle," in 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 1236-1243.

[9] M. Idrissi, M. Salami, and F. Annaz, "A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms," Journal of Intelligent & Robotic Systems, vol. 104, no. 2, p. 22, 2022/01/22 2022.

[10] N. Sethi and S. Ahlawat, "Low-fidelity design optimization and development of a VTOL swarm UAV with an open-source framework," Array, vol. 14, p. 100183, 2022/07/01/ 2022.

[11] K. D. Nguyen, C. Ha, and J. T. Jang, "Development of a New Hybrid Drone and Software-in-the-Loop Simulation Using PX4 Code," in Intelligent Computing Theories and Application, Cham, 2018, pp. 84-93: Springer International Publishing.

[12] A. Sheibani and M. A. Pourmina, "Simulation and Analysis of the Stability of a PID Controller for Operation of Unmanned Aerial Vehicles," in Mechanical Engineering and Technology, Berlin, Heidelberg, 2012, pp. 757-765: Springer Berlin Heidelberg.

[13] F. A. A. Andrade et al., "Unmanned Aerial Vehicles Motion Control with Fuzzy Tuning of Cascaded-PID Gains," Machines, vol. 10, no. 1. doi: 10.3390/machines10010012

[14] R. P. Padhy, S. Verma, S. Ahmad, S. K. Choudhury, and P. K. Sa, "Deep Neural Network for Autonomous UAV Navigation in Indoor Corridor Environments," Procedia Computer Science, vol. 133, pp. 643-650, 2018/01/01/ 2018.

[15] M. Ö. Efe, "Sliding Mode Control for Unmanned Aerial Vehicles Research," in Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics, X. Yu and M. Önder Efe, Eds. Cham: Springer International Publishing, 2015, pp. 239-255.

[16] K. D. Nguyen and T. T. Nguyen, "Vision-Based Software-in-the-Loop-Simulation for Unmanned Aerial Vehicles Using Gazebo and PX4 Open Source," in 2019 International Conference on System Science and Engineering (ICSSE), 2019, pp. 429-432.

[17] V. Parmar, N. Bhatia, S. Negi, and M. Suri, "Exploration of Optimized Semantic Segmentation Architectures for edge-Deployment on Drones," CoRR, vol. abs/2007.02839, 2020 2020.

[18] F. Guo, J. Yang, Y. Chen, and B. Yao, "Research on image detection and matching based on SIFT features," in 2018 3rd International Conference on Control and Robotics Engineering (ICCRE), 2018, pp. 130-134.

[19] L. Tang, S. Ma, X. Ma, and H. You, "Research on Image Matching of Improved SIFT Algorithm Based on Stability Factor and Feature Descriptor Simplification," Applied Sciences, vol. 12, no. 17. doi: 10.3390/app12178448

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 29–32
DOI: 10.15439/2022R52
ISSN 2300-5963 ACSIS, Vol. 33

Optimal tracking controllers with Off-policy Reinforcement Learning Algorithm in Quadrotor

Dinh Duong Pham, Thanh Trung Cao, Tat Chung Nguyen, Phuong Nam Dao

***Abstract*—In this study, the optimal tracking control problem for the quadrotor which is a highly coupling system with completely unknown dynamics is addressed based on data by introducing the reinforcement learning (RL) technique. The proposed Off-policy RL algorithm does not need any knowledge of quadrotor model. By collecting data, which is the states of quadrotor system then using an actor-critic networks (NNs) to solve the optimal tracking trajectory problem. Finally, simulation results are provided to illustrate the effectiveness of proposed method.**

I. Introduction

In recent years, unmanned aerial vehicle (UAV) has been gaining an increasing consideration in research society due to its huge potential in many areas where the appearance of human is hard to achieve, such as: disaster surveillance, agricultural applications,... One of the most effective UAV is quadrotor with the ability of vertically taking off and landing, the versatile adaptation to arbitrary trajectories. So that, the needs of solving the problem of tracking control for a quadrotor has been put under research for years. Moreover, in application, it could be impossible to have the fully knowledge of the system due to the uncertainty of the environment where the quadrotor functions, the unknown loads that quadrotor carries. So the uncertainty is an indispensable part when it comes to control a quadrotor. Recently, many proposed controllers were proposed: PID-controller [5],[6], Linear-quadratic regulator (LQR) Controller [3],[7], Backstepping technique [1],[2], Sliding mode controller [4]. However, they are not considered as optimal control. In [9], this paper proposed a formation optimal control for multiple quadrotors. But the disadvantages of this scheme is that it just tackled the tracking problem for a simple trajectory which is straight line. This paper proposed a control scheme based on Off-policy Reinforcement Learning algorithm to obtain the optimal controllers for tracking problem of a quadrotor with completely unknown knowledge of the system.

II. Preliminaries and Problem Statement

In this section, we present the model of quadrotor and the traditional control scheme. A quadrotor 1 could be described with dynamic equations:

$$\begin{aligned} m\ddot{p} &= T_p R e_{3,3} - mge_{3,3} \\ J\ddot{\Theta} &= \tau - C(\Theta,\dot{\Theta})\dot{\Theta} \end{aligned} \qquad (1)$$

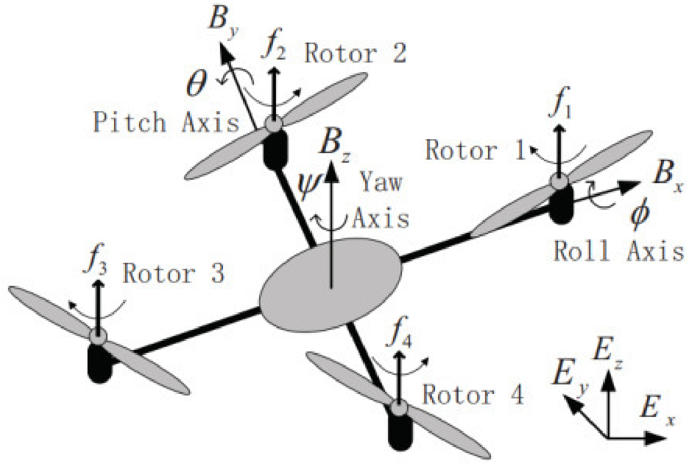


Figure 1. A typical quadrotor

Where:
The position of the center of mass is $p = [p_x, p_y, p_z]^T \in \mathbb{R}^3$. The Euler angles $\Theta = [\phi, \theta, \psi]$. $e_{i,j}$ is the vector which has i numbers of zeros except for number 1 in the j^{th} position.

$$C(\Theta,\dot{\Theta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

We define $T_p \in \mathbb{R}$ is the total force of all propellers $T_p = T_1 + T_2 + T_3 + T_4$ và $\tau = [\tau_\phi, \tau_\theta, \tau_\psi]^T \in \mathbb{R}^3$ is momentum that acts on quadrotor which resolve around x, y, z axis, $T_p = k_w u_z$ and $\tau = [l_\tau k_w u_\phi,\ l_\tau k_w u_\theta,\ k_t u_\psi]^T$. m is the weight, g is gravity, $J = diag(J_x, J_y, J_z)$ with $J_x,\ J_y,\ J_z$ are moments of inertia which resolve around $x,\ y,\ z$ axis respectively. It can be seen that this is 6 DOF which is highly coupling with 4 inputs $u_z, u_\phi, u_\theta, u_\psi$. In fact, $u_z, u_\phi, u_\theta, u_\psi$ are dependent on propellers' velocity:

$$\begin{aligned} u_z &= \omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 \\ u_\phi &= \omega_2^2 - \omega_4^2 \\ u_\theta &= \omega_1^2 - \omega_3^2 \\ u_\psi &= \omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 \end{aligned} \qquad (2)$$

Figure 2 illustrates the typical cascade control strategy for a quadrotor which consists of position controller in the outer loop and attitude controller in the inner loop.

Corresponding Author: Thanh Trung Cao, trung.caothanh@hust.edu.vn, nam.daophuong@hust.edu.vn

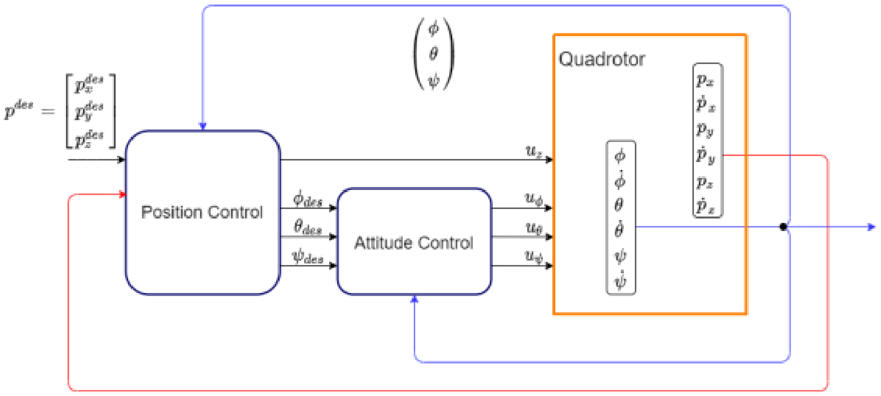

$p^{des} = \begin{bmatrix} p_x^{des} \\ p_y^{des} \\ p_z^{des} \end{bmatrix}$
$\begin{pmatrix} \phi \\ \theta \\ \psi \end{pmatrix}$
Position Control
Attitude Control
Quadrotor
ϕ_{des}
θ_{des}
ψ_{des}
u_z
u_ϕ
u_θ
u_ψ
ϕ, $\dot\phi$, θ, $\dot\theta$, ψ, $\dot\psi$
p_x, $\dot p_x$, p_y, $\dot p_y$, p_z, $\dot p_z$

Figure 2. The principle of controlling quadrotors

Remark 1: The control objective is to obtain the optimal tracking control by the Off-policy Algorithm RL for an unknown dynamics quadrotor. It is worth noting that this strategy has the advantage of solving the optimal tracking control problem for a sophisticated trajectory with no prior knowledge about the system by the iterative algorithm to estimate the optimal controller.

III. Proposed control strategy

A. *Position Controller with Off-policy RL*

The position dynamic can be written:

$$\begin{aligned} \ddot{p} &= m^{-1} k_w u_z R e_{3,3} - g e_{3,3} \\ &= m^{-1} k_w u_p \end{aligned} \tag{3}$$

Note that $u_p = u_z R e_{3,3} - \frac{m}{k_w} g e_{3,3} \in \mathbb{R}^3$ Set $x_p = [p_x, \dot{p}_x, p_y, \dot{p}_y, p_z, \dot{p}_z]^T \in \mathbb{R}$. $A_p = diag(a_p, a_p, a_p) \in \mathbb{R}^{6\times 6}$, $a_p = [0_{2\times 1}\ e_{2,1}]$ and $B_p = m^{-1} k_w [e_{6,2}, e_{6,4}, e_{6,6}]$ Assume the desired trajectory $\dot{x}_{pd}$ has $\dot{x}_{pd} = A_{pd} x_{pd}$ and error $e_p = x_p - x_{pd}$, we could rewrite the expanded system:

$$\dot{X}_p = \begin{bmatrix} \dot{e}_p \\ \dot{x}_{pd} \end{bmatrix} = \begin{bmatrix} A_p & A_p - A_{pd} \\ 0_{6\times 6} & A_{pd} \end{bmatrix} X_p + \begin{bmatrix} B_p \\ 0_{6\times 3} \end{bmatrix} u_p \tag{4}$$

The cost function is chosen:

$$\begin{aligned} V_p(X_p(t)) = \int_t^{\infty} & e^{-\lambda(\tau - t)} (X_p(\tau)^T Q_p X_p(\tau) \\ & + u_p(\tau)^T R_p u_p(\tau)) d\tau \end{aligned} \tag{5}$$

The Off-policy RL algorithm [8] for this optimal control problem is proposed as:

---

1) Initiate:
   Start with an acceptable control input u_p^0 and noise u_{pe} which is added to guarantee PE condition. Collect data and determine a threshold ϵ_p
2) Policy Evaluation
   With $u_p^i(X_p)$ solved from previous iteration, let solve $V_p^{i+1}(X_p)$ và $u_p^{i+1}(X_p)$ from equation:

$$\begin{aligned} & V_p^{i+1}(X_p(t+\delta t)) - V_p^{i+1}(X_p(t)) \\ & = -\int_t^{t+\delta t} [X_p(\tau)^T Q_p X_p(\tau) \\ & \quad + [u_p^i(X_p(\tau))]^T R_p u_p^i(X_p(\tau))] d\tau \\ & \quad + \int_t^{t+\delta t} \lambda V_p^{i+1}(X_p(\tau)) d\tau \\ & \quad + 2\int_t^{t+\delta t} [u_p^{i+1}(X_p(\tau))]^T R_p u_p^i(X_p(\tau)) d\tau \\ & \quad - 2\int_t^{t+\delta t} [u_p^{i+1}(X_p(\tau))]^T R_p [u_p^0(\tau) + u_{pe}] d\tau \end{aligned} \tag{6}$$

3) Policy Improvement
   Continue to iterate until $\|u_p^{i+1} - u_p^i\| < \epsilon_p$

To approximate V_p^i và u_p^i, Critic-Actor NNs ware estimated as:

$$V_p^i(X_p) = w_{Vp}^T \varphi_p(X_p) \tag{7}$$

$$u_p^i(X_p) = w_{up}^T \psi_p(X_p) \tag{8}$$

In detail, $\varphi_p(X_p) \in \mathbb{R}^{l_1}$ and $\psi_p(X_p) \in \mathbb{R}^{l_2}$ are 2 activation function vectors. $w_{Vp} \in \mathbb{R}^{l_1 \times 1}$ and $w_{up} \in \mathbb{R}^{l_2 \times 3}$ are 2 weight vectors respectively. Then, we can apply Least-Square Algorithm to solve (6).
After obtaining optimal $u_p = [u_{px}, u_{py}, u_{pz}]^T$, we could have u_z, desired attitudes angles:

$$\begin{aligned} u_z &= \sqrt{u_{px}^2 + u_{py}^2 + (u_{pz} + u_b)^2} \\ \psi_d &= 0 \\ \phi_d &= arcsin(\frac{u_{px} sin(\psi_d) - u_{py} cos(\psi_d)}{u_z}) \\ \theta_d &= arctan(\frac{u_{px} cos(\psi_d) + u_{py} sin(\psi_d)}{u_{pz} + u_b}) \end{aligned} \tag{9}$$

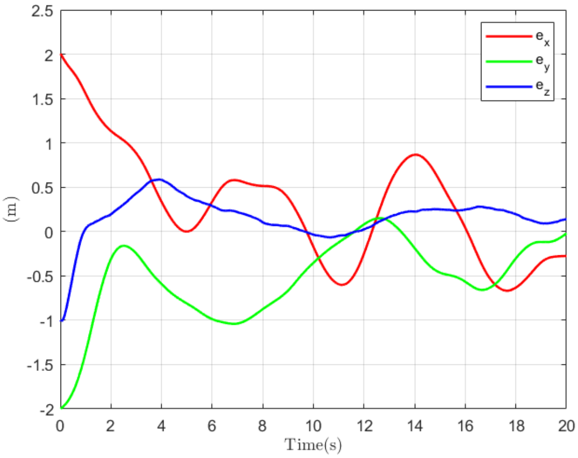


Figure 3. The position tracking error at initial stage

B. *Attitude Controller with Off-policy RL*

In the inner loop, similar control structure is proposed. Set $x_\Theta = [\phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$.Here, we have $\dot{x}_\Theta = F_\Theta x_\Theta + B_\Theta u_\Theta$ and $B_\Theta = [e_{6,2} b_{\Theta 1}, e_{6,4} b_{\Theta 2}, e_{6,6} b_{\Theta 3}] \in \mathbb{R}^{6\times 3}$ $b_{\Theta 1} = J_x^{-1} l_\tau k_w$, $b_{\Theta 2} = J_y^{-1} l_\tau k_w$, $b_{\Theta 3} = J_z^{-1} k_t$. The desired trajectory of attitude was obtained in the outer loop, which could be described as $\dot{x}_{\Theta d} = F_{\Theta d} x_{\Theta d}$. Let have $e_\Theta = x_\Theta - x_{\Theta d}$. Firstly, we have the expanded system:

$$\dot{X}_{\Theta d} = \begin{bmatrix} \dot{e}_\Theta \\ \dot{x}_{\Theta d} \end{bmatrix} = \begin{bmatrix} F_\Theta & F_\Theta - F_{\Theta d} \\ 0_{6\times 6} & F_{\Theta d} \end{bmatrix} X_{\Theta d} + \begin{bmatrix} B_\Theta \\ 0_{6\times 3} \end{bmatrix} u_\Theta \tag{10}$$

The cost function is chosen as:

$$V_\Theta(X_\Theta(t)) = \int_t^\infty e^{-\lambda(\tau - t)} (X_\Theta(\tau)^T Q_e X_\Theta(\tau) + u_\Theta(\tau)^T R u_\Theta(\tau)) d\tau \tag{11}$$

Then, we continue to implement the iterative algorithm as in the previous section.

IV. Simulation

Consider a quadrotor with the desired trajectory is a spiral trajectory

At the first stage, we use 2 PID-controllers for both outer and inner loops to collect data for the next stage of training to obtain the optimal controllers. Note that noises is added to the system to guarantee the PE condition.

The position tracking error in this stage is illustrate in Fig 3.

Then, we use the data as the input to the algorithms which are proposed in the previous section. The convergences of the weights are shown in Fig 4.

After we obtain the weights, estimated optimal controllers are applied to the object. The tracking performance is illustrated in Fig 5.

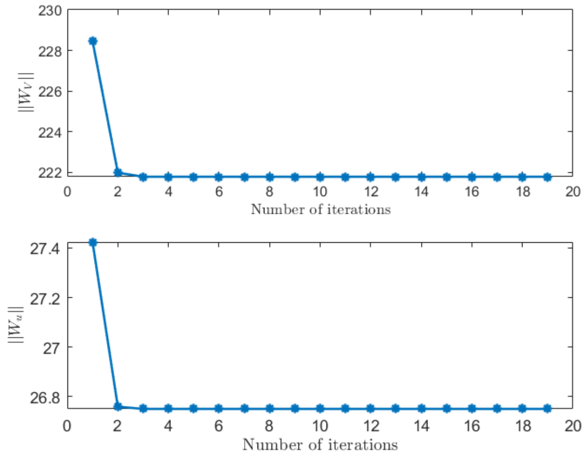


Figure 4. The convergence of weights in position controller

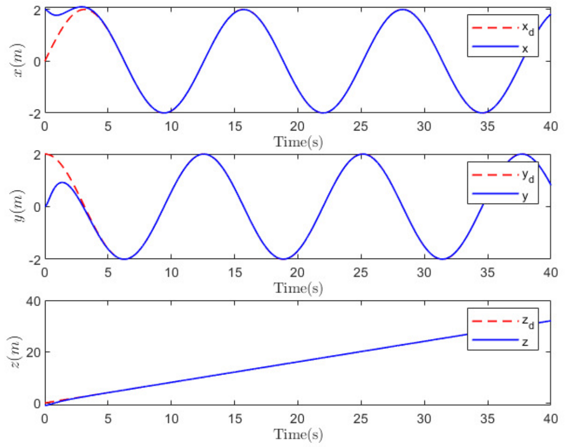


Figure 5. The tracking position of the optimal controllers

V. Conclusion

In this paper, a novel control strategy which consists of the Off-policy RL algorithm was proposed. By collecting data to train two actor-critic networks (NNs) which aim to estimate the optimal controllers which includes position controller and attitude controller, this structure has the advantage of no need of any prior information of the highly coupling system. Finally, simulation results are provided to illustrate the tracking performance of a sophisticated trajectory of the system.

Acknowledgement(s)

Dinh Duong Pham was funded by Vingroup JSC and supported by the Master, PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Institute of Big Data, code VINIF.2021.ThS.43.

REFERENCES

[1] E. Altug, J. Ostrowski, and R. Mahony. Control of a quadrotor helicopter using visual feedback. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 1, pages 72–77 vol.1, 2002.

[2] A. Das, F. Lewis, and K. Subbarao. Backstepping approach for controlling a quadrotor using lagrange form dynamics. *Journal of Intelligent and Robotic Systems*, 56:127–151, 09 2009.

[3] Y. Li and S. Song. A survey of control algorithms for quadrotor unmanned helicopter. In *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pages 365–369, 2012.

[4] C. Mu, C. Sun, and W. Xu. Fast sliding mode control on air-breathing hypersonic vehicles with transient response analysis. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 230, 11 2015.

[5] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006.

[6] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006.

[7] W. Wang, H. Ma, and C.-Y. Sun. Control system design for multi-rotor mav. *Journal of Theoretical and Applied Mechanics*, 51:1027–1038, 01 2013.

[8] G. Xiao, H. Zhang, Y. Luo, and H. Jiang. Data-driven optimal tracking control for a class of affine non-linear continuous-time systems with completely unknown dynamics. *Iet Control Theory and Applications*, 10:700–710, 2016.

[9] W. Zhao, H. Liu, F. L. Lewis, and X. Wang. Data-driven optimal formation control for quadrotor team with unknown dynamics. *IEEE Transactions on Cybernetics*, pages 1–10, 2021.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 33–36
DOI: 10.15439/2022R50
ISSN 2300-5963 ACSIS, Vol. 33

Fuzzy Brain Emotional Controller for Heart Disease Diagnosis

Duc-Hung Pham
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Vietnam.
phamduchunghp@gmail.com

Viet-Ngu Nguyen
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Vietnam.
Ngunguyenviet77@gmail.com

Thi Minh-Tam Le
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Vietnam
leminhtamutehy@gmail.com

***Abstract*—This article provides a new way for classifying heart disease. A classifier using a controller for brain emotional learning and a fuzzy system is presented. The controller's parameter updating laws are built using the gradient descent method. The method's convergence and stability are ensured by the Lyapunov function. Using the heart disease dataset from the University of California, Irvine (UCI), the performance of the system is examined. In addition, a comparison with different classifiers is provided. The outcomes of our experiments illustrate the efficacy of our strategy.**

***Index Terms*—Heart disease prediction, fuzzy system, Brain emotional learning controller.**

I. Introduction

In today's engineering applications, fuzzy inference systems and neural networks are used widely. There is a significant increase in productivity when fuzzy systems and neural networks are combined [1, 2]. The fuzzy inference system and the cerebellar model articulation controller [1] are two examples of popular methods. Other popular methods include the fuzzy inference system and brain-based learning control systems [3, 4], as well as the fuzzy inference system and the cerebellar model articulation controller [1].

The brain emotional learning controller (BELC) [2] was first presented by C. Lucas and colleagues as one of the most effective methods for the control of nonlinear systems. [2] BELC is also utilized in a wide range of applications, some of which include the prediction of time series [3], image encryption and decryption [1, 4], robot control [5], and chaotic synchronization [4].

In recent years, the role of engineering techniques has grown alongside that of sophisticated algorithms for medical diagnosis. The use of computer-assisted diagnosis, also known as CAD, is extremely important in clinical medicine since it has the potential to significantly increase the percentage of patients who are cured through earlier diagnosis. [6-10] A great number of researchers have presented ways with the goal of enhancing the speed and accuracy with which heart illness can be diagnosed. In 2019, Le et al. proposed a type-2 fuzzy neural network (FNN) for the diagnosis of cardiac illness [6]. This method used FNN for datasets from the University of California, Irvine. In the year 2021, Bakhsh proposed a classification system for heart conditions based on an enhanced deep genetic algorithm (EDGA) [7]. The findings of the study indicated that the EDGA is an appropriate tool for diagnosing heart disease. Convolutional Neural Network (CNN) of ResNet-50 was presented by Charles et al. in 2022 for the purpose of medical diagnosis [11]. This CNN has close to 50 layers, which enables it to attain classification performance.

This research makes a unique contribution to the learning model by describing the emotional signal within the learning rules for classification problems. It is possible to accomplish the desired result by making an informed decision regarding the emotional state of the system. More specifically, the generalization quality and the accuracy of the prediction can be improved by making an informed decision regarding the definition of the reinforcing signal. The results of the simulation and the research that was done on them show how effective the technique that was suggested is.

The rest of this paper can be summed up as follows: Section II describes a fuzzy brain emotional learning controller (FBELC), Section III discusses online learning and convergence analysis, Section IV discusses heart disease diagnosis using the proposed FBELC, and Section V concludes the paper.

II. Fuzzy Brain Emotional Learning Controller

Figure 1 displays the structure of FBELC. The fuzzy inference system and five layers make up an FBELC (input, sensory cortex space, weight space, amygdala and orbitofrontal cortex space, and output space).

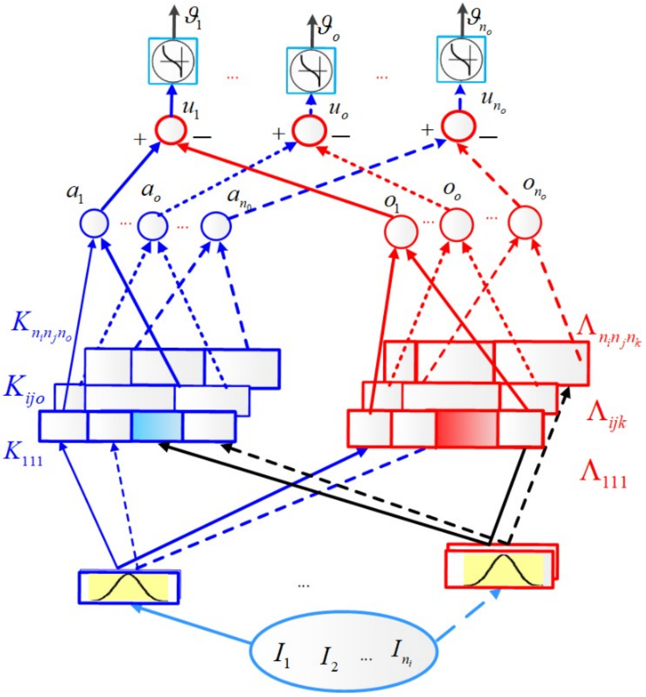


Fig. 1. The structure of FBELC classifier

The fuzzy inference rules are defined as:

If I_1 is G_{1j},..., and I_{n_i} is $G_{n_i j}$, then $a_o = K_{ijo}$ and $o_o = \Lambda_{ijo}$

for $o = 1,2...,n_o$; j=1,2...,n_j; i=1,2...,n_i (1)

where n_i, n_j, n_o are respectively input dimension, layer dimension and output dimension; G_{ij} is the fuzzy set, K_{ijo}, Λ_{ijo} are respectively orbitofrontal cortex weight and amygdala weight, a_o and o_o are amygdala output and orbitofrontal cortex, respectively.

Layer 1: Input space

$I = \left[I_1,\ I_2,...,I_{n_i} \right] \in R^{n_i}$, where n_i is the number of features.

Layer 2: Sensory cortex space:

A Gaussian function is written as

$$G_{ij} = \exp\left(-\frac{(I_i - m_{ij})^2}{2v_{ij}^2} \right) \quad (2)$$

where m_{ij} and v_{ij} are respectively the center and upper dilation.

Layer 3: Weight space K_{ijo} and Λ_{ijo}.

Layer 4: Amygdala and orbitofrontal cortex space

$$a_o = \frac{\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} G_{ij}K_{ijo}}{\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} G_{ij}} \quad (3)$$

$$o_o = \frac{\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} G_{ij}\Lambda_{ijo}}{\sum_{i=1}^{n_i}\sum_{j=1}^{n_j} G_{ij}} \quad (4)$$

Layer 5: Output space

FBELC output is the difference between amygdala and orbitofrontal cortex output. The *o-th* output is then computed as

$$u_o = a_o - o_o \quad (5)$$

Using the sigmoid function for classifying, the output of the FBELC classifier can be written as

$$\vartheta_o = \frac{1}{(1 + \exp(-u_o))} \quad (6)$$

for o=1,2,...,n_o

III. Online Learning and Convergence Analysis

A. Online learning

The updating weights Λ_{ijo} and K_{ijo} are calculated as

$$K_{ijo}(k+1) = K_{ijo}(k) + \Delta K_{ijo}(k) \quad (7)$$

$$\Lambda_{ijo}(k+1) = \Lambda_{ijo}(k) + \Delta\Lambda_{ijo}(k) \quad (8)$$

$$\Delta K_{ijo}(k) = l_K(G_{ij} \max(0, Rd_o(k) - a_o(k)) \quad (9)$$

$$\Delta\Lambda_{ijo}(k) = l_\Lambda(G_{ij}(u_o(k) - Rd_o(k)))) \quad (10)$$

where l_K and l_Λ are the learning rates. Rd_o is the reward signal. Select Rd_o as follows

$$Rd_o(k) = \xi_1(t_o(k) - \vartheta_o(k)) + \xi_2 u_o(k) \quad (11)$$

where t_o is the desired output; ξ_1 and ξ_2 are

factors corresponding to the desired output error and FBELC output, respectively. A cost function is chosen

as $\Phi(k) = \frac{1}{2}\sum_{o=1}^{n_o}(t_o(k) - \vartheta_o(k))^2$ (12)

Using the gradient descent method, obtains

$$\Delta m_{ij} = -l_m \frac{\partial\Phi}{\partial m_{ij}} = -l_m \frac{\partial\Phi}{\partial\vartheta_o}\frac{\partial\vartheta_o}{\partial u_o}\frac{\partial u_o}{\partial G_{ij}}\frac{\partial G_{ij}}{\partial m_{ij}}$$
$$= l_m \sum_{o=1}^{n_o}(t_o - \vartheta_o)\vartheta_o(1-\vartheta_o)(K_{ijo} - \Lambda_{ijo})G_{ij}\frac{2(I_i - m_{ij})}{v_{ij}^2} \quad (13)$$

$$\Delta v_{ij} = -l_v \frac{\partial\Phi}{\partial v_{ij}} = -l_v \frac{\partial\Phi}{\partial\vartheta_o}\frac{\partial\vartheta_o}{\partial u_o}\frac{\partial u_o}{\partial G_{ij}}\frac{\partial G_{ij}}{\partial v_{ij}}$$
$$= l_v \sum_{o=1}^{n_o}(t_o - \vartheta_o)\vartheta_o(1-\vartheta_o)(K_{ijo} - \Lambda_{ijo})G_{ij}\frac{2(I_i - m_{ij})^2}{v_{ij}^3} \quad (14)$$

Then, the updating law is calculated as:

$$m_{ij}(k+1) = m_{ij}(k) + \Delta m_{ij}(k) \quad (15)$$

$$v_{ij}(k+1) = v_{ij}(k) + \Delta v_{ij}(k) \quad (16)$$

B. Convergence Analysis

Define a Lyapunov function as

$$\Psi(k) = \frac{1}{2}\boldsymbol{e}^2(k) \quad (17)$$

where $\boldsymbol{e}(k) = \left[e_1(k),...,e_o(k),...,e_{n_o}(k) \right]$

in which $e_o(k) = t_o - \vartheta_o$

The deviate of Ψ can be written as

$$\Delta\Psi(k) = \Psi(k+1) - \Psi(k)$$
$$= \frac{1}{2}\left(\boldsymbol{e}^2(k+1) - \boldsymbol{e}^2(k)\right)$$
$$= \frac{1}{2}\left(\boldsymbol{e}(k+1) + \boldsymbol{e}(k)\right)\left(\boldsymbol{e}(k+1) - \boldsymbol{e}(k)\right) \quad (18)$$
$$= \frac{1}{2}\left(2\boldsymbol{e}(k) + \Delta\boldsymbol{e}(k)\right)\Delta\boldsymbol{e}(k)$$

Using the Taylor expansion, gets

$$\boldsymbol{e}(k+1) = \boldsymbol{e}(k) + \Delta\boldsymbol{e}(k) \cong \boldsymbol{e}(k) + \left[\frac{\partial\boldsymbol{e}(k)}{\partial\boldsymbol{m}}\right]^T \Delta\boldsymbol{m}$$

Where $\boldsymbol{m} = [m_{11},....,m_{ij},...,m_{n_i n_j}]$

$$\frac{\partial\boldsymbol{e}(k)}{\partial\boldsymbol{m}} = \frac{\partial\boldsymbol{e}(k)}{\partial\boldsymbol{\vartheta}(k)}\frac{\partial\boldsymbol{\vartheta}(k)}{\partial\boldsymbol{m}} = -\frac{\partial\boldsymbol{\vartheta}(k)}{\partial\boldsymbol{m}} \quad (20)$$

$$\Delta\boldsymbol{m} = -l_m\frac{\partial\Psi(k)}{\partial\boldsymbol{m}} = -l_m\frac{\partial\Psi(k)}{\partial\boldsymbol{\vartheta}(k)}\frac{\partial\boldsymbol{\vartheta}(k)}{\partial\boldsymbol{m}} = l_m(\boldsymbol{t} - \boldsymbol{\vartheta})\frac{\partial\boldsymbol{\vartheta}(k)}{\partial\boldsymbol{m}}$$
$$= l_m\boldsymbol{e}(k)\frac{\partial\boldsymbol{\vartheta}(k)}{\partial\boldsymbol{m}}$$

(21)

$$\Delta \boldsymbol{e}(k) = -l_m \boldsymbol{e}(k)\left(\frac{\partial \boldsymbol{\vartheta}(k)}{\partial \boldsymbol{m}}\right)^T \frac{\partial \boldsymbol{\vartheta}(k)}{\partial \boldsymbol{m}}$$

Set $Q = \left(\frac{\partial \boldsymbol{\vartheta}(k)}{\partial \boldsymbol{m}}\right)^T \frac{\partial \boldsymbol{\vartheta}(k)}{\partial \boldsymbol{m}}$, then

$$\Delta \boldsymbol{e}(k) = -l_m \boldsymbol{e}(k) Q \tag{22}$$

From (17) to (22), $\Delta\Psi(k)$ is obtained as:

$$\Delta\Psi(k) = \frac{1}{2}\left(2\boldsymbol{e}(k) - l_m \boldsymbol{e}(k)Q\right)(-l_m \boldsymbol{e}(k)Q) =$$

$$= \frac{1}{2}\boldsymbol{e}(k)\left(2 - l_m Q\right)(-l_m Q)\boldsymbol{e}(k) = -l_m Q \boldsymbol{e}^2(k)\left(1 - \frac{l_m Q}{2}\right)$$

$\Delta\Psi(k) \le 0$ if $1 - \frac{l_m Q}{2} \ge 0 \to 0 \le l_m \le \frac{2}{Q}$ then, the system's stability is guaranteed. Choosing l_v can be proved as similarity.

IV. Heart Disease Diagnosis Using FBELC

In order to evaluate the system, some parameters must be calculated; Sensitivity λ_{SEN}, negative predictive value λ_{NPV}, false positive rate $\lambda_{\text{FP\_rate}}$, Specificity λ_{SPE}, Accuracy λ_{ACC}.

$$\lambda_{\text{SEN}} = \frac{\lambda_{\text{TP}}}{\lambda_{\text{TP}} + \lambda_{\text{FN}}} \tag{24}$$

$$\lambda_{\text{NPV}} = \frac{\lambda_{\text{TN}}}{\lambda_{\text{TN}} + \lambda_{\text{FN}}} \tag{25}$$

$$\lambda_{\text{FP\_rate}} = \frac{\lambda_{\text{FP}}}{\lambda_{\text{TN}} + \lambda_{\text{FP}}} \tag{26}$$

$$\lambda_{\text{SPE}} = \frac{\lambda_{\text{TN}}}{\lambda_{\text{TN}} + \lambda_{\text{FP}}} \tag{27}$$

$$\lambda_{\text{ACC}} = \frac{\lambda_{\text{TP}} + \lambda_{\text{TN}}}{\lambda_{\text{TP}} + \lambda_{\text{TN}} + \lambda_{\text{FP}} + \lambda_{\text{FN}}} \times 100\% \tag{28}$$

where λ_{TP}, λ_{TN}, λ_{FP}, λ_{FN} are respectively true positives, true negatives, false positives and false negatives.

First, the UCI Starlog heart disease dataset, shown in Table I, contains 270 sets of samples, each with 13 features and 1 output target. Second, Fig. 2 depicts the FBELC method's accuracy value of 94.81% of the test set. Third, Fig. 3 depicts the result of the confusion matrix, which yields the following. The values of λ_{TP}=147, λ_{TN} =109, λ_{FP} =11, and λ_{FN} =3 and λ_{ACC}, λ_{SEN}, λ_{SPEC}, λ_{NPV}, and $\lambda_{\text{FP\_rate}}$ are then obtained using (24)-(28). Finally, table II shows comparisons between our method and other methods, demonstrating our method's superior performance.

TABLE I
THE DATASET OF UCI HEART DISEASE

	Characteristic
1	Slope of the peak exercise ST segment
2	Resting blood pressure
3	Serum cholesterol in mg/dl
4	Exercise-induced angina
5	Maximum heart rate achieved
6	Resting electrocardiographic results
7	Fasting blood sugar > 120 mg/dl
8	Old peak
9	Number of major vessels (0–3)
10	Chest pain type
11	Sex
12	Age
13	Thal

(23)

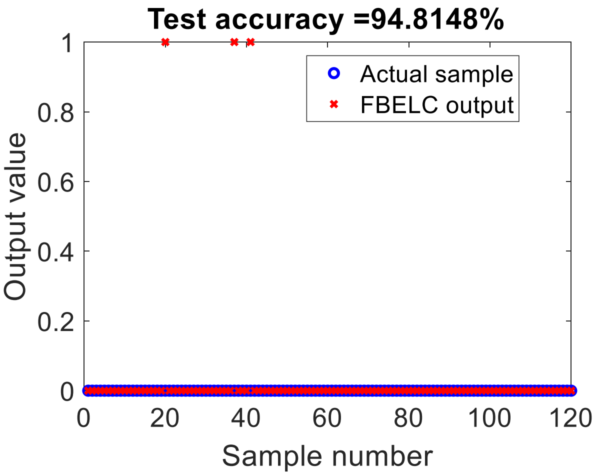


Fig. 2. The accuracy result of heart disease prediction using FBELC

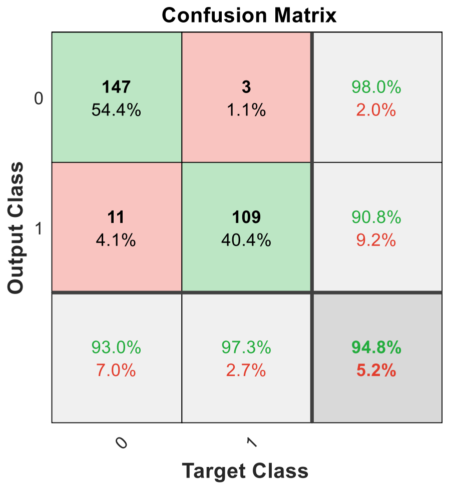


Fig. 3. The confusion matrix of heart disease prediction using FBELC

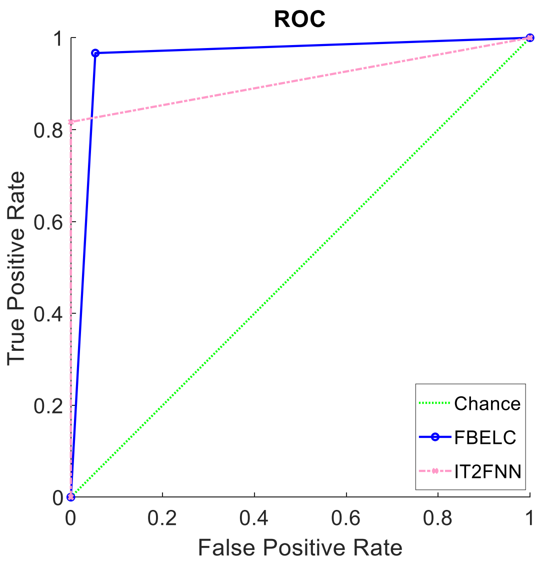


Fig. 4. The ROC of heart disease prediction using FBELC and IT2FNN

TABLE II
CLASSIFICATION PERFORMANCE MEASURE

	ACC	SEN	SPE	NPV	FP_rate
ANN [10]	0.84	0.87	0.79	0.83	0.21
SVM [10]	0.82	0.77	0.89	0.75	0.11
Logistic regression [10]	0.85	0.89	0.81	0.85	0.19
kNN [10]	0.80	0.84	0.76	0.81	0.24
Classification tree [10]	0.77	0.79	0.73	0.79	0.27
Naïve Bayes [10]	0.83	0.85	0.80	0.84	0.20
Kmeans IT2FNN [6]	93.81	--	93.58	--	--
Our method	94.81	0.89	0.9467	0.97	0.09

V. CONCLUSION

For classification, this study successfully proposed an FBELC. This work makes a contribution by combining a BELC with a fuzzy inference system and applying it to heart disease diagnosis. The fuzzy set and the novel setting of the optimization value of the BELC's reward signal, in particular, can improve classification efficiency. The simulation results show that the proposed algorithm has a high degree of generalization and accuracy while remaining simple and easy to implement. As a result of the findings, the proposed classifier appears to be a promising alternative for medical diagnosis. Some optimization algorithm can be applied to find the optimal learning rates such as improved particle swarm, modified grey wolf optimizer. Then, the proposed FBELC can be used in practical experiments in the future.

REFERENCES

[1] C. M. Lin, D. H. Pham, and T. T. Huynh, "Encryption and Decryption of Audio Signal and Image Secure Communications Using Chaotic System Synchronization Control by TSK Fuzzy Brain Emotional Learning Controllers," *IEEE Transactions on Cybernetics,* pp. 1-15, 2021.

[2] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami, "Introducing BELBIC: brain emotional learning based intelligent controller," *Intelligent Automation Soft Computing,* vol. 10, no. 1, pp. 11-21, 2004.

[3] H. S. A. Milad and J. J. I. A. Gu, "Expanded neo-fuzzy adaptive decayed brain emotional learning network for online time series predication," vol. 9, pp. 65758-65770, 2021.

[4] C.-M. Lin, D.-H. Pham, and T.-T. Huynh, "Synchronization of Chaotic System Using a Brain-Imitated Neural Network Controller and Its Applications for Secure Communications," *IEEE Access,* vol. 9, pp. 75923-75944, 2021.

[5] T.-T. Huynh, C.-M. Lin, T.-L. Le, V.-P. Vu, and F. Chao, "Self-organizing double function-link fuzzy brain emotional control system design for uncertain nonlinear systems," *IEEE Transactions on Systems, Man, Cybernetics: Systems,* 2020.

[6] T.-L. Le, T.-T. Huynh, L.-Y. Lin, C.-M. Lin, and F. Chao, "A K-means Interval Type-2 Fuzzy Neural Network for Medical Diagnosis," *International Journal of Fuzzy Systems,* vol. 21, no. 7, pp. 2258-2269, 2019.

[7] A. A. Bakhsh, "High-performance in classification of heart disease using advanced supercomputing technique with cluster-based enhanced deep genetic algorithm," *The Journal of Supercomputing,* pp. 1-22, 2021.

[8] V. B. Charles, D. Surendran, SureshKumar, and Control, "Heart disease data based privacy preservation using enhanced ElGamal and ResNet classifier," *Biomedical Signal Processing,* vol. 71, p. 103185, 2022.

[9] S. Qiao *et al.*, "RLDS: An explainable residual learning diagnosis system for fetal congenital heart disease," *Future Generation Computer Systems,* vol. 128, pp. 205-218, 2022/03/01/ 2022.

[10] A. K. Dwivedi, "Performance evaluation of different machine learning techniques for prediction of heart disease," *Neural Computing and Applications,* vol. 29, no. 10, pp. 685-693, 2018.

[11] V. B. Charles, D. Surendran, A. J. B. S. P. SureshKumar, and Control, "Heart disease data based privacy preservation using enhanced ElGamal and ResNet classifier," vol. 71, p. 103185, 2022.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 37–41
DOI: 10.15439/2022R38
ISSN 2300-5963 ACSIS, Vol. 33

An Edge Computing Collaboration Solution for Internet of Vehicles

Vu Khanh Quy
Hung Yen University of Technology and Education
Hungyen, Vietnam
quyvk@utehy.edu.vn

Dang Van Anh
Hung Yen University of Technology and Education
Hungyen, Vietnam
vananh271285@gmail.com

Vi Hoai Nam
Hung Yen University of Technology and Education
Hungyen, Vietnam
vihoainam@gmail.com

Nguyen Minh Quy
Hung Yen University of Technology and Education
Hungyen, Vietnam
minhquy@utehy.edu.vn

Anh-Ngoc Le
Swinburne Vietnam, FPT University
Hanoi, Vietnam
ngocla2@fee.edu.vn

***Abstract* - The advent of 5^{th} generation communication systems (5G) in the early 21^{st} century has realized real-time Internet of Things applications. 5G has capable of providing network services with extremely-high throughput and extremely low delay and allows a huge device number to connect together based on Internet infrastructure, forming the Internet of Things (IoT). In recent years, IoT has been applied in a variety of fields serving humans, such as smart cities, smart agriculture, e-healthcare, smart education, military, and IoT ecosystems. One of the main challenges of IoT applications is computing solutions to reduce service response times. In this study, we propose an Edge Computing Collaboration Solution for the Internet of Vehicles (IoV). Our solution proposes a small database that allows edge computing servers of IoVs to store each other's information. When the mobile end-users move to the new edge servers' managed coverage, properties related to the EC service are exchanged between the edge servers. The results have shown that our proposed solution improves significantly service response time, by up to 10-20%, compared to the existing solutions.**

***Index Terms*—5G, Internet of Things, Edge Computing, Internet of Vehicles (IoVs).**

I. Introduction

Smart devices are becoming more popular and necessary tools in modern society. The communication systems between network devices are increasingly diverse and complex in solutions and technologies. The 5^{th} generation networks, also known as $5G$, are attractive and topical study topics [1]. The $5G$ network enables the delivery of services with ultra-high throughput and ultra-low latency and allows hundreds of billions of mobile devices to connect. 5G realizes the concept of the IoT and forms emerging applications serving humans in smart cities, smart agriculture, smart healthcare, and other smart ecosystems [2].

According to Cisco's forecast, the number of connected IoT devices to the Internet will be over 500 billion by 2030 [3]. As a result, these devices will generate huge amounts of data. Traditional data processing models and methods are infeasible [4]. For recent decades, cloud computing (CC) is the dominant technology in all fields. CC has proven outstanding capability and ability by providing robust computing services with high reliability. One limitation of CC is high latency time, so it is unachievable to real-time IoT solutions [5]. To address this problem, edge computing (EC) was proposed. EC brings the abilities of the cloud closer to end-users, reducing service response times and realizing real-time IoT applications [6], as shown in **Fig. 1**.

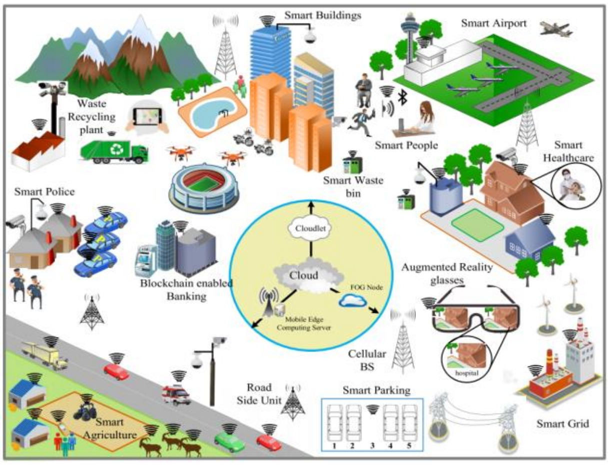


Fig. 1. An Illustration of EC-based Smart Cities [7].

In recent, IoT-based smart city applications have been focused on research. Smart cities provide a wide range of convenient services for citizens, including smart energy management, smart healthcare, intelligent transportation systems and other utility services. It can be said that smart cities are the combination of the internet infrastructure, Internet of things and big data processing to provide citizens with increasingly superior smart utilities [7].

In this work, we introduce a collaboration solution between EC servers to improve performance and reduce latency for real-time IoVs applications in smart cities. The rest of this paper is organized as follows: in Section 2, we present related works. The proposed solution is introduced in Section 3. Section 4 presents simulation results and analysis. The conclusion and open research issues is presented in Section 5.

II. Related Works

In recent, smart cities is one of the emerging fields that have attracted great attention from the research community in both industry and academia. The concept of smart cities is very diverse. According to [8], smart cities are divided into six components, including *citizens*, *economy*, *governance*, *transportation*, *environment* and *living*, see **Fig. 2**.

Aiming to present a full picture of the recent smart cities applications based on IoT, we perform a literature review of proposals in three years in this area.

- In the field of citizens, proposals focus on aspects to provide a more comfortable living for humans. Specifically, the authors have proposed a framework to enhance security [9], biometrics [10] and sports activities [11].

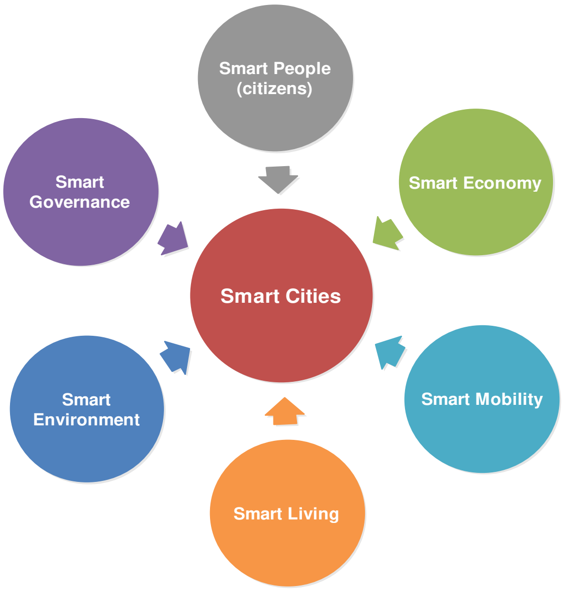


Fig. 2. The key components of smart cities.

- In the field of the economy, solutions focus on aspects to develop a sustainable economy, including improving seaport operations [12], charging solutions for IoE vehicles and EC-based resource allocation solutions [13].
- In the field of mobility, studies focus on solving the challenges of intelligent transportation systems, such as resources allocation solutions for IoE vehicles [14], optimal routing and estimation for vehicles [15].
- In the field of environments, proposals focus on solutions to provide a green environment for residents, such as optimal routing based on the quality of air metrics [16], smart irrigation solutions [17] and green environmental management architectures [18].
- In the field of smart living, studies focus on solutions to improve the life quality of residents, such as travel recommendations [19], resident control and monitoring [20] and healthcare [21].
- In the field of governance, proposals focus on proposing governance solutions, such as infrastructure management [22], complex event solutions [23] and voting solutions [24].

The surveyed results have shown that the vision of smart cities still has many challenges. The proposals are still in the primitive stage and are limited to each specific sector. More research and proposals with integrated solutions need to be employed in this field. Studies have also demonstrated that the heart of smart cities is the citizens. Therefore, the awareness, applicability and adoption of new technologies, as well as the active participation of residents, are major problems. However, in our opinion, although there are still many challenges, smart cities will be an inevitable development trend in the near future.

III. THE PROPOSED SOLUTION

A. *Motivation*

We consider an IoV system based on edge computing for smart cities. This system connects vehicles, IoT devices, sensors, applications and people based on the Internet to provide smart utility services for citizens, as described in Section II. Specifically, sensors collect information. According to the convention architecture, the information is obtained and processed at other layers such as EC and CC [25] based on connections within their coverage. Consequently, the information is sent to the edge servers, see **Fig. 3**. However, the IoVs in smart cities, vehicles are often mobile. This leads to disconnection to the edge server when vehicles move out of the edge server's coverage. Then, vehicles have to establish a connection to another edge server and re-request its edge services. When a vehicle connects to a new edge server, the services that were performed at the old edge server are discarded and re-requested at the new edge server. As a result, response times and system resources will increase. With the huge vehicle numbers in smart cities, the system resource and power consumption will be huge. This is the motivation for us to do this research.

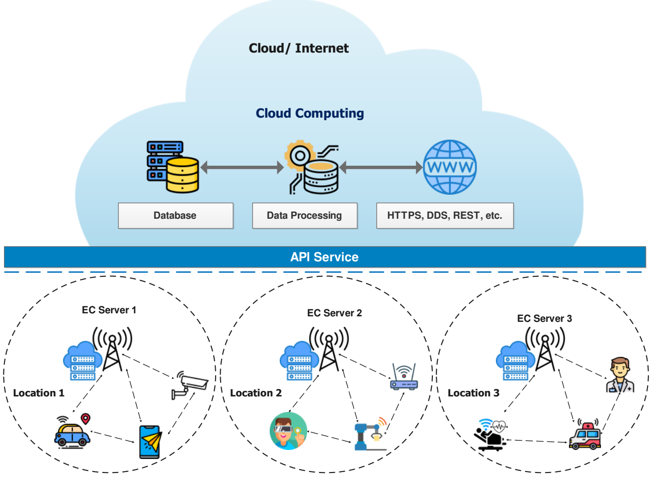


Fig. 3 The EC-based IoT architecture for smart cities.

B. *Proposed Network Architecture*

In existing architecture, the system consists of three layers: things, edge and cloud layers, as presented in **Fig. 3**. It should be recalled that, in a network system, the two basic transactions are searching and routing. These transactions are invoked to serve service requests derived from the Things layer with objectives including minimal service response time, saving energy and system resource consumption. In the proposed architecture, we establish an information map between the EC centres, as shown in **Fig. 4**.

The information map is defined as follows: An information map is a small database to store properties and services that the EC servers have provided. When the end-user moves from

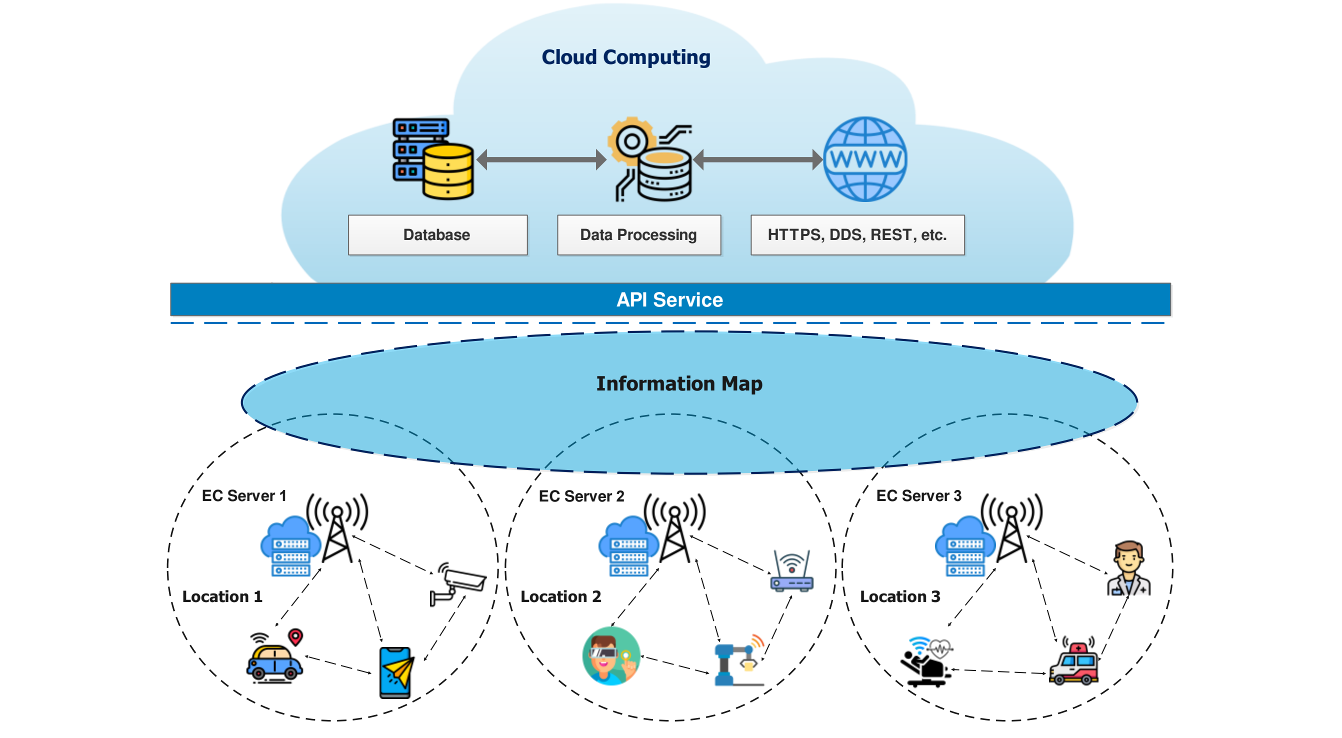


Fig. 4 The proposed EC-based IoT architecture for smart cities.

one coverage to another, an EC server, instead of having to launch a search process on a cloud to fulfil service requests that were served previously or other executed EC servers, now, EC servers simply searches within the information map. Consequently, service response times, energy and system resource consumption will be significantly reduced. The proposed architecture is presented in Fig. 4.

C. *The effectiveness of our solution*

In this study, we define some mathematical symbols as follows:

- Symbol n is the number of mobile nodes that must roam while is service request progressing.
- Symbol T and T' are the service request and response time at each layer, respectively. T and T' are the same value in all coverage.
- Symbol T_{S1}, T_{S2}, and T_{S3} are the searching time at the things, edge and cloud layers, respectively.
- Symbol T_{R1}, T_{R2} and T_{R3} are the routing time at the things, edge and cloud layers, respectively
- Symbol T'_{S1}, T'_{S2}, and T'_{S3} are the routing time to return results at things, edge and cloud layers, respectively.

When the mobile node moves in coverage, the service response time cost of a transaction is determined as follows:

$$T = \sum_{i=1}^{n}(T_{S1} + T_{S2} + T_{S3}) + \sum_{i=1}^{n}(T_{R1} + T_{R2} + T_{R3}) \quad (1)$$

$$T'_1 = T'_2 = \sum_{i=1}^{n}(T'_{R1} + T'_{R2} + T'_{R3}) \quad (2)$$

$$T = (T_1 + T'_1) \quad (3)$$

Suppose that, while the mobile end-user initializes a service request, the system will perform the searching and routing transactions at things, edge and cloud, respectively, to respond to this request. In-while, the mobile end-user moves to new coverage. In the traditional architecture, allsearching and routing transactions will be re-performed on new coverage. Consequently, time and energy consumption have been determined as follows,

$$T = 2 \times (T_1 + T'_1) \quad (4)$$

In our proposed model, to determine the service response time, T_1 and T'_1 values are still defined in Eq. (1) and (2), respectively. However, when a mobile user roams to new coverage, due to the support of information maps, mobile users are still able to use the searched results that have been performed in the old coverage. Therefore, T_2 and T'_2 values are determined according to Eq. (5) and Eq. (6).

$$T_2 = \sum_{i=1}^{n}(T_{S1} + T_{S2}) + \sum_{i=1}^{n}(T_{R1} + T_{R2}) \quad (5)$$

$$T'_2 = \sum_{i=1}^{n}(T'_{R1} + T'_{R2}) \quad (6)$$

IV. Simulation and Analysis

Aiming to clarify the effectiveness of the proposed solution, we set up an EC-based IoT system consisting of 1.000 mobile end-user nodes. We have assumed that the percentage of nodes roaming in while transactions are processing in a range $\lambda_{max} \in [10 - 25]\%$. Moreover, to focus on clarifying the efficiency of the solution, we have also assumed that the costs of routing and searching for different coverage are the same and represented by numbers.

- *The traditional solution*: apply Eq. (1)-(4).

- *The proposed solution*: T_1 and T'_1 values still applied Eq. (1)-(4), while T_2 and T'_2 values applied Eq. (5)-(6).

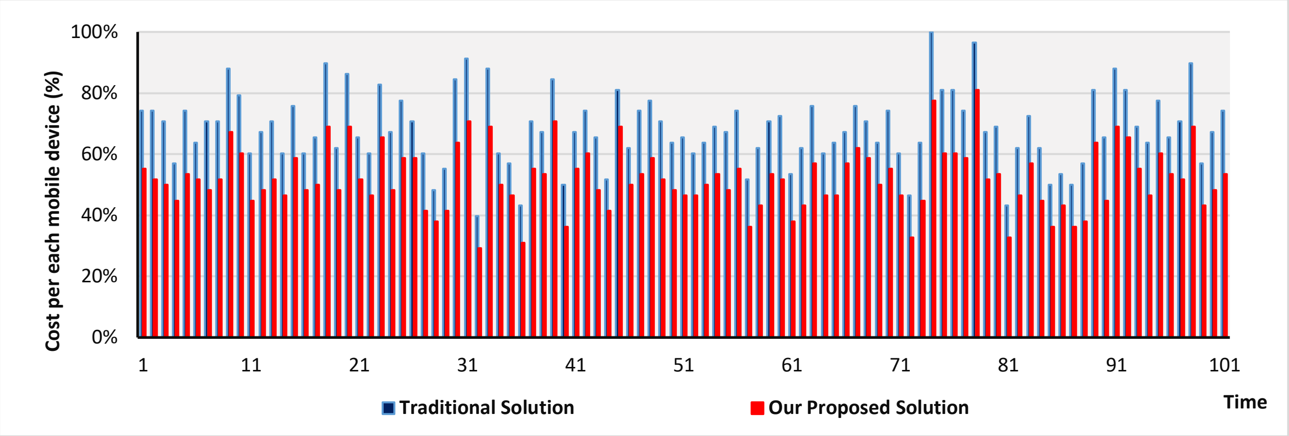


Fig. 5. The cost-effectiveness of the proposed method per each device.

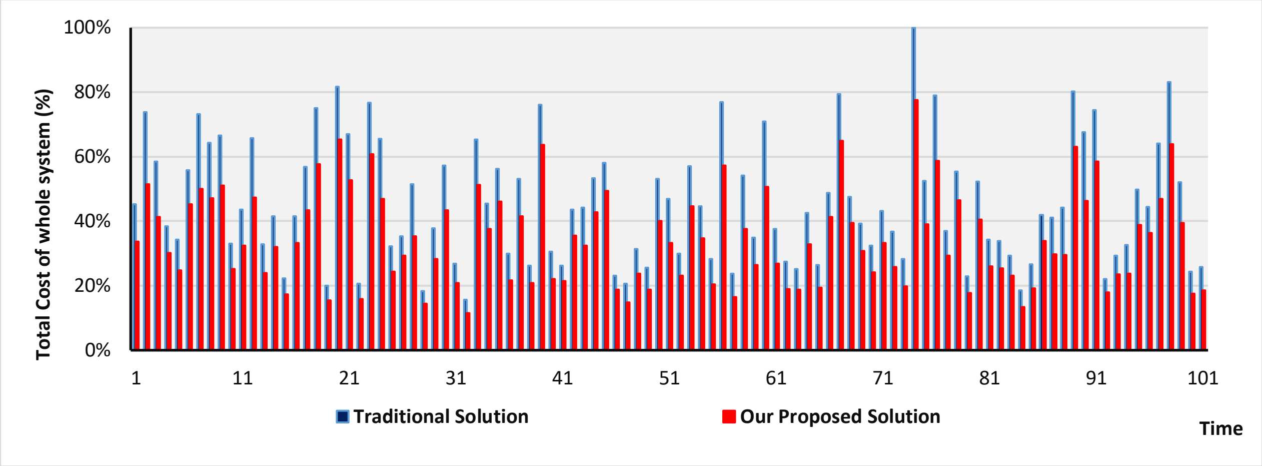


Fig. 6. The cost-effectiveness of the proposed method for the whole system.

Fig. 5 presents a performance comparison of two solutions in terms of the average service response time per service request at the things layer. The simulation results have shown that the proposed solution reduces the service response time by $[10 - 20]\%$ compared to the existing solutions.

Fig. 6 presents a performance comparison in terms of the service response time of the whole system. The results have shown that the new solution significantly reduces the response time of all requests from vehicles compared to the existing solution. The effectiveness of this solution is becoming more apparent as the number of increased roaming service requests corresponds to an increase in mobile vehicle numbers.

Besides the effectiveness of reducing service response time. In another aspect, the energy consumption of the whole system will also be significantly improved. This problem will be shown in our future studies.

V. Conclusion and Open Issues

In this study, we focus on studying a collaboration solution between EC servers to reduce response time for real-time IoVs applications in smart cities. Our proposed solution operates at the application layer between edge computing servers by establishing a small data map to store properties of edge services that are performed by edge servers. This solution is very effective in reducing response time and enhancing the performance of IoVs applications in smart cities. The results have indicated that our solution outperforms compared to existing algorithms. From the privacy and security perspectives, the proposed information map must be against attacks or fake information maps. The problems will be presented by us in future works.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

We have conducted the research, analyzed the data, and performed simulations together. All authors had approved the final version.

References

[1] S. V. Balkus, H. Wang, B. D. Cornet, C. Mahabal, H. Ngo, H. Fang, “A Survey of Collaborative Machine Learning Using 5G Vehicular Communications,” IEEE Communications Surveys & Tutorials, 24(2), pp. 1280-1303, 2022.

[2] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, M. Ur-Rehman, “Challenges, Applications, and Future of Wireless Sensors in Internet of Things: A Review,” IEEE Sensors Journal, 22(6), pp. 5482-5494, 2022.

[3] Y. B. Zikria, R. Ali, M. K. Afzal, et al., “Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions,” Sensors, vol. 2021, Article ID 1174, 21 pages, 2021.

[4] B. Liu, Z. Luo, H. Chen and C. Li, “A Survey of State-of-the-art on Edge Computing: Theoretical Models, Technologies, Directions, and Development Paths,” IEEE Access, vol. 10, pp. 54038-54063, 2022.

[5] S. Xu, Y. Li, S. Guo, C. Lei, D. Liu, X. Qiu, “Cloud–Edge Collaborative SFC Mapping for Industrial IoT Using Deep Reinforcement Learning,” IEEE Transactions on Industrial Informatics, 18(6), pp. 4158-4168, 2022.

[6] J. Du, C. Jiang, A. Benslimane, S. Guo and Y. Ren, "SDN-Based Resource Allocation in Edge and Cloud Computing Systems: An Evolutionary Stackelberg Differential Game Approach," IEEE/ACM Transactions on Networking, 30(4), pp. 1613-1628, 2022.

[7] L. U. Khan et al., “Edge-Computing-Enabled Smart Cities: A Comprehensive Survey,” IEEE Internet of Things Journal, 7(10), pp. 10200-10232, 2020.

[8] A. Kirimtat, O. Krejcar, A. Kertesz, et al., “Future Trends and Current State of Smart City Concepts: A Survey,” IEEE Access, vol. 8, pp. 86448-86467, 2020.

[9] H. Zhao, Y. Zhang, X. Huang, Y. Xiang, C. Su, “A Physical-Layer Key Generation Approach Based on Received Signal Strength in Smart Homes,” IEEE Internet of Things Journal, 9(7), pp. 4917-4927, 2022.

[10] A. Mitra, D. Bigioi, S. P. Mohanty, P. Corcoran, E. Kougianos, “iFace 1.1: A Proof-of-Concept of a Facial Authentication Based Digital ID for Smart Cities,” IEEE Access, vol. 10, pp. 71791-71804, 2022.

[11] K. Zhang, H. Chen, H. -N. Dai, H. Liu and Z. Lin, “SpoVis: Decision Support System for Site Selection of Sports Facilities in Digital Twinning Cities,” IEEE Transactions on Industrial Informatics, 18(2), pp. 1424-1434, 2022.

[12] K. L. A. Yau, S. Peng, J. Qadir, et al., “Towards Smart Port Infrastructures: Enhancing Port Activities Using Information and Communications Technology,” IEEE Access, vol. 8, pp. 83387 - 83404, 2020.

[13] S. Chavhan, N. Dubey, A. Lal, et al., “Next-Generation Smart Electric Vehicles Cyber Physical System for Charging Slots Booking in Charging Stations,” IEEE Access, vol. 8, pp. 160145 - 160157, 2020.

[14] E. ElGhanam, H. Sharf et al., “On the Coordination of Charging Demand of Electric Vehicles in a Network of Dynamic Wireless Charging Systems,” IEEE Access, vol. 10, pp. 62879 - 62892, 2022.

[15] J. Qiu, L. Du, D. Zhang, et al., “Nei-TTE: Intelligent Traffic Time Estimation Based on Fine-Grained Time Derivation of Road Segments for Smart City,” IEEE Transactions on Industrial Informatics, vol. 16, pp. 2659 - 2666, 2020.

[16] E. Ghaffari, A. M. Rahmani, M. Saberikamarposhti, “An Optimal Path-Finding Algorithm in Smart Cities by Considering Traffic Congestion and Air Pollution,” IEEE Access, vol. 10, pp. 55126 - 55135, 2022.

[17] A. Aldegheishem, N. Alrajeh, L. García, et al., “SWAP: Smart WAter Protocol for the Irrigation of Urban Gardens in Smart Cities,” IEEE Access, vol. 10, pp. 39239 - 39247, 2022.

[18] M. Ghahramani, M. C. Zhou, A. Molter, F. Pilla, “IoT-Based Route Recommendation for an Intelligent Waste Management System,” IEEE Internet of Things Journal, vol. 9, pp. 11883 - 11892, 2022.

[19] F. Huang, J. Xu, J. Weng, “Multi-Task Travel Route Planning With A Flexible Deep Learning Framework,” IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 3907-3918, 2021.

[20] J. R. Santana, L. Sánchez, P. Sotres, et al., “A Privacy-Aware Crowd Management System for Smart Cities and Smart Buildings,” IEEE Access, vol. 8, pp. 135394 - 135405, 2020.

[21] R. Zhou, X. Zhang, X. Wang, G. Yang, N. Guizani and X. Du, “Efficient and Traceable Patient Health Data Search System for Hospital Management in Smart Cities,” IEEE Internet of Things Journal, 8(8), pp. 6425-6436, 2021.

[22] Y. Bai, Q. Hu, S. H. Seo, et al., “Public Participation Consortium Blockchain for Smart City Governance,” IEEE Internet of Things Journal, vol. 9, pp. 2094 - 2108, 2020.

[23] B. Khazael, H. T. Malazi, S. Clarke, “Complex Event Processing in Smart City Monitoring Applications,” IEEE Access, vol. 9, pp. 143150 - 143165, 2020.

[24] G. Rathee et al., “On the Design and Implementation of a Blockchain Enabled E-Voting Application Within IoT-Oriented Smart Cities,” IEEE Access, vol. 9, pp. 34165 - 34176, 2021.

[25] H. Wang et al., “Architectural Design Alternatives Based on Cloud/Edge/Fog Computing for Connected Vehicles,” IEEE Communications Surveys & Tutorials, 22(4), pp. 2349-2377, 2020.

DOI: 10.15439/2022R37
ISSN 2300-5963 ACSIS, Vol. 33

Outage Performance of Full-Duplex Dual-hop Relaying System With Energy Harvesting Using Fountain Codes

Nguyen Hong Viet*, Dang The Hung†, Nguyen Thi Thanh Hoai‡, Nguyen Hoang Anh†,
Nguyen Hoanh Viet†, and Gia-Thinh Vo§,
* Air Force Officer's College, Nha Trang, Khanh Hoa, Vietnam
† Telecommunications University, Nha Trang, Khanh Hoa, Vietnam
‡ Le Quy Don High School, Ha Tinh, Vietnam
§ Thu Dau Mot University, Binh Duong, Vietnam
Corresponding author: danghung8384@gmail.com

***Abstract*—In this paper, we study a dual-hop relaying system exploiting fountain codes, where both the source node and the destination node have one antenna, while the relay node is a device with two antennas. Moreover, the relay node is a energy limited device, so it must harvest the radio frequency signals to assist the source node to transmit its data to the destination node with full-duplex mode which increases double channel capacity. We analyze the system performance through the derived exact outage probability expression over Rayleigh fading channels. Finally, Monte-Carlo simulations are performed to confirm the theoretical results of the proposed model.**

***Index Terms*—Fountain codes, dual-hop relaying system, full-duplex, energy harvesting.**

I. INTRODUCTION

Recently, the mobile users enhance connectivity quality, widen the coverage area, data transmission, and process information among devices to devices together, such as Internet of Thing (IoT) networks, smart cities, automation driving car [1]. However, multiple service platforms are supported which create many challenges as high data rates, low latency, high energy, and spectrum efficiency. To solve these problem, many new technologies have been researched and proposed to be applied in wireless communications, e.g. dual-hop/multi-hop systems [2], [3], non-orthogonal multiple access [4], short packet communications [5], massive MIMO [6], cognitive radio [7], energy harvesting (EH) from radio frequency signals [8], and full duplex (FH) transmission [9]. In particular, EH considered as a potential solution which can prolong the operation time of the devices, it is very suitable for machine with limited power sources such as sensor nodes, multi-hop relaying networks [10]. Moreover, FH help to improve spectrum efficiency due to the devices with FD can receive and transmit signals in band at the same time which increases double channel capacity when comparing with traditional half-duplex mode [11]. Besides, self-interference cancellation (SIC) scheme used in FD devices to improve the bit error rate (BER) performance. However, it is difficult to design and implement antennas to completely eliminate interference in FD systems which always exist the residual self-interference (RSI) level to increase the noise floor in the wireless system.

There are some literatures combining the advantages of FD and EH in relaying communications which have attracted much attention of researchers to enhance the reliability, capacity, delay time, energy efficiency and coverage area [12]- [16]. Particularly, the authors in [12] have considered a dual-hop two-way full-duplex network with EH at the relay applying amplify-and-forward (AF) scheme to forward the transmission data and spatial modulation at the source. The results showed that the BER performance and the spectrum efficiency with FD mode reach better when comparing with the normal half-duplex one. In [13], the authors studied the power allocation methods in a two-way FD system under impact of RSI via the exact expressions of optimal power allocation parameter and presented that the performance of studied protocol outperforms the regular power allocation one. The FD-EH relay system performance through the derived exact closed-form expression such as the symbol error probability has also analyzed in [14]. In the work [15], the authors considered a FD-EH relay system via the exact expressions such as outage probability (OP) and symbol error rate (SER) over cascaded Rayleigh fading channels. The results present that the performances for OP and SER which are significant lower when comparing with ones over Rayleigh fading channels. The work in [16] evaluated the outage probability in FD-EH relaying networks in which the destination has multi-antennas using selection combining (SC) or maximal ratio combining (MRC) technique to improve the quality for received signals.

Fountain codes (FC) can adapt with different channel conditions which recently has attracted the attention of many researchers [17]- [21]. Different with fixed-rate codes, a transmitter in FC do not require the knowledge of channel state information (CSI) before transmitting encoded packets, so that the receivers can maintain flexible decoding performance. The Fountain encoder can generate an unlimited number of encoded fountain packets, and then sends these encoded packets to the pre-determined receivers. If the receivers obtain enough fountain packets, they can decode the original data [18]- [21].

To the best of our knowledge, there are no literatures that refer to exploit FC in FD-EH dual-hop relaying system. Therefore, in this paper we carry out research the FD dual-hop system performance using FC with EH based TS method at the relay node to enhance the power consumption, capacity level and reliable transmission. We reach the exact expression for the outage probability (OP) over Rayleigh fading channels. These mathematical expressions will be verified through Monte Carlo simulations.

The structure of our paper is arranged as follows: In section II, we represent the proposed model. In section III, we examine the system performance. Section IV gives simulation results and theoretical analysis ones. Finally, we conclude the paper in section V.

II. SYSTEM MODEL

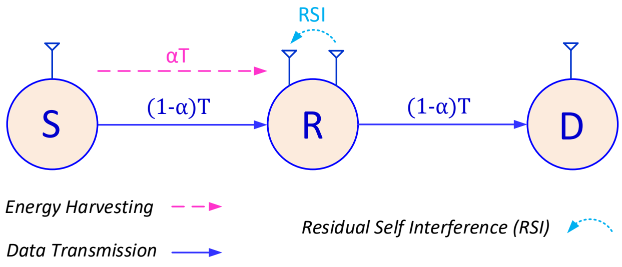


Fig. 1. System model of the proposed protocol.

The considered model shows in Figure 1, in which a source (S) node wants to send its data to a destination (D) node with assistance a relay (R) node by decode-and-forward (DF) technique over Rayleigh fading channels. We also assume that S and D are one antenna with half-duplex (HD) action mode while R is equipped two antennas with full-duplex (FD) approach in the same frequency. Moreover, the relay node is a energy-limited device, and R must harvest the radio frequency signals energy from S through time-switching (TS) method. Therefore, the information transmission from S to D which split two time slots in which energy harvesting portion is αT, and data transmission part is $(1-\alpha)T$, where $\alpha \in (0,1)$ is TS factor, and T is the transmission period from S to D in the block signal.

Using the Fountain code, S divides its data into K packets. Each packet will be appropriately encoded and then sent sequentially to D in orthogonal time slots. We assume that the considered system is limited in delay time, and S will finish its transmission after transmitting $N_{\max}$ encoded packets to D. In order perform to decode the original data, D must exactly obtain at least H data packet, where $H = (1+\varepsilon)K, H \le N_{\max}$ and ε is a constant depending on the code design [19]. Let denote N_{D} as the number of encoded packets that D can receive after $N_{\max}$ time slots. If $N_{\mathrm{D}} \ge H$, D can successfully the original information, and contrary the outage occurs if $N_{\mathrm{D}} < H$.

We denote that $h_{\mathrm{SR}}, h_{\mathrm{RD}}$, and h_{RR} are the channel factors between links S $\to$ R, R $\to$ D, and the residual self-interference at R, respectively. We also assume the channels of devices as Rayleigh fading, so that the channel gains $\gamma_{\mathrm{SR}} = |h_{\mathrm{SR}}|^2$, $\gamma_{\mathrm{RD}} = |h_{\mathrm{RD}}|^2$, and $\gamma_{\mathrm{RR}} = |h_{\mathrm{RR}}|^2$, have exponential distribution with characteristic parameters λ_{SR}, λ_{RD}, and λ_{RR}, whose cumulative distribution function (CDF) and probability density function (PDF) are expressed as

$$F_X(x) = 1 - \exp(-\omega x), \tag{1}$$

$$f_X(x) = \omega \exp(-\omega x), \tag{2}$$

where $X \in \{\gamma_{\mathrm{SR}}, \gamma_{\mathrm{RD}}, \gamma_{\mathrm{RR}}\}$ and $\omega \in \{\lambda_{\mathrm{SR}}, \lambda_{\mathrm{RD}}, \lambda_{\mathrm{RR}}\}$.

The energy harvesting at the relay node is given as

$$E_h^R = \eta \alpha T P_{\mathrm{S}} \gamma_{\mathrm{SR}}, \tag{3}$$

where $\eta \in (0,1)$ is the conversion energy efficiency, P_{S} is the transmission power of the source.

The transmission power at the relay node is given by

$$P_{\mathrm{R}} = \frac{\eta \alpha T P_{\mathrm{S}} \gamma_{\mathrm{SR}}}{(1-\alpha)T} = \mu P_{\mathrm{S}} \gamma_{\mathrm{SR}}, \tag{4}$$

where $\mu = \eta\alpha/(1-\alpha)$.

Next, we have the received signal at R and D, respectively as

$$y_{\mathrm{R}} = h_{\mathrm{SR}}\sqrt{P_{\mathrm{S}}}x_{\mathrm{S}} + h_{\mathrm{RR}}\sqrt{P_{\mathrm{R}}}x_{\mathrm{R}} + n_{\mathrm{R}}, \tag{5}$$

$$y_{\mathrm{D}} = h_{\mathrm{RD}}\sqrt{P_{\mathrm{R}}}x_{\mathrm{R}} + n_{\mathrm{D}}, \tag{6}$$

where x_{S} and x_{R} are transmission signals of the source node and the relay node, n_{R} and n_{D} are additive white Gaussian noise (AWGN) with the zero mean and variance N_0 (also assume that all devices of the system which have the same variance N_0). From (5), we can determine the power of signal self-interference at R as

$$\mathrm{E}\{P_{\mathrm{R}}\gamma_{\mathrm{RR}}\} = \mu P_{\mathrm{S}} \mathrm{E}\{\gamma_{\mathrm{RR}}\gamma_{\mathrm{SR}}\}, \tag{7}$$

where $\mathrm{E}\{.\}$ is the statistical expectation operator.

We assume that the relay node is equipped with self-interference cancellation (SIC). However, in practice the signal self-interference at R will not completely eliminate due to the imperfection of hardware that will still have a part, called residual self-interference (RSI), it is a Gaussian random variable which has variance as

$$\delta_{\mathrm{RSI}} = \kappa \mu P_{\mathrm{S}}, \tag{8}$$

where κ is the SIC efficiency at R.

The signal-to-interference-plus-noise-ratio (SINR) at R and D is shown as

$$\Psi_{\mathrm{SR}} = \frac{\gamma_{\mathrm{SR}} P_{\mathrm{S}}}{\delta_{\mathrm{RSI}} + N_0}, \Psi_{\mathrm{RD}} = \frac{\gamma_{\mathrm{RD}} P_{\mathrm{R}}}{N_0}, \tag{9}$$

For the DF relaying protocol, the end-to-end of the system depends on the lowest SINR level. The equivalent SINR of the system is as follows:

$$\Psi_{\mathrm{e2e}} = \min(\Psi_{\mathrm{SR}}, \Psi_{\mathrm{RD}}). \tag{10}$$

Next, the outage probability (OP) value at D is presented as

$$\mathrm{OP} = \Pr(N_{\mathrm{D}} < H | N_{\max}). \tag{11}$$

III. OUTAGE PERFORMANCE

Each encoded packet will be decoded successfully if the end-to-end data rate is larger than a predetermined target rate which is given by C_{th}. Otherwise, the encoded packet will not be obtained correctly. Therefore, the probability that the destination cannot accurately decode one encoded packet as

$$\begin{aligned}\rho_{\mathrm{D}} &= \Pr\left[(1-\alpha)\log_2\left(1+\Psi_{\mathrm{e2e}}\right) < C_{\mathrm{th}}\right] \\ &= \Pr\left(\Psi_{\mathrm{e2e}} < \gamma_{\mathrm{th}}\right), \end{aligned} \tag{12}$$

where C_{th} is the desired transmission rate of the system, and $\gamma_{\mathrm{th}} = 2^{C_{\mathrm{th}}/(1-\alpha)} - 1$.

We have the probability of a encoded packet which is decoded correctly as $\Pr\left(\Psi_{\mathrm{e2e}} > \gamma_{\mathrm{th}}\right) = 1 - \rho_{\mathrm{D}}$. Combining (9), (10), and (12), ρ_{D} is rewritten as follows:

$$\rho_{\mathrm{D}} = \Pr\left[\min\left(\Psi_{\mathrm{SR}}, \Psi_{\mathrm{RD}}\right) < \gamma_{\mathrm{th}}\right]. \tag{13}$$

We can implement (13) as below

$$\rho_{\mathrm{D}} = 1 - \Pr\left(\Psi_{\mathrm{SR}} > \gamma_{\mathrm{th}}, \Psi_{\mathrm{RD}} > \gamma_{\mathrm{th}}\right). \tag{14}$$

Let $X = |h_{\mathrm{SR}}|^2$ and $Y = |h_{\mathrm{RD}}|^2$, we can rewrite (14) follow as

$$\begin{aligned}\rho_{\mathrm{D}} &= 1 - \Pr\left(\frac{P_{\mathrm{S}}}{\delta_{\mathrm{RSI}} + N_0}X > \gamma_{\mathrm{th}}, \frac{\mu P_{\mathrm{S}}}{N_0}XY > \gamma_{\mathrm{th}}\right) \\ &= 1 - \underbrace{\Pr\left(X > \gamma_{\mathrm{th}}\frac{\left(\delta_{\mathrm{RSI}} + N_0\right)}{P_{\mathrm{S}}}, XY > \gamma_{\mathrm{th}}\frac{N_0}{\mu P_{\mathrm{S}}}\right)}_{I(a,b)}. \end{aligned} \tag{15}$$

To calculate ρ_{D}, the first we need to find out integral of $I(a,b)$. Applying the conditional probability theorem [22], we have

$$\begin{aligned} I(a,b) &= \Pr\left(X > a, XY > b\right) \\ &= \int_a^\infty \left[1 - F_Y\left(\frac{b}{x}\right)\right] f_X(x)\,dx, \end{aligned} \tag{16}$$

where $a = \frac{\gamma_{\mathrm{th}}\left(\delta_{\mathrm{RSI}} + N_0\right)}{P_{\mathrm{S}}}$, $b = \frac{\gamma_{\mathrm{th}}}{\Delta\mu}$, and $\Delta = P_{\mathrm{S}}/N_0$ is transmit signal-to-noise ratio (SNR).

From (1) and (2), the integral $I(a,b)$ can be rewritten as

$$I(a,b) = \lambda_{\mathrm{SR}} \int_a^\infty \exp\left(-\lambda_{\mathrm{SR}}x\right)\exp\left(-\lambda_{\mathrm{RD}}\frac{b}{x}\right)dx. \tag{17}$$

With the help of [22, eq. (1.211.1)], we have

$$\exp\left(-\lambda_{\mathrm{RD}}\frac{b}{x}\right) = \sum_{m=0}^{\infty}\frac{\left(-\lambda_{\mathrm{RD}}b\right)^m}{m!}x^{-m}. \tag{18}$$

Substituting [18] into [17], which yields

$$I(a,b) = \lambda_{\mathrm{SR}}\sum_{m=0}^{\infty}\frac{\left(-\lambda_{\mathrm{RD}}b\right)^m}{m!} \times \int_a^\infty x^{-m}\exp\left(-\lambda_{\mathrm{SR}}x\right)dx. \tag{19}$$

Applying [22, eq. (3.351.2)], we have

$$\int_a^\infty x^{-m}\exp\left(-\lambda_{\mathrm{SR}}x\right)dx = \left(\lambda_{\mathrm{SR}}\right)^{m-1}\Gamma\left(1-m, \lambda_{\mathrm{SR}}a\right), \tag{20}$$

where $\Gamma(\cdot,\cdot)$ is the upper incomplete Gamma function [22, eq. (8.350.2)]. Replacing [20] into [19], we obtain the integral of $I(a,b)$ as

$$I(a,b) = \lambda_{\mathrm{SR}}\sum_{m=0}^{\infty}\frac{\left(-\lambda_{\mathrm{RD}}b\right)^m}{m!} \times \left(\lambda_{\mathrm{SR}}\right)^{m-1}\Gamma\left(1-m, \lambda_{\mathrm{SR}}a\right). \tag{21}$$

Similarly, replacing [21] into [15], we get the exact probability of ρ_{D} as follows:

$$\rho_{\mathrm{D}} = 1 - \lambda_{\mathrm{SR}}\sum_{m=0}^{\infty}\frac{\left(-\lambda_{\mathrm{RD}}b\right)^m}{m!} \times \left(\lambda_{\mathrm{SR}}\right)^{m-1}\Gamma\left(1-m, \lambda_{\mathrm{SR}}a\right). \tag{22}$$

As mentioned in (11), the exact expression of OP can be provided as

$$\begin{aligned}\mathrm{OP} &= \Pr\left(N_{\mathrm{D}} < H | N_{\max}\right) \\ &= \sum_{N_{\mathrm{D}}=0}^{H-1} C_{N_{\max}}^{N_{\mathrm{D}}}\rho_{\mathrm{D}}^{N_{\mathrm{D}}}\left(1-\rho_{\mathrm{D}}\right)^{N_{\max}-N_{\mathrm{D}}}. \end{aligned} \tag{23}$$

We observed from (23) that the possible values of N_{D} are 0 to $H-1$, and $C_{N_{\max}}^{N_{\mathrm{D}}}$ are cases for each value of N_{D}.

IV. SIMULATION RESULTS

In this section, we perform Monte-Carlo simulations based Matlab software which validate the math expressions in Section III and evaluate the system's features. In all simulations, we set the parameters as the number of required Fountain packets that need to be reached to restore the original data (H), target rate (C_{th}), path loss exponent factor (β), the energy conversion efficiency of energy harvesting phase (η), total block time (T) by $3, 1, 3, 1, 1$, and the coordinates of the node in a two-dimensional plane Oxy as source, relay, and destination node are $(0,0)$, $(0.5,0)$, and $(1,0)$, respectively. For all of the figures, we use 5×10^5 trials over Rayleigh fading channels, and the simulation results and theoretical ones are notated by Sim and Theory.

Figure 2 shows the value of OP in terms of Δ $(\Delta = P_{\mathrm{S}}/N_0)$ (dB). In this figure, the SIC efficiency at the relay node ($\kappa = -30$ (dB)), maximum number of time slots for transmission from S to D $(N_{\max} = 5)$. Observing at this figure, we can see that OP value decreases when increasing the transmit SNR (Δ). Because transmit power level of the source increases which increases the ability receiving of enough required encoded packets to reconstruct the original information at the destination. The second remarkable point in Figure 2 is the value of OP increases when decreasing the time duration of energy harvesting α, for example, the value of $\alpha = 0.1$, OP is the highest when comparing with OP at the values of $\alpha \in (0.2, 0.3)$, and OP is the smallest when $\alpha = 0.3$.

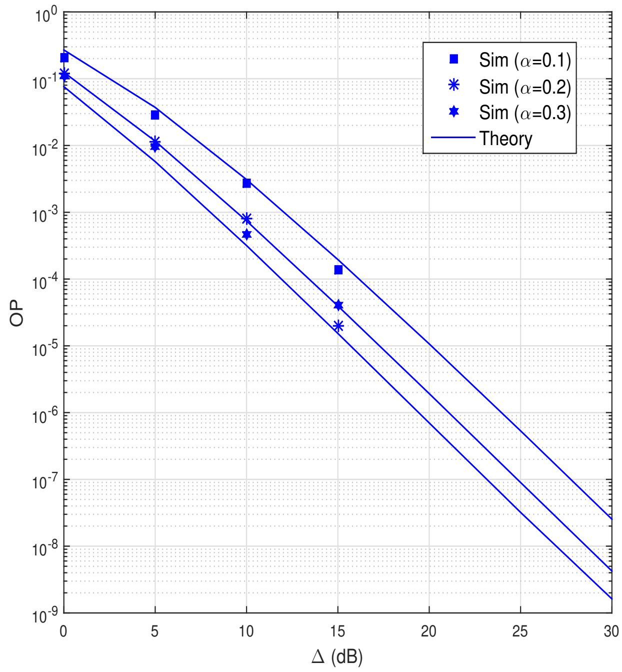


Fig. 2. OP plots versus Δ (dB) when $\kappa = -30$ (dB), $N_{\max} = 5$.

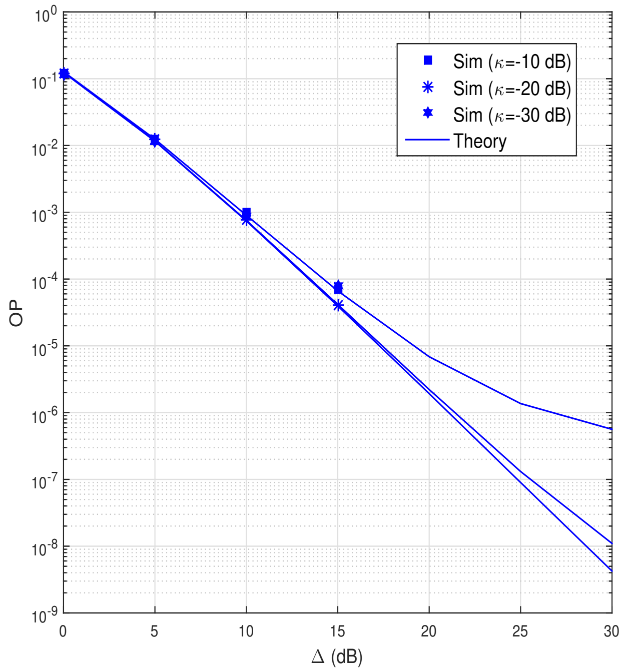


Fig. 4. OP plots versus Δ (dB) when $\alpha = 0.2$, $N_{\max} = 5$.

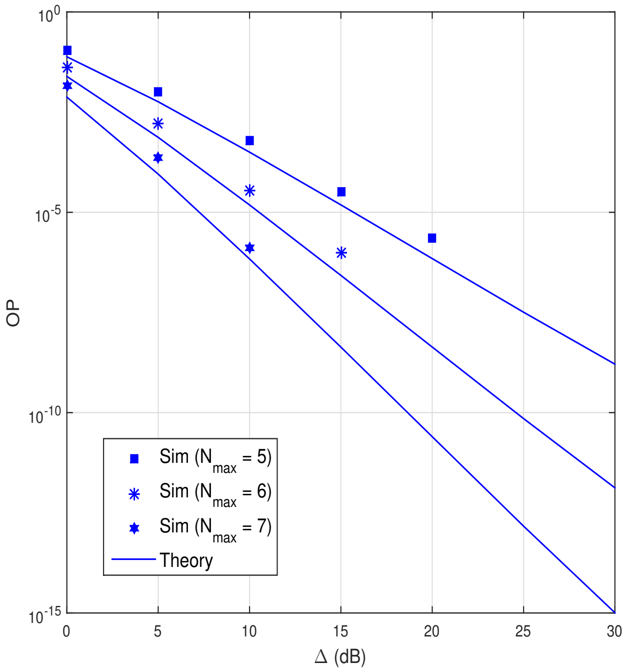


Fig. 3. OP plots versus Δ (dB) when $\kappa = -30$ (dB), $\alpha = 0.3$.

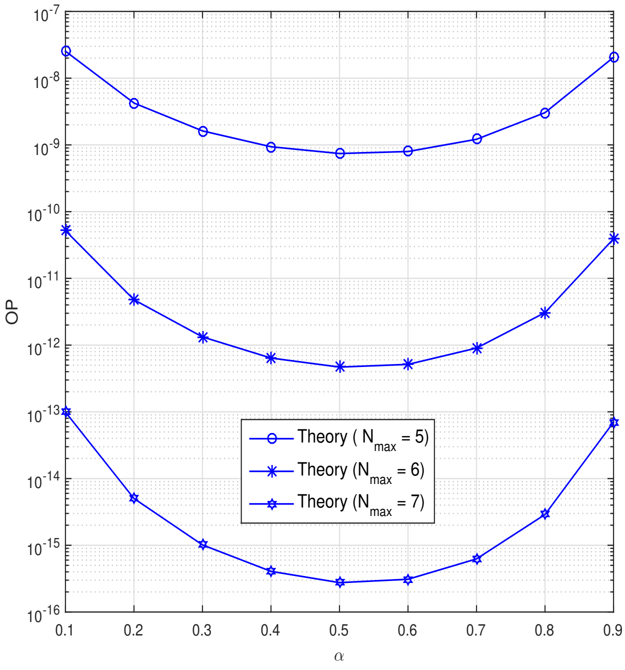


Fig. 5. OP plots versus α when $\kappa = -30$ (dB), $\Delta = 30$ (dB).

In figure 3 shows OP based on Δ in dB when $\kappa = -30$ (dB) and $\alpha = 0.3$. We can see that the value of OP decreases when increasing the levels of Δ. Moreover, OP significantly decreases as increasing the value of $N_{\max}$. However, when $N_{\max}$ increases which makes to increase the system delay time.

In figure 4 investigates the impact of the SIC level on the performance of the considered protocol, i.e., $\kappa = -10$ dB, $\kappa = -20$ dB, $\kappa = -30$ dB, and $\alpha = 0.2$, $N_{\max} = 5$. It can seen that OP value significantly decreases when increasing SIC level and the transmit SNR (Δ). Therefore, RSI has seriously affect on the system performance.

In figure 5, we present OP as a function of α when $\kappa = -30$ dB, and $\Delta = 30$ dB. We can realize that OP decreases when increasing the value of $N_{\max}$, and OP is the lowest at the value $\alpha = 0.5$. Moreover, when increasing the maximum number of time slots $(N_{\max})$ that means the delay time of the system increases which is an important measurement standard in wireless communication systems.

V. CONCLUSIONS

In this paper, we evaluated the system performance of the considered FD dual-hop relaying protocol using Fountain codes, in which the relay node collects the RF signals from the source, in terms of outage probability expression over Rayleigh fading channels. The analytical results show that the performance of the SIC, time switching ratio factor α, choosing a suitable transmission power level, and the number of maximum time slots for data transmission are important parameters that determine the main performance of the system and should be carefully considered when designing and setting the system.

REFERENCES

[1] S. Malathy, et al. "Routing constraints in the device-to-device communication for beyond IoT 5G networks: a review," *Wirel. Netw.,* vol. 27, no. 5, pp. 3207-3231, 2021.

[2] M. O. Hasna and M.-S. Alouini, "End-to-end performance of transmission systems with relays over Rayleigh-fading channels," *IEEE Trans. Wirel. Commun.,* vol. 2, no. 6, pp. 1126-1131, 2003.

[3] M. Hasna and M.-S. Alouini, "A performance study of dual-hop transmissions with fixed gain relays," *IEEE Trans. Wirel. Commun.,* vol. 3, no. 6, pp. 1963-1968, Nov. 2004.

[4] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A survey of non-orthogonal multiple access for 5G," *IEEE Commun. Surv. Tutor.,* vol. 20, no. 3, pp. 2294-2323, 2018.

[5] G. Durisi, T. Koch, and P. Popovski, "Toward massive, ultrareliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE,* vol. 104, no. 9, pp. 1711-1726, 2016.

[6] D. Borges, P. Montezuma, R. Dinis, and M. Beko, "Massive MIMO techniques for 5G and beyond-opportunities and challenges," *Electronics,* vol. 10, no. 14, pp. 1667, 2021.

[7] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.,* vol. 6, no. 4, pp. 13-18, Aug. 1999.

[8] T. X. Doan, T. M. Hoang, T. Q. Duong, and H. Q. Ngo, "Energy harvesting-based D2D networks in the presence of interference and ambient RF sources," *IEEE Access,* Mar. 2017.

[9] Z. Zhongshan, C. Xiaomeng, L. Keping, A. V. Vasilakos, and L. Hanzo, "Full duplex techniques for 5G networks: self-interference cancellation, protocol design, and relay selection," *IEEE Communications Magazine,* vol. 53, no. 5, pp. 128-137, 2015.

[10] S. Ulukus et al., "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications,* vol. PP, no. 99, pp. 1-1, 2015.

[11] V.-D. Nguyen, T. Q. Duong, H. D. Tuan, O.-S. Shin, and H. V. Poor, "Spectral and energy efficiencies in full-duplex wireless information and power transfer," *IEEE Trans. on Commun.,* vol. 65, no. 5, pp. 2220-2233, 2017.

[12] A. Koc, I. Altunbas, and E. Basar, "Two-way full-duplex spatial modulation systems with wireless powered AF relaying," *IEEE Wirel. Commun. Lett.,* vol. 7, no. 3, pp. 444-447, 2018.

[13] Y. Jingrui, L. Xuefang, and Y. Qinghai, "Power allocation of two way full-duplex AF relay under residual self-interference," in *Communications and Information Technologies (ISCIT),* 2014 14th International Symposium on, 2014, pp. 213-217.

[14] B. C. Nguyen, T. M. Hoang, and P. T. Tran, "Performance analysis of full-duplex decode-and-forward relay system with energy harvesting over nakagami-*m* fading channels," *International Journal of Electronics and Communications,* vol. 98, pp. 114-122, 2019.

[15] B. C. Nguyen, X. N. Tran, T. M. Hoang, "Performance analysis of full-duplex vehicle-to-vehicle relay system over double-rayleigh fading channels," *Mob. Netw. Appl.,* vol. 25, pp. 363-372, 2020.

[16] P. T. Tin, T. N. Nguyen, D. H. Tran, M. Voznak, V. D. Phan, and S. Chatzinotas, "Performance enhancement for full-duplex relaying with time-switching-based SWIPT in wireless sensors networks," *Sensors,* vol. 21, no. 11, pp. 3847, 2021.

[17] D. J. C. Mackay, "Fountain codes," *IEEE Proc. Commun.,* vol. 152, pp. 1062-1068, Dec. 2005.

[18] J. Castura and Y. Mao, "Rateless coding for wireless relay channels," *IEEE Trans. Wirel. Commun.,* vol. 6, no. 5, pp. 1638-1642, May 2007.

[19] T. T. Duy and H.Y. Kong, "Secondary spectrum access in cognitive radio networks using rateless codes over rayleigh fading channels," *Wirel. Pers. Commun.,* vol. 77, no. 2, pp. 963-978, Jul. 2014.

[20] X. Wang, W. Chen, and Z. Cao, "A rateless coding based multi-relay cooperative transmission scheme for cognitive radio networks," in *IEEE Globecom, Honolulu,* HI, USA, Nov. 2009, pp. 164-169.

[21] X. Di, K. Xiong, P. Fan, and H. C. Yang, "Simultaneous wireless information and power transfer in cooperative relay networks with rateless codes," *IEEE Trans. Veh. Technol.,* vol. 66, no. 4, pp. 2981-2996, Apr. 2017.

[22] D. Zwillinger, *Table of integrals, series, and products. Elsevier,* 2014.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 49–54
DOI: 10.15439/2022R47
ISSN 2300-5963 ACSIS, Vol. 33

Innovative Development of a Flying robot with a Flexible Manipulator for Aerial Manipulations

Yavor Yotov
Institute of Mechanics. Bulgarian Academy of Sciences
Sofia, Bulgaria
javor.jotov@imbm.bas.bg

Nikolay Zlatov
Institute of Mechanics. Bulgarian Academy of Sciences
Sofia, Bulgaria
zlatovn@hotmail.com

Georgi Hristov
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
ghristov@uni-ruse.bg

Plamen Zahariev
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
pzahariev@uni-ruse.bg

Chi Hieu Le
Faculty of Science & Engineering
University of Greenwich
Kent, United Kingdom
c.h.le@gre.ac.uk

James Gao
Faculty of Science & Engineering
University of Greenwich
Kent, United Kingdom
j.gao@gre.ac.uk

Anh My Chu
Institute of Simulation Technology, Le Quy Don Technical University
Hanoi, Vietnam
ghristov@uni-ruse.bg

Ho Quang Nguyen
Institute of Engineering & Technology
Thu Dau Mot University
Binh Duong, Vietnam
quangnh@tdmu.edu.vn

Le Minh Huynh
Department of Science & Technology
Saigon Hi-Tech Park
Ho Chi Minh City, Vietnam
hlminh.shtp@gmail.com

Trung Thanh Bui
Faculty of Mechanical Engineering
Hung Yen University of Technology and Education. Hung Yen, Vietnam
buitrungthanh@gmail.com

Jamaluddin Mahmud
College of Engineering, Universiti Teknologi MARA, Shah Alam
Selangor, Malaysia
jm@uitm.edu.my

Michael S. Packianather
School of Engineering
Cardiff University
Cardiff, United Kingdom
packianatherms@cf.ac.uk

***Abstract*—This paper presents an innovative development of a flying robot or an aerial robot, with a flexible manipulator, called the Dexterous Aerial Robotic System (DFTS), for aerial manipulations, especially for inspections and reparations of various structures such as wind turbines, power lines and open gas pipelines, decorations and painting of high industrial chimneys and walls of high buildings, as well as transport and delivery of courier shipments, relocation and manipulation of assemblies and units in inaccessible or dangerous environments. The proposed DFTS consists of two independent but interconnected systems or functional units, which have two main separate functions respectively, including a basic carrying function, and a precise positioning and stabilization function. The system with a basic carrying function is actually the main flying system, the unmanned aerial vehicle (UAV); it is remotely controlled and piloted. Meanwhile, the aerial manipulation platform, called the vertical take-off and landing platform VTOL, which is an active flying platform with 6 degrees of freedom (DOF) is used for positioning and stabilization; and it is attached to the UAV via the soft link. With the use of a long soft link, the problems which are caused by the air turbulent flows generated by the UAV are minimized, and the aerial manipulations of objects are safely controlled and operated. The VTOL which is equipped with a grasping mechanism was successfully developed, prototyped and tested. The experimental results showed that, the developed VTOL can self-stabilize with the inclination angle of being up to 8 degrees.**

***Index Terms*—Flying robot, aerial robot, flexible manipulator, aerial manipulation, unmanned aerial vehicle, UAV, soft robot link, design, product development**

I. Introduction

With the technological and scientific trends of moving towards Smart Manufacturing and Industry 4.0, there has been an emerging need of developing multifunctional robotic systems and smart solutions for a wide range of industrial applications [1,2, 10-14]. For applications related to aerial manipulations, there have been lots of efforts and studies to develop flying robots or aerial robots. Basically, one aerial robot has two main functional units or systems: the unmanned aerial vehicle (UAV) and the robotic manipulator(s) or robotic arm(s) which have the capability of directly interacting with objects in the surrounding environment [3,4]. The robotic manipulators of the aerial robot are optimally designed for implementing specific functions and industrial applications, including inspections and reparations of various structures, such as wind turbines, power lines and open gas pipelines, decorations and painting of high industrial chimneys and walls of high buildings, as well as transport and delivery of courier shipments, relocation and manipulation of assemblies and units in inaccessible or dangerous environments [3-7].

One of the key challenges to design aerial robots is to dynamically and optimally control the aerial manipulator (s) in which its center of gravity needs to remains unchanged, and the dynamic impacts from the resultant forces and loads need to be minimized during the aerial manipulations. This is more challenging when the additional flying platform with the multifunctional robotic arm(s) is attached to the UAV, the main flying system, via the soft link. The uneven or changing forces and loads applied on the aerial robot during aerial manipulations need to be carefully considered when developing dynamic models of aerial robots.

This study presents innovative development of an aerial robot, called the Dexterous Aerial robotic System (DFTS), that has a flexible robotic manipulator, called the vertical take-off and landing platform (VTOL) which is pivotally attached to the UAV via the soft link; and it can be therefore considered as an active 6-DOF soft-link flying platform. The VTOL is equipped with a controllable multifunctional gripper.

The rest of the paper is organized as follows. Section II presents materials and methods with the detailed descriptions of the proposed DFTS, as well as the dynamics and control issues. Section III presents the experimental demonstrations and results. Finally, Section IV presents the summaries, brief discussions, conclusions and further studies.

II. Materials and Methods

A. *A design concept of the Dexterous Aerial Robotic System*

The proposed DFTS is described as shown in Fig.1, in which the active flying platform VTOL with the controllable multifunctional gripper is connected to the main flying system UAV via a soft link. The lifting force is provided by VTOL, the additional flying platform, which has a smaller mass and the size. Figure 1(c) describes the functional units of the active flying platform VTOL, including the following key elements: a support arm, a universal joint, a self-stabilized gripper, motors, propellers with motors and Electronic Speed Controllers (ECS), and a control unit.

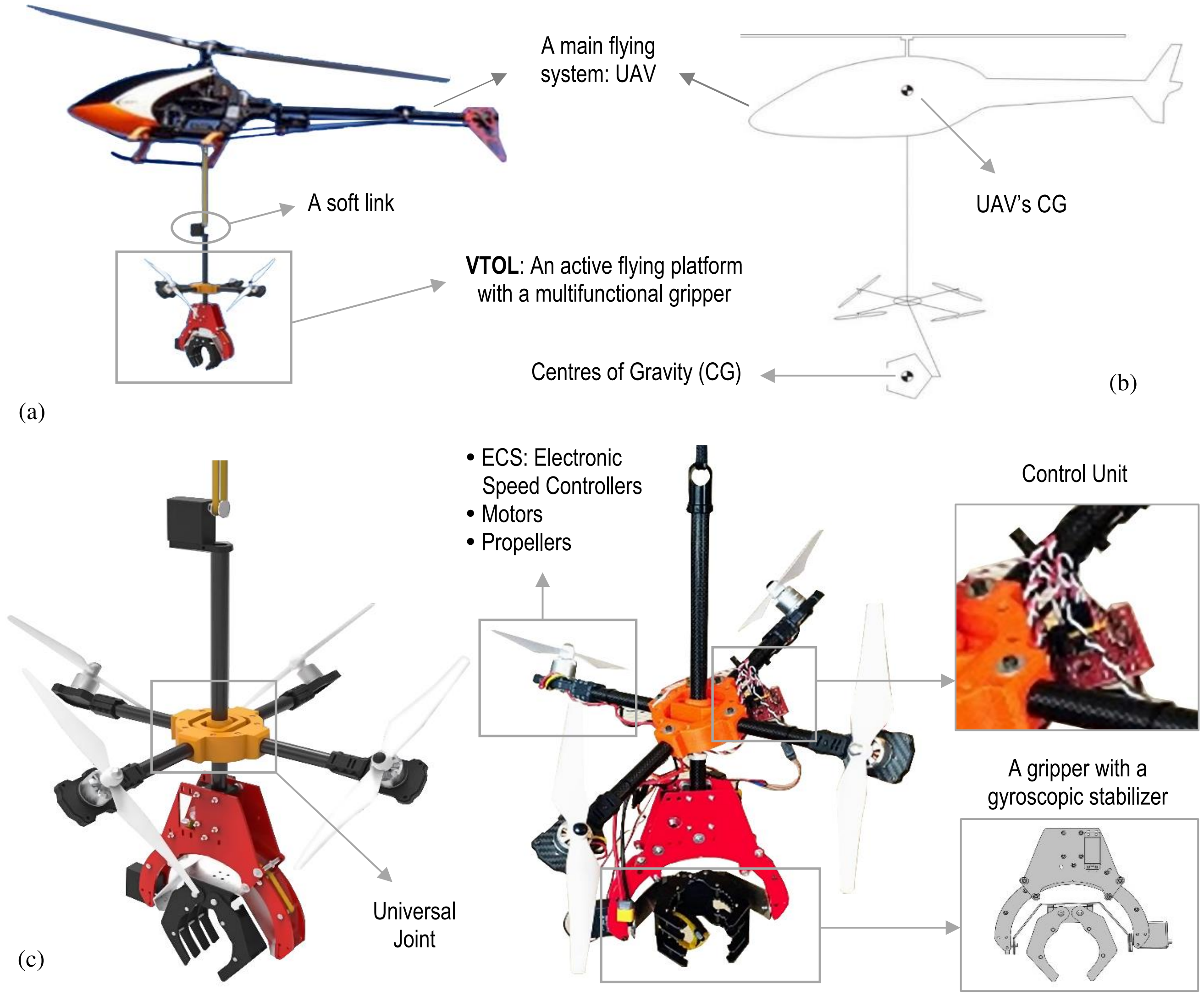

A main flying system: UAV
A soft link
VTOL: An active flying platform with a multifunctional gripper
(a)
UAV's CG
Centres of Gravity (CG)
(b)
• ECS: Electronic Speed Controllers
• Motors
• Propellers
Control Unit
A gripper with a gyroscopic stabilizer
Universal Joint
(c)

Fig.1. (a): A proposed DFTS include the active flying platform VTOL with the controllable multifunctional gripper connected to the main flying system UAV via a soft link. (b): Centers of gravity (CG). (c): Functional units of the active flying platform VTOL.

The DFTS is designed with the main functions of positioning and stabilizing the objects in space with a minimal impact from the dynamic forces and loads during the aerial manipulations. It also means that, the main centers of gravity of the whole DFTS system including the UAV and VTOL, should remain unchanged, or optimally controlled with a minimal impact from the dynamic forces and loads as well as other related variants. It is important to note that the control of the UAV influences the control of the VTOL, and their centers of gravity need to be always located in one vertical axis as described in Fig.1 (b)).

The support structure of the VTOL is a circular carbon tube, in which a multifunctional gripper is fixed to one end, and the other end is the soft link attached to the UAV. The support structure is designed with additional functions of carrying the power supply battery pack and cables.

Figure 2 presents the 3D design of a universal joint which is one of the key elements of the VTOL. To make it convenient for developing the proof-of-concept prototype, the universal joint shown in Fig.1 (c) is a modified base of the UAV. The universal joint can provide the inclinations on the X and Y axes (Figs. 2), with the rotations of being up to +/- 40 degrees.

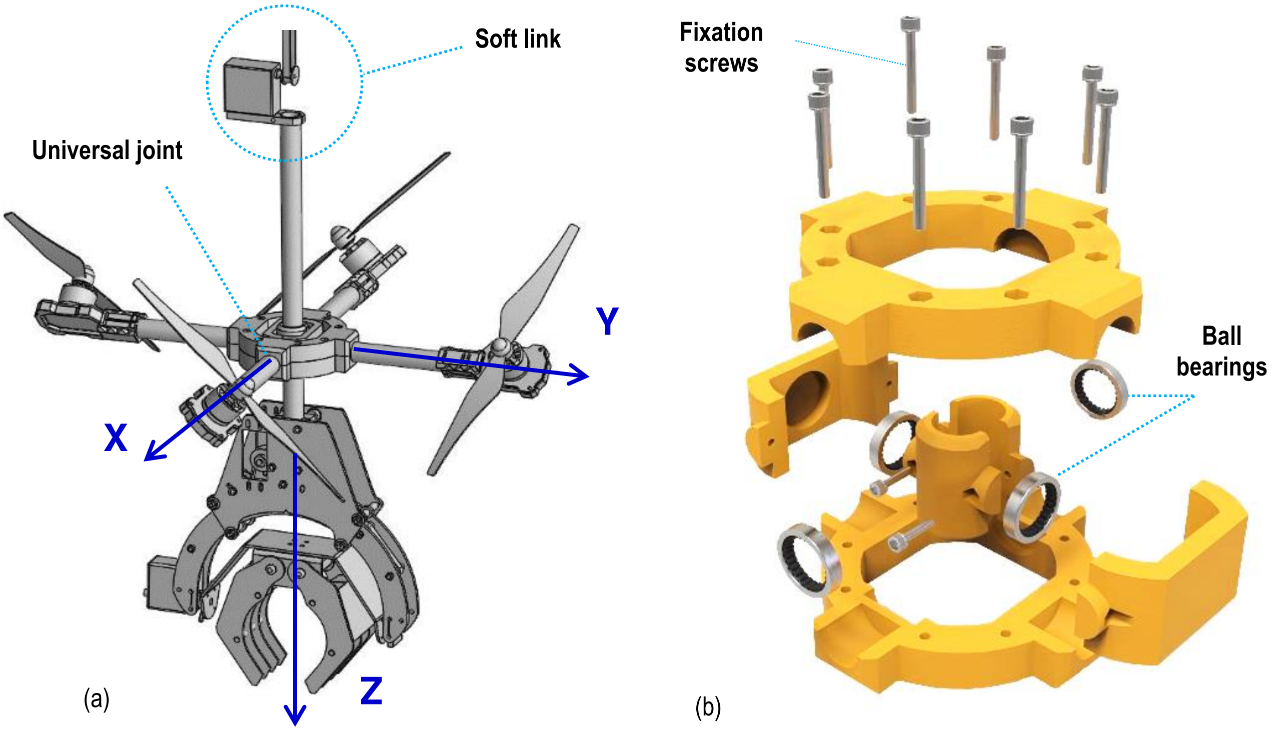


Fig.2. (a): The VTOL with the controllable multifunctional gripper which is connected to the UAV via a soft link. (b): The exploded view with the key parts of the universal joint.

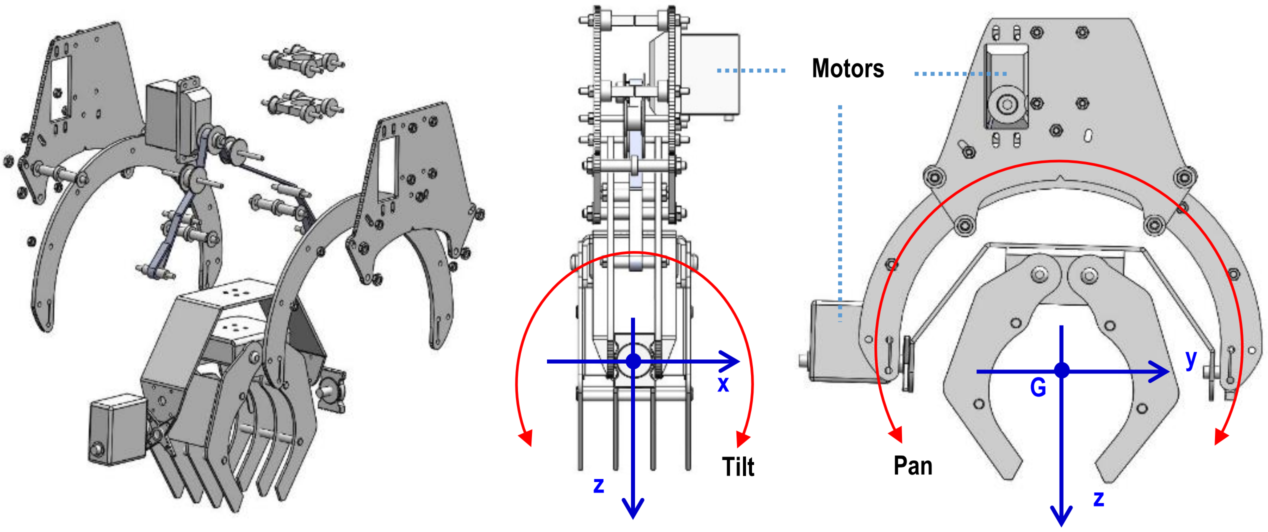


Fig.3. A 3D design and an exploded view of a gripper with two jaws, in which the pan and tilt rotations around x and y axes of the local coordinate system G are operated by servo motors.

Figure 3 presents a 3D design and an exploded view of a gripper with two jaws, which are operated by servo motors. When moving in space, the position of a gripper which is relative to the earth's coordinate system is not fixed. Therefore, the continuous positioning of a gripper is required, and this is accomplished by using a gyroscopic stabilizer, a gyroscopically self-stabilizing unit, which is made of composite materials and metals, and the rotations around x-axis (pan rotation) and y-axis (tilt rotation) of the local coordinate system G (Fig. 3) are operated by servo motors.

The propulsion unit of the active flying platform VTOL include the following elements: four motors, four propellers and one ECS (Electronic Speed Controller). The positioning and movement of the propellers and rotors are done with the following order, based on the main coordinate system XYZ as shown in Figs 3 and 5: (a) Rotor # 1 rotates in the positive direction to the Z axis; it is parallel to the XY plane, and at an angle of -45 degrees to the X axis; (b) Rotor # 2 rotates in the negative direction to the Z axis; it is parallel to the XY plane, and at an angle of -135 degrees to the X axis; (c) Rotor # 3 rotates in the same direction as Rotor # 1; it is parallel to the XY plane, and at an angle of 135 degrees to the X axis; and (d) Rotor # 4 rotates in the same direction as Rotor # 2; it is parallel to the XY plane, and at an angle of 45 degrees to the X axis.

B. Dynamics & Control of the Dexterous Aerial Robotic System

1. Weight and inertial forces

The dynamic behaviour of the proposed the aerial robot DFTS, consisting of the UAV with one or more robotic arms, is closely dependent on the inertial moments and the weight of the elements of the DFTS. This means that the moment of inertia changes significantly and the overall centre of gravity changes constantly, leading to the reaction forces and movements that cause instability of the aerial robot, especially the active flying platform VTOL.

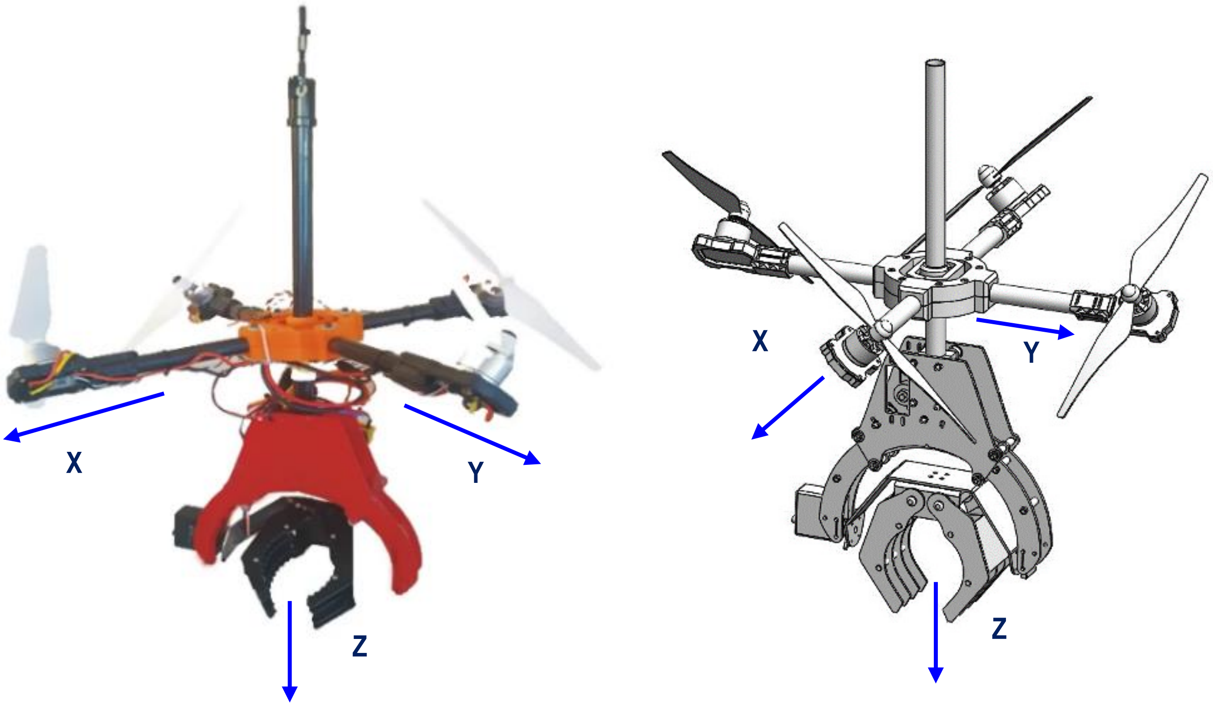


Fig.4. A coordinate system of the active flying platform VTOL with the origin of the coordinate system is centered at CG of the VTOL.

The equations of the dynamic model of the robotic arm with M joints can be derived from the Denavit-Hartenberg formalism; and they can be expressed in a compact matrix form. With the proposed design of the DFTS as described in Section II (A), the lifting force is absorbed by the main flying system (UAV), the active flying platform VTOL can be considered as a flying system with the horizontal orientation only. In this way, mathematically, we have the following equation:

$$\boldsymbol{M_a(q)\ddot{q} + C_a(q,\dot{q})\dot{q} + G_a(q) = \tau}$$

Where $\boldsymbol{M}$ is inertial moment matrix, $\boldsymbol{C}$ is the centrifugal force, $\boldsymbol{g}$ is the vector of gravity.

$$\boldsymbol{q = [q_y\, q_1^T\, q_2^T\, q_m^T]^T}$$

The input signal $\boldsymbol{\tau}$ can be represented as:

$$\boldsymbol{\tau = [\tau_y\, \tau_1^T\, \tau_2^T\, \tau_m^T]^T}$$

Where $\boldsymbol{q = [q_1\, q_2 \ldots q_M]\, T}$ are the angles of deflection of the joints and $\boldsymbol{M_a \in R^{MxM}}$, $\boldsymbol{C_a \in R^M}$ и $\boldsymbol{G_a \in R^M}$ are the values of weight, Coriolis and gravitational forces, respectively. The vector $\boldsymbol{\tau \in R^M}$ is the thrust applied by the motors of each joint.

It is assumed that the whole system works as one. Then we can consider the forces as the sum of the constant inertial moment matrix $\boldsymbol{I_b}$ of the UAV, and the variable inertial moment matrix $\boldsymbol{I_a}$ of the gripper:

$$\boldsymbol{I_{total} = I_b + I_a}$$

It is clearly seen that, the position of the centers of gravity changes, and it depends on the position of the gripper in particular, and the positions of the active flying platform VTOL in general.

2. The coordinate system of the UAV

For convenient computations, the coordinate system of the active flying platform VTOL is shown in Fig.4; the origin of the coordinate system is centered at the center of gravity of the VTOL.

The axis X starts from the center of gravity (CG) and towards the nose of the UAV. The axis Y starts from the center of gravity and to the right of the UAV. The axis Z starts from the center of gravity and downwards from the VTOL following the right-hand rule.

III. Simulations, Demonstrations and Results

The proof-of-concept prototypes of the DFTS in general and the VTOL in particular were virtually and physically developed and tested. The simulations and experiments were implemented, with the focus on the following objectives: (i) To determine the working envelope of the DFTS at different soft link lengths, (ii) To determine the time to reach the service zone when deviating from the Z axis with and without the load; and (iii) To determine the stability of the system when it is operated with a 360-degree rotation of the active flying platform VTOL.

The dynamics of the VTOL and the control of a gyroscopically self-stabilizing unit are simulated with the use of Simulink® (The MathWorks, Inc: www.mathworks.com). To make it convenient for the experiments, the VTOL is operated in the manual mode.

It is noted that, the inclination angle is dependent on the length of the soft link (Fig.1 (a)). The experiments were done to determine the working envelope of the DFTS at different soft link lengths at different soft link lengths, in which the system remains stable. The active flying platform VTOL was practically mounted to a fixed tripod, to ensure the free oscillatory motion of the VTOL in the horizontal plane. The experimental results showed that, the active flying platform VTOL can self-stabilize with the inclination angle of being up to 8 degrees, as shown in Fig. 5.

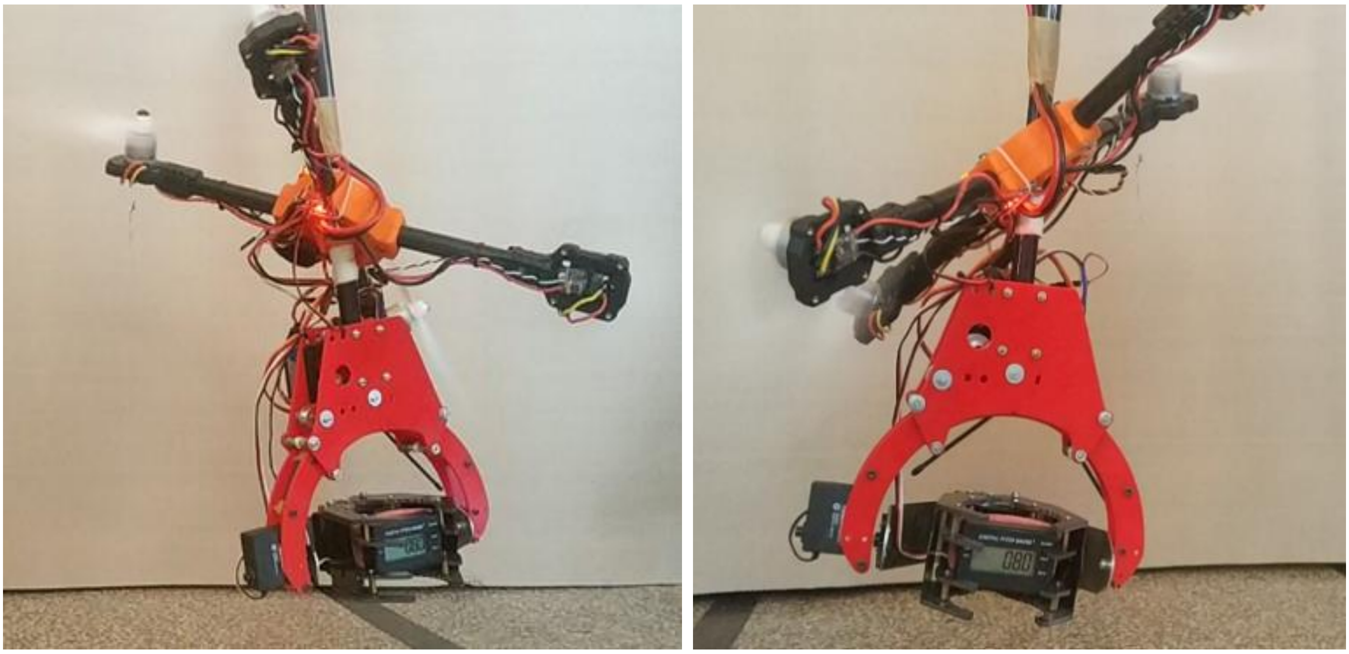


Fig.5. The demonstration of the experiments in which the active flying platform VTOL can self-stabilize with the inclination angle of being up to 8 degrees.

IV. Summary, Discussions and conclusions

It has been well-recognized that the potential applications of aerial robots in industries are huge, including the use of aerial robots in warehousing and manufacturing operations as well as inspections, rescue operations, monitoring and surveillance [3-5, 7-9]. Currently, most of the aerial robots are manually operated and controlled, and there is a growing interest of developing autonomous aerial robots in recent years.

Depending on the specific applications and technical requirements of interacting with objects, the aerial robots can be classified with two main interaction types: the direct interaction and the indirect interaction. The aerial robots with the direct interactions are the robotic systems that are equipped the specific mechanism for a specific type of aerial manipulations, such as the aerial robots which were developed in the European Commission funded project, AEROARMS [9]. There are limitations and challenges when developing aerial robots with the direct interactions. Firstly, there are dynamic dependencies that limit the maximum payload of the aerial robot and the weight of the object that is manipulated by the aerial robot. Secondly, there are the problems caused by the turbulent airflows. Finally, there are the risks of colliding with the rotating elements of the UAV.

To minimize or avoid the above-mentioned weak points of the aerial robots with the direct interactions, there has been a growing interest of developing aerial robots with the indirect interactions, in which an object is remotely or indirectly manipulated with the use of robotic arm(s) or arial manipulation unit(s), which are attached to the UAV via a long flexible or soft link. Such connection solutions via the flexible or soft links make it possible to perform aerial manipulation missions; and it is especially effective and safe, when manipulating objects in difficult and in accessible environments.

In this paper, an innovative development of an aerial robot DFTS, was proposed to indirectly manipulate objects in space. The newly proposed DFTS consists of two independent but interconnected systems or functional units, which have two main separate functions respectively, including (i) a basic carrying function which is actually the main flying system, the unmanned aerial vehicle (UAV), remotely controlled and piloted, and (ii) a precise positioning and stabilization function which is the active flying platform VTOL with 6-DOF arial manipulations, and it is attached to the UAV via the soft link. With the use of a long soft link for the aerial robot with 6-DOF arial manipulations, the problems which are caused by the air turbulent flows generated by the UAV are minimized, and the aerial manipulations of objects are safely controlled and effectively operated. The VTOL which is equipped with a grasping mechanism was successfully developed, prototyped and tested. The experimental results showed that, the developed VTOL can self-stabilize with the inclination angle of being up to 8 degrees.

The outcomes from this study will be used as the foundation to fully develop the aerial robots for the specific applications in industries, with the focus on the following: inspections and reparations of various structures such as wind turbines, power lines and open gas pipelines, decorations and painting of high industrial chimneys and walls of high buildings, transport and delivery of courier shipments, relocation and manipulation of assemblies and units in inaccessible or dangerous environments, warehousing and manufacturing operations in smart factories.

The key challenges and difficulties that need to be solved as well as the future implementations of a study can be summarized as follows: (i) Robust and optimal control of aerial robots with indirect interactions, in which its centers of gravity remains unchanged, and the dynamic impacts from the resultant forces and loads are minimized during the aerial manipulations; (ii) Development of optimal dynamic models of aerial robots when uneven or changing forces and loads are applied on the aerial robot during aerial manipulations; (iii) Simulations and optimizations to enhance the performance of aerial robots, taking into account the issues related to stabilizations during the aerial manipulations, enhancement of mechanical strengths, and energy saving via topology optimization, sustainable design and manufacturing principles; and (iv) Integration with smart systems and collaborative robots for applications in Smart Manufacturing and Industry 4.0.

ACKNOWLEDGMENT

This work has been accomplished with the financial support from the grant number BG05M2OP001-1.002-0011-C02, financed by the Bulgarian Operational Programme "Science and Education for Smart Growth" (SESG) (2014-2020), and co-financed by the European Union through the European Structural and Investment Funds (ESIF). This work was also supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Smart Crop Production” approved by Decision of the Ministry Council №866/26.11.2020 г; and it has been partly supported for sustainable and collaborative research partnerships by the Vingroup Innovation Foundation (VINIF) under the grant number VINIF.2020.NCUD.DA059, the research foundation of Le Quy Don Technical University, and Thu Dau Mot University, Vietnam.

REFERENCES

[1] Chi Hieu Le *et al.* (2020) Challenges and conceptual framework to develop heavy-load manipulators for smart factories. International Journal of Mechatronics and Applied Mechanics, 2020 (8), 209-216.

[2] Daniel Arey *et al.* (2021) Lean industry 4.0: a digital value stream approach to process improvement. Procedia Manufacturing, 54, 2021, 19-24.

[3] AbdullatifBaba (2022) A new design of a flying robot, with advanced computer vision techniques to perform self-maintenance of smart grids. Journal of King Saud University - Computer and Information Sciences: 34 (5), 2022, 2252-2261.

[4] Nursultan Imanberdiyev *et al.* (2022) Design, development and experimental validation of a lightweight dual-arm aerial manipulator with a COG balancing mechanism. Mechatronics: 82, 2022, 102719.

[5] Jian Liu *et al.* (2021) A multi-finger robot system for adaptive landing gear and aerial manipulation. Robotics and Autonomous Systems: 146, 2021, 103878.

[6] Anh My Chu *et al.* (2020)) A novel mathematical approach for finite element formulation of flexible robot dynamics. Mechanics Based Design of Structures and Machines, An International Journal, 50 (11), 2022, 3747-3767.

[7] Alejandro Suarez *et al.* (2018) Design of a lightweight dual arm system for aerial manipulation. Mechatronics 50, 2018, 30–44.

[8] Hai-N. Nguyen *et al.* (2018) A novel robotic platform for aerial manipulation using quadrotors as rotating thrust generators, IEEE Trans. Robot 34 (2), 2018, 353–369.

[9] AEROARMS (2022) Aerial robotic system integrating multiple arms and advanced manipulation capabilities for inspection and maintenance. Available at https://cordis.europa.eu/project/id/644271 [Access: 9/2022].

[10] Anh My Chu *et al.* (2020) Kinematic and dynamic modelling for a class of hybrid robots composed of m local closed-loop linkages appended to an n-link serial manipulator. Applied Sciences 10(7), 2567.

[11] Anh My Chu *et al.* (2019) An efficient finite element formulation of dynamics for a flexible robot with different type of joints. Mechanism and Machine Theory, 134, 2019, 267-288.

[12] Anh My Chu (2019) Novel robot arm design and implementation for hot forging press automation. International Journal of Production Research, 57 (14): 4579-4593.

[13] Arey Daniel *et al.* (2019) An Investigation into the Adoption of Automation in the Aerospace Manufacturing Industry. In: Advances in Manufacturing Technology XXXIII. IOS Press, 87-92.

[14] Minh Duc Vu *et al.* (2021) a conceptual digital twin for cost-effective development of a welding robotic system for smart manufacturing. Lecture Notes in Mechanical Engineering. Springer Nature, 1018-1025.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 55–61 DOI: 10.15439/2022R36 ISSN 2300-5963 ACSIS, Vol. 33

Development of Ensemble Tree Models for Generalized Blood Glucose Level Prediction

Aashima
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
aashimakodan@gmail.com

Shashank Bhargav
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
shashankbhargav55@gmail.com

Shruti Kaushik
RxDataScience, Inc.
USA – 277709
shrutikaushik15@gmail.com

Varun Dutt
Indian Institute of Technology Mandi
Kamand, Himachal Pradesh, India – 175005
varun@iitmandi.ac.in

***Abstract*—Type-1 diabetes (T1D) patients must carefully monitor their insulin doses to avoid serious health complications. An effective regimen can be designed by predicting accurate blood glucose levels (BGLs). Several physiological and data-driven models for BGL prediction have been designed. However, less is known on the combination of different traditional machine learning (ML) algorithms for BGL prediction. Furthermore, most of the available models are patient-specific. This research aims to evaluate several traditional ML algorithms and their novel combinations for generalized BGL prediction. The data of forty T1D patients were generated using the Automated Insulin Dosage Advisor (AIDA) simulator. The twenty-four hour time-series contained samples at fifteen-minute intervals. The training data was obtained by joining eighty percent of each patient's time-series, and the remaining twenty percent time-series was joined to obtain the testing data. The models were trained using multiple patients' data so that they could make predictions for multiple patients. The traditional non-ensemble algorithms: linear regression (LR), support vector regression (SVR), k-nearest neighbors (KNN), multi-layer perceptron (MLP), decision tree (DCT), and extra tree (EXT) were evaluated for forecasting BGLs of multiple patients. A new ensemble, called the Tree-SVR model, was developed. The BGL predictions from the DCT and the EXT models were fed as features into the SVR model to obtain the final outcome. The ensemble approach used in this research was based on the stacking technique. The Tree-SVR model outperformed the non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and other novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN). This research highlights the utility of designing ensembles using traditional ML algorithms for generalized BGL prediction.**

***Index Terms*—diabetes, time-series, generalization, machine learning, stacking**

I. Introduction

Diabetes mellitus is a major global health concern as it is growing rapidly [1]. It has been reported that the adult population suffering from diabetes has tripled in the past two decades [1]. In 2019, this number was estimated to be 9.3%, which was an astounding 483 million of adults aged 20-79 years [1].

The normal blood glucose levels (BGLs) in a healthy adult after 8 hours of fasting should be between 70 mg/dl and 100 mg/dl [2]. The human pancreas maintains BGLs in this narrow range by releasing glucagon and insulin. Type-1 diabetes (T1D) is an incurable metabolic disorder characterized by high BGLs, caused by low or no insulin production by the pancreas. Complications due to T1D include cardio-vascular diseases, nerve, kidney, and eye damage. However, it has been shown that proper management can significantly reduce the complications and high costs related with diabetes [3]. T1D patients commonly use finger prick tests to measure their BGLs and adjust insulin doses multiple times in a day [3]. Diabetes management is incredibly difficult due to data inadequacy and, in some patients, due to improper data interpretation [3]. Thus, good BGL prediction models can provide great value for T1D patients.

The literature for BGL prediction contains physiological models [4][5], neural networks (NNs) [3] [6]-[21] and traditional machine learning (ML) algorithms [22]-[26]. Physiological models are pretty accurate but they require deep understanding of glucose metabolism, as the model parameters should be set only by an expert [2]. Therefore, data-driven models have an advantage that they can be used by individuals without any expert knowledge.

The major contributions of this study include development of novel combinations of traditional ML algorithms, and BGL prediction for multiple patients by using generalized models. Few research studies have combined different traditional ML algorithms using an ensemble approach. Furthermore, most of the existing research is patient-specific i.e., the models were trained to predict BGLs for a single patient at a time. The goal of this study is to evaluate several traditional ML algorithms for generalized BGL prediction of T1D patients, and combine them using an ensemble approach based on the stacking technique. Specifically, this research proposes an ensemble (Tree-SVR) of decision tree (DCT), extra tree (EXT) and support vector regression (SVR). The predictions from the DCT and the EXT models were fed as inputs to the SVR model to obtain final predictions. The Tree-SVR model is compared with traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and other novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN).

This paper is ordered as: mention the literature in BGL prediction, data description, a brief explanation of the numerous regression models used in this study and their detailed calibration process, the results from different non-ensemble and ensemble models, discussions, conclusion and the implications of this study.

II. Background

Commonly used physiological models include the Meal Simulation Model of the Glucose-Insulin System [5] and the Automated Insulin Dosage Advisor (AIDA) [4].

Hamdi et al. [6], Pappada et al. [7], Asad et al. [11], and Kushner et al [12] utilized NNs for predicting BGLs of T1D patients. They predicted BGLs in the horizon of 15 minutes, 50-80 minutes and 60-240 minutes, respectively. Recurrent neural networks (RNNs) were implemented by Sandham et al. [8][9] for providing short-term therapy to T1D patients. Martinsson et al. [10] utilized a RNN which was trained to learn parameters of an uni-variate Gaussian output distribution, for making BGL predictions up to an hour. Munoz-Organero [13] implemented a RNN based on long short-term memory (LSTM) cells on AIDA data. Rabby et al. [14] used stacked LSTM with Kalman smoothing for predicting BGLs of 6 T1D patients. Genetic algorithm (GA) was applied on the output of RNN, LSTM, bidirectional LSTM, stacked LSTM, and gated recurrent unit to achieve improved performance by Kim et al. [15].

Zhu et al. [16] introduced a dilated RNN, which was shown to outperform autoregressive models (ARX), conventional NNs and SVR model. Li et al. [17] developed a combination of incremental learning with echo state networks (ESN) and feedback ESNs, which was shown to perform better than conventional methods. The ESN belongs to the RNN family. Wang et al. [18] showed that their proposed LSTM network optimized with improved particle swarm optimization (IPSO) and variational modal decomposition (VDM) performed better than LSTM, VDM-LSTM, VDM-PSO-LSTM for 56 diabetes patients. Zhu et al. [19] used convolutional neural network (CNN) and RNN to design a generative adversarial network (GAN) for BGL prediction.

Assadi et al. [20] implemented extended Kalman filter (EKF), ARX and extreme learning machine (ELM) for predicting BGLs predictions of 20 AIDA patients. The Levenberg-Marquardt algorithm was used by Robertson et al. [3] to train an Elman RNN model for making BGL predictions. A hybrid model based on grammatical evolution (GE) and physiological models was designed by Contreras et al. [2].

Robertson et al. [3] and Munoz-Organero [13] showed that their models were trained by using only one patient's data, and they did not perform well for predicting BGLs of other patients. Thus, it is imperative to use data of multiple patients to account for inter-individual variability [3]. Monte-Moreno [21] presented a system for simultaneous noninvasive blood pressure and BGL estimation. The models, designed using photoplethysmography (PPG) and ML techniques, did not need calibration over time and patients. Pappada et al. [7] developed generalized NNs that were trained on 11-17 patients' data, and evaluated using the remaining unseen data that was not included in the NN calibration.

Georga et al. [23] evaluated SVR models on 12 real patients' data. Hamdi et al. [22] looked into differential equations (DE) and SVR for predicting BGLs of 12 real patients. Another patient-specific SVR model was designed by Plis et al. [24] and Bunescu et al. [25], where input features were generated using a standard physiological model of BGL dynamics. Monte-Moreno [21] evaluated ridge linear regression (LR), random forest (RF), multi-layer perceptron (MLP), and SVR models for BGL prediction. Mordvanyuk et al. [26] implemented patient-specific k-nearest neighbors (KNN) model for sequential T1D data, considering only carbohydrates intake, bolus dose and preprandial BGL as input features. Maged et al. [27] evaluated several ML and DL patient-specific models, and found that that EXT regressor and ANN performed the best.

Saiti et al. [28] evaluated ensemble algorithms: linear, bagging and boosting meteregressor to show that they performed better than the individual component models for BGL prediction. Ma et al. [29] combined the residual compensation network (RCN) and the autoregressive moving average (ARMA) model for predicting BGLs in the horizon of 30 minutes and 60 minutes. Xie et al. [30] compared the performance of several models: Elastic Net, gradient boosting trees, RF, SVR, vanilla LSTM, temporal convolutional network (TCN) etc. with classical ARX model for predicting BGLs of T1D patients.

Recent studies on BGL prediction are moving towards data-driven models [2]. However, to the best of our knowledge, few researches have combined different traditional ML algorithms using an ensemble approach. Most of the existing research is patient-specific i.e., the models were trained for an individual and used to obtain predictions for the same. Furthermore, very less is known on the usage of EXT algorithm for BGL prediction. In this research, the stacking ensemble approach was used with a slight variation. The predictions from DCT and EXT models were fed as input to other non-ensemble models (LR, SVR, KNN, and MLP) to obtain the final predictions. BGLs for multiple patients can be predicted by using these models.

III. Methodology

A. Data

The AIDA simulator [31] was used to generate the data used in this research. It is a freeware simulator to observe the effects of glucose-insulin interaction in T1D patients [31]. It assumes that the patient's pancreas produces no insulin [31]. The program has been designed for teaching, demonstration and self-learning purposes [31]. The AIDA model provides interactive virtual patient scenarios by using comprehensive glucose and insulin sub-models based on mathematical differential equations [31]. The user can change the input variables

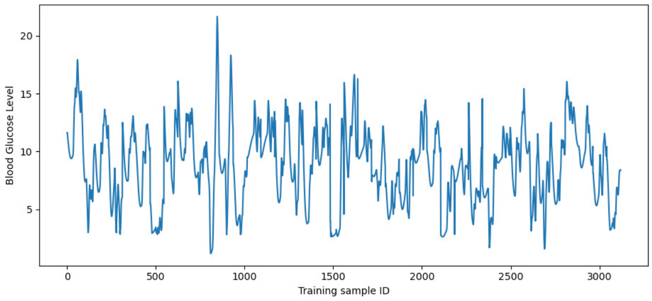


Fig. 1. Blood glucose levels of all training samples

for a patient scenario, and immediately see the impact on their BGLs [31].

The 24-hour time-series data for 40 T1D patients contained BGLs at 15-minute intervals. The BGL depended on 13 input features: patients' weight (kilograms), timestamp (hours) (0-24), carbohydrates intake (grams), short-acting injection (units) (effect in 2-5 hours), intermediate and long-acting injection (units) (effect in 24-48 hours), type of medication (units), kidney functioning renal glucose threshold (RTG) (mmol/l) (the kidneys start to excrete glucose into urine when blood glucose concentration reaches RTG), kidney function renal (ml/min) (a measure of of how well kidneys operate), liver insulin sensitivity (mmol/l), lower and upper glucose limit. The injections were taken by patients 15 minutes prior to their meal to lower BGLs.

There were 97 samples in the time-series data for each patient. The training data was obtained by stitching the initial eighty percent of each time-series. Similarly, the testing data was obtained by stitching the remaining twenty percent of each time-series. This resulted in 3120 training samples and 760 testing samples. The time-series for different patients were stitched because we aimed to develop a generalized model. Instead of personalized models, a single model should be able to learn and predict BGLs for multiple patients. Figure 1 shows the stitched training data, where the y-axis represent the BGLs, which are highly fluctuating in nature.

B. Error Metric

In this research we aimed to estimate BGL, which is a real value. The distance between the prediction values and the actual values is used to determine the quality of a regression model. Root mean squared error (RMSE) [32], which is a popular metric for regression analysis, was utilized in this study. The RMSE is obtained by taking the square root of the mean of squared differences between all the actual and predicted values. The error is defined by

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(y_i - y_i')^2}{n}} \tag{1}$$

where n is the total number of samples, y_i is the predicted value and y_i' is the actual value. The results obtained from different models for different sets of hyperparameters were compared with each other by using RMSE.

C. Machine Learning Models

This research implemented traditional ML non-ensemble models and combined them with each other using an ensemble approach. Scikit-learn in Python [33] was used to implement all models except MLP. MLP was implemented using Keras in Python [34]. Model performance is influenced by the choice of hyperparameters [35]. Therefore, all models were trained with diverse sets of hyperparameters. The hyperparameters which obtained the lowest RMSE on the training data were chosen. The random seed for all algorithms was fixed, so that the results could be reproduced. The calibration process is explained as follows.

1) Linear Regression: It is a supervised algorithm that targets to find the best fit line, which has the least total error from all data points [36]. The error is the distance of a data point (training sample) from the line. The line in n-dimensional space is parameterized by n coefficients [36]. There are no main hyperparameters.

2) K-Nearest Neighbors: It is a supervised algorithm that stores all the training samples and estimates the outcome for a testing sample by using the target values of the K nearest neighbors [37]. The nearest neighbors among the training samples are calculated using their distance from the test sample.

The main hyperparameters are the number of nearest neighbors (K) and the distance metric [37]. The distance metric was varied as Euclidean, Manhattan, and Minkowski. The value of K was varied 5-99 (in steps of 1).

3) Support Vector Regression: It is a supervised algorithm which is based on the support vector machines (SVMs). The SVM model aims to fit a hyper-plane in the higher-dimensional feature space, such that the margin (minimum distance) of class boundaries is maximized [38]. The model is penalized by an objective function upon misclassification or if a sample lies within the margin [38]. The support vector regression (SVR) algorithm uses this principle for regression problems.

The main hyperparameters are kernel, gamma, and the degree of the regularization [39]. The kernel specifies the shape of the hyper-plane and gamma specifies the kernel coefficient [39]. The degree of regularization (C) is used to control overfitting on training data. The kernel was varied as gaussian, sigmoid, and polynomial. For gaussian kernel, C was varied 0.1-2.0 (in steps of 0.1), 10-1000 (in steps of 10), and 1000-7000 (in steps of 1000), respectively. For sigmoid kernel, C was varied 0.1-2.0 (in steps of 0.1). For polynomial kernel, degree was varied 1-10 (in steps of 1). For the three degree values that obtained the lowest training RMSE, C was varied 0.1-2.0 (in steps of 0.1), 10-1000 (in steps of 10), and 1000-7000 (in steps of 1000). The gamma parameter was set to 'scale', which is the default value in scikit-learn [39].

4) Multi-layer Perceptron: A multi-layer perceptron is a feed-forward fully-connected NN containing at least three layers: an input layer, a hidden layer, and an output layer [40].

A layer is an array of perceptrons. A perceptron is a simple computational unit that calculates output from weighted inputs by applying a non-linear activation function [41]. The weights are randomly initialized and then updated by backpropagation [40]. The aim is to minimize the gap between actual and predicted output [40].

The various hyperparameters include number of hidden layers, number of nodes in each layer, learning rate, activation function, error metric, regularization penalty, and the training algorithm [40][41]. Each training algorithm has its own additional hyperparameters. The learning rate is used to determine the step size of weight updation and regularization is used to control overfitting on the training data [41].

We evaluated three MLP models. The first layer for each model was the input layer containing 13 nodes, because each sample had 13 input features. The specifications for each layer are described as [number of nodes, activation function], and the different layers are written in the order of occurrence. The first model had 3 layers - [13,none], [32,relu] and [1,relu]. The second model had 4 layers - [13,none], [32,tanh], [64,tanh] and [1,relu]. The third model had 5 layers - [13,none], [32,tanh], [64,tanh], [32,tanh] and [1,relu]. The training algorithm (optimizer), weight initializer, and regularizer in each model were chosen to be adam [42], glorot normal [43] and L1-L2 [44], respectively. For each model, number of epochs was varied as 10, 50, 100, and 200, with and without regularization.

5) Decision Tree: It is a supervised algorithm [45]. It has a hierarchical, tree structure with decision nodes and branches [45]. An attribute is associated with each decision node, and the node splits into two or more branches [45]. A set of attribute value(s) is associated with each branch [45]. The target value is placed at the leaf node, which has no further branches [45]. During training, the data is broken into smaller subsets at each node [45]. The aim is to attain maximum homogeneity at each decision node [45]. To call a subset as completely homogeneous, it should contain instances with similar values [45]. For obtaining the output for a test sample, the algorithm travels down the tree by following the decision rules present at each node.

Maximum depth and minimum split are the main hyperparameters [46]. The maximum distance i.e., the number of branches from the root node to a leaf node of the tree, is called the maximum depth [46]. The minimum number of samples that must be available at each node to attempt a split is called the minimum split [46]. Unpruned and fully grown trees can be obtained by using the default values of these parameters [46]. The maximum depth was varied 1-50 (in steps of 1), for each minimum split value of 2, 3, and 4.

6) Extra Tree: An extremely randomized tree is similar to DCT, but while splitting the training dataset at each decision node, random splits are drawn from the training data present at that node and the candidate attributes are also selected randomly [47].

Maximum depth and minimum split are the main hyperparameters [47]. The maximum depth was varied 1-50 (in steps of 1), for each minimum split value of 2, 3, and 4.

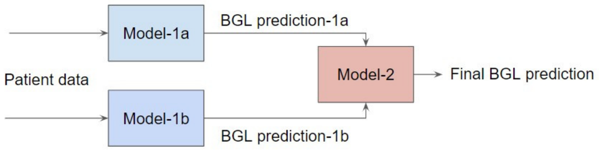


Fig. 2. The ensemble structure used in this research.

7) Ensemble: In this research, we designed ensembles inspired by the stacking approach. Stacking is an ensemble approach in which the predictions from multiple base models are aggregated using a meta-model [48]. The meta-model is usually a naive algorithm [48]. The model is trained using k-fold cross-validation [48]. The $k-1$ folds are used for training the base models, and the meta-model is trained by using the remaining 1-fold [48]. The ensemble designed in this research is slightly different: the meta-model is not a naive algorithm, and k-fold cross validation is not used for training.

Figure 2 is a visual representation of the ensemble structure. The first level (level-1) contains the base models (model-1a and model-1b), and the second level (level-2) contains the meta-model (model-2). The level-1 models were first trained on the training data. Then, the obtained predictions were used to train the level-2 model. For a test sample, the predictions obtained from level-1 models were fed into level-2 model to obtain the final outcome.

The calibrated non-ensemble models were selected to be the constituent models of an ensemble. The best two models were selected as model-1a and model-1b, and model-2 was varied as the remaining models. Since the optimal parameters were already known for model-1a and model-1b from the previous step, the intermediate level-1 predictions in the ensemble were fixed. Thus, only model-2 required calibration. The model-2 was trained to estimate the final BGL value by using the BGL estimates calculated by model-1a and model-1b as input features.

IV. Results

The traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) were calibrated to obtain the optimal values for hyperparameters. In the KNN model, the optimal hyperparameters were observed to be Manhattan distance metric with K as 5. In the SVR model, the train RMSE obtained by the gaussian kernel decreased monotonically for increasing values of C. The train RMSE obtained by the sigmoid kernel increased with increasing values of C. The train RMSE obtained by the polynomial kernel was nearly the same for degrees 6, 7 and 8. For each degree value, the train RMSE was observed to decrease monotonically for increasing values of C. The optimal hyperparameters were observed to be the gaussian kernel with C as 7000. In the DCT model, a minimum split of 2 achieved the best training results. For maximum depth greater than 20, the train RMSE was observed to be constant. Therefore, 20 was chosen as the optimal maximum depth. In the EXT model, a minimum split of 2 achieved the best training results. For maximum depth greater than 25, the train RMSE was observed to be constant. Therefore, 25 was chosen as the optimal maximum depth. The three MLP models obtained the best results at 200 epochs with regularization. The second MLP model with 4 layers obtained the lowest train RMSE among all the evaluated MLP models.

Table 1 presents the performance of traditional non-ensemble models (LR, SVR, KNN, MLP, DCT, and EXT) and the optimal hyperparameters obtained by calibration. Table 2 presents the performance of novel Tree variants (Tree-LR, Tree-MLP, and Tree-KNN, and Tree-SVR). The results are presented in an increasing order of test RMSE. Figure 5 presents the plot of the actual BGLs and the predicted BGLs by the Tree-SVR ensemble model for all the testing samples.

TABLE I
PERFORMANCE AND OPTIMAL HYPERPARAMETERS FOR NON-ENSEMBLE MACHINE LEARNING MODELS

Model	Optimal hyperparameters	Train RMSE	Test RMSE
DCT	maximum depth = 20, minimum split = 2	0.202	2.207
EXT	maximum depth = 25, minimum split = 2	0.199	2.207
KNN	K = 5, distance metric = Manhattan	0.667	2.374
MLP	4 layers = [13,none], [32,tanh], [64,tanh], [1,relu]	2.050	2.507
SVR	kernel = gaussian, gamma = scale, C = 7000	2.160	2.642
LR	None	2.908	2.777

TABLE II
PERFORMANCE OF DESIGNED ENSEMBLE MODELS

Model-1a	Model-1b	Model-2	Train RMSE	Test RMSE
DCT	EXT	SVR	0.202	2.201
DCT	EXT	KNN	0.201	2.206
DCT	EXT	LR	0.199	2.207
DCT	EXT	MLP	0.224	3.139

V. DISCUSSIONS

The DCT model and the EXT model outperformed other models (LR, SVR, KNN, and MLP) by obtaining equal test RMSEs of 2.207. Because of the considerable gap in train and test RMSE, the DCT, EXT, and KNN models can be considered to have overfitting. The SVR and MLP models can be said to have little or no overfitting. On the other hand, the LR model can be considered to have underfitting as it had higher train RMSE as compared to test RMSE. It performed poorly with highest train and test RMSEs of 2.908 and 2.777, respectively. This was because the LR algorithm finds a linear relationship, whereas the BGLs are highly fluctuating (Fig. 1).

The DCT and the EXT models performed the best among the non-ensemble models, therefore, they were chosen as level-1 models for the ensembles. The level-2 model was varied as LR, SVR, KNN and MLP. These novel ensembles were named the Tree variants (Tree-LR, Tree-KNN, Tree-MLP, and Tree-SVR). Because of the considerable gap in train and test RMSE, all the Tree variants can be considered to have overfitting. The Tree-KNN and the Tree-SVR models performed slightly better than the DCT and the EXT models. They obtained test RMSEs of 2.206 and 2.201, respectively. The Tree-SVR model performed the best with a test RMSE of 2.201. The improvement in results from the non-ensemble models to the novel ensemble is slight. However, it shows a promise that combination of traditional ML algorithms with each other can obtain better BGL predictions, thus provide value to T1D patients.

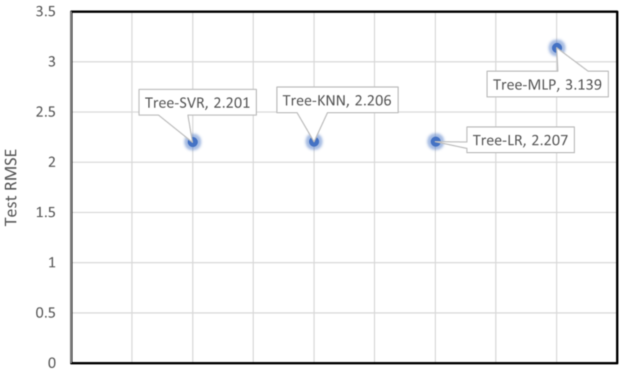


Fig. 3. Testing performance of designed ensemble models

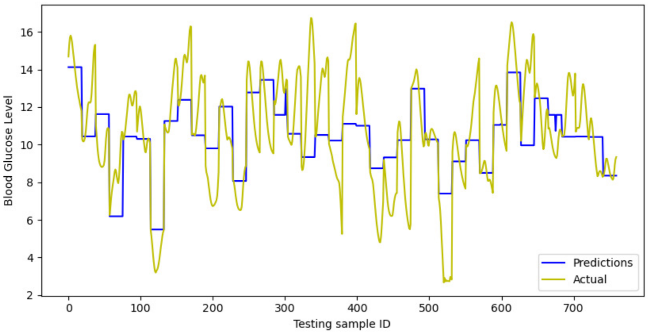


Fig. 4. Predicted and actual blood glucose levels for Tree-SVR ensemble model

The models were evaluated and tested on less amount of data, which is one of the limitations of this study. There were only 3120 training samples and 760 testing samples. The smaller number of samples did not allow validation data to be obtained. Therefore, hyperparameter tuning was performed by using only training data, which is not an ideal choice. The AIDA simulator is an educational program, thus, the use of virtual patients' data is another limitation of this research. However, the BGL values were oscillating. Therefore, the proposed Tree model can be called scalable, and it will be able to model similar data for real patients.

VI. CONCLUSIONS

Data-driven models for BGL prediction aim to assist in designing an effective regimen, thus, preventing serious health complications associated with diabetes. This study evaluated traditional non-ensemble models: linear regression (LR), multi-layer perceptron (MLP), support vector regression (SVR), k-nearest neighbors (KNN), decision tree (DCT), and extra tree (EXT) for generalized BGL prediction of type-1 diabetes (T1D) patients. Novel ensembles were designed, inspired by the stacking approach, where the predictions from the DCT and the EXT models were fed to a non-ensemble model for final BGL prediction. The AIDA simulator was used to generate 24-hour data of 40 virtual patients [31]. Eighty percent of each time-series was stitched together for training generalized models. The root mean squared error (RMSE) was used to gauge model performance.

The DCT and the EXT models outperformed the other non-ensemble models (LR, SVR, KNN, and MLP). The ensemble (Tree-SVR) of DCT, EXT and SVR outperformed all the non-ensemble models and the Tree variants (Tree-LR, Tree-KNN, and Tree-MLP) evaluated in this research.

This research has various implications for T1D patients. The patients' inability to interpret data often compromises diabetes management [3]. The models proposed in this research can help individuals in predicting future BGLs without expert knowledge about the model and glucose metabolism. Neural networks (NNs) are widely used in research studies for diabetes and BGL prediction. However, NNs often require large datasets to give promising results, and deep learning can be expensive [49]. The state-of-the-art NNs are often trained on data containing thousands, or even millions of observations [49]. This study has used only traditional ML algorithms, which can work well even with datasets containing smaller number of samples[49]. Few studies have combined different traditional ML algorithms with each other for BGL prediction; but this study has developed an ensemble approach with these algorithms. Majority of the available studies have been patient-specific, and the developed models calculated predictions on an individual basis. However, this study has developed generalized models that were shown to predict BGLs of multiple patients.

Moving forward, this research can be conducted by using lesser number of input features. The models can be evaluated on larger T1D datasets with real patients. Motivated by the improvement shown by ensemble models, more traditional ML algorithms can be combined with each other using several ensemble approaches for BGL prediction.

ACKNOWLEDGMENT

This research was supported by a research grant from RxDataScience, Inc., USA, to the Indian Institute of Technology Mandi, India. The project was supported by grants (award: IITM/CONS/RxDSI/VD/33) to Varun Dutt.

REFERENCES

[1] IDF Diabetes Atlas, 9th edition (2019). http://www/diabetesatlas.org

[2] Iván Contreras, Silvia Oviedo, Martina Vettoretti, Roberto Visentin, and Josep Vehí. 2017. Personalized blood glucose prediction: A hybrid approach using grammatical evolution and physiological models. PLOS ONE 12 (11 2017), e0187754. https://doi.org/10.1371/journal.pone.0187754

[3] Gavin Robertson, Eldon Lehmann, William Sandham, and David Hamilton. 2011. Blood Glucose Prediction Using Artificial Neural Networks Trained with the AIDA Diabetes Simulator: A Proof-of-Concept Pilot Study. J. Electrical and Computer Engineering 2011 (05 2011). https://doi.org/10.1155/2011/681786

[4] E. D. Lehmann and T. Deutsch. 1992. A physiological model of glucose-insulin interaction in type 1 diabetes mellitus. Journal of biomedical engineering 14 3 (1992), 235–42.

[5] S. M. Lynch and B. W. Bequette. 2001. Estimation-based model predictive control of blood glucose in type I diabetics: a simulation study. Proceedings of the IEEE 27th Annual Northeast Bioengineering Conference (Cat. No.01CH37201) (2001),79–80.

[6] T. Hamdi, J. Ben Ali, N. Fnaiech, V. Di Costanzo, F. Fnaiech, E. Moreau, and J.Ginoux. 2017. Artificial neural network for blood glucose level prediction. In 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C). 91–95. https://doi.org/10.1109/SM2C.2017.8071825

[7] Scott Pappada, Brent Cameron, and Paul Rosman. 2008. Development of a neural network for prediction of glucose in type I diabetes patients. Journal of diabetes science and technology 2 (09 2008), 792–801. https://doi.org/10.1177/193229680800200507

[8] Sandham WA, Nikoletou D, Hamilton DJ, Patterson K, Japp A, MacGregor C. Blood glucose prediction for diabetes therapy using a recurrent artificial neural network. In: Proceedings, EUSIPCO-98, IX European Signal Processing Conference, Rhodes Is-land, Greece, 1998; Vol. 11: pp. 673-676.

[9] W. A. Sandham, D. J. Hamilton, A. Japp and K. Patterson, "Neural network and neuro-fuzzy systems for improving diabetes therapy," Proceedings of the 20th Annual Inter-national Conference of the IEEE Engineering in Medicine and Biology Society. Vol.20 Biomedical Engineering Towards the Year 2000 and Beyond (Cat. No.98CH36286), Hong Kong, China, 1998, pp. 1438-1441 vol.3, doi: 10.1109/IEMBS.1998.747154.

[10] John Martinsson, Alexander Schliep, Björn Eliasson, and Olof Mogren. 2020. Blood Glucose Prediction with Variance Estimation Using Recurrent Neural Networks. Journal of Healthcare Informatics Research 4 (03 2020). https://doi.org/10.1007/s41666-019-00059-y

[11] Muhammad Asad, Usman Qamar, Babar Zeb, Aimal Khan, and Younas Khan. 2019. Blood Glucose Level Prediction with Minimal Inputs Using Feedforward Neural Network for Diabetic Type 1 Patients (ICMLC '19). Association for Computing Machinery, New York, NY, USA, 182–185. https://doi.org/10.1145/3318299.3318354

[12] Taisa Kushner, Marc D. Breton, and Sriram Sankaranarayanan.Multi-Hour Blood Glucose Prediction in Type 1 Diabetes: A Patient-Specific Approach Using Shallow Neural Network Models.Diabetes Technology & Therapeutics.Dec 2020.883-891.

[13] Mario Munoz-Organero. 2020. Deep Physiological Model for Blood Glucose Prediction in T1DM Patients. Sensors 20 (07 2020), 3896. https://doi.org/10.3390/s20143896

[14] Rabby, M.F., Tu, Y., Hossen, M.I. et al. Stacked LSTM based deep recurrent neural network with kalman smoothing for blood glucose prediction. BMC Med Inform Decis Mak 21, 101 (2021). https://doi.org/10.1186/s12911-021-01462-5

[15] Kim, Dae-Yeon &Choi, Dong-Sik & Kang, Ah & Woo, Jiyoung & Han, Yechan & Chun, Sung Wan & Kim, Jaeyun. (2022). Intelligent Ensemble Deep Learning System for Blood Glucose Prediction Using Genetic Algorithms. Complexity. 2022. 1-10. 10.1155/2022/7902418.

[16] Zhu, T., Li, K., Chen, J. et al. Dilated Recurrent Neural Networks for Glucose Forecasting in Type 1 Diabetes. J Healthc Inform Res 4, 308–324 (2020). https://doi.org/10.1007/s41666-020-00068-2

[17] Ning Li, Jianyong Tuo, Youqing Wang, Menghui Wang, Prediction of blood glucose concentration for type 1 diabetes based on echo state networks embedded with incremental learning, Neurocomputing, Volume 378, 2020, Pages 248-259, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2019.10.003

[18] W. Wang, M. Tong and M. Yu, "Blood Glucose Prediction With VMD and LSTM Optimized by Improved Particle Swarm Optimization," in IEEE Access, vol. 8, pp. 217908-217916, 2020, doi: 10.1109/ACCESS.2020.3041355.

[19] Zhu, T; Yao, X; Li, K; Herrero, P; Georgiou, P; (2020) Blood glucose prediction for type 1 diabetes using generative adversarial networks. In: Bach, K and Bunescu, R and Marling, C and Wiratunga, N, (eds.) Proceedings of the 5th International Workshop on Knowledge Discovery in Healthcare Data co-located with 24th European Conference on Artificial Intelligence (ECAI 2020). (pp. pp. 90-94). : Santiago de Compostela, Spain

[20] Khaoula Assadi, Takoua Hamdi, F. Fnaiech, J. M. Ginoux, and E. Moreau. 2017. Estimation of blood glucose levels techniques. 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C) (2017), 75–80.

[21] Enric Monte-Moreno. 2011. Non-invasive estimate of blood glucose and blood pressure from a photoplethysmograph by means of machine learning techniques.Artificial intelligence in medicine 53 (06 2011), 127–38. https://doi.org/10.1016/j.artmed.2011.05.001

[22] Takoua Hamdi, Jaouher Ben Ali, Véronique Di Costanzo, Farhat Fnaiech, Eric Moreau, and Jean-Marc Ginoux. 2018. Accurate prediction of continuous blood glucose based on support vector regression and differential evolution algorithm. Biocybernetics and Biomedical Engineering 38, 2 (2018), 362 – 372. https://doi.org/10.1016/j.bbe.2018.02.005

[23] Eleni Georga, Vasilios Protopappas, Diego Ardigò, Michela Marina, Ivana Zavaroni, Demosthenes Polyzos, and Dimitrios Fotiadis. 2012. Multivariate Prediction of Subcutaneous Glucose Concentration in Type 1 Diabetes Patients Based on Support Vector Regression. IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society 17 (09 2012). https://doi.org/10.1109/TITB.2012.2219876

[24] K. Plis, Razvan C. Bunescu, C. Marling, J. Shubrook, and F. Schwartz. 2014. A Machine Learning Approach to Predicting Blood Glucose Levels for Diabetes Management. In AAAI Workshop: Modern Artificial Intelligence for Health Analytics.

[25] R. Bunescu, N. Struble, C. Marling, J. Shubrook, and F. Schwartz. 2013. Blood Glucose Level Prediction Using Physiological Models and Support Vector Regression. In 2013 12th International Conference on Machine Learning and Applications, Vol. 1. 135–140. https://doi.org/10.1109/ICMLA.2013.30

[26] Natalia Mordvanyuk, F. Torrent-Fontbona, and B. López. 2017. Prediction of Glucose Level Conditions from Sequential Data. In CCIA. .

[27] Maged, Youssef & Atia, Ayman. (2022). The Prediction Of Blood Glucose Level By Using The ECG Sensor of Smartwatches. 406-411. 10.1109/MIUCC55081.2022.9781730.

[28] Kyriaki Saiti, Martin Macaš, Lenka Lhotská, Kateřina Štechová, Pavlína Pithová, Ensemble methods in combination with compartment models for blood glucose level prediction in type 1 diabetes mellitus,Computer Methods and Programs in Biomedicine, Volume 196, 2020, 105628, ISSN 0169-2607, https://doi.org/10.1016/j.cmpb.2020.105628

[29] Ma, Ning et al. “Online Blood Glucose Prediction Using Autoregressive Moving Average Model with Residual Compensation Network.” KDH@ECAI (2020)

[30] Xie, Jinyu & Wang, Qian. (2020). Benchmarking Machine Learning Algorithms on Blood Glucose Prediction for Type I Diabetes in Comparison With Classical Time-Series Models. IEEE Transactions on Biomedical Engineering. PP. 10.1109/TBME.2020.2975959.

[31] AIDA, http://www.2aida.org/

[32] James Moody. What does RMSE really mean?,

[33] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. 2012. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12 (01 2012).

[34] Keras webpage https://keras.io/guides/sequential_model/

[35] Wikipedia contributors. 2020. Hyperparameter (machine learning) — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Hyperparameter_(machine_learning)&oldid=984957886

[36] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[37] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

[38] Towards Data Science webpage https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms

[39] Scikit-learn webpage https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[40] Wikipedia webpage https://en.wikipedia.org/wiki/Multilayer_perceptron

[41] Machine learning mastery webpage https://machinelearningmastery.com/neural-networks-crash-course/

[42] Keras webpage https://keras.io/api/optimizers/adam/

[43] Keras webpage https://keras.io/api/layers/initializers/

[44] Keras webpage https://keras.io/api/layers/regularizers/

[45] Dr. Saed Sayad. Decision Tree - Regression, https://www.saedsayad.com/decision_tree_reg.htm

[46] scikit-learn developers. sklearn.tree.DecisionTreeRegressor, https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[47] scikit-learn developers. sklearn.tree.ExtraTreeRegressor, https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html

[48] Medium webpage https://medium.com/@supun.setunga/stacking-in-machine-learning-357db1cfc3a

[49] George Seif. 2018. https://towardsdatascience.com/three-reasons-that-you-should-not-use-deep-learning-15bec517b622

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 63–68
DOI: 10.15439/2022R14
ISSN 2300-5963 ACSIS, Vol. 33

Performance Evaluation Of Multi-Hop Relaying IoTs Networks Using Hop-By-Hop Cooperative Transmission Under Impact of Co-channel Interference

Ngo Hoang An
Industrial University of
Ho Chi Minh City
and Ho Chi Minh University
of Food Industry
Ho Chi Minh city, Vietnam
annh@hufi.edu.vn

Nguyen Ngoc Son
Faculty of Electronics Technology
Industrial University of
Ho Chi Minh City
Ho Chi Minh city, Vietnam
nguyenngocson@iuh.edu.vn

Nguyen Quang Sang
Science and Technology Application for
Sustainable Development Research Group
Ho ChiMinh City University of Transport
Ho Chi Minh city, Vietnam
sang.nguyen@ut.edu.vn

Doan Van Dong
Faculty of Electrical and
Electronic Engineering
Ho ChiMinh City
University of Transport
Ho Chi Minh city, Vietnam
dongdv@hcmutrans.edu.vn

Lam-Thanh Tu
Communication and Signal Processing
Research Group, Faculty of Electrical
and Electronics Engineering
Ton Duc Thang University
Ho Chi Minh city, Vietnam
tulamthanh@tdtu.edu.vn

Tran Trung Duy
Department of Electrical Engineering
Posts and Telecommunications Institute of
Technology
Ho Chi Minh city, Vietnam
trantrungduy@ptithcm.edu.vn

***Abstract*—The performance of the multi-hop relaying networks applied to Internet of Things (IoT) networks is investigated in the present work. To ameliorate the reliability of the whole network, for each hop, we employ cooperative communications subject to co-channel interference. In particular, several cooperative schemes are applied in the present work namely, incremental relaying, selection combining, and maximal ratio combining. In this context, we derive the exact closed-form expressions of end-to-end (e2e) outage probability (OP) of the whole systems over Rayleigh fading distribution. Numerical results based on the Monte-Carlo method are then employed to not only confirm the accuracy of the proposed mathematical framework but also to illustrate the advantages of the considered schemes compared with the conventional multi-hop relaying without using cooperative communications.**

***Index Terms*—Cooperative communication, multi-hop relaying, co-channel interference, outage probability.**

I. Introduction

Cooperative communication and relaying techniques [1]-[5] are widely used in self-organized networks (such as IoTs networks) to enhance performance. So far, the cooperative relaying has been applied into various developed wireless networks such as secure communication at physical layer [6], wirelessly EH (energy harvesting) [7], NOMA (Non-Orthogonal Multiple Access) [8], IRS (Intelligent Reflecting Surface) [9], underlay spectrum sharing cognitive radio [10], etc. Different with the conventional relaying technique, a destination node using cooperative communication can exploit the direct link for obtaining higher diversity order [11]. In addition, the destination can use MRC combiner [11] or SC combiner [12] to decode the received data. However, disadvantage of cooperative communication is that two orthogonal time slots are used for each data transmission. In [11], [13], the authors proposed an incremental cooperation (IC) approach, where the relay node was only used when the direct transmission was not successful. As a result, IC enhances spectrum usage efficiency, as compared with conventional cooperative communication [11], [13].

For far source-destination distance, multi-hop relaying (MHR) [14]-[18] that uses multiple intermediate relays is often employed. However, performance of the conventional MHR methods in [14]-[18] is severely degraded over multi-path fading and co-channel interference (CCI) channels. In [19]-[21], the authors evaluated and optimized performance of various MHR schemes operating on the CCI environments. In [22]-[28], the authors proposed diversity-aided MHR models to further enhance performance for the MHR networks. Particularly, the authors of [22]-[25] measured e2e outage probability (OP) of cooperative multi-hop relaying schemes, in which the source and relay nodes cooperate together to exploit the spatial diversity. However, it is too difficult to deploy these schemes into the IoTs networks because of a requirement of high synchronization and high storage capacity at the relays. In [26]-[28], the authors introduced path-selection methods to obtain higher diversity order for multi-path MHR networks. In particular, the optimal path in [26]-[28] is the one which provides the highest e2e channel capacity. In addition, it is difficult to implement these path-selection approaches in the IoTs networks because they require perfect channel-coefficient estimation on all the paths.

Different with [19]-[28], this paper considers the MHR networks using hop-by-hop cooperative communication. References [29]-[30] are the most relevant to this paper. Indeed, reference [29] used the cooperative transmission to obtain better e2e OP performance for cognitive MHR networks with hardware imperfection. Different with [29], this paper does not consider the cognitive networks, but investigates impact of CCI on the e2e OP. Published work [30] also applied hop-by-hop cooperative communication for the MRH networks using Fountain codes. Next, we will summarize the main motivation and main contribution of this paper as follows:

- Different with [29]-[30], we consider various cooperative transmission methods: i) in the first one, named MHR-SC, the receiver at each hop uses SC to decode the received data; ii) in the second one, named MHR-MRC, MRC is employed by the receiver at each hop; in the third one, named MHR-IC, the IC technique is applied at each hop.

- For MHR-IC, the receiver at each hop does not use SC and MRC. We also propose a simple time allocation strategy for the hop-by-hop cooperative transmission to improve the OP performance for MHR-IC.

- We provide exact formulas of the e2e OP of MHR-SC, MHR-MRC and MHR-IC over CCI and Rayleigh fading channels, and use computer simulations to verify these derived formulas.

- The obtained results show that the considered schemes obtain much better OP performance than the conventional MHR one (named MHR-WoCC).

In the following, we will present the proposed scheme model in Section II, performance evaluation in Section III, the computer simulations and the theoretical results in Section IV, and conclusion in Section V.

II. System Model

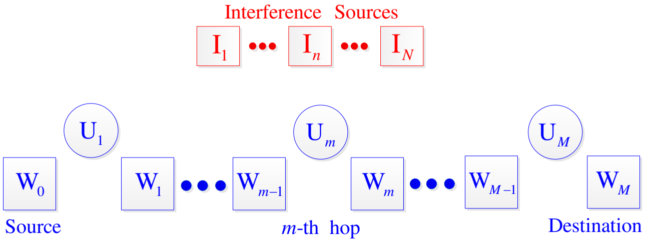


Fig. 1. The proposed MHR IoTs scenarios.

Fig. 1 demonstrates operation principle of MHR-SC, MHR-MRC and MHR-IC, where the source (W_0) transmits its data to the destination (W_M) with assistance of $M-1$ decode-and-forward (DF) relays denoted as W_1, W_2, …, W_{M-1}, where $M \geq 1$. The $\mathrm{W}_0 \to \mathrm{W}_M$ route is previously established by the network layer [31]-[32], and is used to send the source data to W_M. Considering the $m-$th hop $(m=1,2,...,M)$; W_{m-1} communicates with W_m via assistance of a coopertive node (U_m). Note that U_m is one of the nodes which are in radio range of W_{m-1} and W_m. In addition, considering ultra-dense IoTs networks [33]-[34], and hence there always exists at least one cooperative node at each hop. Assume that N CCI sources (named I_n $(n=1,2,...,N)$) cause co-channel interference on the receivers U_m and W_m.

Assume that all the IoTs nodes have single antenna; the $\mathrm{W}_0 \to \mathrm{W}_M$ communication has to use M time slots. If the e2e delay is 1 time unit, then time allocated for each time slot is $\tau = 1/M$.

Considering the $m-$th hop in MHR-WoCC; W_{m-1} transmits the data to W_m, without using the help of U_m.

For MHR-SC and MHR-MRC, the transmisson at the $m-$th hop is split into two equal phases with equal duration of $\tau/2$. In particular, in the first one, W_{m-1} sends the data to U_m and W_m. Then, U_m attempts to decode the received data, and it will forward the data to W_m at the second one if the decoding status is successful. If W_m can receive the data from both W_{m-1} and U_m, this node in MHR-SC (MHR-MRC) will use SC (MRC) for the decoding operation.

In MHR-IC, duration of the first and second sub-time slots is allocated by $\alpha\tau$ and $(1-\alpha)\tau$, respectively, where α $(0<\alpha \leq 1)$ is a pre-designed value. Particularly, in the first sub-time slot, W_{m-1} sends the data to U_m and W_m. At the end of the first sub-time slot, if the decoding status at W_m is successful, it will feedback a ACK message to inform, and then will transmit the data to W_{m+1} $(m+1 \leq M)$. If W_m fails to decode the data, it will send back a NACK message to request the help from U_m. In this case, U_m will retransmit the data to W_m at the second sub-time slot if the decoding status of U_m is correct. As mentioned above, W_m in MHR-IC will not use any the combining technique. Moreover, we can observe that if $\alpha = 1$, MHR-IC becomes MHR-WoCC.

Let $g_{\mathrm{T,R}}$ denote channel gain of the $\mathrm{T} \to \mathrm{R}$ Rayleigh fading channel, where T is a transmitter $\left(\mathrm{T} \in \{\mathrm{W}_{m-1}, \mathrm{I}_n, \mathrm{U}_m\}\right)$ and R is a receiver $\left(\mathrm{R} \in \{\mathrm{W}_m, \mathrm{U}_m\}\right)$; cumulative distribution function (CDF) and probability density function (PDF) of $g_{\mathrm{T,R}}$ are written, respectively as

$$F_{g_{\mathrm{T,R}}}(x) = 1-\exp\left(-\lambda_{\mathrm{T,R}}x\right), f_{g_{\mathrm{T,R}}}(x) = \lambda_{\mathrm{T,R}}\exp\left(-\lambda_{\mathrm{T,R}}x\right), \quad (1)$$

where $\lambda_{\mathrm{T,R}} = \left(d_{\mathrm{T,R}}\right)^{\beta}$, with $d_{\mathrm{T,R}}$ is distance between T and R, and β is path-loss exponential [11].

Under impact of CCI, the channel capacity between W_{m-1} and W_m in MHR-WoCC can be formulated as

$$C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-WoCC}} = \frac{1}{M}\log_2\left(1+\frac{Pg_{\mathrm{W}_{m-1},\mathrm{W}_m}}{\sum_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right). \quad (2)$$

In (2), we assume that transmit power of the W_{m-1} and U_m transmitters is *P*, and that of the CCI sources is *Q*, for all *m* and *n*. In addition, it is also assumed that variance of Gaussian noises at all the receivers (R) is σ_0^2.

Considering MHR-SC and MHR-MRC; the channel capacity of the $\mathrm{W}_{m-1} \to \mathrm{W}_m$ and $\mathrm{W}_{m-1} \to \mathrm{U}_m$ links can be expressed, respectively as

$$C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\mathrm{Z}} = \frac{1}{2M}\log_2\left(1+\frac{Pg_{\mathrm{W}_{m-1},\mathrm{W}_m}}{\sum_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right), \quad (3)$$

$$C^{\mathrm{Z}}_{\mathrm{W}_{m-1},\mathrm{U}_m}=\frac{1}{2M}\log_2\left(1+\frac{Pg_{\mathrm{W}_{m-1},\mathrm{U}_m}}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{U}_m}+\sigma_0^2}\right),\quad (4)$$

where $Z\in\{\text{MHR-SC},\text{MHR-MRC}\}$.

When W_m in MHR-SC (MHR-MRC) uses SC (MRC) to decode the received data, the channel capacity obtained at W_m can be formulated, respectively as

$$C^{\text{MHR-SC}}_{\mathrm{W}_m,\mathrm{SC}}=\frac{1}{2M}\log_2\left(1+\frac{P\max\left(g_{\mathrm{W}_{m-1},\mathrm{W}_m},g_{\mathrm{U}_m,\mathrm{W}_m},\right)}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right),\quad (5)$$

$$C^{\text{MHR-MRC}}_{\mathrm{W}_m,\mathrm{MRC}}=\frac{1}{2M}\log_2\left(1+\frac{P\left(g_{\mathrm{W}_{m-1},\mathrm{W}_m}+g_{\mathrm{U}_m,\mathrm{W}_m}\right)}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right).\quad (6)$$

We note that the channel gains $g_{\mathrm{I}_n,\mathrm{W}_m}$ are the same in the first and second sub-time slots. As a result, as performing SC or MRC, W_m does not need to estimate $g_{\mathrm{I}_n,\mathrm{W}_m}$.

In MHR-IC, the channel capacity of the $\mathrm{W}_{m-1}\rightarrow\mathrm{W}_m$, $\mathrm{W}_{m-1}\rightarrow\mathrm{U}_m$ and $\mathrm{U}_m\rightarrow\mathrm{W}_m$ links is given, respectively as

$$C^{\text{MHR-IC}}_{\mathrm{W}_{m-1},\mathrm{W}_m}=\frac{\alpha}{M}\log_2\left(1+\frac{Pg_{\mathrm{W}_{m-1},\mathrm{W}_m}}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right),\quad (7)$$

$$C^{\text{MHR-IC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}=\frac{\alpha}{M}\log_2\left(1+\frac{Pg_{\mathrm{W}_{m-1},\mathrm{U}_m}}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{U}_m}+\sigma_0^2}\right),\quad (8)$$

$$C^{\text{MHR-IC}}_{\mathrm{U}_m,\mathrm{W}_m}=\frac{1-\alpha}{M}\log_2\left(1+\frac{Pg_{\mathrm{U}_m,\mathrm{W}_m}}{\sum\limits_{n=1}^{N}Qg_{\mathrm{I}_n,\mathrm{W}_m}+\sigma_0^2}\right).\quad (9)$$

III. Performance Analysis

This section evaluates the e2e OP of the Z scheme, where $Z\in\{\text{MHR-WoCC},\text{MHR-SC},\text{MHR-MRC},\text{MHR-IC}\}$. At first, we assume that the T → R link is outage if channel capacity $C^{Z}_{\mathrm{T,R}}$ is below a pre-determined threshold (C_{th}). Otherwise, (i.e., $C^{Z}_{\mathrm{T,R}}\geq C_{\mathrm{th}}$), assume that the receiver R can correctly decode the data received from the transmitter T.

Hence, we can formulate the e2e OP of the Z scheme as

$$\mathrm{OP}^{\mathrm{Z}}_{\mathrm{e2e}}=1-\prod_{m=1}^{M}\left(1-\mathrm{OP}^{\mathrm{Z}}_{m}\right),\quad (10)$$

where $\mathrm{OP}^{\mathrm{Z}}_{m}$ is OP at the $m-$th hop of the Z scheme.

A. The MHR-WoCC Scheme

From (2), $\mathrm{OP}^{\text{MHR-WoCC}}_{m}$ can be expressed as

$$\begin{aligned}\mathrm{OP}^{\text{MHR-WoCC}}_{m}&=\Pr\left(C^{\text{MHR-WoCC}}_{\mathrm{W}_{m-1},\mathrm{W}_m}<C_{\mathrm{th}}\right)\\&=\int_0^{+\infty}\!\!...\int_0^{+\infty}F_{g_{\mathrm{W}_{m-1},\mathrm{W}_m}}\left(\theta_1\sum_{n=1}^{N}x_n+\chi_1\right)\quad (11)\\&\times f_{g_{\mathrm{I}_1,\mathrm{W}_m}}\left(x_1\right)...f_{g_{\mathrm{I}_N,\mathrm{W}_m}}\left(x_N\right)dx_1...dx_N,\end{aligned}$$

where $\rho_1=2^{MC_{\mathrm{th}}}-1,\theta_1=\dfrac{Q\rho_1}{P},\chi_1=\dfrac{\sigma_0^2\rho_1}{P}$.

For ease of presentation and analysis, we can assume that the random variables $g_{\mathrm{I}_n,\mathrm{U}_m}$ $\left(g_{\mathrm{I}_n,\mathrm{W}_m}\right)$ are independent and identical, i.e., $\lambda_{\mathrm{I}_n,\mathrm{U}_m}=\lambda_{\mathrm{I},\mathrm{U}_m}$ and $\lambda_{\mathrm{I}_n,\mathrm{W}_m}=\lambda_{\mathrm{I},\mathrm{W}_m}$, $\forall m,n.$

Now, substituting CDF of $g_{\mathrm{W}_{m-1},\mathrm{W}_m}$ and PDF of $g_{\mathrm{I}_n,\mathrm{W}_m}$ into (11), after some careful manipulation, we obtain

$$\begin{aligned}&\mathrm{OP}^{\text{MHR-WoCC}}_{m}=\\&1-\left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m}+\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_1}\right)^{N}\exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_1\right).\end{aligned}\quad (12)$$

B. The MHR-SC Scheme

In this scheme, $\mathrm{OP}^{\text{MHR-SC}}_{m}$ can be formulated as

$$\begin{aligned}\mathrm{OP}^{\text{MHR-SC}}_{m}&=\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}<C_{\mathrm{th}}\right)\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{W}_m}<C_{\mathrm{th}}\right)\\&+\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}\geq C_{\mathrm{th}}\right)\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_m,\mathrm{SC}}<C_{\mathrm{th}}\right).\end{aligned}\quad (13)$$

Using (3)-(4), similar to (11)-(12), we can compute $\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{W}_m}<C_{\mathrm{th}}\right)$ and $\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}<C_{\mathrm{th}}\right)$, respectively as

$$\begin{aligned}&\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{W}_m}<C_{\mathrm{th}}\right)=\\&1-\left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m}+\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_2}\right)^{N}\exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_2\right),\end{aligned}\quad (14)$$

$$\begin{aligned}&\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}<C_{\mathrm{th}}\right)=\\&1-\left(\frac{\lambda_{\mathrm{I},\mathrm{U}_m}}{\lambda_{\mathrm{I},\mathrm{U}_m}+\lambda_{\mathrm{W}_{m-1},\mathrm{U}_m}\theta_2}\right)^{N}\exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{U}_m}\chi_2\right).\end{aligned}\quad (15)$$

where $\rho_2=2^{2MC_{\mathrm{th}}}-1,\theta_2=\dfrac{Q\rho_2}{P},\chi_2=\dfrac{\sigma_0^2\rho_2}{P}$. In addition, we note that $\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}\geq C_{\mathrm{th}}\right)=1-\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_{m-1},\mathrm{U}_m}<C_{\mathrm{th}}\right)$.

For $\Pr\left(C^{\text{MHR-SC}}_{\mathrm{W}_m,\mathrm{SC}}<C_{\mathrm{th}}\right)$, using (5), we can write

$$\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-SC}} < C_{\text{th}}\right) = \int_0^{+\infty} \ldots \int_0^{+\infty} F_{T_{\max}}\left(\theta_2 \sum_{n=1}^{N} x_n + \chi_2\right) \times f_{g_{\mathrm{I}_1,\mathrm{W}_m}}\left(x_1\right) \ldots f_{g_{\mathrm{I}_N,\mathrm{W}_m}}\left(x_N\right) dx_1 \ldots dx_N, \quad (16)$$

where $T_{\max} = \max\left(g_{\mathrm{U}_m,\mathrm{W}_m}, g_{\mathrm{W}_{m-1},\mathrm{W}_m}\right)$ whose CDF is

$$\begin{aligned} F_{T_{\max}}(x) &= F_{g_{\mathrm{W}_{m-1},\mathrm{W}_m}}(x) F_{g_{\mathrm{U}_m,\mathrm{W}_m}}(x) \\ &= 1 - \exp\left(-\lambda_{\mathrm{U}_m,\mathrm{W}_m} x\right) - \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m} x\right) \\ &\quad + \exp\left(-\left(\lambda_{\mathrm{U}_m,\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\right) x\right). \end{aligned} \quad (17)$$

Combining (1), (16) and (17), and after calculating the integrals, we obtain

$$\begin{aligned} &\Pr\left(C_{\mathrm{W}_m,\mathrm{SC}}^{\text{MHR-SC}} < C_{\text{th}}\right) = \\ &1 - \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{U}_m,\mathrm{W}_m}\theta_2}\right)^N \exp\left(-\lambda_{\mathrm{U}_m,\mathrm{W}_m}\chi_2\right) \\ &- \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_2}\right)^N \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_2\right) \\ &+ \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \left(\lambda_{\mathrm{U}_m,\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\right)\theta_2}\right)^N \\ &\times \exp\left(-\left(\lambda_{\mathrm{U}_m,\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\right)\chi_2\right). \end{aligned} \quad (18)$$

C. The MHR-MRC Scheme

Similar to MHR-SC, $\mathrm{OP}_m^{\text{MHR-MRC}}$ can be expressed as

$$\begin{aligned} &\mathrm{OP}_m^{\text{MHR-MRC}} = \\ &\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-MRC}} < C_{\text{th}}\right) \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-MRC}} < C_{\text{th}}\right) \\ &+ \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-MRC}} \geq C_{\text{th}}\right) \Pr\left(C_{\mathrm{W}_m,\mathrm{MRC}}^{\text{MHR-MRC}} < C_{\text{th}}\right). \end{aligned} \quad (19)$$

We note that $\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-MRC}} < C_{\text{th}}\right) = \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-SC}} < C_{\text{th}}\right)$ and $\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-MRC}} < C_{\text{th}}\right) = \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-SC}} < C_{\text{th}}\right)$. For $\Pr\left(C_{\mathrm{W}_m,\mathrm{MRC}}^{\text{MHR-MRC}} < C_{\text{th}}\right)$, we have

$$\Pr\left(C_{\mathrm{W}_m,\mathrm{MRC}}^{\text{MHR-MRC}} < C_{\text{th}}\right) = \int_0^{+\infty} \ldots \int_0^{+\infty} F_{T_{\text{sum}}}\left(\theta_2 \sum_{n=1}^{N} x_n + \chi_2\right) \times f_{g_{\mathrm{I}_1,\mathrm{W}_m}}\left(x_1\right) \ldots f_{g_{\mathrm{I}_N,\mathrm{W}_m}}\left(x_N\right) dx_1 \ldots dx_N, \quad (20)$$

where $T_{\text{sum}} = g_{\mathrm{W}_{m-1},\mathrm{W}_m} + g_{\mathrm{U}_m,\mathrm{W}_m}$. To find CDF of T_{sum}, we can use moment generating function (MGF) method proposed in [35].

Moreover, we note that the cooperative relay U_m is selected so that $d_{\mathrm{W}_{m-1},\mathrm{U}_m}$ and $d_{\mathrm{U}_m,\mathrm{W}_m}$ should be shorter than $d_{\mathrm{W}_{m-1},\mathrm{W}_m}$, or $\lambda_{\mathrm{U}_m,\mathrm{W}_m} < \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}$. Hence, we can obtain CDF of T_{sum} as

$$\begin{aligned} F_{T_{\text{sum}}}(x) &= 1 - \frac{\lambda_{\mathrm{U}_m,\mathrm{W}_m}}{\lambda_{\mathrm{U}_m,\mathrm{W}_m} - \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}} \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m} x\right) \\ &\quad - \frac{\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}}{\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m} - \lambda_{\mathrm{U}_m,\mathrm{W}_m}} \exp\left(-\lambda_{\mathrm{U}_m,\mathrm{W}_m} x\right). \end{aligned} \quad (21)$$

Plugging (1), (20) and (21) together, we have

$$\begin{aligned} &\Pr\left(C_{\mathrm{W}_m,\mathrm{MRC}}^{\text{MHR-MRC}} < C_{\text{th}}\right) = \\ &1 - \frac{\lambda_{\mathrm{U}_m,\mathrm{W}_m}}{\lambda_{\mathrm{U}_m,\mathrm{W}_m} - \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}} \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_2}\right)^N \\ &\times \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_2\right) \\ &- \frac{\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}}{\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m} - \lambda_{\mathrm{U}_m,\mathrm{W}_m}} \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{U}_m,\mathrm{W}_m}\theta_2}\right)^N \\ &\times \exp\left(-\lambda_{\mathrm{U}_m,\mathrm{W}_m}\chi_2\right). \end{aligned} \quad (22)$$

D. The MHR-IC Scheme

In this scheme, $\mathrm{OP}_m^{\text{MHR-IC}}$ can be formulated as

$$\begin{aligned} \mathrm{OP}_m^{\text{MHR-IC}} &= \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-IC}} < C_{\text{th}}\right) \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-IC}} < C_{\text{th}}\right) \\ &\quad + \Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-IC}} \geq C_{\text{th}}\right) \\ &\quad \times \underbrace{\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-IC}} < C_{\text{th}}, C_{\mathrm{U}_m,\mathrm{W}_m}^{\text{MHR-IC}} < C_{\text{th}}\right)}_{J}. \end{aligned} \quad (23)$$

Similarly, we can obtain the following results:

$$\begin{aligned} &\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{W}_m}^{\text{MHR-IC}} < C_{\text{th}}\right) = \\ &1 - \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_3}\right)^N \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_3\right), \end{aligned} \quad (24)$$

$$\begin{aligned} &\Pr\left(C_{\mathrm{W}_{m-1},\mathrm{U}_m}^{\text{MHR-IC}} < C_{\text{th}}\right) = \\ &1 - \left(\frac{\lambda_{\mathrm{I},\mathrm{U}_m}}{\lambda_{\mathrm{I},\mathrm{U}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{U}_m}\theta_3}\right)^N \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{U}_m}\chi_3\right). \end{aligned} \quad (25)$$

where $\rho_3 = 2^{\frac{MC_{\text{th}}}{\alpha}} - 1, \theta_3 = \frac{Q\rho_3}{P}, \chi_3 = \frac{\sigma_0^2\rho_3}{P}$.

Considering the probability J marked in (23); using (7) and (9), which yields

$$\begin{aligned} J = \int_0^{+\infty} \ldots \int_0^{+\infty} & F_{g_{\mathrm{W}_{m-1},\mathrm{W}_m}}\left(\theta_3 \sum_{n=1}^{N} x_n + \chi_3\right) \\ & \times F_{g_{\mathrm{U}_m,\mathrm{W}_m}}\left(\theta_4 \sum_{n=1}^{N} x_n + \chi_4\right) \\ & \times f_{g_{\mathrm{I}_1,\mathrm{W}_m}}\left(x_1\right) \ldots f_{g_{\mathrm{I}_N,\mathrm{W}_m}}\left(x_N\right) dx_1 \ldots dx_N, \end{aligned} \quad (26)$$

where $\rho_4 = 2^{\frac{MC_{\text{th}}}{1-\alpha}} - 1, \theta_4 = \frac{Q\rho_4}{P}, \chi_4 = \frac{\sigma_0^2\rho_4}{P}$.

Substituting CDFs and PDFs given in (1) into (26), and after some careful calculation, we finally obtain

$$\begin{aligned} J = 1 &- \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_3} \right)^N \exp\left(-\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_3\right) \\ &- \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \lambda_{\mathrm{U}_m,\mathrm{W}_m}\theta_4} \right)^N \exp\left(-\lambda_{\mathrm{U}_m,\mathrm{W}_m}\chi_4\right) \\ &+ \left(\frac{\lambda_{\mathrm{I},\mathrm{W}_m}}{\lambda_{\mathrm{I},\mathrm{W}_m} + \left(\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\theta_3 + \lambda_{\mathrm{U}_m,\mathrm{W}_m}\theta_4\right)} \right)^N \\ &\times \exp\left(-\left(\lambda_{\mathrm{W}_{m-1},\mathrm{W}_m}\chi_3 + \lambda_{\mathrm{U}_m,\mathrm{W}_m}\chi_4\right)\right). \end{aligned} \quad (27)$$

IV. SIMULATION RESULTS

Section IV presents Monte-Carlo simulations to check exactness of the formulas derived in Section III, and to compare the e2e OP performance of the considered schemes. In this section, coordinate of the nodes is fixed as follows : $\mathrm{W}_m\left(\dfrac{m}{M},0\right)$, $\mathrm{U}_m\left(\dfrac{2u-1}{2M},0\right)$, $\mathrm{I}_n(0.5,1)$, where $m = 0,1...,M$, $u = 1,2,...,M$, $n = 1,2,...,N$. In addition, we fix the path-loss exponential by $\beta = 3$, the outage threshold by $C_{\mathrm{th}} = 0.25$, the number of CCI sources by $N = 2$. We also assume that transmit power of all the transmitters is the same, i.e., $P = Q$, and we denote $\Psi = P/\sigma_0^2$ as the transmit SNR.

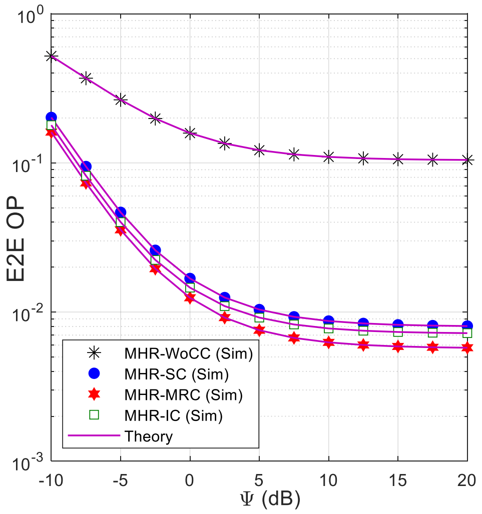


Fig. 2. E2e OP as a function of Ψ (dB) when $M = 4$ and $\alpha = 0.6$.

Fig. 2 presents the e2e OP as a function of Ψ in dB with $M = 4$ and $\alpha = 0.6$. As we can see, the OP performance of all the schemes is better as Ψ increases. However, at high Ψ regimes, all the OP values converge to saturation values due to impact of CCI. Fig. 2 also shows that MHR-SC, MHR-MRC and MHR-IC obtain much better performance than MHR-WoCC. It is due to the fact that MHR-SC, MHR-MRC and MHR-IC use the hop-by-hop cooperative transmission, which significantly enhances reliability of the data transmission. Next, it is seen that MHR-MRC obtains the best performance due to the optimal combiner used, while OP of MHR-IC is between those of MHR-MRC and MHR-SC.

Fig. 3 shows the e2e OP as a function of M with $\Psi = 10$ (dB) and $\alpha = 0.45$. Similar to Fig. 2, OP of MHR-SC, MHR-MRC and MHR-IC is much lower than that of MHR-WoCC, and MHR-MRC obtains the best performance. However, in Fig. 3, MHR-SC outperforms MHR-IC, and this means that the value of α significantly impacts on the OP performance of MHR-IC. Fig. 3 also shows that there exists the optimal values of the number of hops at which the e2e OP of the considered schemes is lowest.

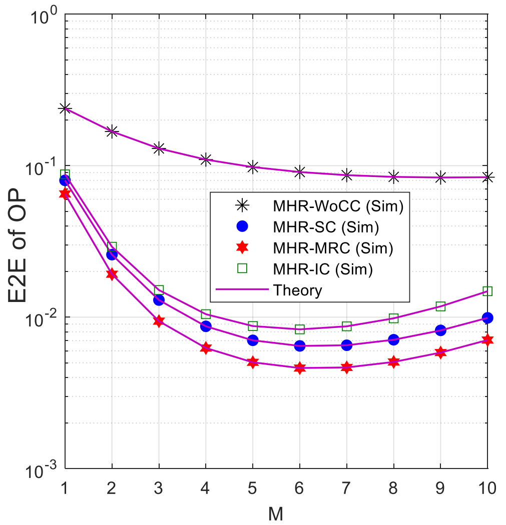


Fig. 3. E2e OP as a function of M when $\Psi = 10$ (dB) and $\alpha = 0.45$.

Fig. 4 investigates impact of α on the e2e OP of MHR-IC when $\Psi = 10$ (dB) and $M = 6$. As observed, there exists an optimal value of α at which the OP performance of MHR-IC is best. In particularthe optimal value of α in this figure is 0.55. It is also seen from Fig. 4 that MHR-MRC still provides the best OP performance, and MHR-IC outperforms MHR-SC when $0.5 \le \alpha < 0.65$.

It is worth noting from Figs. 2-4 that the Monte-Carlo based simulation results match well with the analytical ones, which validate our derived expressions.

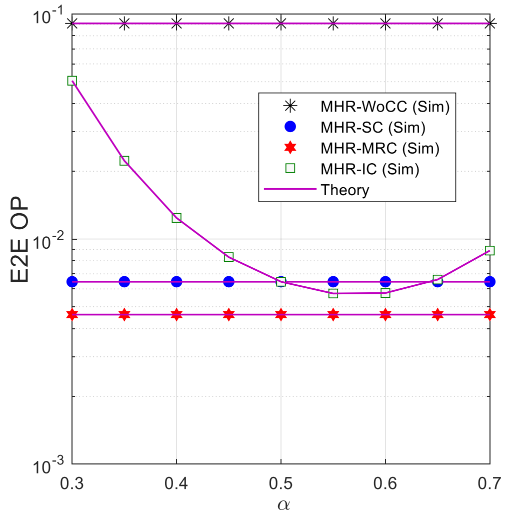


Fig. 4. E2e OP as a function of α when $\Psi = 10$ (dB) and $M = 6$.

V. Conclusion

This paper evaluated the e2e OP of the cooperative transmission aided MHR IoTs networks under impact of CCI via both simulation and analysis. The results presented that using hop-by-hop cooperative communication significantly enhanced the OP performance for the MHR IoTs networks, as compared with the conventional MHR scheme (MHR-WoCC). Moreover, to further enhance the OP performance for the considered schemes, the number of hops of the source-destination route should be designed optimally.

References

[1] P. N. Son and T. T. Duy, "Performance Analysis of Underlay Cooperative Cognitive Full-duplex Networks with Energy-Harvesting Relay," Computer Communications, vol. 122, pp. 9-19, Jun. 2018.

[2] H. D. Binh and S. Q. Nguyen. "Outage Performance of Energy Harvesting DF Relaying NOMA Networks," Mobile Networks and Applications, vol. 23, no. 6, pp. 1572-1585, Dec. 2018.

[3] S. Q. Nguyen, H. T. Nguyen, D. V. Dong and W.-J. Hwang, "Exact Outage Analysis of Cognitive Energy Harvesting Relaying Networks Under Physical Layer Security," EAI Endorsed Transactions on INIS, vol. 6, no. 18, pp. 1-15, March 2019.

[4] T. H. Nguyen, et al, "Performance Analysis and Optimization of the Coverage Probability in Dual Hop LoRa Networks With Different Fading Channels," IEEE Access, vol. 8, pp. 107087-107102, Jun. 2020.

[5] L. T. Tu, et al, "Performance Evaluation of Incremental Relaying in Underlay Cognitive Radio Networks with Imperfect CSI," in Proc. of 2020 IEEE ICCE, 2021, pp. 472-477.

[6] L. -T. Tu and A. Bradai, "On the Performance of Physical Layer Security of RIS-aided Communications," in Proc. of 2021 IEEE CAMA, 2021, pp. 570-574.

[7] P. X. Nguyen, et al., "Backscatter-Assisted Data Offloading in OFDMA-Based Wireless-Powered Mobile Edge Computing for IoT Networks," IEEE Internet of Things Journal, vol. 8, no. 11, pp. 9233-9243, Jun. 2021.

[8] H. D. Hung, et al, "Security-Reliability Trade-off Analysis for Rateless Codes-Based Relaying Protocols Using NOMA, Cooperative Jamming and Partial Relay Selection," IEEE Access, vol. 9, pp. 131087-131108 , Sept. 2021.

[9] T. Van Chien, et al, "Outage Probability Analysis of IRS-Assisted Systems Under Spatially Correlated Channels," IEEE Wireless Communications Letters, vol. 10, no. 8, pp. 1815-1819, Aug. 2021.

[10] Pham Ngoc Son, "Joint impacts of Hardware impairments, Imperfect CSIs, and Interference constraints on Underlay Cooperative Cognitive Networks with reactive relay selection," Telecommunication Systems, vol. 71, no. 1, pp. 65-76, May 2019.

[11] J. N. Laneman, D. N. C. Tse and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," IEEE Transactions on Information Theory, vol. 50, no. 12, pp. 3062-3080, Dec. 2004.

[12] P. M. Nam, H. D. Hung, T. T. Duy and L. T. Thuong, "Security-Reliability Tradeoff of MIMO TAS/SC Networks using Harvest-to-Jam Cooperative Jamming Methods With Random Jammer Location," ICT Express, 2022. Doi: 10.1016/j.icte.2021.11.003

[13] T. T. Duy and H.Y. Kong, "Performance Analysis of Incremental Amplify-and-Forward Relaying Protocols with Nth Best Partial Relay Selection under Interference Constraint," Wireless Personal Communications, vol. 71, no. 4, pp. 2741-2757, Aug. 2013.

[14] G. Amarasuriya, C. Tellambura and M. Ardakani, "Asymptotically-Exact Performance Bounds of AF Multi-Hop Relaying over Nakagami Fading," IEEE Transactions on Communications, vol. 59, no. 4, pp. 962-967, Apr. 2011.

[15] M. Mao, N. Cao, Y. Chen and Y. Zhou, "Multi-Hop Relaying Using Energy Harvesting," IEEE Wireless Communications Letters, vol. 4, no. 5, pp. 565-568, Oct. 2015.

[16] T. T. Duy and H.Y. Kong, "Secrecy Performance Analysis of Multihop Transmission Protocols in Cluster Networks," Wireless Personal Communications (WPC), vol. 82, no. 4, pp. 2505-2518, Jun. 2015.

[17] A. Chaaban and A. Sezgin, "Multi-Hop Relaying: An End-to-End Delay Analysis," IEEE Transactions on Wireless Communications, vol. 15, no. 4, pp. 2552-2561, Apr. 2016.

[18] N. Q. Sang and H. Y. Kong. "Exact Outage Analysis Of The Effect Of Co-Channel Interference on Secured Multi-hop Relaying Networks," International Journal of Electronics, vol. 103, no. 11, pp. 1822-1838, Aug. 2016.

[19] H. Yu and G. L. Stuber, "General Decode-and-Forward Cooperative Relaying with Co-Channel Interference in Shadowed Nakagami Fading Channels," IEEE Transactions on Wireless Communications, vol. 11, no. 12, pp. 4318-4327, Dec. 2012.

[20] J. A. Hussein, et al, "Performance Analysis of a Multi-Hop UCRN With Co-Channel Interference," IEEE Transactions on Communications, vol. 64, no. 10, pp. 4346-4364, Oct. 2016.

[21] E. Chen, M. Xia, D. B. da Costa and S. Aïssa, "Multi-Hop Cooperative Relaying With Energy Harvesting From Cochannel Interferences," IEEE Communications Letters, vol. 21, no. 5, pp. 1199-1202, May 2017.

[22] J. Zhang and T. M. Lok, "Performance Comparison of Conventional and Cooperative Multihop Transmission," in Proc. of IEEE WCMC, 2006, pp. 897-901.

[23] G. Farhadi and N. C. Beaulieu, "Fixed Relaying Versus Selective Relaying in Multi-hop Diversity Transmission Systems," IEEE Transactions on Communications, vol. 58, no. 3, pp. 956-965, March 2010.

[24] P. T. Tin, et al, "Secrecy Performance Enhancement for Underlay Cognitive Radio Networks Employing Cooperative Multi-hop Transmission With and Without Presence of Hardware Impairments," Entropy, vol. 21, no. 2, Feb. 2019.

[25] P. M. Nam, et al, "Outage Performance of Power Beacon-Aided Multi-Hop Cooperative Cognitive Radio Protocol Under Constraint of Interference and Hardware Noises," Electronics, vol. 9, no. 6, pp. 1-19, Jun. 2020.

[26] M. R. Bhatnagar, "Performance Analysis of a Path Selection Scheme in Multi-Hop Decode-and-Forward Protocol," IEEE Communications Letters, vol.16, no. 12, pp. 1980-1983, Dec. 2012.

[27] M. R. Bhatnagar, R. K. Mallik and O. Tirkkonen, "Performance Evaluation of Best-Path Selection in a Multihop Decode-and-Forward Cooperative System," IEEE Transactions on Vehicular Technology, vol. 65, no. 4, pp. 2722-2728, Apr. 2016.

[28] T. D. Hieu, T. T. Duy and B.-S. Kim, "Performance Enhancement for Multi-hop Harvest-to-Transmit WSNs With Path-Selection Methods in Presence of Eavesdroppers and Hardware Noises," IEEE Sensors Journal, vol. 18, no. 12, pp. 5173 - 5186, Jun. 2018.

[29] T. T. Duy and V.N.Q. Bao, "Performance Analysis of Cooperative-based Multi-hop Transmission Protocols in Underlay Cognitive Radio with Hardware Impairment," VNU Journal of Computer Science and Communication Engineering, vol. 31, no. 2, pp. 15-28, 2015.

[30] N. T. Huan, et al, "Incremental Cooperation Based Multi-hop Relaying Scheme With Fountain Codes, Wirelessly Energy Harvesting and Partial Relay Selection," in Proc. of ATC, Oct. 2022, pp. 1-6.

[31] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," in Proc. of WMCSA'99, Feb. 1999, pp.90-100.

[32] B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, vol. 353, pp.153-181, 1996.

[33] L. Yu, J. Wu and P. Fan, "Energy Efficient Designs of Ultra-Dense IoT Networks With Nonideal Optical Front-Hauls," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 7934-7945, Oct. 2019.

[34] S. K. Sharma and X. Wang, "Toward Massive Machine Type Communications in Ultra-Dense Cellular IoT Networks: Current Issues and Machine Learning-Assisted Solutions," IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 426-471, Firstquarter 2020.

[35] A Papoulis. *Probability, Random Variables, and Stochastic Processes*. 4th Ed., 2022.

DOI: 10.15439/2022R10
ISSN 2300-5963 ACSIS, Vol. 33

Dealing with Imbalanced Data for GPS Trajectory Outlier Detection

Nguyen Van Chien
Faculty of Information Technology
Ho Chi Minh City University of Transport
Ho Chi Minh City, Vietnam
chiennguyensrdn@gmail.com

Van-Hau Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
haunv@utehy.edu.vn

Le Van Quoc Anh[1]
Faculty of Information Technology
Ho Chi Minh City University of Transport
Ho Chi Minh City, Vietnam
anh@ut.edu.vn

***Abstract*—Detecting abnormal GPS trajectories derived by the mobility of people, cars, buses, and taxis plays a crucial role in developing applications for intelligent transportation systems. Outlier detection based on classification models is among promising approaches but it faces the imbalanced data problem, where instances labeled as abnormal have a very low number of observations. In this paper, we propose a framework that employs methods to deal with imbalanced data to the problem of GPS trajectory outlier detection. Our experiments show that dealing with imbalanced data beforehand can improve the performance of outlier detection models.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

For the past few years, with the popularity of smart devices with positioning technology like GPS (Global Position System), more and more GPS trajectory data are collected and available for analysis. Basically, trajectory datasets contain sequences of time-stamped points, each of which consists of latitude, longitude and altitude information. Such data represent space-time information for tracking moving objects.

Mining from GPS trajectory data plays an important role to develop applications in intelligent transportation systems. Among data mining tasks from GPS trajectory data, outlier detection, which aims to identify abnormal moving behaviors, has received considerable attention [1]. Due to the characteristics of the GPS trajectory data and there is no clear definition of trajectory anomaly, the outlier detection task is facing many challenges. One direction to solve the unclear definition of trajectory anomaly is to allow users specify which trajectories are abnormal. In this way, trajectories or subsequences of trajectories are labeled as normal or abnormal, and then the labeled datasets are used to train a machine learning model. The biggest advantage of such an approach is that the machine learning models capture the same definition of trajectory anomaly as humans do. However, one drawback of this approach is that training datasets may contain less abnormal instances than normal ones. This leads to the problems of imbalanced data in machine learning.

The data set is Imbalanced when the majority label component is much superior to the remaining label component [2]. Using imbalanced data for training a machine learning model poses a challenge since the model can bias towards the majority class. This results in models that have inferior predictive performance, especially for the minority class [3]. In the problem of GPS trajectory outlier detection, there is more sensitive to classification errors for the abnormal instances than the normal ones.

To handle imbalanced data, several approaches have been proposed, such as changing the performance metrics, modified the algorithms or the data. In this paper, we study significant techniques for dealing with imbalanced data and apply them to GPS trajectory data. The results show that such techniques can be utilized and integrated to the mining process to improve the performance of the outlier detection from GPS trajectory data.

II. Background and Related Work

A. GPS trajectory data

This paper uses the following concepts to develop the issue:

- **GPS point**: we represent a GPS point by a tuple *<id, latitude, longitude, timestamp>*, where *id* is the identifier of the moving object; and the last three components describe the position and the timestamp of the moving object.
- **GPS trajectory:** a series of GPS points that have the same *id* component and the points are arranged in time order.
- **GPS dataset or GPS log:** a set of GPS trajectories.

B. Outlier detection from GPS trajectory data as classification problem

In this paper, we focus on approaches to the problem of outlier detection from GPS data that are based on classification methods. In this way, sub-trajectories are labeled as normal or abnormal in some way, for example a manual method or using histogram-based approaches. We describe the framework in detail in Section III.

[1] Corresponding author. Email: anh@ut.edu.vn

Since the above approach depends on manually labeling, the number of instances labeled as abnormal has a very low number of observations in comparison with the number of instances labeled as normal. Therefore, we need to handle imbalanced data in this case.

C. Handling Imbalanced data

Resampling methods in disequilibrium learning applications alter the data set with mechanisms and techniques for a more balanced distribution [2]. Basic classification algorithms have better performance on balanced data sets, as mentioned in previous studies [3],[4].

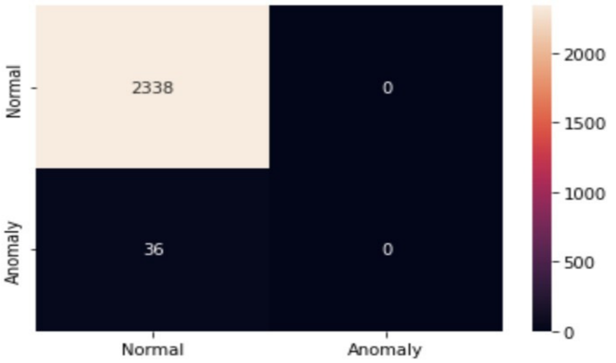


Figure 2. The original data classification model confusion matrix (SVM).

Figure 2 shows that the prediction rate of the majority class is very high, but the minority class is very low. Therefore, even distribution between classes is very important for classification algorithms.

Sampling Methods for Imbalanced Learning

● **Tomek Links(T-Link)**

Tomek links is a data cleaning technique to remove overlapping, noise applied from sampling methods.

```
$x \in S_{maj}$ , $y \in S_{min}$
d(x,y) distance between x and y
If there is no instance k, such that d(x,y)<d(x,k) or d(x,y)< d(y,k):
    (x, y) is a T-Link
Else:
    x or y is nosie or x and y is near border
```

So use Tmoke Link to "cleanup" unwanted noisy data, discarding until all the closest neighbor pairs that are farthest apart belong to a class. The removal of noisy data makes the classifier better and improves the performance of the system.

● **Synthetic Minority Oversampling Technique (SMOTE)**

SMOTE is an progressive technique of over-sampling developed by Chawala. Procedure consists of steps [2]:

```
● For each sampler $X_0$ in minority class:
– Pick one of its K nearest neighbors X ∈ minority class
– Create Z is a new sampler, as follows:
        $Z = X_0 + w(X - X_0)$        (1)
where w is Random Uniform range [0, 1].
(1) => Z is a random point on the line with equation:
    $X_0 + w(X - X_0) = 0$
```

The samples synthesized according to (1) is a point on the line connecting X_0under consideration and K-nearest is randomly selected X.

● **Adaptive Synthetic Sampling (ADASYN)**

ADASYN, on the different hand, uses a systematic procedure to adaptively create extra pieces of synthetic data according to their allotments. The algorithm is described in detail [5]. The main concept of the ADASYN algorithm is to use density distribution as a measure to automatically decide how many aggregate samples need to be developed for each minority example by adaptively modifying the weights of the examples different minority examples to compensate for the unequal distribution[6].

● **Random Oversampling and Undersampling**

One of the familiar approaches was to use resampling techniques to construct the dataset balanced. Oversampling or oversampling can be applied to the resampling of the data set. Reducing the number of elements in the data set is the idea of undersampling. The process of oversampling is the multiplication of minority cases by duplicating or repeating some cases. [7].

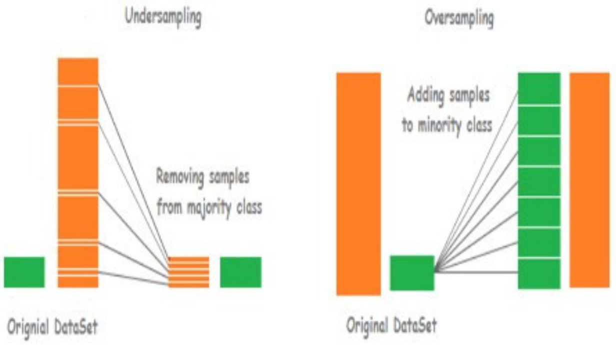

Undersampling
Removing samples from majority class
Orignial DataSet
Oversampling
Adding samples to minority class
Original DataSet

Figure 3. Differences between undersampling and oversampling

● **SVM-SMOTE (SVMs)**

A variant of the SMOTE algorithm which uses an SVM algorithm to detect samples to use for developing new synthetic samples as presented in [8]. SVM uses the concepts of support vector, maximize soft-margin, and hyperplane to classify between data samples, to improve performance, reduce classification errors during data resampling [9].

III. Proposed Framework

In this section, we describe our framework for detecting sub-trajectories with anomalous motion behavior. The framework consists of four steps.

The first step is a processing step that extracts trajectories by time frame. The second step is a feature extraction. We derive objects in the form of a tuple <*s,m,d,t,l,v*> when *s* is the start point, *m* is a midpoint, *d* is the destination point, *t* is the timestamp, *l* is the distance length, and *v* is the velocity. Each object is labeled as normal or abnormal by humans or by an automatic method like histogram-based method as described in [10].

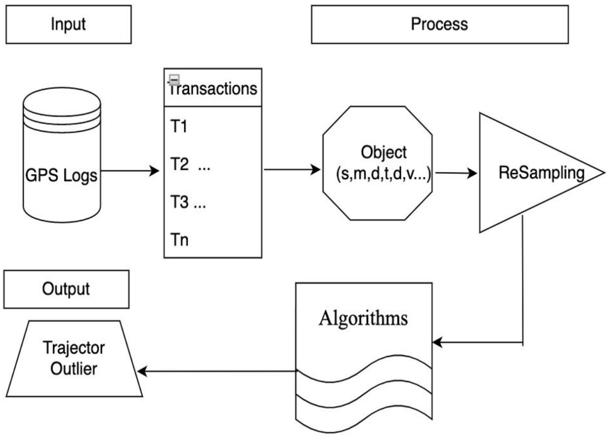


Figure 1. Framework to detect abnormal movement behavior

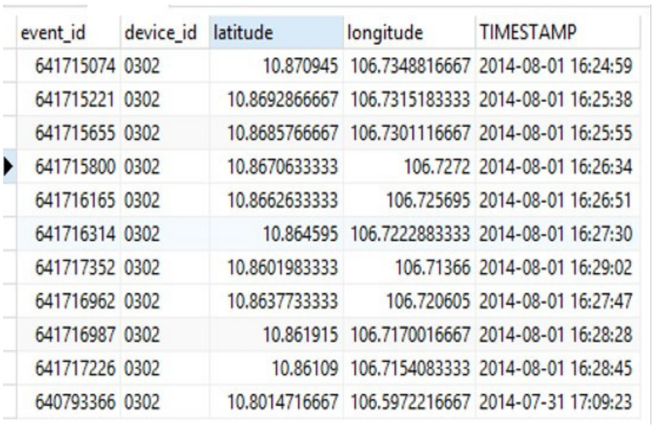
The third step applies techniques to deal with imbalanced data. Finally, in the last step, one can apply any classification algorithm to classify the objects as abnormal or not.

IV. Experimental Results

D. Dataset Description and Experimental Setup

We use two data sources to demonstrate our proposed approach. The first dataset is provided by the OTS transport service monitoring company. Itinerary data is exploited on Ho Chi Minh City routes. The dataset has 411 vehicles, mined from June 01, 2015, to June 07, 2015. This dataset is the same as the dataset using in [10],[11].

The second dataset is provided by Kaggle website. This data is exploited on Beijing routes. The dataset has 10,357 taxis, mined from June 01, 2015, to June 07, 2015. The total number of points in this dataset is about 15 million and the total distance of the trajectories reaches to 9 million kilometers. Figure 4 demonstrates the format of the GPS Log of the data in Ho Chi City, where each record represent a GPS point in a trajectory.



	event_id	device_id	latitude	longitude	TIMESTAMP
	641715074	0302	10.870945	106.7348816667	2014-08-01 16:24:59
	641715221	0302	10.8692866667	106.7315183333	2014-08-01 16:25:38
	641715655	0302	10.8685766667	106.7301116667	2014-08-01 16:25:55
▶	641715800	0302	10.8670633333	106.7272	2014-08-01 16:26:34
	641716165	0302	10.8662633333	106.725695	2014-08-01 16:26:51
	641716314	0302	10.864595	106.7222883333	2014-08-01 16:27:30
	641717352	0302	10.8601983333	106.71366	2014-08-01 16:29:02
	641716962	0302	10.8637733333	106.720605	2014-08-01 16:27:47
	641716987	0302	10.861915	106.7170016667	2014-08-01 16:28:28
	641717226	0302	10.86109	106.7154083333	2014-08-01 16:28:45
	640793366	0302	10.8014716667	106.5972216667	2014-07-31 17:09:23

Figure 4. GPS Log collected from a vehicle tracking device

E. Results

We give an example of an oversampling technique. Its main goal is class balance, by random repetition of minority samples.

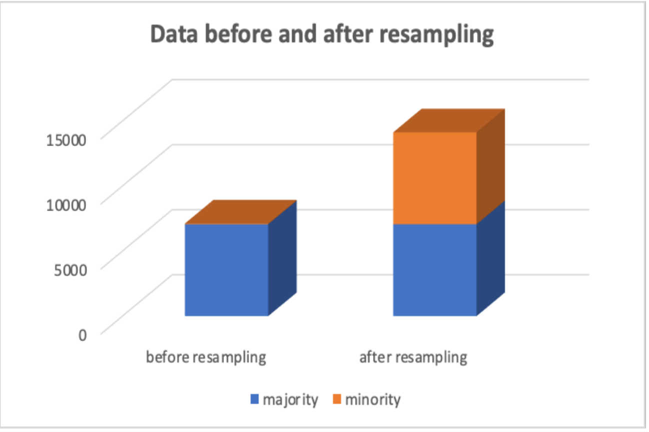


Figure 5. Data before and after resampling (oversampling)

Figure 5 shows how the class target is distributed after using this method on our dataset and it equals to 7,005.

However, this technique has two limitations. First, it will rise the probability of over-fitting, as it creates the same reproductions of the minority class instances [2]. Second, it makes the learning process take longer if the initial data set is very large, but balanced.

In this paper, we only give the imbalance handling solutions in order to show that it will achieve better minority class recognition performance. The need for preprocessing, resampling data is very important for the problem of data imbalance in machine learning.

The performance of the model is evaluated using measures such as Weighted accuracy, F-scoere, G-means. The following is a summary of each measure:

$$Sensitivity: The\ True\ Positive\ rate\left(TP\right)=\frac{TP}{TP+FN}$$

$$Specificity: The\ True\ Negative\ rate\left(TN\right)=\frac{TN}{FP+TN}$$

$$G-mean=\sqrt{Sensitivity*Specificity}$$

$$Weighted\ accuracy=0.5*(Sensitivity+Specificity)$$

$$Precíion=\frac{TP}{TP+FP};\ Recall=\frac{TP}{TP+FN}$$

$$F-Score=2\frac{Precíion*Recall}{Precíion+Recall}$$

F. *GPS route data Ho Chi Minh City-Viet Nam*

For real datasets, we obtain GPS Log from a company providing vehicle tracking services, called OTS. Itinerary data is exploited on Ho Chi Minh City routes. The dataset has 411 vehicles, mined from June 01, 2015, to June 07, 2015 [10].

Table I. Performance measures G-mean, F-score, Weighted accuracy (algorithm SVM).

SVM	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.6325	0.0507	0.1620
SMOTE/T-Link	0.6351	0.0515	0.1632
Over-Sampling	0.6390	0.0545	0.1683
Over/T-Link	0.6646	0.0587	0.1748
ADASYN	0.6328	0.0508	0.1621
SVMSMOTE	0.5865	0.0687	0.1968

Table II. Performance measures G-mean, F-score, Weighted accuracy (algorithm Logistic Regression).

LR	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.5526	0.0355	0.1343
SMOTE/T-Link	0.5523	0.0355	0.1343
Over-Sampling	0.5261	0.0324	0.1284
Over/T-Link	0.5399	0.0340	0.1315
ADASYN	0.5370	0.0337	0.1308
SVMSMOTE	0.5192	0.0351	0.1390

Table III. Performance measures G-mean, F-score, Weighted accuracy (algorithm Random Forest)

RF	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.6670	0.1032	0.2404
SMOTE/T-Link	0.6523	0.0958	0.2315
Over-Sampling	0.5274	0.1000	0.7020
Over/T-Link	0.5132	0.0500	0.4963
ADASYN	0.6504	0.0932	0.2279
SVMSMOTE	0.5558	0.0952	0.2674

Summary of results when applying data resampling techniques when applied to different data classification algorithms:

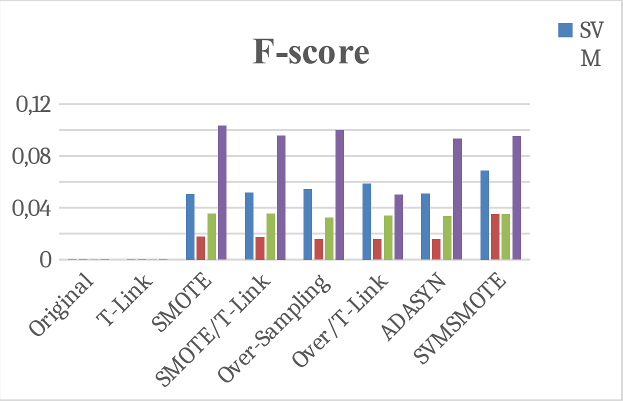


Figure 6. Weighted accuracy of various Machine learning algorithms using various sampling techniques

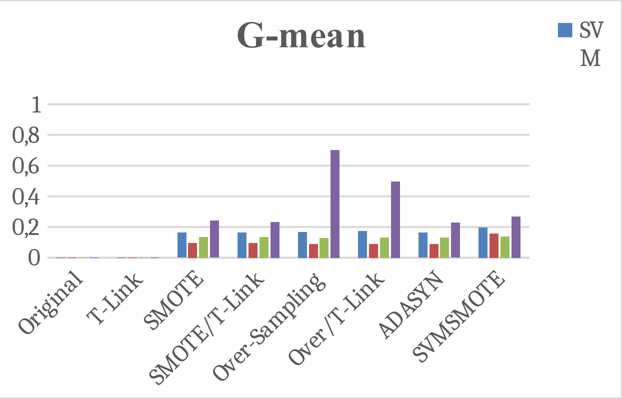


Figure 7. F-score of various Machine learning algorithms using various sampling techniques

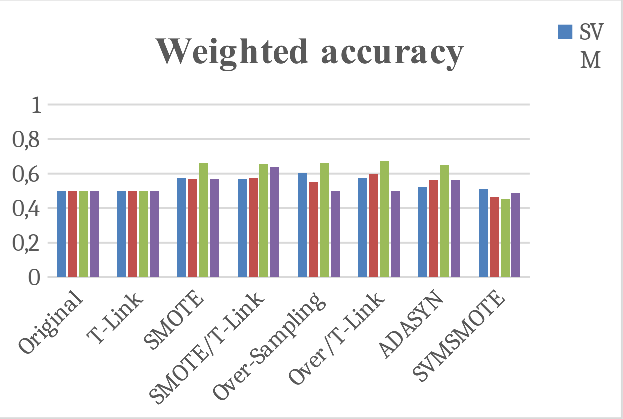


Figure 8. G-mean of various Machine learning algorithms using various sampling techniques

The results show that when applying data resampling techniques, such as minority reinforcement sampling, the majority group reduction, the model predicts better than the minority group on all the classification algorithms being studied. However, even though the prediction accuracy of the majority class was slightly reduced. But the indicators of sensitivity and accuracy of the model such as Weighted Accuracy, G-mean, F-score have changed markedly Table I Table II Table III. Achieving the original purpose of the article detected unusual journeys.

SVM: G-means F-score and Weighted accuracy the best results with SVM-SMOTE resampling technique of 19.68%, 6.87%, and 58.65%, respectively.

LR: G-means the best results with SVM-SMOTE resampling technique of 13.9%. F-score and Weighted-accuracy the best results with SMOTE resampling technique of 3.55%, 55.26% respectively.

RF: G-means the best results with Over-Sampling resampling technique of 70.2%. F-score and Weighted-accuracy the best results with SMOTE/T-Link resampling technique of 9.58%, 65.23% respectively.

In each different classification algorithm, there will be data resampling techniques suitable for each algorithm.

G. *GPS route data Shanghai-China*

We have used the dataset provided by the competition from Kaggle website. This data is exploited on Beijing routes. The dataset has 10,357 taxis, mined from June 01, 2015, to June 07, 2015.

Table IV. Performance measures G-mean, F-score, Weighted accuracy (algorithm SVM)

SVM	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.5709	0.0320	0.1276
SMOTE/T-Link	0.5683	0.0316	0.1269
Over-Sampling	0.6042	0.0336	0.1307
Over/T-Link	0.5746	0.0313	0.1261
ADASYN	0.5219	0.0266	0.1162
SVMSMOTE	0.107	0.0259	0.1161

Table V. Performance measures G-mean, F-score, Weighted accuracy (algorithms Logistic Regression)

LR	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.6574	0.0389	0.1408
SMOTE/T-Link	0.6553	0.0386	0.1403
Over-Sampling	0.6596	0.0373	0.1378
Over/T-Link	0.6733	0.0413	0.1345
ADASYN	0.6490	0.0377	0.1386
SVMSMOTE	0.4505	0.0	0.0

Table VI. Performance measures G-mean, F-score, Weighted accuracy (algorithms Random Forest).

RF	Weighted accuracy	F-score	G-mean
Original	0.5	0.0	0.0
T-Link	0.5	0.0	0.0
SMOTE	0.5655	0.0469	0.1600
SMOTE/T-Link	0.6351	0.0515	0.1632
Over-Sampling	0.4989	0.0	0.0
Over/T-Link	0.500	0.0	0.0
ADASYN	0.5634	0.0455	0.1573
SVMSMOTE	0.4847	0.0	0.0

Summary of results when applying data resampling techniques when applied to different data classification algorithms:

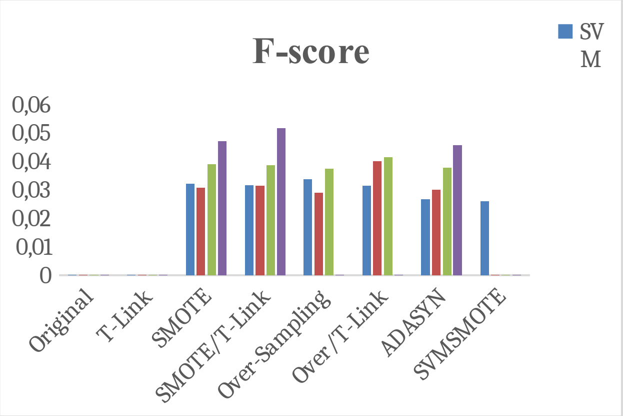


Figure 9. Weighted accuracy of various Machine learning algorithms using various sampling techniques

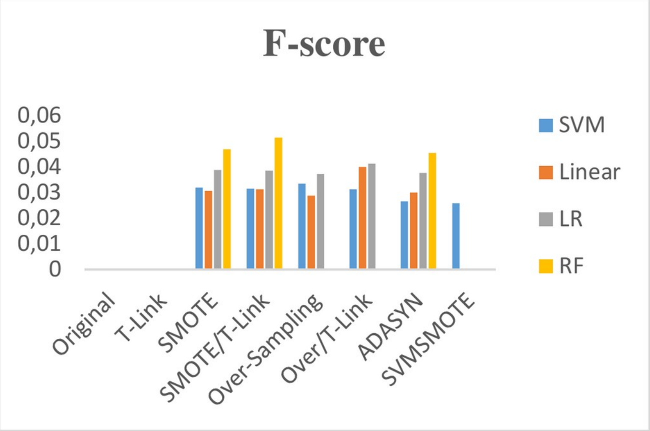


Figure 10. F-score of various Machine learning algorithms using various sampling techniques.

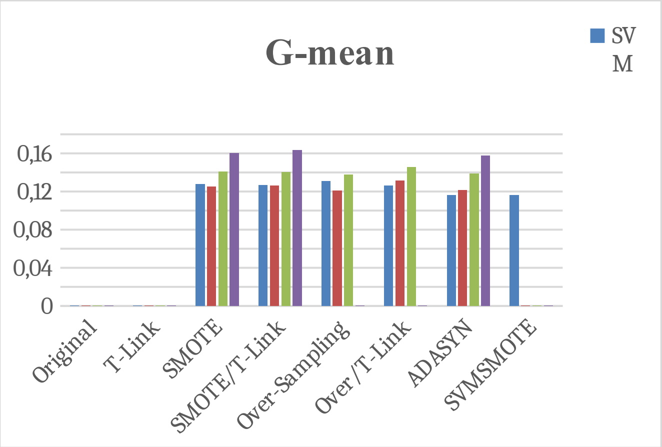


Figure 11 G-mean of various Machine learning algorithms using various sampling techniques

SVM: G-means F-score and Weighted accuracy the best results with SMOTE resampling technique of 12.76%, 3.2%, and 57.09%, respectively.

LR: G-means, F-score and Weighted accuracy achieved the best results with Over/T-Link resampling technique of 13.45%, 4.13%, and 67.33%, respectively.

RF: G-means the best results with SMOTE/T-Link resampling technique of 16.32%. F-score and Weighted-accuracy the best results with SMOTE resampling technique of 5.15%, 65.51% respectively.

Summary: Figure (6-11) We grouped by model evaluation techniques, for each resampling technique for each machine learning algorithm, to show that applying resampling techniques achieves results high when predicting the minority class.

Table (I-IV): G-mean, F-score all algorithms that do not apply resampling have the value 0, because the prediction of the minority cases is incorrect.

Weighted accuracy also shows that performance is improved when resampling techniques are applied, the results are show in Table (I-IV) and Figure (6-11).

Looking at (Table I-IV), our data showed that using SMOTE and T-Link as a combined sampling method has a better performance than T-Link sampling and the original data.

V. Conclutions and Future Work

In this paper, we solve imbalanced data problems in GPS outlier detection and prove that it will achieve better performance for overall process. The need for preprocessing, and resampling data is very important for the problem of data imbalance in machine learning. We demonstrate the effectiveness with real datasets, i.e., GPS logs from vehicle tracking services. We apply resampling techniques to the classification models and compare them with the original (unbalanced) data set. We find that the application of resampling techniques achieves better results for minority class prediction. For future work, we are planning to apply deep learning techniques, learn the optimal threshold of data resampling to deal with class imbalance to improve the predictive model.

References

[1] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 316-324.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research,* vol. 16, pp. 321-357, 2002.

[3] V. S. Spelmen and R. Porkodi, "A review on handling imbalanced data," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018: IEEE, pp. 1-11.

[4] P. Nair and I. Kashyap, "Hybrid pre-processing technique for handling imbalanced data and detecting outliers for KNN classifier," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019: IEEE, pp. 460-464.

[5] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, 2008: IEEE, pp. 1322-1328.

[6] L. Gautheron, A. Habrard, E. Morvant, and M. Sebban, "Metric learning from imbalanced data," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019: IEEE, pp. 923-930.

[7] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: overview study and experimental results," in *2020 11th international conference on information and communication systems (ICICS)*, 2020: IEEE, pp. 243-248.

[8] H. M. Nguyen, E. W. Cooper, and K. Kamei, "Borderline over-sampling for imbalanced data classification," in *Proceedings: Fifth International Workshop on Computational Intelligence & Applications*, 2009, vol. 2009, no. 1: IEEE SMC Hiroshima Chapter, pp. 24-29.

[9] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[10] C. Nguyen, T. Dinh, V.-H. Nguyen, N. P. Tran, and A. Le, "Histogram-based Feature Extraction for GPS Trajectory Clustering," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems,* vol. 7, no. 22, 2020.

[11] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "T-drive: Enhancing driving directions with taxi drivers' intelligence," *IEEE Transactions on Knowledge and Data Engineering,* vol. 25, no. 1, pp. 220-232, 2011.

DOI: 10.15439/2022R27
ISSN 2300-5963 ACSIS, Vol. 33

Automated Code Generation from Use cases and the Domain Model

1st Minh-Hue Chu
Hung Yen University of Technology and Education, Vietnam
Email: huectm@gmail.com

2nd Anh-Hien Dao
Hung Yen University of Technology and Education, Vietnam
Email: hienda@gmail.com

***Abstract*—In this paper, we propose a method to automatically generate source code files from a use case model and a domain class diagram named USLSCG (Use case Specification Language (USL) based Code Generation). In our method, a use case scenario is precisely specified by a USL model. The USL model and the domain class diagram then are used as inputs to generate source code files automatically. These source code files include classes following three-layer applications and a SQL script file to create a database and store procedures.**

***Index Terms*—Generate source code, USL, Use case, USLSCG**

I. Introduction

The software development life cycle is divided into some main stages. In the first stage, software requirements are documented in the SRS (Software Requirement Specification) document. These requirements are usually documented by UML (Unified Modeling Language) models and statements in the natural language. In the second state, design documents then are built from the SRS document. Models in design documents present different design views, for example, database designs, architecture designs, object designs, user interface designs, etc. Next state, the design models are implemented into the code source. Finally, the testing activity is performed to ensure the quality of software products [1]. The input of design and test stages are the software requirements in the SRS document that are usually documented by use case diagrams and textual use case descriptions in the template-based natural language [2]. Design models are then input for programmers to transform into source code files. These activities are usually performed manually by developers. Firstly, They will read software requirement specification documents which are typically several hundred pages to build analysis models, design models, and test cases. They then transform design models into source code files. However, in software development, requirements usually change during development. So, when the software requirements change, analysis models, design models, source code, and test cases must be rebuilt.

To reduce the time and cost of software development, automation solutions are proposed and developed. A major challenge for automation in software development is software requirements described in the natural language and modeled by models that are not precise enough. proposed a model To deal with this challenge, [3] proposed a DSML (Domain-Specific Modeling Language) named USL (Use case Specific Language) to precisely specify textual use case descriptions for automation aims in software development. In the research, we discussed abilities to generate analysis, design models, and test cases automatically from USL models. In the previous research [4], we also proposed a method named USLTG to generate test cases from the USL models. In another research [5], we also proposed a method named USLCG. The USLCG method allows generating design class diagrams automatically from USL models and the domain class diagram. In this paper, we focus on generating source code files automatically from USL models and the domain class diagram. These source code classes conform to design class diagrams generated in the previous research [5]. We named this method USLSCG. Firstly, functional requirements are captured by UML use case diagrams and USL models which specify use case descriptions precisely. In addition, the Entities of the system are captured by a domain class diagram in UML. Secondly, for each use case, USLSCG transforms the corresponding inputs above into source code classes of the use case. Besides, we also generate automatically a SQL script file containing T-SQL statements for creating a relational database and store procedures of the database.

To summarize, the main contributions of this paper are:

- the USLSCG method to generate automatically source code files from use cases and the domain class diagram;
- a set of rules to map action types into source code classes and methods of classes.
- algorithms to transform use cases into source code files
- a generator to realize the USLSCG method.

The rest of this paper is organized as follows. Section II introduces the background and motivation for developing USLSCG. Section III shows our proposed approach. Section IV explains how to generate source code files of the USLSCG method. Section V briefly discusses the tool support of USLSCG. Section VI discusses related works. The paper is closed with a conclusion and future works.

II. Background and Motivation

In this section, we first discuss the basic knowledge that we use in this research. We then present our motivations.

a) Use cases: A use case describes a sequence of interactions between a system and an external actor that results in the actor being able to achieve some outcome of value [6]. In the SRS documents, use cases are commonly used for capturing and structuring the functional requirements of software systems. Use cases are widely modeled by UML use case diagrams and each use case is loosely structured by textual descriptions following the structure as in [2]. Use case models are central models in software development. These models will be used as inputs to build different software artifacts including activity diagrams, class diagrams, sequence diagrams, source code, functional test cases, and so on. For example, Figure 1 shows a simplified use case model of an ATM system, Table 1 shows a specification of the use case Withdraw describing event flows of this use case. In this paper, we used the Withdraw use case for illustrative examples.

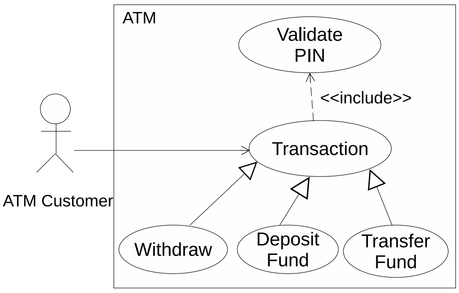


Fig. 1. A simplified use case model of the *ATM* system.

TABLE I
A TEMPLATE-BASED DESCRIPTION OF THE *Withdraw* USE CASE

Use case name: *Withdraw*
Brief description: The customer withdraw cash.
Primary actors: Customer
Precondition: The Insert Card use-case was success.
Postcondition: If the use-case was successful, the system updates the balance, dispenses the cash, prints a receipt for the user. If not, the system displays an error message.
Trigger: User selects the Withdraw function.
Special requirement: There is no special requirement.

Basic flow

1. The customer enters the withdrawal amount.
2. The system validates that the ATM has enough funds in the user account. If the user account has not enough funds go to step 2a.1.
3. The system generate the withdrawal transaction information.
4. The system sends the withdrawal transaction information to the Bank system.
5. The Bank system gets the withdrawal transaction information.
6. The Bank system sends the withdrawal transaction approval to the ATM system.
7. The system gets the withdrawal transaction approval from the Bank system. If the bank do not approve the withdrawal transaction, then go to step 7a.1.
8. The system updates the balance of the user account; The system dispenses the cash in the cash dispenser.
9. The customer gets the cash from the cash dispenser.
10. The system records the withdrawal transaction information.
11. The withdrawal transaction ends.

Alternate flows

2a. If the user account has not enough money.
 1. The system displays an error message on the customer console and go to step 1.
7a. The bank do not approve the withdrawal transaction.
 1. The system displays an error message on the customer console.
 2. The system records the withdrawal transaction information.
 3. The withdrawal transaction end.

b) Domain model: A domain model captures entities in a system. In software development, this model is usually specified by a UML class diagram including three types of elements: (1) domain conceptual classes, (2) attributes, and (3) relationships among conceptual classes. Domain conceptual classes represent objects used in the system use cases. Figure 2 shows the ATM domain model is picked from the work [7]. shows the ATM domain model in UML. In this research, a domain model in the UML class diagram is called a domain class diagram.

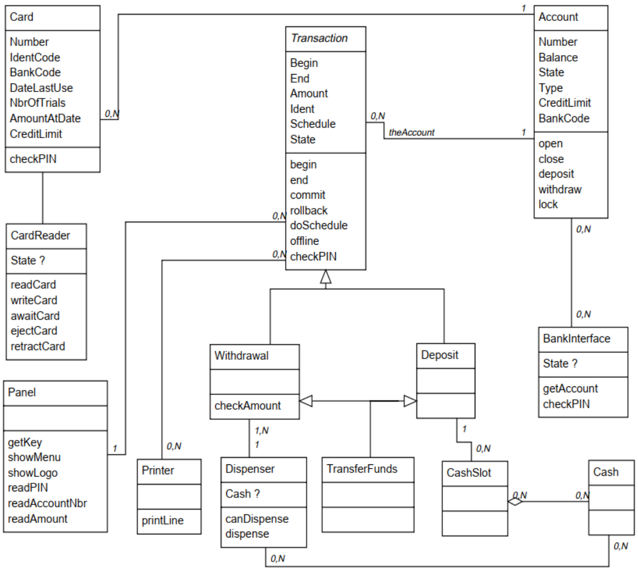


Fig. 2. The domain class diagram of the *ATM* system.

c) Three-layer architecture: Layers indicate the logical separation of components. A Layered architecture concentrates on grouping related functionality within an application into distinct layers that are stacked vertically on top of each other. Each layer has unique namespaces and classes. In the three-layer architecture, there are three layers. The first layer is the presentation layer where users can interact with the application. The second layer is the business logic layer. This layer is the middle layer - the heart of the application. It contains all the business logic of the application and describes how business objects interact with each other, where the presentation layer and data access layer can indirectly communicate with each other. The third layer is the data access layer. This layer enforces rules regarding accessing data, providing simplified access to data stored in persistent storage, such as SQL Server, and MySQL. It is noteworthy that this layer only focuses on data access instead of data storage. Besides, we have an extra layer called the business objects layer. This layer contains objects that are used in the application and common helper functions (without logic) used for all layers. In the three-layer architecture, the business objects layer is optional. However, as we follow the OOP, we should reduce the duplicate codes as much as possible. Therefore, using this layer to keep common codes instead of holding them in each layer is essential. In this article, we choose three-layer architecture to build template files for our solution

In software development, the source code of an application is transformed by programmers from design models manually. However, when software requirements are changed late, the design and source code must be rebuilt. To reduce efforts in software development, design models and source code need to be generated from software requirements automatically. A motivating question is how source code can be generated automatically from a use case what generated source code follows the design class models? This solution helps to semi-automate the implementation process. In order to automatically transform use cases and the domain model into source code files, we need to address the following main challenges.

- How do generate the source code of classes from a use case specification? Here, each generated class belongs to one of three layers (Presentation, Business Logic, Data Access). Besides, the domain class diagram is transformed into classes in the business objects layer.
- How are methods of source code classes defined? Operations of classes are defined based on the message passing among objects, i.e, objects collaborate together to realize a use case. We also define the parameters of the generated methods.

III. Overview of our approach

Our USLSCG approach is illustrated in Figure 3. This approach uses USL to specify each use case description by a USL model conforming to the USL metamodel as shown in Figure 4. We take a USL model, a UML class model capturing domain concepts of the system, and source code templates as inputs to generate source code automatically. In SubSection III-A, we briefly explain the language USL. Our USLSCG method includes two main steps: Step 1 aims to extract actions with constraints from a USL model; Step 2 takes inputs including extracted actions, domain classes, and source code templates aim to generate source code classes and a SQL Script file. A detailed explanation of these steps is presented in Section IV.

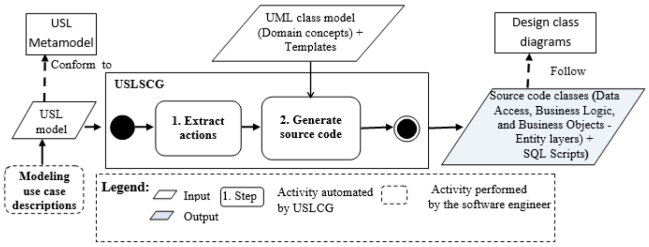

USL Metamodel
Conform to
USL model
Modeling use case descriptions
UML class model (Domain concepts) + Templates
USLSCG
1. Extract actions
2. Generate source code
Design class diagrams
Follow
Source code classes (Data Access, Business Logic, and Business Objects - Entity layers) + SQL Scripts)
Legend:
Input
Output
1. Step
Activity automated by USLCG
Activity performed by the software engineer

Fig. 3. Overview of the USLSCG Approach.

A. Capturing Use Case descriptions in USL models

The USL language is a DSML to specify use cases precisely. This language was introduced in other research [3]. The USL is defined based on the metamodeling technique. The metamodel of the USL was determined as in Figure 4. The USL approach aims to specify use case descriptions as USL models that could be automatically transformed into other software artifacts, including analysis, design models, source code, and test cases. The following explains how to specify a use case description in natural language by a USL model:

- The use-case-overview field is specified by the *DescriptionInfo* object properties.
- Steps of the basic flow are specified by *FlowStep*s, that are connected by *BasicFlowEdge*s and *ControlNode*s. These steps are either *ActorStep*s or *SystemStep*s.
- Steps of each alternate flow are specified by *FlowStep*s that are linked by *ControlNode*s and *AlternateFlowEdge*s.
- An *ActorStep* can include one or more *ActorRequest*s and *ActorInput*s.
- A *SystemStep* can include one or more *SystemInput*s, *SystemDisplay*s, *SystemRequest*s, *SystemState*s, *SystemOutput*s, *SystemInclude*s, and *SystemExtend*s.
- Use case constraints, guard conditions, and actions in *SystemStep*s or *ActorStep*s are captured by *Constraint*s associated with *InitialNode*, *FinalNode*, *BasicFlowEdge*s, *AlternateFlowEdge*s, and *Action*s.

The USL model representing the use case *Withdraw* is shown as in Figure 5. This model captures the description fields of the use case *Withdraw* by the use-case-overview field, the basic flow by steps from *s1* to *s10*, the alternate flow *2a* by steps *s11*, the alternate flow *7a* by steps *s12* and *s13*, the actions in the steps by actions from *a1*. . .*a14*, the guard constraints to select between the flows by Constraints from *g1*. . .*g4*, and the postcondition of actions by Constraints from *p1*. . .*p3*. In particular, *s1*, *s5*, *s6*, *s9* are *ActorStep*s; the other *Step*s are *SystemStep*s; *a1* and *a6* are *ActorInput*s (the actor of *a1* is a person, the actor of *a6* is an external system); *a5* and *a10* (requesting object of *a10* is a device) are *ActorRequest*s; *a3* is a *SystemOperation*; *a4*, *a7*, and *a9* are *SystemRequest*s; *a2*, *a8*, *a11*, *a14* are *SystemState*s; *a12* and *a13* are *SystemOutput*s.

IV. Transforming Use case into Source Code

In this section, we present about two steps of USLSCG as shown in Figure. 3 and explain how USLSCG can automatically generate source code files from a USL model and a UML domain class diagram.

A. Extraction of Actions

In the first step of USLSCG as in previous research, we develop an algorithm named *ExtActions* to extract a set of actions from a USL model, as shown in Algorithm 1. Algorithm *ExtActions* takes *D*, a USL model, as input. The output of this Algorithm is a set of Actions.

Example IV.1. Algorithm *ExtActions* will return thirteen Actions (*a1* . . . *a13*) from the use case *Withdraw*. The extracted actions belong to one of nine action types as in the part (d) of Figure. 4.

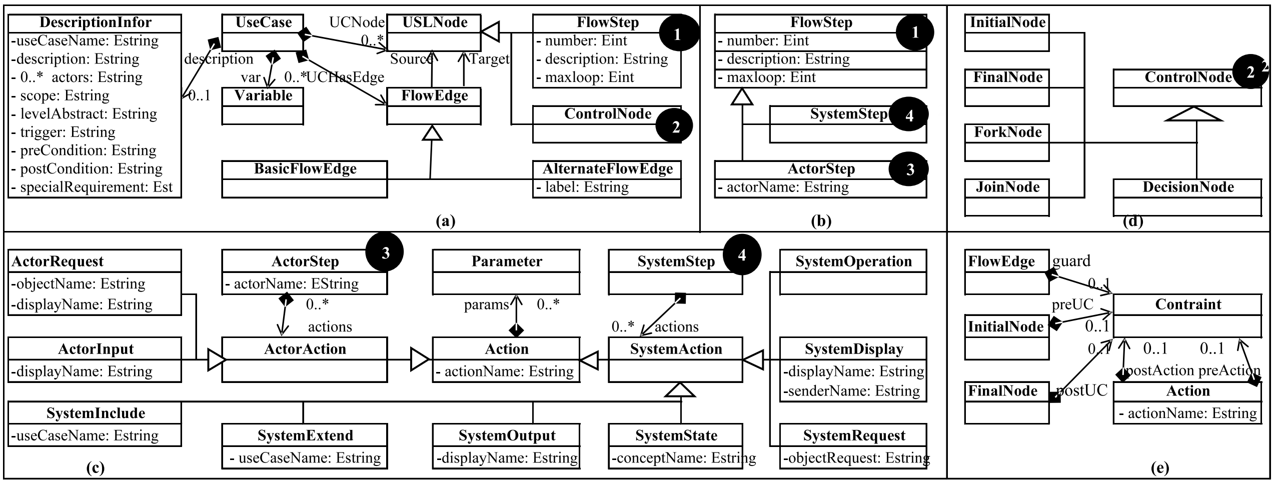
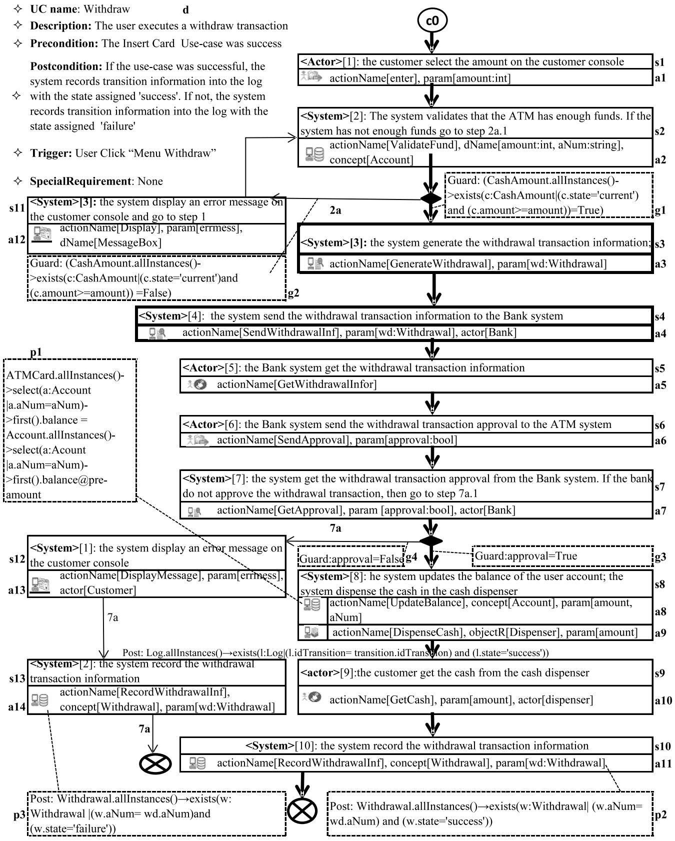


Fig. 4. The USL metamodel.


✧ **UC name**: Withdraw

✧ **Description:** The user executes a withdraw transaction

✧ **Precondition:** The Insert Card Use-case was success

✧ **Postcondition:** If the use-case was successful, the system records transition information into the log with the state assigned 'success'. If not, the system records transition information into the log with the state assigned 'failure'

✧ **Trigger:** User Click "Menu Withdraw"

✧ **SpecialRequirement**: None

<Actor>[1]: the customer select the amount on the customer console

actionName[enter], param[amount:int]

<System>[2]: The system validates that the ATM has enough funds. If the system has not enough funds go to step 2a.1

actionName[ValidateFund], dName[amount:int, aNum:string], concept[Account]

Guard: (CashAmount.allInstances()->exists(c:CashAmount|(c.state='current') and (c.amount>=amount))=True)

<System>[3]: the system display an error message on the customer console and go to step 1

actionName[Display], param[errmess], dName[MessageBox]

Guard: (CashAmount.allInstances()->exists(c:CashAmount|(c.state='current')and (c.amount>=amount)) =False)

<System>[3]: the system generate the withdrawal transaction information;

actionName[GenerateWithdrawal], param[wd:Withdrawal]

<System>[4]: the system send the withdrawal transaction information to the Bank system

actionName[SendWithdrawalInf], param[wd:Withdrawal], actor[Bank]

ATMCard.allInstances()->select(a:Account |a.aNum=aNum)->first().balance = Account.allInstances()->select(a:Acount |a.aNum=aNum)->first().balance@pre-amount

<Actor>[5]: the Bank system get the withdrawal transaction information

actionName[GetWithdrawalInfor]

<Actor>[6]: the Bank system send the withdrawal transaction approval to the ATM system

actionName[SendApproval], param[approval:bool]

<System>[7]: the system get the withdrawal transaction approval from the Bank system. If the bank do not approve the withdrawal transaction, then go to step 7a.1

actionName[GetApproval], param [approval:bool], actor[Bank]

<System>[1]: the system display an error message on the customer console

actionName[DisplayMessage], param[errmess], actor[Customer]

Guard:approval=False

Guard:approval=True

<System>[8]: he system updates the balance of the user account; the system dispense the cash in the cash dispenser

actionName[UpdateBalance], concept[Account], param[amount, aNum]

actionName[DispenseCash], objectR[Dispenser], param[amount]

Post: Log.allInstances()→exists(l:Log|(l.idTransition= transition.idTransition) and (l.state='success'))

<System>[2]: the system record the withdrawal transaction information

actionName[RecordWithdrawalInf], concept[Withdrawal], param[wd:Withdrawal]

<actor>[9]:the customer get the cash from the cash dispenser

actionName[GetCash], param[amount], actor[dispenser]

<System>[10]: the system record the withdrawal transaction information

actionName[RecordWithdrawalInf], concept[Withdrawal], param[wd:Withdrawal]

Post: Withdrawal.allInstances()→exists(w: Withdrawal |(w.aNum= wd.aNum)and (w.state='failure'))

Post: Withdrawal.allInstances()→exists(w:Withdrawal| (w.aNum= wd.aNum) and (w.state='success'))

Fig. 5. The USL model specifies the use case *Withdraw*.

B. Generating automatic source code files

In order to generate source code files automatically, We use the Acceleo project. Specifically, Acceleo is a model

Algorithm 1: ExtActions

```
ExtActions(D)
Input: D is a USL model
Output: la is a set of actions extracted from D
BEGIN
    la ← ∅;
    foreach s in D.USLNodes do
        if s is SystemStep then
            foreach a in ((SystemStep)s).SystemActions do
                la ← la ∪ a;
        if s is ActorStep then
            foreach a in ((ActorStep)s).ActorActions do
                la ← la ∪ a;
END
```

transformation language M2T of the Eclipse framework [8]. Acceleo can automatically generate source code files by using templates such as illustrated Figure 6.

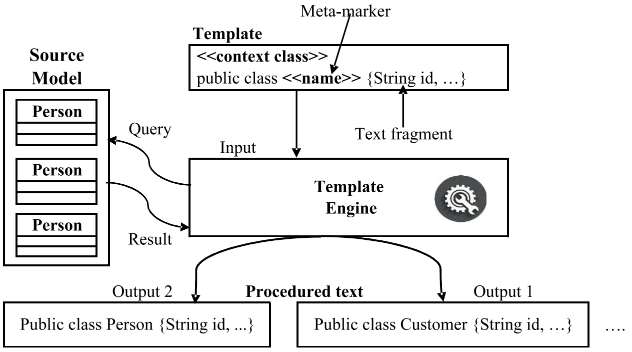


Fig. 6. Templates are used as inputs for the Accleo Project to automatically generate source code files[9].

Firstly, USLSCG uses a *Business Object template file*, a *SQL Script template file* as inputs for the Acceleo project to generate source code classes at the business objects layer, and a SQL script file automatically. Specifically, each class in the UML domain class diagram is mapped to a business object class, and each attribute of a UML class or association relationship between two UML classes is transformed into a property of the code class. Each generalization relationship is mapped into the inheritance relationship between the child and parent class. In addition, the generated SQL script file is a file script including statements to create a database, tables, and store procedures.

Secondly, USLSCG uses *Data Access template files* as inputs for the Acceleo project to generate classes automatically at the data access layer. In particular, each class in the UML domain class diagram is mapped to a code interface and class at the data access layer. To use in different situations and use cases, classes at the data access layer can be generated full of methods. We use a *Data Access frame template* shown as List 1 to generate classes at the data access layers. In addition, each method has one template to replace *DataAccess* part of the frame template, for example, List 2 is a template of the Delete method.

```
using System;
using System.Data;
using System.Collections.Generic;
using Example.Common;
using {{ProjectName}}.DataModel;
using System.Linq;
namespace {{ProjectName}}.DataAccess {
    public partial class {{tableName}}Repository : I
    {{tableName}}Repository
  {
    private IDatabaseHelper _dbHelper;
    public {{tableName}}Repository(IDatabaseHelper
    dbHelper)
    {
      _dbHelper = dbHelper;
    }
    {{DataAccess}}
    }
}
```

Listing 1. DataAccess Template

```

/// <summary>
/// Delete records in the table {{tableName}}
/// </summary>
/// <param name="json_list_id">List id want to
    delete</param>
/// <param name="updated_by">User made the deletion
    </param>
/// <returns></returns>
public List<{{tableName}}Model> Delete(string
    json_list_id,Guid updated_by)
  {
      string msgError = "";
      try
   {
    var dt = _dbHelper.
  ExecuteSProcedureReturnDataTable(out msgError, "
  sp_{{tablename}}_delete_multi",
          "p_json_list_id", json_list_id,
          "p_updated_by", updated_by);
        if (!string.IsNullOrEmpty(msgError))
    {
          throw new Exception(msgError);
        }
        return dt.ConvertTo<{{tableName}}
  Model>().ToList();
      }
   catch (Exception ex)
   {
        throw ex;
      }
    }
```

Listing 2. Template of Delete method in the Data Access class

Finally, *Business Logic* and *Presentation template files* are used as inputs for USLSCG to generate classes at the business logic and presentation layer automatically. We develop Algorithm 2 to create these classes automatically. Note that the article only generates the source code class at the presentation layer and does not discuss the interface design based on the domain classes because this problem has been handled very well by editors supporting the programming languages.

Algorithm 2: GenBP

```
GenBP(la, tems)
Input: la, a set of actions;
tems are template files;
Output: lcbps, classes at business and presentation layer
BEGIN
Create a business class following templates corresponding to the use case
  named BC;
foreach a in la do
    switch $TypeOf(a)$ do
        case $ActorInput$ do
            if Business(a.ActorName) don't exit then
                Create a presentation class corresponding to
                  a.ActorName;
            CreatePresentationMethod(a);
        case $ActorRequest$ do
            if Presentation(a.ActorName) don't exist then
                Create a presentation class corresponding to
                  a.ActorName;
            if a.RequestObjectType is system then
                CreatePresentationMethod(a);
        case
          $SystemOperation$ or $SystemExtend$ or $SystemRequest$
          do
            CreateBusinessMethod(a);
        case $SystemDisPlay$ or $SystemOutput$ do
            CreatePresentationMethod(a);
        case $SystemState$ do
            CreateBusinessMethod(a);
END
```

Algorithm *GenBP* takes the set of constrained actions *la* which are extracted from Algorithm *GenActions* as input. The output of *GenBP* is classes conforming to class diagram *ACD* in the research [5].

Firstly, Algorithm *GenBP* generates a business class called *BC* and a presentation class called *PC* for each USL model. Nextly, Algorithm *GenBP* travers actions in *la* and transforms each action to corresponding methods of the *BC* or PC. Algorithm *GenBP* employs the functions Presentation(a.ActorName), CreatePresentationMethod(a), CreateBusinessMethod(a) that are explained below.

- The function *Presentation (a.ActorName)* checks whether a presentation class corresponding to the actor of action *a* exists. this function will return true if this class exists.
- The function *CreatePresentationMethod(a)* generates a method for the presentation class *PC* corresponding to

action *a*. The generated method is based on properties *ActionName* and *Parameters* of action *a*.
- The function *CreateBusinessMethod(a)* generates a method for the business class *BC*. The generated method is based on the properties *ActionName* and *Parameters* of action *a*.

Specifically, methods CreatePresentationMethod(a), CreateBusinessMethod(a) will use a template corresponding to the type of action *a*. For example, List 3 shows a template corresponding to the action *SystemOperation* mapped to the update method. List 4 shows the generated business class of use case Withdrawal.

```
using {{ProjectName}}.Common;
using {{ProjectName}}.Common.Caching;
using {{ProjectName}}.DataAccess
namespace {{ProjectName}}.Business {
    public partial class {{tableName}}Business : I{{
    tableName}}Business {
        private I{{tableName}}Repository _res;
    private ICacheProvider _redis;

        public {{tableName}}Business(I{{tableName}}
    Repository {{tableName}}Res, ICacheProvider
    redis)
    {
            _res = {{tableName}}Res;
      _redis = redis;
        }

    {{BusinessLogic}}
    }
}
```

```
/// <summary>
/// Update information in the table{{tableName}}
/// </summary>
/// <param name="model">the record updated</param>
/// <returns></returns>
 public bool Update({{tableName}} model)
    {
            return _res.Update(model);
        }
```

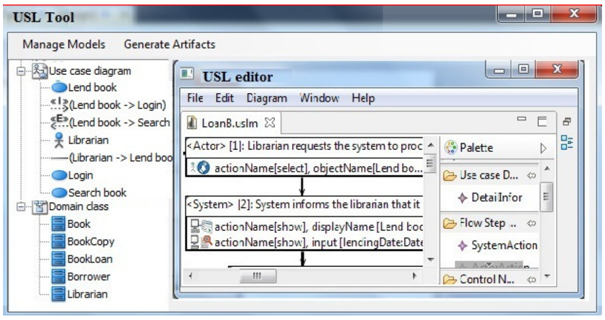
Listing 3. Update method template of business class

```
using System;
using System.Collections.Generic;
using ATM.BusinessObject;
using ATM.DataAccess;

namespace ATM.BusinessLogic
{
    public class WithdrawalBusiness :
    IWithdrawalBusiness
    {
        private IWithdrawalRepository _res;
        public EmployeeBusiness(
    IWithdrawalRepository res)
        {
            _res = res;
        }

        /// <summary>
        /// Validate the record Withdrawal
        /// </summary>
        /// <param name="model">the record validated
     </param>
        /// <returns></returns>
        public bool ValidateFund(Withdrawal model)
        {
            return _res.Validate(model);
        }

        /// <summary>
        /// Create a new record Withdraw
        /// </summary>
        /// <param name="model">The record is
    recorded </param>
        /// <returns></returns>
        public bool GenerateWithdraw(Withdrawal
    model)
        {
            if (model.Withdrawal_id == null || model
    .Withdrawal_id == Guid.Empty.ToString())
            { var c_guid = Guid.NewGuid().ToString()
    ; model.Withdrawal_id = c_guid; }
            return _res.Create(model);
        }

        //...

    }
}
```

Listing 4. The business class is generated for use case Withdraw

V. Tool Support

We have added a code generation feature named USLSCG for the USL tool in the research [3] as depicted in Figure 7. The USL tool allows the integration of use cases into model-driven software engineering (MDSE). Firstly, this tool reads use case diagrams and a UML domain class diagram. Secondly, we specify each use case description as a USL model by the USL Editor tool. Finally, the code generator USLSCG reads the source code template files and transforms the UML domain class diagram into source code classes at the business objects layer, data access layer, and a SQL script file and it also transforms each use case into source code classes in presentation and business layer by using model transformation languages Model to Text (M2T).



<Actor> [1]: Librarian requests the system to proc
actionName[select], objectName[Lend bo...
<System> [2]: System informs the librarian that it
actionName[show], displayName [Lend boc
actionName[show], input [lendingDate:Date

Fig. 7. The support tool USL contains the source code generation feature.

The architecture of the USLSCG generator is shown as in Figure 8. This tool takes as input a USL model and a UML domain class diagram and source code templates. The output of this tool is classes at presentation, business logic, data access, business object layers; a SQL script file to create Database and store procedures. In order to build the USLSCG generator, we implement two modules, Module (1) and Module (2) as shown

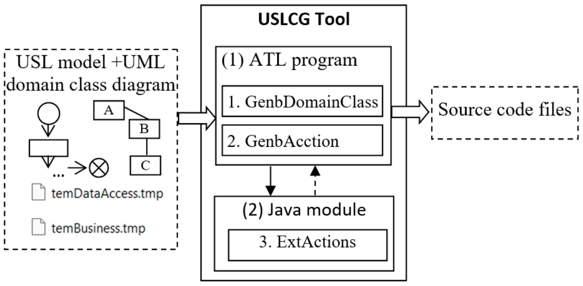


Fig. 8. The architecture of the generator USLSCG.

in Figure. 8. Module (1) is implemented by an Acceleo project. Module (1) aims to parse the USL model in Java and Its output is taken as input for Module (2). Module (2) extracts actions from the USL model and returns a list of actions for Module (1). Finally, Module (1) transforms the inputs into source code files by transformations in Acceleo. Firstly, Module (1) reads the UML domain class diagram and template files as inputs for Module (1) functions. Function 1.GenbDomainClass gets inputs including the UML domain class diagram, a SQL script template file, an object template file, and data access template files to generate a SQL script file, classes in the business objects layer, and classes at the data access layer, respectively. Next step, Module (1) read each USL model to pass Module (2) written in java to extract actions from the USL model and return a list of actions for Module (1). Function 2.GenbAcction reads presentation template files, business template files and actions returned from Module (2) to generate classes at the presentation and business logic layer.

VI. Related work

We position our work in the automatic generation of source code classes from use cases and a UML domain class diagram. Within this context, source code classes are often manually built from design class diagrams of use cases. In order to automatically generate source code classes from a use case model to reduce cost when software requirements are changed, several approaches [10], [11], [12], [13] have been proposed.

Sunitha *el al.* [11] proposed a methodology to automatically code generate from state chart diagrams. This paper presents a method to convert hierarchical states, concurrent and history states to Java code with a design pattern-based approach. Particularly, each state of the system (or object) will be mapped to a source code class. However, in the object-oriented approach, each object type is abstracted into a class, and each state is usually an obtainable value of a class property.

Francu *el al.* [10] presented a method that allows generating an implementation of a system from the use cases written in a natural language. This paper did not show the results of the generator for Entity Managers which are used services for classes of the above layers. Additionally, this paper does not handle one-to-many or many-to-many associations among the classes in the domain model. Besides, extracted verbs do not classify so arguments of the procedures do not be generated precisely.

Fatolahi *el al.* [12] proposed a semi-automated method for the generation of web-based applications from use cases. However, this approach is not fully automatic, they need to interact with the developer to obtain the appropriate value of required user parameters. Compared with all the works above, our approach allows generating classes containing full parameters following three layers architecture. Besides, a script file is also generated which creates the database and its store procedures. This file is not created in all of the above research.

VII. Conclusion

In this work, we proposed an automatic method for generating source code files from a use case specification represented as a USL model, a domain class diagram, and template files. Generated source code files contain classes of the application, a SQL script file to create the database, and store procedures for the application. We also developed the generator USLSCG to realize our method. In the future work, we will improve the generator USLSCG to generate various kinds of applications. Furthermore, generated source code files in different programming languages will also be supported in our next version of the generator USLSCG.

References

[1] I. Sommerville, *Software Engineering*, 10th ed. Boston: Pearson, Mar. 2015.

[2] A. Cockburn and H. a. Technology, “WRITING EFFECTIVE USE CASES,” *Addison-Wesley*, p. 113, 2001.

[3] C. Hue and D.-H. Dang, “USL: A Domain-Specific Language for Precise Specification of Use Cases and Its Transformations,” *Informatica*, vol. 42, Sep. 2018.

[4] C. Hue, D.-H. Dang, and N. Binh, *A Transformation-Based Method for Test Case Automatic Generation from Use Cases*, Nov. 2018, pages: 257.

[5] M.-H. Chu and D.-H. Dang, “Automatic Extraction of Analysis Class Diagrams from Use Cases,” in *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*, Nov. 2020, pp. 109–114, iSSN: 2164-2508.

[6] K. Wiegers and J. Beatty, *Software Requirements, 3rd Edition*, ser. 3rd Edition, Aug. 2013.

[7] G. Ksters, H.-w. Six, and M. Winter, “Validation and verification of use cases and class models,” 04 2001.

[8] Laurent Goubet and Laurent Delaigue, “Acceleo/Getting Started - Eclipsepedia.” [Online]. Available: https:// wiki.eclipse.org/Acceleo

[9] a. M. W. Marco Brambilla, Jordi Cabot, *Model-Driven Software Engineering in Practice*, 2nd ed. Morgan & Claypool, 2017.

[10] J. Francu and P. Hnetynka, “Automated Code Generation from System Requirements in Natural Language,” *e-Informatica*, vol. 3, pp. 72–88, Jan. 2009.

[11] S. E. V. and P. Samuel, “Automatic Code Generation From UML State Chart Diagrams,” *IEEE Access*, vol. 7, pp. 8591 – 8608, Jan. 2019.

[12] A. Fatolahi, S. S. Somé, and T. C. Lethbridge, “Towards A Semi-Automated Model-Driven Method for the Generation of Web-based Applications from Use Cases,” 2008.

[13] H. Ikeda, H. Nakagawa, and T. Tsuchiya, “Towards Automatic Facility Layout Design Using Reinforcement Learning,” Sep. 2022, pp. 11–20. [Online]. Available: https://annals-csis.org/proceedings/2022/drp/25.html

DOI: 10.15439/2022R35
ISSN 2300-5963 ACSIS, Vol. 33

Chronic kidney disease diagnosis using Fuzzy Knowledge Graph Pairs-based inference in the extreme case

Pham Minh Chuan
Faculty of Information Technology
Hung Yen University of technology
and education
HungYen, Vietnam
chuanpm@utehy.edu.vn

Luong Thi Hong Lan,
Tran Manh Tuan
Faculty of Computer Science and
Engineering
Thuyloi University
Hanoi, Vietnam
lanlh@tlu.edu.vn, tmtuan@tlu.edu.vn

Nguyen Hong Tan
Faculty of Information
Technology
Information Technology and
Communication University
Thainguyen, Vietnam
nhtan@ictu.edu.vn

Cu Kim Long
School of Information
Communication Technology,
Hanoi University of Science and
Technology (HUST), Hanoi, Vietnam;
Information Technology Center,
Ministry of Science and Technology
(MOST), Hanoi, Vietnam
longck.2006@gmail.com

Pham Van Hai
School of Information Communication
Technology
Hanoi University of Science and
Technology
Hanoi, Vietnam
haipv@soict.hust.edu.vn

Le Hoang Son
VNU Information Technology
Institute
Vietnam National University
Hanoi, Vietnam
sonlh@vnu.edu.vn

***Abstract*—Chronic kidney disease is one of the diseases with high morbidity and mortality, commonly occurring in the general adult population, especially in people with diabetes and hypertension. Scientists have researched and developed intelligent medical systems to diagnose chronic kidney disease. Nevertheless, healthcare services remain low in resource-limited areas, and general practitioners are very short of clinical experience. Identifying chronic kidney disease in clinical practice remains challenging, especially for the general practitioner. This study proposes a model to develop a model for improving the efficiency of differential diagnosis. This paper presents a model consisting of a fuzzy knowledge graph pairs-based inference mechanism by accumulating the new rules to enrich the fuzzy rule base. A real-world dataset is gathered in Dien Bien hospital to evaluate the performance of our proposed model.**

***Index Terms*—fuzzy knowledge graph, fuzzy inference system, chronic kidney disease, health sector.**

I. Introduction

Chronic kidney disease (CKD) has arisen as one of the main reasons leading to death recently. According to the World Health Organization, the number of patients infected with CKD is rising, impacting an estimated 800 million individuals [1]. An urgent question for medical professionals is how CKD can be detected early to reduce the burden on doctors and medical facilities. In this case, an intelligent health system aimed at the early detection of CKD is considered a valuable and appropriate solution. The high number of affected individuals and the significant adverse impact of CKD should motivate enhanced endeavors of many researchers for better prevention and treatment.

As CKD gradually develops, early recognition and successful dealing are simply treatments to decrease the death rate. It is required to address CKD problems through work concentrating on predicting diseases. If we diagnose early CKD, we have accomplished early remediation for CKD patients.

In the past decade, scientists and developers have researched and published many intelligent models or methods to predict CKD using various techniques. Ma et al. [2] proposed a deep-learning algorithm for predicting CKD at an early stage. In [3], the authors list articles on the early prediction of CKD using AI approaches. In another study, Mehdi Hosseinzadeh et al. [4] proposed a model for predicting CKD using machine learning in an intelligent environment in smart cities with the help of cloud computing technology.

On the other hand, as the electronic healthcare dataset overgrows, fuzzy techniques and fuzzy inference systems (FIS) are becoming more common for accurate and early diagnosis of common diseases in patients. For instance, some of the new proposed intelligent models can be introduced as Fuzzy inference system [5,6,7], Adaptive neuro-fuzzy inference system [8,9], Knowledge graph [10], Fuzzy knowledge graph (FKG) [11,14,21], Mamdani Complex fuzzy inference system (M-CFIS) [12,13], and so on. However, existing intelligent techniques used in these new methods have limitations when applied in decision-making support systems with limited input data.

Recently, Lan et al. [11] introduced an M-CFIS-FKG model that combines M-CFIS and FKG to improve the reference speed and experimental time in testing data of the M-CFIS. Although M-CFIS-FKG has been enhanced in calculation and inference time, its low accuracy still limits the model. Then, Long et al. [14] proposed a new model (the FKG-Pairs), to overcome this shortcoming. The FKG-Pairs model has improved the precision of M-CFIS-FKG; However, the model still has several limitations in the case of a shortage of knowledge or too-small fuzzy rule base.

This research aims to develop a novel model that extends reference-based fuzzy knowledge graphs by accumulating the new rules to enrich the fuzzy rule base to predict CKD. Even though some related works are available in the literature, this emphasizes the uses in real scenarios where the dataset has been collected in the Dien Bien hospital in Vietnam. In addition, the proposed model solves existing limitations in FKG-Pairs.

Significantly, the contributions and novelty of this report are highlighted as follows:

- Giving the proposed model that executes reference based on FKG-Pairs in an extreme case in which too-small fuzzy rule base (so called FKG-Extreme model).

- Studying to apply the proposed model to the real-world dataset collected from Dien Bien hospital in Vietnam.

- Illustrating our proposed model's effectiveness and potential for real-world application by comparing experience between our proposed method and the state-of-the-art method (namely FKG-Pairs).

The rest of this paper is organized as follows. The background preliminaries are presented in Section II. Section III describes our proposed model for predicting CKD. Section IV shows our experimental results and comparion with the state-of-the-art method. Conclusions and future works are given in Section V.

II. Preliminaries

This section briefly presents the background preliminaries used in this work.

A. Fuzzy Inference System

The fuzzy inference system (FIS) is a principal component of building a fuzzy logic system. It utilizes the fuzzy set theory and fuzzy rules to display the qualitative factor of humans about knowledge and thinking. Then it uses fuzzy reasoning processes to find the output regarding crisp inputs.

The general FIS framework has four main parts: a set of fuzzy rules, a fuzzy inference engine, a fuzzification database, and a defuzzification. Fuzzification supports applying multiple fuzzification methods and transforms the crispy input into fuzzy input. This process involves the fuzzy membership function and maps the factual information into a fuzzy value (i.e., the value between 0 and 1). The fuzzy inference engine contains fuzzy rules that form IF-THEN. It executes inference operations on the rules captured in the fuzzy rule base. The inference engine applies fuzzy rules from a knowledge base and produces the fuzzy output between 0 and 1.

Defuzzification is the inverse fuzzification process. It transforms the fuzzy results into crisp output. This process can use many methods, such as Max-membership, Centroid Method, Mean-Max membership, and so on.

B. Fuzzy Knowledge Graph Pairs

The Fuzzy Knowledge Graph Pairs (FKG-Pairs) [14] is introduced in 2022. It was a graph in which nodes represent the attribute's and output's labels, and edges represent the re-

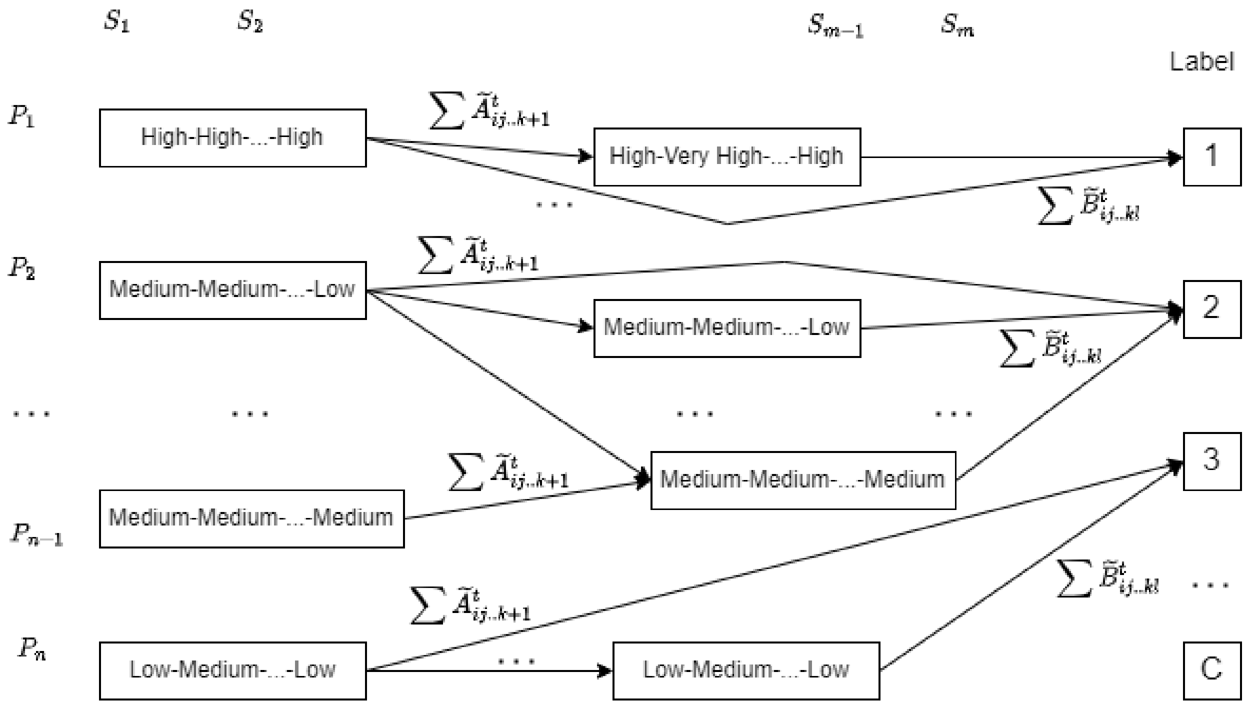


Fig. 1 A representation of the FKG-Pairs.

lation among nodes. Fig. 1 shows a simple representation of the FKG-Pairs.

Given the fuzzy rule base, including n rules, m attributes, and C output labels. The weights of edges connected among the attribute's labels $\left(\widetilde{A}^{t}_{ij\ldots k}\right)$, the weights of edges connected among the attribute's labels and the output's labels $\left(\widetilde{B}^{t}_{ij\ldots k}\right)$ are calculated by using Equations (1) and (2), respectively, as follows [14]:

$$\widetilde{A}^{t}_{ij\ldots k} = \frac{\left|\widetilde{X}_i \rightarrow \widetilde{X}_j \rightarrow \cdots \rightarrow \widetilde{X}_{k+1} \in rule\, t\right|}{|R|} \tag{1}$$

$$where\; t=\overline{1,n}; 1 \leq i < j < \cdots < k \leq m-1.$$

$$\tilde{B}^{t}_{ij\ldots kl} = \left(\sum \tilde{A}^{t}_{ij\ldots k+1}\right) * MIN\left(\frac{|\tilde{X}_i \rightarrow l\ in\ rule\ t|}{|R|}, \frac{|\tilde{X}_j \rightarrow l\ in\ rule\ t|}{|R|}, \ldots, \frac{|\tilde{X}_k \rightarrow l\ in\ rule\ t|}{|R|}\right), \tag{2}$$

$$where\; t=\overline{1,n}; 1 \leq i < j < \cdots < k \leq m-1; l=\overline{1,C}.$$

III. Proposed Model

To enhance the inference capability in the extreme cases in which the uncompleted-information input dataset or the too-small fuzzy rule base, we present a novel model (so called FKG-Extreme model) that combines fuzzy knowledge graph and FIS to predict CKD. The detail of our proposed model is presented in Fig. 2.

The proposed model consists of the main steps as follows:

Step 1: Collecting the dataset of patients with CKD or Non-CKD. The expertise of a specialist doctor has checked this dataset.

Step 2. After obtaining the dataset, the dataset will be preprocessed, such as normalizing the dataset and removing the noise and blank data.

Step 3: Some parameters of our model is initialized, such as the current rule base $R_{Cur} = \varnothing$, time step $t=\overline{1,T}_{max}$; $\theta = \theta_0 \%$.

Step 4: Splitting the dataset *D* into two training and testing sets, in which the training set to train the model and the testing set is used to check the model's effectiveness. The training data is selected as a percentage of the original dataset, $D_{Train} = \theta \% * D$. The rest is used for testing data D_{Test}.

Step 5: Applying the rule-generated mechanism on FIS to the training and testing data that obtain a set of rules $\left(R_{Train}\right)$, and (R_{Test}) respectively.

Step 6: Updating the $R_{Cur} = R_{Cur} + R_{Train}$.

Step 7: Constructing the FKG-Pairs based on the current rule base $\left(R_{Cur}\right)$ after updating.

Step 8: Repeating steps 1 to 6 until $t \geq T_{max}$.

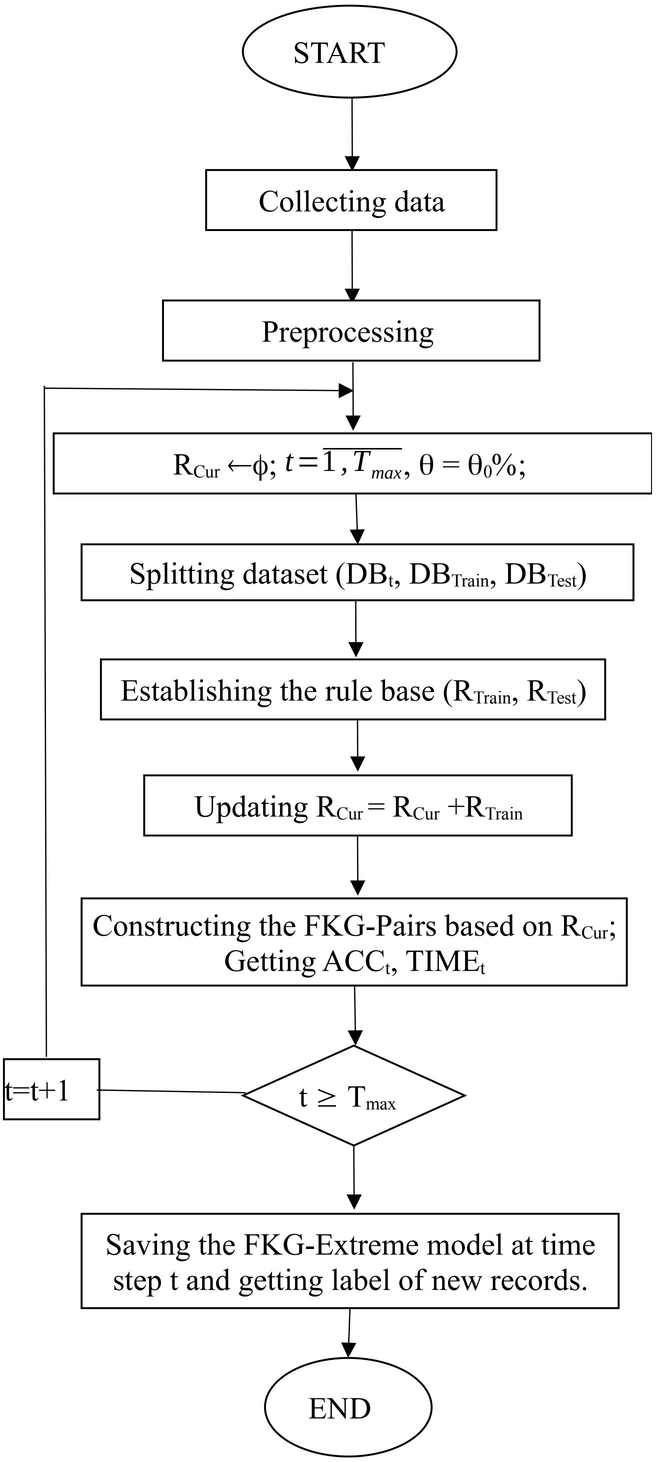

START
Collecting data
Preprocessing
R$_{Cur}$ ←ϕ; $t=\overline{1,T}_{max}$, θ = θ$_0$%;
Splitting dataset (DB$_t$, DB$_{Train}$, DB$_{Test}$)
Establishing the rule base (R$_{Train}$, R$_{Test}$)
Updating R$_{Cur}$ = R$_{Cur}$ +R$_{Train}$
Constructing the FKG-Pairs based on R$_{Cur}$; Getting ACC$_t$, TIME$_t$
t=t+1
t ≥ T$_{max}$
Saving the FKG-Extreme model at time step t and getting label of new records.
END

Fig. 2. Proposed FKG-Extreme model for CKD.

Step 9: Saving the FKG-Extreme model at the time step and getting the label of new records.

IV. The experimental results

With the purpose of illustrating the effectiveness of the proposed model, experiments were run to assess CKD. The group of physicians evaluated every patient and gave a result. The expert opinion was then compared with the results obtained from our proposed model.

A. Experimental dataset and environment

The dataset was collected from DienBien hospital with fifteen clinical and subclinical attributes (such as age, gender, RBC, HGB, HCT, PLT, and so on) given in Table I. It included 3652 patients (in which 2063 patients in non-CKD and 1589 patients in CKD).

B. The experimental results

This section aims to estimate the implementation of FKG-Extreme in CKD diagnosis. We have executed using Dell PC with a Core i5 processor for executing in MATLAB 2014. To demonstrate the effectiveness of the proposed model, the proposed method (FKG-Extreme) is compared with the state-of-the-art method (FKG-Pairs) on the CKD dataset.

The evaluation measures are time-consuming and accuracy to assess these models' implementation. Accuracy is a metric that describes how the model accomplishes across all classes. It is beneficial when all types are of equal importance. It is computed as the ratio of correct forecasts to the total number of predictions (see Equation (3)).

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (3)$$

TABLE I. LIST OF ATTRIBUTES IN CKD DATASET.

No.	Attribute name	Valued range
1.	Age	1 - 100 years
2.	Gender	0: Female; 1: Male
3.	Red Blood Cell (RBC)	1.33 - 138 T/L
4.	Hemoglobin (HGB)	3.87 - 186 g/L
5.	Hematocrit (HCT)	11.3 - 126 % (0.35 - 21.41 L/L)
6.	Platelet Count (PLT)	13 - 2141 G/L
7.	White Blood Cell (WBC)	1.11 - 215.58 G/L
8.	Neutrophil (NEUT)	0.01 - 98 %N
9.	Lymphocyte (LYMPH)	0.5 - 98.1 %L
10.	Sodium (Na$^+$)	5.3 - 172.1 mmol/L
11.	Potassium (K$^+$)	2.24 - 103.2 mmol/L
12.	Total Protein	29 - 565.31 g/L
13.	Albumin	-1.0 - 141.7 g/L
14.	Ure	0.5 - 107.42 mmol/L
15.	Creatinin	4.26 - 8632 µmol/L
16.	**ICD code (Output labels)**	**0: Non-CKD** **1: CKD**

The experimental scenario is described as follows: With the proposed method, the data set is divided into 20 equal parts (corresponding to 20-time steps), and each data part is divided the training and testing sets by 5% and 95%, respectively. To assume responsibility for the flexibility of the proposed method, we execute the test ten times and compute the average for each time step. With the FKG-Pairs, we only perform 5% for training data and 95% for testing data.

The results of applying two models for the CKD dataset are visually presented in Figures 3 and 5, respectively. Figure 3 compares the accuracy between FKG-Pairs method and FKG-Extreme method. The results show that the prediction accuracy is significantly improved when enhancing the rule sets in the previous steps for the latter, especially when the data used for the training data set is minimal compared to the testing data set.

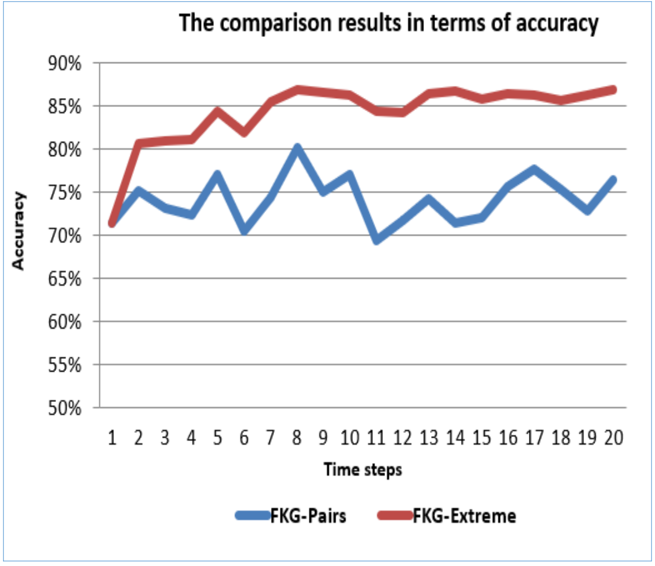


Fig. 2. Comparison results of accuracy between FKG-Pairs and FKG-Extreme in each time step.

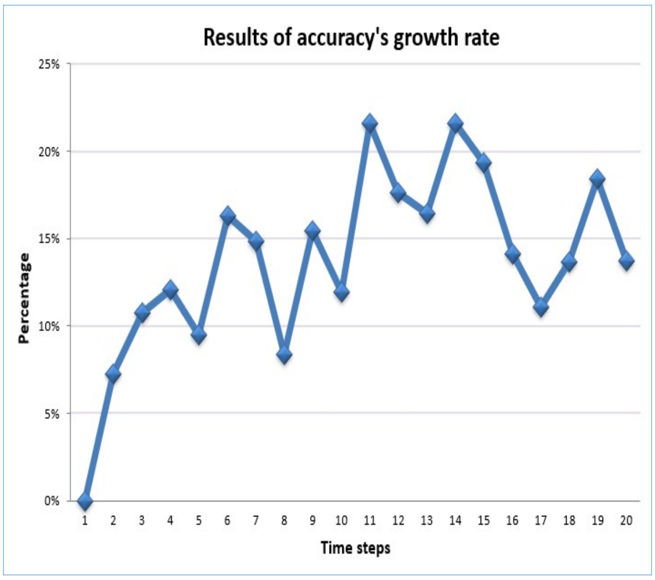


Fig. 3. Accuracy's growth rate of FKG-Extreme vs. FKG-Pairs.

With the aim to illustrate the effectiveness of applying the FKG-Extreme model, we compute the growth ratio in the accuracy of the FKG-Extreme compared to FKG-Pairs due to each time step. The details of the results are represented in Figure 4. As shown in Figure 4, it is clear that the growth percentage ratio of accuracy is from over 5% to approximately 25%. And most of the growth rate due to each step increase from 10% to up. It proves that the proposed method has dramatically improved the model's accuracy in extreme case in which the training data is very small (only 5% rate).

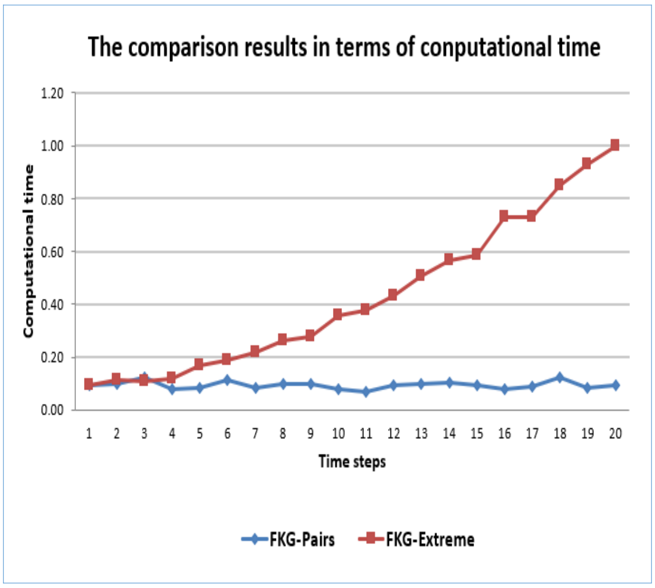


Fig. 4. Comparison results of time consumed between FKG-Pairs and FKG-Extreme in each time steps.

The estimation time on the CKD dataset obtained by the two models is demonstrated in Figure 5. For the proposed method, the time-consuming in most of the time steps is always higher than that of the FKG-Pairs method. This is evident results because of using the accumulative mechanism for the rule base.

TABLE II. Comparison Average Of Accuracy And Execution Time Between Fkg-Pairs And Fkg-Extreme.

Method	**Accuracy**	**Time (s)**
FKG-Pairs	74.13%	**1.8766**
FKG-Extreme	**84.23%**	8.6078

Table II describes the average value of 20-time steps in accuracy and computational time indices of two methods that test on the real-world CKD dataset. Results are handled as the average of k-fold cross-validation (with k=10). It is clear that the average accuracy for the FKG-Extreme reached 84.22% compared to 74.13% for the FKG-Pairs. The accuracy rate for each step is about 12% (see Fig. 4). However, the time-consuming of proposed method is much higher than that of FKG-Pairs method. These results demonstrate that applying the accumulative mechanism for the rule base on FKG can improve the power of the FKG-Pairs-based inference systems in terms of accuracy in extreme case.

V. Conclusions and Future Works

CKD is one of the most difficult mortal diagnoses to predict with great accuracy and precision. Developing an application for the diagnosis of CKD will contribute pramatically for medical experts in treating critical situations and people who have poor health sevices. This study aims to see how well reason based on FKG-Pairs, in extreme cases, in which too-small training data or fuzzy rule base. Then, the model performs on a real-world dataset to diagnose CKD. We combined the fuzzy knowledge graph and FIS to simplify the uncertainties observed in the dataset. Primarily our proposed model works effectively in extreme cases of lack of knowledge, uncompleted-information input dataset, or the too-small fuzzy rule base. Finally, a dataset is gathered in Dien Bien hospital to evaluate the performance of our model in improving the model's accuracy in extreme cases. This has practical significance to applying the FKG-Pairs in the case of a lack of knowledge of training data or new systems.

Some future works are identified for reseachers or deveplopers community in the near future to improve the proposed model such as: (i) Improving the proposed method for extreme cases by applying the Q-learning technique to recommend the best action (in which sampling and splitting methods are considered very important, especially in the health sector); (ii) Studying to develop easy and convenient applications based on clinical symptoms and subclinical testing signs datasets in the real world.

Acknowledgment

This research is funded by Ministry of Education and Training under the grant number B2022-SKH-01.

References

[1] Kovesdy, C. P. (2022). Epidemiology of chronic kidney disease: an update 2022. Kidney International Supplements, 12(1), 7-11.

[2] Ma, F., Sun, T., Liu, L., & Jing, H. (2020). Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network. Future Generation Computer Systems, 111, 17-26.

[3] Khade, A. A., Vidhate, A. V., & Vidhate, D. (2021, October). A comparative analysis of applied AI techniques for an early prediction of chronic kidney disease. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 1386-1392). IEEE.

[4] Hosseinzadeh, M., Koohpayehzadeh, J., Bali, A. O., Asghari, P., Souri, A., Mazaherinezhad, A., ... & Rawassizadeh, R. (2021). A diagnostic prediction model for chronic kidney disease in internet of things platform. Multimedia Tools and Applications, 80(11), 16933-16950.

[5] Sharma, P. K., Sachdeva, A., & Bhargava, C. (2021). Fuzzy logic: A tool to predict the Renal diseases. Age, 1, 100.

[6] Lin, H. C., Hung, P. H., Hsieh, Y. Y., Lai, T. J., Hsu, H. T., Chung, M. C., & Chung, C. J. (2022). Long-term exposure to air pollutants and increased risk of chronic kidney disease in a community-based population using a fuzzy logic inference model. Clinical Kidney Journal.

[7] Ahmed, T. I., Bhola, J., Shabaz, M., Singla, J., Rakhra, M., More, S., & Samori, I. A. (2022). Fuzzy logic-based systems for the diagnosis of chronic kidney disease. BioMed Research International, 2022.

[8] Damodara, K., & Thakur, A. (2021, March). Adaptive neuro fuzzy inference system based prediction of chronic kidney disease. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 973-976). IEEE.

[9] Abiyev, R. H., Idoko, J. B., & Dara, R. (2021, August). Fuzzy Neural Networks for Detection Kidney Diseases. In International Conference on Intelligent and Fuzzy Systems (pp. 273-280). Springer, Cham.

[10] Long, C. K., Trung, H. Q., Thang, T. N., Dong, N. T., & Van Hai, P. (2021). A knowledge graph approach for the detection of digital human profiles in big data. Journal of Science and Technology: Issue on Information and Communications Technology, 19(6.2), 6-15.

[11] Lan, L. T. H., Tuan, T. M., Ngan, T. T., Giang, N. L., Ngoc, V. T. N., & Van Hai, P. (2020). A new complex fuzzy inference system with fuzzy knowledge graph and extensions in decision making. Ieee Access, 8, 164899-164921.

[12] Selvachandran, G., Quek, S. G., Lan, L. T. H., Giang, N. L., Ding, W., Abdel-Basset, M., & De Albuquerque, V. H. C. (2019). A new design of Mamdani complex fuzzy inference system for multi-attribute decision-making problems. IEEE Transactions on Fuzzy Systems, 29(4), 716-730.

[13] Tuan, T. M., Lan, L. T. H., Chou, S. Y., Ngan, T. T., Son, L. H., Giang, N. L., & Ali, M. (2020). M-CFIS-R: Mamdani complex fuzzy inference system with rule reduction using complex fuzzy measures in granular computing. Mathematics, 8(5), 707.

[14] Long, C. K., Van Hai, P., Tuan, T. M., Lan, L. T. H., Chuan, P. M., & Son, L. H. (2022). A novel fuzzy knowledge graph pairs approach in decision making. Multimedia Tools and Applications, 1-30.

[15] Long Cu Kim and Hai Pham Van (2018), "Intelligent Collaborative Decision Model for Simulation of Disaster Data in Cities and Urbanlization", International Journal of Advanced Research (IJAR), Vol. 6, Issue 07.

[16] C. K. Long et al., (2020), "A Big Data Framework for eGovernment in Industry 4.0", Open Computer Science, ISSN: 2299-1093.

[17] PHAM, Hai Van; TIEN, Dong Nguyen. Hybrid Louvain-Clustering Model Using Knowledge Graph for Improvement of Clustering User's Behavior on Social Networks. In: The International Conference on Intelligent Systems & Networks. Springer, Singa-pore, 2021. p. 126-133.

[18] DINH, Xuan Truong; PHAM, Hai Van. Social Network Analysis Based on Combining Probabilistic Models with Graph Deep Learning. In: Communication and Intelligent Systems. Springer, Singapore, 2021. p. 975-986.

[19] Pham, H.V.; Thanh, D.H.; Moore, P. Hierarchical Pooling in Graph Neural Networks to Enhance Classification Performance in Large Datasets. Sensors 2021, 21, 6070. https://doi.org/10.3390/s21186070.

[20] Hai Van Pham, Long Kim Cu, (2020), "Intelligent Rule-based Support Model Using Log Files in Big Data for Optimized Service Call Center Schedule", Proceedings of International Conference on Research in Intelligent Computing in Engineering, ISBN 978-981-15-2780-7.

[21] C.K.Long et al. (2021), "Disease Diagnosis in the Traditional Medicine: A Novel Approach based on FKG-Pairs", Journal of Research and Development on Information and Communication Technology, Vol. 2021(2), pp. 59-68.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 89–94
DOI: 10.15439/2022R03
ISSN 2300-5963 ACSIS, Vol. 33

Analysis and Prediction for Air Quality Using Various Machine Learning Models

To-Hieu Dao[1,2]
[1]*EEE, Phenikaa University*
[2]*GU of Science & Technology*
Ha Noi, Viet Nam
hieu.daoto@phenikaa-uni.edu.vn

Hoang Van Nhat
Electrical & Electronic Engineering
Phenikaa University
Ha Noi, Viet Nam
van.nhathoang@aalto.fi

Hoang Quang Trung
Electrical & Electronic Engineering
Phenikaa University
Ha Noi, Viet Nam
trung.hoangquang@phenikaa-uni.edu.vn

Vu Hoang Dieu[1,2]
[1]*EEE, Phenikaa University*
[2]*GU of Science & Technology*
Ha Noi, Viet Nam
dieu.vuhoang@phenikaa-uni.edu.vn

Nguyen Thi Thu
Electrical & Electronic Engineering
Ha Noi University of Industry
Ha Noi, Viet Nam
thunt@haiu.edu.vn

Duc-Nghia Tran
Institute of Information Technology
Viet Nam Academy of Science & Technology
Ha Noi, Viet Nam
nghiatd@ioit.ac.vn

Duc-Tan Tran*
Electrical & Electronic Engineering
Phenikaa University
Ha Noi, Viet Nam
tan.tranduc@phenikaa-uni.edu.vn

***Abstract*—Air pollution has been a concern in recent years. Measuring the extent of pollution is important to know about the air quality. Previous research has used machine learning algorithms to forecast the Air Quality Index (AQI) in specific locations. Even though that research achieved quite reliable results, they still have some drawbacks that need to be taken into consideration, such as low accuracy or lack of data analysis.On a public dataset, we used Random Forest, XGBoost, and Neural Network to build a machine learning model for the purpose of making predictions about the air quality index (AQI) in a number of cities located in India. The performances of these models were evaluated by using their score errors, Root Mean Square Error (RMSE), and Coefficient Of Determination (R^2). This paper demonstrates the analysis of air pollutants from the dataset, which is an effective way to enhance the model's performance.**

***Index Terms*—Machine Learning, AQI, Data Analysis**

I. Introduction

Recent economic and social developments have had an effect on various environmental variables, including the land, water resources, and air. Because of this, wireless sensor network-based air quality monitoring is a popular research topic [1], [2]. According to WHO [3], seven million deaths were related to air pollution each year.

Based on the computation of pollutants that are harmful to human health, the air pollution level index can be created [4]. The Air Quality Index (AQI) is the name of this index, which ranges from 0 to 500. A high AQI is not good for people. There are distinct ways to calculate the AQI, such as using the formula or using machine learning techniques. In 2018, study led by Samir Lemes and colleagues demonstrated the disparity between several approaches to estimate the AQI by calculating and ranking AQI values according to certain criteria [5]. They then used these parameters to calculate the levels of air pollution in two different parts of Bosnia and Herzegovina. The final result of their work illustrates the comparison of AQI values on the same dataset, which was obtained by using different methods of US AQI, EU AQI, and SAQI_11 standards.

Machine learning, while on the other hand, has demonstrated its superior effectiveness by combining knowledge from various fields, such as statistics, artificial intelligence, and computer science [6]–[8]. In recent years, the use of machine learning to predict AQI values has become common and has piqued the interest of researchers [9]–[11]. Although the models constructed in the experiments performed well, they still have some limitations, such as filling in missing values, feature significance analysis, and feature creation to fully exploit the dataset. The final results were then evaluated using three machine algorithms with different validation criteria.

II. AQI Calculation And Dataset

A. AQI Calculation

In order to get the AQI value, there are several different methods to calculate it worldwide. For example, the AQI formula for China, which is based on the National Ambient

Corresponding(*): Duc-Tan Tran. Email: tan.tranduc@phenikaa-uni.edu.vn
Address: Phenikaa University, Ha Noi, Viet Nam

Air Quality Standard of China (NAAQS-1996), differs from the AQI calculation method defined by the US Environmental Protection Agency (1994) and from the method developed by India (NAAQS Dependent Air Quality Index) [4]. Therefore, in this work, we used the estimating formula for China as the reference in comparison with using the Machine Learning approach.

Based on the method proposed by NAAQS-2012, the components used in the formula include 6 pollutants ($PM10, PM2.5, SO_2, Ozone, NO_2$, and CO) and 7 indexes, including the maximum 8-hour Ozone concentration (mg/m^3), the maximum 1-hour Ozone concentration, and the daily average concentration of $SO_2, NO_2, CO, PM10$, and $PM2.5$. The calculation by Eqa. (1) for each individual pollutant is:

$$AQI = \frac{AQI_h - AQI_l}{BP_h - BP_l} \times (C_Q - BP_l) + AQI_l \qquad (1)$$

Where, C_Q is the pollutant Q's daily mean value;The pollution levels for substance Q are, respectively, AQI_h and AQI_l, with the corresponding estimated highs and lows being BP_h and BP_l. The final AQI value is the largest value in the AQI series by Eqa. (2), obtained after completing each AQI math operation:

$$AQI = \max(AQI_0, AQI_1, ...AQI_n) \qquad (2)$$

Where n is the number of pollutants considered. In this experiment, we decided to use the Machine Learning approach to predict the AQI value because the first method is quite time consuming, and complicated. Most importantly, the dataset is not always available to be calculated by the formula, which requires information on pollutant concentrations both daily and hourly.

B. Dataset

This research employed a publicly accessible dataset containing 29531 instances of Indian air quality. This data set was collected over a six-year period (January 2015 to June 2020), allowing us to evaluate proposed air quality calculation methods. Each instance has had the average daily AQI and some other pollutants from different stations in cities across India. The Central Pollution Control Board [12], the official website of the Government of India, provides the dataset. There are 12 features that have been recorded, including some significant air pollution contaminants like particulate matter ($PM2.5$ and $PM10$), ozone (O_3), nitrogen oxides (NO), NO_x, nitrous dioxide (NO_2), sulfur dioxide (SO_2), carbon monoxide (CO) emissions, amoniac NH_3 and other chemical occurrences (benzen, toluene, xylene). However, there are some important features that contribute mostly to the value of the AQI, which are particulate matter ($PM2.5$ and $PM10$), CO, NO_2, and SO_2. On top of that, NO_x, which is associated with acid rain, photochemical smog, and tropospheric ozone destruction, is another indicator for AQI prediction [13].

In the dataset, time plots are significant for some analysis related to changes in AQI over months and years, which helps us choose an effective method to predict the AQI value along with time series. We did some analysis regarding the changes in all data features and the AQI value according to year in Fig. 1. The total value is the sum of all pollutants recorded in all cities at different times throughout the given period. It is evident to note that there is an upward trend when it comes to the pollutants and AQI values throughout the 6-year period. The last 3 years from 2018 to 2020 witnessed the highest figures of these pollutants. According to this, 2019 and 2020 are the most polluted years recorded, in which the AQI value and particulate matter peaked in October, November, and December.

According to [4], there are some main pollutants that lead to high degrees of air pollution. Thus, we used the total value recorded in five main indexes, including AQI, $PM10$, $PM2.5$, CO, and NO_2 to rank the most polluted cities. The visualization is displayed in Fig. 2.

Among the five most polluted cities above, they all recorded high levels of five pollutants, which are AQI, $PM10$, $PM2.5$, CO, and NO_2. On average, Ahmedabad is the most polluted city when it comes to the AQI value, at almost 450 on average. Second in terms of pollution are Delhi, Patna, Gurugram, Lucknow, and so on, which had a high degree of pollutants including AQI, $PM10$, $PM2.5$, and NO_2. However, the CO value was record-high in only Ahmedabad, with more than 20, whereas in other cities this substance only ranged from 0 to approximately 2.

III. Data Preprocessing And Method

The data preprocessing is the first and most important step, which not only results in a good validation result but also improves the predictive performance of the model later on. This stage often includes missing data imputation, removing strange datapoints, feature engineering techniques, and feature selection. The two first steps help us have a full set of data, improving the accuracy of the models. Meanwhile, selecting useful features can reduce running time, minimize overfitting while running the model.

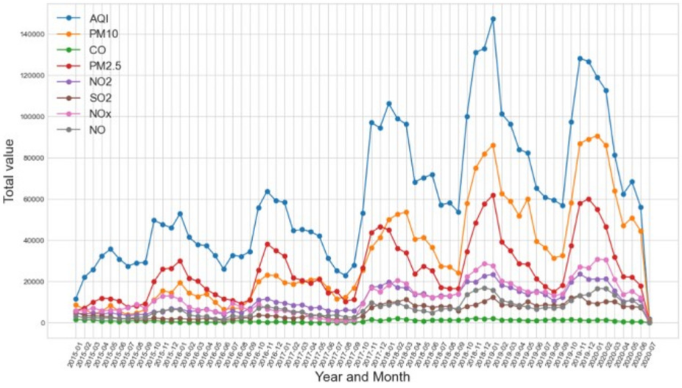


Fig 1. Air pollutants depicted by time.

TAB I
STATISTICS OF NULL VALUES

	Pollutants	Value	Percentage (%)
1	Xylene	18109	61.3
2	$PM10$	11140	37.7
3	NH_3	10328	35.0
4	Toluene	8041	27.2
5	Benzene	5623	19.0
6	AQI	4681	15.9
7	$PM2.5$	4598	15.6
8	NO_x	4185	14.2
9	O_3	4022	13.6
10	SO_2	3854	13.1
11	NO_2	3585	12.1
12	NO	3582	12.1
13	CO	2059	7.0

A. Missing Data Imputation

Tab. I shows the missing value percentage for each column in the dataset. From the given result, the missing data proportion is distributed mostly in xylene, $PM10$, NH_3, and toluene, which are 61.3%, 37.7%, 35.0%, and 27.2%, respectively. Data loss rates in other situations range from 12% to 19%. This issue can be resolved in a number of ways, including by eliminating dropped data points or by adding the most common value from each case to the missing data. In this study, the K-Nearest Neighbors Imputer (KNNImputer) technique is used to overcome losing information [14]. At this stage, each sample's missing values are imputed by the mean value, calculated from 3-neighbors nearest data points in the dataset. This technique was used because it is easy to use and works well. It is also more accurate than simple imputation.

B. Feature Engineering

New features that are created based on the features in the dataset can be very helpful to improve the performance of the model. In this step, we used the mathematical transform, which groups some existing features into a new one that has a good association with the target. In the dataset, we came up with 3 new features by using this method. The first one is "$ParticulateMatters_i$", which was made by adding the value of $PM10$ to $PM2.5$ together. The second new feature is "$Nito_i$", obtained by the sum of NO_2, NO_x, and NO. Finally, the attribute $average\_N_i$ is the average value of N previous AQI data points. In addition, year and month are two time features extracted from data information. Three new numerical features are depicted by Eqa. (3)-(5) below:

$$ParticulateMatters_i = PM10_i + PM2.5_i \tag{3}$$

$$Nito_i = (NO_2)_i + NO_i + (NO_x)_i \tag{4}$$

$$average\_N_i = \frac{\sum\limits_{k=i-N}^{i-1} AQI_k}{N} \tag{5}$$

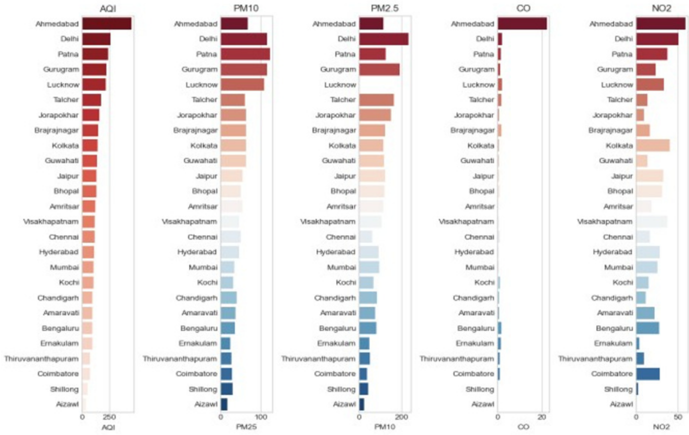


Fig 2. Most polluted cities towards five pollutants.

C. Method

Our method to estimate the AQI value used nine main features, which were reached in the previous stages. We implemented steps in order to get the AQI prediction. Firstly, after being preprocessed as well as experienced data selection and data engineering, selected features were divided into 2 subsets called Training set and Test set. In which Training set accounted for 80% of the total dataset while Test set held the remaining volume, at 20%. The aim of the division is to validate the model's performance later on. In this work, we used three machine learning algorithms, namely Random Forest Regression, Gradient Boosted Regression, and Neural Network Regression, to train three models on the training set. Then, the trained models could be applied to the test set to deal with the unknown data, and get the target prediction. Finally, we used some criteria to assess its effectiveness with regard to model prediction. Fig 3 shows the steps we undertook in our study:

1) Random Forest Regression (RFR): This algorithm is a synthetic prediction algorithm that integrates many different models to create more efficient models. Random Forest (RF) consists of many decision trees, each of which predicts a certain object well and is different from the others [15]. By averaging the results, we were able to significantly reduce the number of overfitting while maintaining the model's good predictive score. The steps are as follows:

Step 1. From the initial dataset, we need to build several subsets of data. The technique used to do this task is called the bootstrapping method, in which, from $n\_samples$ data points, we repeatedly choose random data points with replacement. The result is n datasets called bootstrap samples, which have the same size as the original dataset but in which some data points will be absent or some will be repeated [16]. This method guarantees that each bootstrap sample is modestly different from the others.

Step 2. For each new dataset, build a decision tree with a slight modification: instead of choosing the best test for a specified node, in each node we randomly choose k features

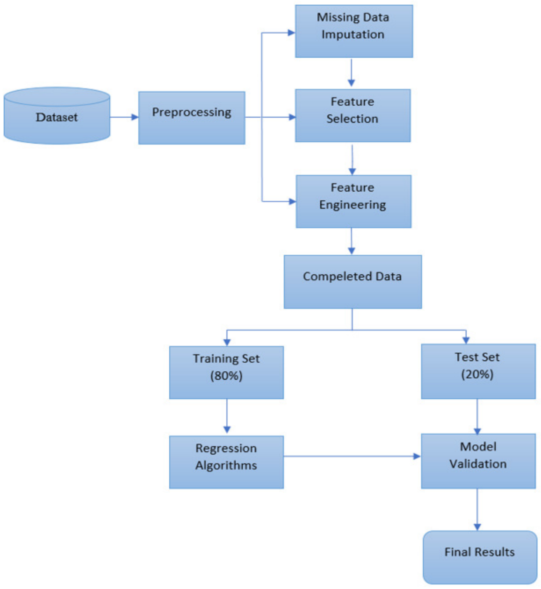


Fig 3. Machine Learning prediction steps.

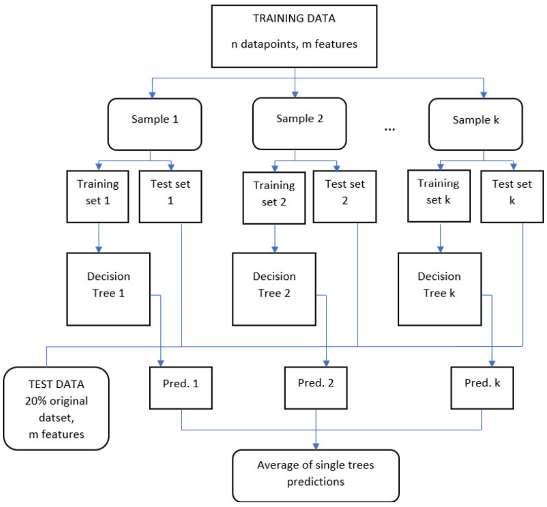


Fig 4. Flow chart of Random Forest Regression.

(out of the total $n\_features$, where $k < n$), and choose the best split among these chosen attributes. By doing this, each tree will perform differently on k distinct selected features, leading to different performances each time.

Step 3. To make a prediction on the unknown dataset, the algorithm uses the predictions obtained in **step 2** and averages the results to get the final prediction.

Fig. 4 and Fig. 5 demonstrate the flowchart of the Random Forest Model based on the research of Lingjian Yang in 2017 [17], and the flowchart of Decision Tree Model which is a part of the Random Forest Algorithm based on the work done in 2017 by Ibrahim A Ibrahim [18].

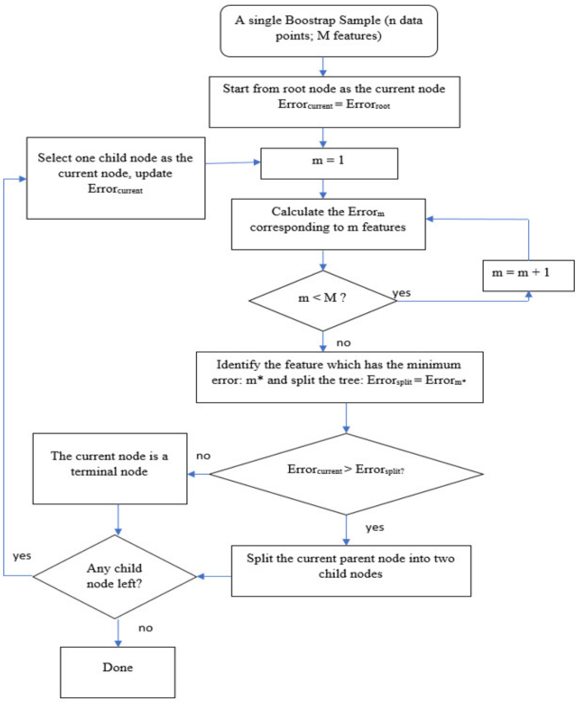

A single Boostrap Sample (n data points; M features)
Start from root node as the current node $\text{Error}_{\text{current}} = \text{Error}_{\text{root}}$
Select one child node as the current node, update $\text{Error}_{\text{current}}$
m = 1
Calculate the Error_{m} corresponding to m features
m = m + 1
m < M ?
yes
no
Identify the feature which has the minimum error: m* and split the tree: $\text{Error}_{\text{split}} = \text{Error}_{\text{m*}}$
The current node is a terminal node
no
$\text{Error}_{\text{current}} > \text{Error}_{\text{split}}$?
yes
yes
Any child node left?
Split the current parent node into two child nodes
no
Done

Fig 5. Flowchart of each bootstrap Sample in Random Forest.

2) Gradient Boosted Regression (XGBoost): In this study, Gradient Boosted Regression (XGBoost) [19] was implemented as a variant of Gradient Boosted Regression Trees. Similar to RF, Gradient Boosted (GB) models are built by many simple decision trees (weak learners), with a depth of one to five. However, the idea behind this model is that each tree can better predict and correct the mistakes of the previous ones. This results in the overall performance of the GB model being improved by adding more trees, and it can make more accurate predictions than the RF model if the parameters are set up meticulously. Therefore, XGBoost requires high accuracy and reliability from datasets. However, it requires careful tuning of the parameters and takes a long time to run.

3) Neural Network: Neural Network or Multilayer Perceptrons is a type of linear model that uses various stages of processing to get the final output. A multilayer model can perform efficiently with a large dataset, constructing a very complex model [20]. However, the models require quite a bit of running time as well as meticulous fine-tuning of the parameters. There are some parameters in this model that we have to take into account while implementing. First, hidden layer sizes (HLS), which is the number of hidden layers in the models; the number of units in each hidden layer; and the regularization, which is used to control the model's complexity.

D. Validation

In this study, overall performance was assessed using three indexes: 1) Mean Absolute Error (MAE), which is the absolute difference between the observed value (y_i) and the predicted result ($\hat{y}_i$). The lower the MAE, the closer the predicted result is to the actual value, and $MAE = 0$

is the ideal value; 2) $RMSE$ [21] is the average of the difference between $\hat{y}_i$ and y_i. The lower the $RMSE$, similar to MAE, the closer $\hat{y}_i$ is to y_i. The higher the $RMSE$, the more dispersed the $\hat{y}_i$ values are over a wider range; 3) The coefficient of determination (R_2) [22] has a value range of 0 to 1, indicating how close the predictions $\hat{y}_i$ are to the true value y_i of the model. When $R_2 = 1$, the ideal prediction is understood because it perfectly fits the real data and maximizes performance. In contrast, as R_2 approaches zero, the model becomes less reliable. In summary, a good model is satisfied when the $RMSE$, MAE, and R_2 are low. Eqa. (6)-(8) determines the above three indexes.

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{7}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y}_i - y_i)^2} \tag{8}$$

Where, $\bar{y}_i$ is the average AQI value at data point i.

IV. RESULTS AND DISCUSSION

In the study, the target of the model is the AQI in various Indian cities. In order to get the most precise prediction, we split the dataset into 2 parts: a training set containing 80% of the total data, which was used to train the model; and a test set holding the rest of the data points, which was used in the validation step. In order to choose the parameters for each model, we used a method called Grid Search [23], which tried different values of parameters in the model and then chose the optimized set containing the best ones.

A. Prediction of the AQI Applying Random Forest Regression

In this research, we employed two values: $n\_estimators$ and $n\_features$. The number of decision trees included in the model is $n\_estimators$, and the subset of features in each decision tree is $n\_features$. We applied Grid Search to locate the $n\_stimators$ parameter. Meanwhile, $n\_features$ were obtained by taking the square of the total features. According to that, the best combination of parameters used in the work was $n\_estimators = 500$ and $n\_features = 3$. The criteria for validating the model were: $MAE = 19.18$, $R2 = 0.94$, and $RMSE = 33.22$.

B. Performance Of XGBoost Model In Predicting AQI

For the XGBoost model, we also used Grid Search to choose two parameters, $n\_estimators$ and $learning\_rate$, which are the number of trees and the rate at which a tree can fix the mistakes of the previous ones. Meanwhile, the third parameter n_jobs, the selected set of parameters was $n\_estimators = 300$, $learning\_rate = 0.02$, and $n\_jobs = 4$. The model's statistical criteria were: $MAE = 18.98$, $R^2 = 0.942$, and $RMSE = 32.6$.

TAB II
PERFORMANCE OF MODEL BASED ON 3 CRITERIA.

Methods	MAE	R^2	$RMSE$
Random Forest (RFR)	19.18	0.94	33.22
XGBoost	18.98	0.942	32.6
Neural Network	22.36	0.928	36.39

C. AQI Prediction Of Neural Network Model

According to the previous section, we had two parameters in Neural Network models: hidden layer sizes (HLS) and the number of units in each hidden layer (α). Using the Grid Search method, we obtained the optimal values for the two parameters with $HLS = 50$, and $\alpha = 0.5$. The results of the validation criteria were $MAE = 22.36$, $R^2 = 0.928$, and $RMSE = 36.39$. Tab. II displays the comparison of the three models' performances over the three corresponding criteria.

As can be seen from the result, while Random Forest and XGBoost got performances that are approximately the same in both three criteria, Neural Network, however, performed less efficiently with the same conditions. The MAE and $RMSE$ of this model are much higher, at 23.36 and 36.39, respectively, yet $R_2 = 0.928$ is lower than that of the two other algorithms. As a result, XGBoost is the most effective among the three models when it comes to the statistical criteria, with $MAE = 18.98$, $R_2 = 0.942$, and $RMSE = 32.6$. Fig. 6 shows the comparison between the result of the XGBoost model's prediction and actual AQI values with 500 samples. Look at this diagram. The predicted value line (orange) closely follows the actual value line (blue). The distinction is insignificant. This is consistent with the $MAE = 18.98$ value that we measured.

Huixiang Liu *et al.* [10] used two indices to examine the difference in AQI prediction in Beijing, China: correlation (R_2) and mean of difference ($RMSE$). They used Support Vector Regression (SVR) and Random Forest Regression (RFR) in their study and obtained two sets of indices ($R2 = 0.9760, RMSE = 94.4918$) and ($R2 = 0.8401, RMSE = 83.6716$). These two indexes are also used by Chao Song and Xiaoshuang Fu in their paper [[24]. They integrated a set of algorithms into the one called Combination Forecasting Model (CFM) to get the predictions of AQI in Zengzhou and Shanghai, China. Their results finally reached $RMSE = 36.89$ and $R_2 = 0.86$ for the dataset collected in Zhengzhou, and ($RMSE = 35.32, R_2 = 0.72$) for the other location – Shanghai. Even though these works are different from our research because of the dataset, Machine Learning algorithms, and some other criteria used to evaluate the models, it is suggested that our models achieved quite good results compared to those of other research when assessed using the same criteria.

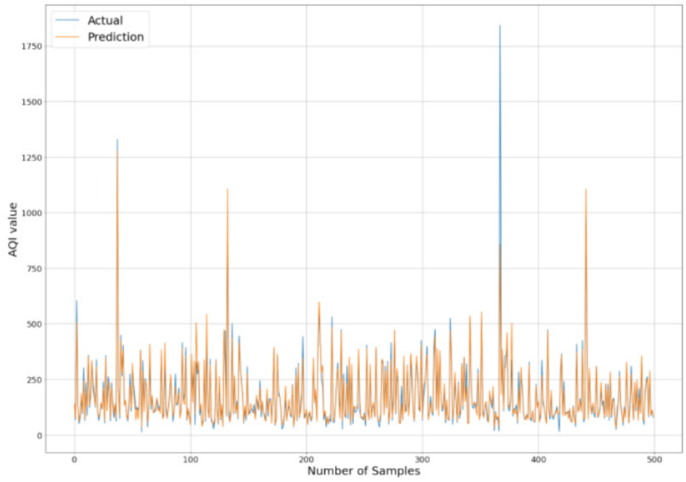


Fig 6. Correlation between AQI values predicted and measured using the XGboost algorithm.

V. CONCLUSIONS

Air quality has a direct impact on human life and society as a whole. As a result, not only the government, but also individuals and organizations, must work together to prevent environmental pollution, particularly air pollution. As a result, the AQI index is needed to evaluate air quality, and it can also be used to design and produce intelligent meteorological monitoring devices. In this paper, the air pollution indicators in many Indian cities were analyzed and predicted in this study using real data on pollutants provided by the Indian government. The study's findings demonstrated that, while all three models provide good predictive results, the XGB model outperforms the others. Meanwhile, the Neural Network model, which requires careful tuning parameters, and much operating time, was not as effective as XGB and RFR. To conclude, the study showed the data analysis and transformation, then built three models for AQI predictions, which attempted to improve the performance in terms of each model's accuracy. In the future, we expect to develop algorithms on devices that use low-power microcontrollers to predict air quality remotely [6], [8].

REFERENCES

[1] D.-C. Nguyen, T. Duc-Tan, and D.-N. Tran, "Application of compressed sensing in effective power consumption of WSN for landslide scenario," in *2015 Asia Pacific Conference on Multimedia and Broadcasting*, 2015, pp. 1–5.

[2] D. T. Pham, D. C. Nguyen, V. V. Pham, B. C. Doan, and D. T. Tran, "Development of a Wireless Sensor Network for Indoor Air Quality Monitoring," in *The 2015 International Conference on Integrated Circuits, Design, and Verification*, Vietnam, 2015, pp. 178–183.

[3] H. Gu, W. Yan, E. Elahi, and Y. Cao, "Air pollution risks human mental health: an implication of two-stages least squares estimation of interaction effects," *Environmental Science and Pollution Research*, vol. 27, no. 2, pp. 2036–2043, 2020.

[4] S. Kumari and M. K. Jain, "A Critical Review on Air Quality Index," *Environmental Pollution*, vol. 77, pp. 87–102, 2018.

[5] S. Lemeš, "Air Quality Index (AQI)—comparative study and assessment of an appropriate model For B&H," in *2th Scientific/Research Symposium with International Participation 'Metallic And Nonmetallic Materials*. MNM, 2018, pp. 282–291.

[6] N. H. Van, P. Van Thanh, D. N. Tran, and D.-T. Tran, "A new model of air quality prediction using lightweight machine learning," *International Journal of Environmental Science and Technology*, 2022. [Online]. Available: https://doi.org/10.1007/s13762-022-04185-w

[7] N. C. Minh, T. H. Dao, D. N. Tran, Q. H. Nguyen, T. T. Nguyen, and D. T. Tran, "Evaluation of Smartphone and Smartwatch Accelerometer Data in Activity Classification," in *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, 2021, pp. 33–38.

[8] N. T. Thu, T.-h. Dao, B. Q. Bao, D.-n. Tran, P. V. Thanh, and D.-T. Tran, "Real-Time Wearable-Device Based Activity recognition Using Machine Learning Methods," *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 321–333, 2022. [Online]. Available: https://dx.doi.org/10.12785/ijcds/120126

[9] J. K. Sethi and M. Mittal, "A new feature selection method based on machine learning technique for air quality dataset," *Journal of Statistics and Management Systems*, vol. 22, no. 4, pp. 697–705, 2019. [Online]. Available: https://doi.org/10.1080/09720510.2019.1609726

[10] H. Liu, Q. Li, D. Yu, and Y. Gu, "Air quality index and air pollutant concentration prediction based on machine learning algorithms," *Applied Sciences (Switzerland)*, vol. 9, no. 19, 2019.

[11] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L. Vanneschi, "A Machine Learning Approach to Predict Air Quality in California," *Complexity*, vol. 2020, pp. 1–23, 2020. [Online]. Available: https://doi.org/10.1155/2020/8049504

[12] P. Bhawan and E. A. Nagar, "Central Pollution Control Board," pp. 1–93, 2019.

[13] R. R. Dickerson, D. C. Anderson, and X. Ren, "On the use of data from commercial NOx analyzers for air pollution studies," *Atmospheric Environment*, vol. 214, no. June, p. 116873, 2019. [Online]. Available: https://doi.org/10.1016/j.atmosenv.2019.116873

[14] S. Zhang, "Nearest neighbor selection for iteratively kNN imputation," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2541–2552, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2012.05.073

[15] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7473 LNCS, pp. 246–252, 2012.

[16] Y. Fang and J. Wang, "Selection of the number of clusters via the bootstrap method," *Computational Statistics and Data Analysis*, vol. 56, no. 3, pp. 468–477, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.csda.2011.09.003

[17] L. Yang, S. Liu, S. Tsoka, and L. G. Papageorgiou, "A regression tree approach using mathematical programming," *Expert Systems with Applications*, vol. 78, pp. 347–357, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2017.02.013

[18] I. A. Ibrahim, T. Khatib, A. Mohamed, and W. Elmenreich, "Modeling of the output current of a photovoltaic grid-connected system using random forests technique," *Energy Exploration and Exploitation*, vol. 36, no. 1, pp. 132–148, 2018.

[19] Y. Wang, Z. Pan, J. Zheng, L. Qian, and M. Li, "A hybrid ensemble method for pulsar candidate classification," *Astrophysics and Space Science*, vol. 364, no. 8, 2019.

[20] D. Faraggi and R. Simon, "A neural network model for survival data," *Statistics in Medicine*, vol. 14, no. 1, pp. 73–82, 1995.

[21] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005.

[22] A. H. Murphy, "The coefficients of correlation and determination as measures of performance in forecast verification," *Weather and Forecasting*, vol. 10, no. 4, pp. 681–688, 1995.

[23] A. C. Müller and S. Guido, *Introduction To Machine Learning With Python: A Guide For Data Scientists*. O'Reilly Media, Inc., 2016.

[24] C. Song and X. Fu, "Research on different weight combination in air quality forecasting models," *Journal of Cleaner Production*, vol. 261, p. 121169, 2020. [Online]. Available: https://doi.org/10.1016/j.jclepro.2020.121169

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 95–101 DOI: 10.15439/2022R25 ISSN 2300-5963 ACSIS, Vol. 33

A Big Data Platform for Real-Time Video Surveillance

Thi-Thu-Trang Do
School of Information and Communication Technology
Hanoi University of Science and Technology
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
trangdtt@utehy.edu.vn

Quang-Thinh Dam
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
thinhquangshin@gmail.com

Tai-Huy Ha
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
hataihuy2001@gmail.com

Quyet-Thang Huynh
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
thanghq@soict.hust.edu.vn

Kyungbaek Kim
Department of Artificial Intelligence Convergence
Chonnam National University
Gwangju, Korea
kyungbaekkim@jnu.ac.kr

Van-Quyet Nguyen*
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
quyetict@utehy.edu.vn

***Abstract*—Nowadays, smart house facilities are strongly developed with the support of multiple security cameras to protect not only a house but also a building. A large amount of video data is produced by these cameras every day. Therefore, traditional data management systems face challenges in collecting, storing, and analyzing big video data. In such systems, it is difficult to find objects and their actions from video surveillance in the building because of either the consuming time or the lack of intelligent technology support. In this paper, we propose a novel big data platform for real-time video surveillance analysis based on the combination of distributed data frameworks and intelligent video processing libraries. The proposed platform is able to collect both real-time video streams and historical video data by using Kafka and Spark Structured Streaming frameworks. Furthermore, the proposed platform provides an intelligent video processing module for object detection by using OpenCV, YOLO, and Keras libraries. To evaluate the proposal, we deploy the proposed big data platform and implement a web interface to support end-user to analyze video surveillance. Through the results of the initial video querying implementation, we show the viability of the proposed platform.**

***Index Terms*—Spark Structured Streaming, Kafka, Video Querying, Video Streaming, Video Surveillance**

I. Introduction

Recently, the volume of video data has increased dramatically on the internet from various sources such as Youtube, Facebook, and Tiktok. These unstructured video data are reservoirs of knowledge and have a direct relation to real-world events. It provides information about people's interactions and behaviors. Moreover, real-time video streams can help in behavior analysis whether it is of traffic or human patterns. The development of technology has also led to the development of security and healthcare systems. A large amount of video surveillance data is stored so that it can be processed when any event occurs. However, manually analyzing video surveillance will take a lot of time and effort. Therefore, video analysis platforms are researched and developed to manage and analyze these video data. These usually studied to evaluate and optimize data transmission throughput and speed. Bunrong Leang et al. [1] proposed a Hadoop ecosystem for supporting to several features in the manufacturing industry. Because the author only use Apache Hadoop and Kafka, ecosystem is quite limited in speed. To address this issue, we take advantage of Spark - a fast and general engine for large-scale data processing. Ayae Ichinose et al. [2] proposed a Streaming Video Engine (SVE) for uploading and processing videos in a distributed manner. That is a framework for real-time video analysis on Spark from which draw some conclusions about the dependence of throughput with the number of brokers or topic partitions. Kai Yu et al. [3] introduced a Video parsing and evaluation platform using Spark Structured Streaming and YARN with Kafka to solve problems existing in ordinary video surveillance systems. However, this platform does not support for extracting objects or properties of objects.

Mark Hamilton et al. [4] proposed a distributed image

* Corresponding author.

processing library which integrates OpenCV with Spark and Cognitive Toolkit - a deep learning library. However, this study is also limited to process images and does not provide any video processing APIs. Lei Huang et al. [5] introduced a method based on convolutional neural networks for recognizing objects in traffic video data. The video data is stored and processed by applying Spark. Kut et al. [6] proposed solutions based on both Hadoop and Spark for detecting edge using canny operator and line using Hough transform. Thus, most of the literature lacks the support for distributed feature extraction APIs.

To address this problem, a distributed video analysis framework is necessary. Md Azher Uddin et al. [7] proposed SIAT - a distributed video analysis framework for intelligent video surveillance. SIAT uses state-of-the-art distributed computing technologies to handle real-time video streams and batch video analytics to ensure scalability, efficiency, and fault tolerance. However, this framework does not have user interface. Melenli et al. [8] introduced a distributed image processing framework in real-time using Apache Kafka and Spark. They also used OpenCV and YOLOv3 to perform human detection and measured the distance between people. Besides, the authors have created an user interface which is used to define cameras, create notifications, or access real-time dashboards.

In this paper, we propose a novel big data platform for real-time video surveillance analysis based on the combination of distributed data frameworks and intelligent video processing libraries. The proposed platform be able to collect, analyze and store both real-time video streams and historical video data. We also implement a web interface to support end-user to analyze video surveillance and retrieve information from database.

II. Related Work

Real-time data processing and big data analytics have been attracting much attention recently [9, 10]. In particular, the development of the surveillance field has generated a huge of data daily leading to organizations must devise ways of handling this information since the existing techniques did not handle efficiently such a volume of data created at such a high rate. For example, Syafrudin et al. [11] introduced a real-time monitoring system that utilizes IoT-based sensors, big data processing, and a hybrid prediction model for the automotive industry. A Hadoop ecosystem is proposed to support several features in the manufacturing industry [1]. Zhang et al. [12] also proposed a cloud-based architecture that used Apache Kafka and Storm for real-time processing and Hadoop based MapReduce framework for batch video data processing. However, the Hadoop ecosystem and Kafka have a drawback which is the limitation of speed. It is the reason why Spark is studied and developed [13, 14].

Zhou et al. [15] introduced a distributed architecture, which can measure and monitor online Internet traffic. Ayae Ichinose et al. [2] proposed a Streaming Video Engine (SVE) - a framework for real-time video analysis on Spark, while Kai Yu et al. [3] introduced a video parsing and evaluation platform to solve problems existing in ordinary video surveillance systems. On the other hand, extracting objects or their properties of them are also attracted much attention and research.

BigDL [16] - a distributed deep learning framework using Spark is recently introduced, which allows users to develop deep learning applications. In this reference, they showed efficiency with object detection and image feature extraction. In addition, BigDL supports immensely efficient and scalable distributed training. Mark Hamilton et al. [4] also proposed a distributed image processing library that integrates OpenCV with Spark and Cognitive Toolkit. However, they do not provide support for video processing. In contrast, Huang et al. [5] proposed a method based on convolutional neural networks for recognizing objects in traffic video data and using Spark to store and process this information. However, most of the existing proposals do not yet support the retrieval of information from the analyzed and stored data.

Kut et al. [6] proposed solutions based on both Hadoop and Spark for detecting edge using the canny operator and line using Hough transform, while SIAT [7] is proposed by Md Azher Uddin is a distributed video analysis framework for intelligent video surveillance. However, they do not implement a user interface. Melenli et al. [8] introduced a distributed image processing framework in real-time using Apache Kafka and Spark, which can detect humans, measure the distance between people and implement a user interface.

III. Design of a Big Data Platform for Video Surveillance

A. Architecture of the Proposed Platform

We propose a platform for collecting and analyzing multiple video data in real-time with four components including Data Collector, Data Storage, Intelligent Video Data Processor, and Data Query Module. Figure 1 presents the architecture of our proposed big data platform. The first component is the Data Collector which contains two sub-components: Kafka cluster and producers. Brokers in the Kafka cluster are managed by Zookeeper. Producers have the responsibility for extracting data from real-time videos and then sending them to the brokers to store. The collected data from video by the Data Collector will be consumed by the third component - the Intelligent Video Data Processor which has a processing mission using Spark Structured Streaming and SparkSQL. Raw data after being consumed by the processing module is contained by the second module - The Data Storage. The processed data by the Intelligent Video Data Processor will be sent to the fourth component - the Data Query Module via the HTTP server. Then, the HTTP server will send back the data to the Data Storage – SQL Server to store for serving the Data Query module. We just save several frames in SQL Server to reduce the vast number of frames, instead, we save it into Hbase - a distributed NoSQL database and it saves the data as a file in Hadoop Distributed File System (HDFS), so the system can serve query tasks more rapidly. The web client in the Data Query module is a user interface that supports humans to query data from the SQL server.

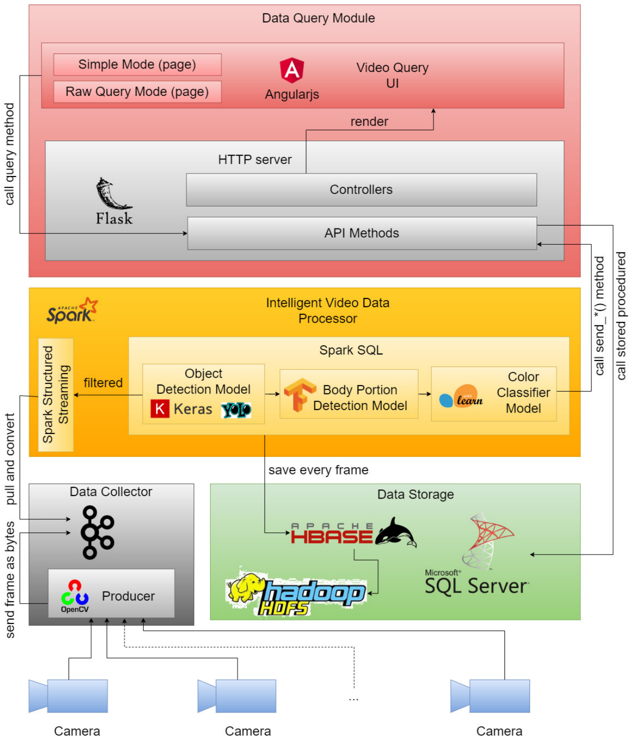


Fig. 1. The system architecture.

B. Design of Real-Time Data Collector Module

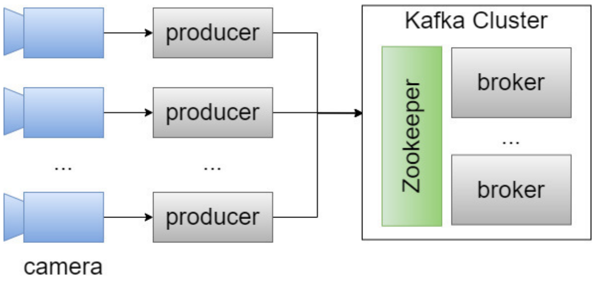


Fig. 2. The Kafka system.

Figure 2 shows our design for collecting data using Kafka. This module is designed to collect the generated data from cameras. We do not want any data loss when receiving data from producer, so we design this component according to replicas and use Zookeeper to manage nodes in the cluster. We all know that the generated data from cameras are videos that are huge collections of images. These images move fast enough for us to see the things call animations. Producers have the responsibility for collecting these images and then converting them to three-dimension arrays called frames by a powerful computer vision library – OpenCV. The problem is that a large number of frames are generated every second. Therefore, instead of sending all of them, we just send one per two seconds to ensure that we have enough bandwidth to transfer and the brokers have enough storage to store them. However, we have a second problem: the matrices can not be transferred by the producers. So, we convert them to bytes strings to send more easily.

C. Design Choice of Data Storage Module

We design two sub-components in the Data Storage module. The first component is Distributed NoSQL database named Hbase. We use this database because it is fast when reading, writing, or random data. Furthermore, it can store massive data because it bases on the Hadoop Distributed File System (HDFS) which stores data safely and distributedly. The second component is SQL Server, we use it to store the processed data by the Intelligent Video Data Processor module.

1) Design of Basic Distributed Storage: We need basic storage for storing raw frame data at every time. These data are filtered by the Intelligent Video Data processor module. These raw data are usually sequentially contained in text files. In this case, our type of data is byte-string which is a fixed-length array of bytes. A byte is an exact integer between 0 and 255 inclusive. These raw data are the basic data for processing and we use Hbase as basic storage for them.

2) Design of Storage for Querying Data: In our platform, we also consider an easy API for querying data with SQL language. For this, we use SQL Server because it is simple to implement and easy to integrate into our platform. Our platform is designed to trace objects and extract properties from them, so we have to design a schema that is suitable for this case. We have six tables to contain information extracted by the Intelligent Video Data Processor module.

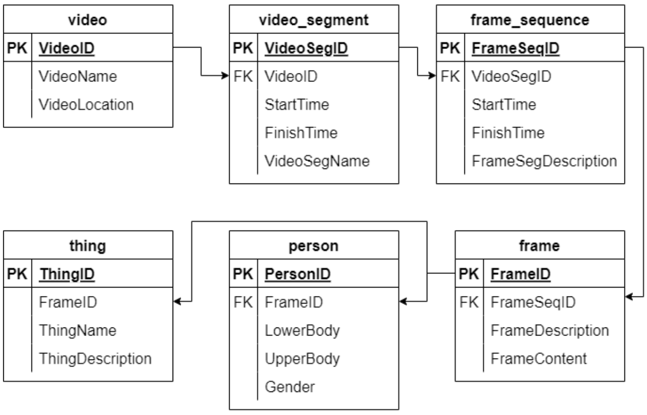


Fig. 3. Database schema.

- **Video**. It contains information about cameras which were installed in our system. Each camera has four properties particularly: ”VideoId” - a unique key to distinguish from others, ”VideoName”- the name of the camera, ”VideoLocation” - the name of the location where the camera was installed.
- **Video_segment**. It contains information about segments of each video. Per ten minutes, we create a new segment. The table has five properties: VideoSegId - a unique key to distinguish from others, VideoId - the camera’s key

producing this segment, StartTime and EndTime are times when video starts and stops, VideoSegName - the name of the video.

- **Frame_sequence**. This table contains information about the period when the processing module detected an object in a frame. For each object detected by the processing module, we create a new frame_sequence record.
- **Frame**. The fourth table is Frame which contains frames from cameras that have objects detected by the processing module. We have to encode the frames to base64 to store them in the database more easily.
- **Person** and **Thing**. Person and thing contain information about objects detected by the processing module. Table ”person” contains information about the properties of humans and table ”thing” contains information about anything that is not humans. The Fig.3 illustrates a schema of the database.

D. Design Choice of Intelligent Video Data Processor Module

In order to consume the data stream from the Kafka cluster, we design this module including AI models. The AI models have the responsibility for detecting objects that appear in frames sent by Kafka, then extracting properties from them. We use a built-in library – request to send output data to the HTTP. Leveraging the real-time querying power of SparkSQL, we built a user-defined function (UDF) and used models inside server.

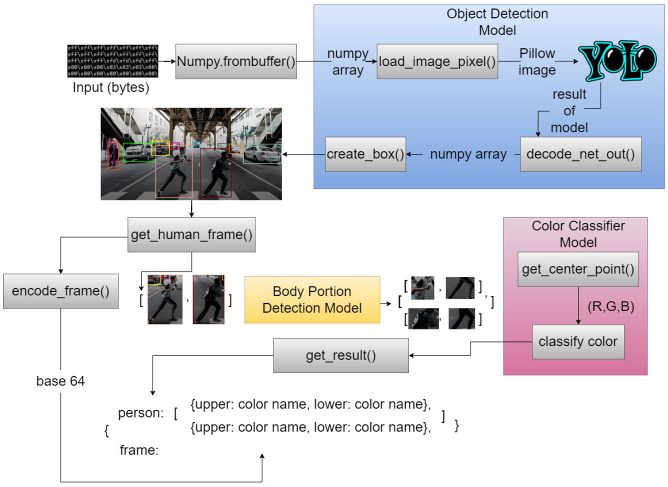


Fig. 4. Processing progress.

1) Design Choice of Image Processing models: Figure 4 demonstrates the processing sequence of the Spark Structured Streaming program. The input of the Intelligent Video Data processor module is a collection of byte strings filtered by SparkSQL after receiving data from Kafka clusters. So, we use a powerful computer vision library – OpenCV to convert them to frame type (NumPy array) because AI models can not calculate byte string data. In order to detect objects in the converted frame, we use a pre-train AI model – YOLOv3. But, there is a problem that YOLO can not integrate with Spark Structured Streaming. Because it is developed base on the DarkNet platform which creates a new thread whenever it runs. However, Spark Structured Streaming does not accept any thread except its, therefore, we have to convert the DarkNet model to another type of model that can be excepted by Spark Structured Streaming. So, we use Keras to convert the YOLOv3 model to the Keras model. We then use body portion detection to detect the upper body portion and lower body portion of human images classified. To train this model, we use a pre-train model called ”Tflite Model Maker” which is developed by Google with 20000 labeled images by Labelimg tools. To collect images for training the model, we downloaded some videos recorded by security cameras from Youtube which were public. Then we use the Yolov3 model to get a human object displayed in each frame in each video and save them to a folder. After that, we used the LabelImg tools to label the images. In each human frame, start point from the shoulder to the end of the back, we labeled this part as ”upper” and from the end of the back to the ankle we labeled it as ”lower”. For the train set, we use 10000 images, 5000 for the test set, and 5000 for validating set. Then, we use the Tflite model maker to train our custom model with the labeled dataset. The model after training has an accuracy is 0.81. Finally, we use a simple machine learning model to classify colors in body portion images. It is implemented by using the KNN algorithm built in Scikit-learn. The output of the whole processing process and related information (object properties) are sent to the HTTP server. Especially, the out frames are converted to base64 encoding to be easy to store in the database. Figure 5 shows the image process and information extraction process.

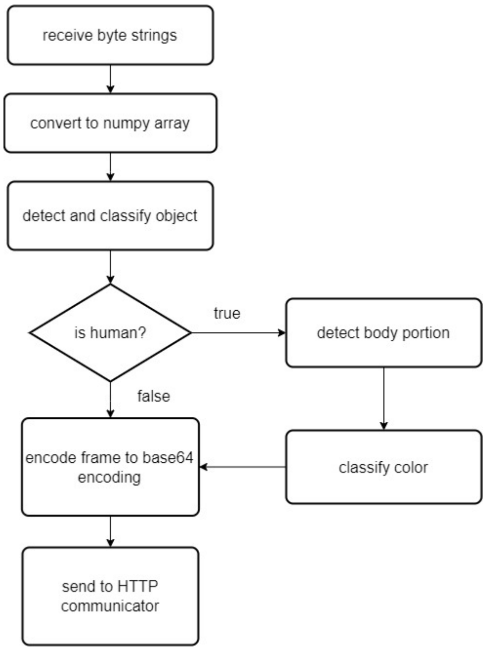


Fig. 5. Processing logic.

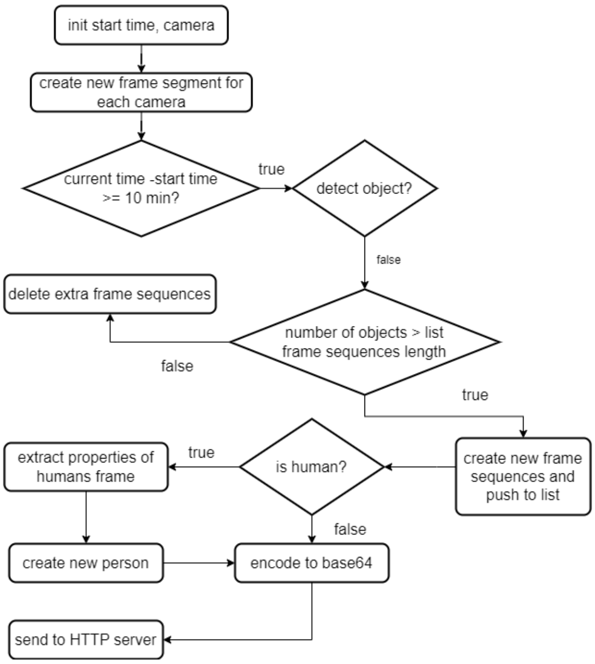

init start time, camera
create new frame segment for each camera
current time -start time >= 10 min?
true
detect object?
false
delete extra frame sequences
number of objects > list frame sequences length
false
true
extract properties of humans frame
true
is human?
create new frame sequences and push to list
false
create new person
encode to base64
send to HTTP server

Fig. 6. Database business.

2) Design of Database Business: The Fig.6 shows the design of the database business. As we design the schema of the database, we need to design a business that is suitable for storing data in the database. First, we initialize the start time, and information about the cameras installed in our system. We then create new frame segments for each camera. Because, after 10 minutes, we need to create a new segment, we need to check the running time by subtracting the starting time from the current time. If the running time is greater than or equal to 10 minutes, new frame segments are created. Next, we will check the objects detected by the YOLO model. If the number of detected objects is greater than the number of frame sequences, new frame sequences will be created. The number of new sequences is equal to the number of extra objects. If the number of detected objects is smaller than the number of sequences, the extra sequences will be removed. Then, for human frames, we extract properties from them. After that, we convert the frames to base64 including things' frames. Finally, we send this information to the HTTP server.

E. Design Choice of Data Query module

We need a web API to query data from the database and transfer data from the data processing module to the database. To do this, we choose the Flask library to create one because of its lightweight and simplicity. Our API has 6 HTTP methods as the following:

- **send_segments** method. It is used to send segment information from the data processing module to the HTTP server.
- **send_sequences** method. It is used to send frame sequence information to the server.
- **send_frames** method. The send_frames has the responsibility for transferring base64 frames from the processing module.
- **send_people** and **send_things** methods. The send_people method and send_things method are used to send information about things and humans' properties.
- **index** method. It is used to return an interface to the user.

These methods use PyOdbc - a built-in library to execute stored procedures written in the database. The query method with its parameter – command will send a command to the SQL server to execute the command and then receive the result.

IV. Implementation and Experiments

A. Environment Settings

We deployed the platform in a cluster including 5 machines, in which, two machines are used to collect data from two cameras and three other machines are to install Hadoop and Spark workers as a cluster. The cluster includes two workers and a master. The master machine has 4GB RAM and 4 CPUs and workers have 2GB RAM and 2 CPUs on each machine. We use another machine that has 4GB RAM and 4 CPUs to install the SQL server database and the HTTP server. And the 2 machines which are used to collect data, which we use to deploy our Kafka cluster, have the same configuration as the workers.

B. Implementation of Web User Interface for Querying Video Data

We designed a web client to support the users to query data from SQL Server more easily. To build this client, we used AngularJS and call the API we built. The client has two modes: Simple Mode and Raw Query Mode. In addition, to run the platform we had to add the cameras' information to the database first and configuration file. Then, we started the SQL server and HTTP server. After that, we started the Hadoop system and Spark Structured Streaming job with our workers. Finally, we started all cameras and the Kafka cluster.

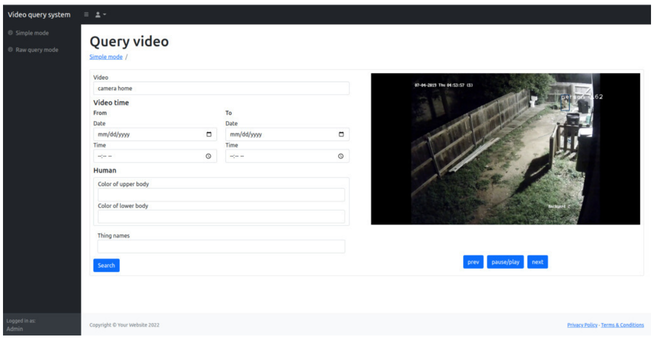
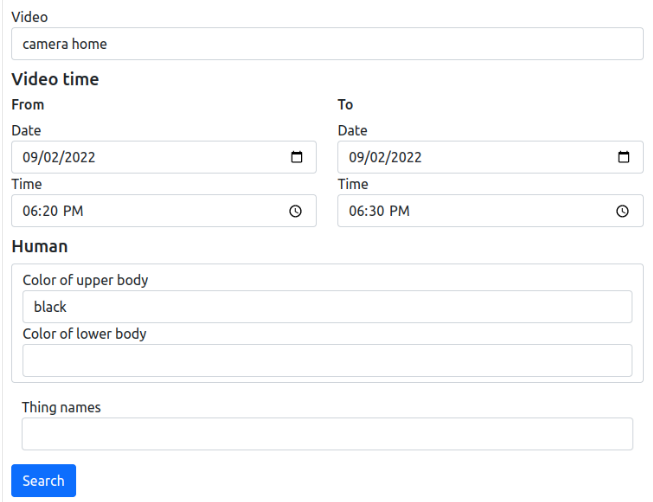

Video
camera home

Fig. 7. The user interface of Simple Mode.

1) Simple Mode: The user interface shown in Fig.7 has two sections. The first section is the input of searching. It is used to type the properties of an object and the information about place and time. The first component of the first section is a list of cameras that can choose to specify which camera the users want to query. The second component is ”Video Time” advocating the users to query the start and end time of the video to limit the range of time in which the object appeared. The third component is used to query the properties of humans appearing in the camera. The last component supports the users to query things existing in the camera.

The second section displays the result of the search as a list of images. We designed the second section with three buttons to move between images or auto-play as a video. For example, we queried the data from the camera named ”home camera” from 18:20 to 18:30 at 2nd December to find the person who wore a black shirt. Every search inputs are shown in Fig.8 and the result of searching is demonstrated in Fig.9



Video
camera home

Video time

From
Date
09/02/2022
Time
06:20 PM

To
Date
09/02/2022
Time
06:30 PM

Human

Color of upper body
black

Color of lower body

Thing names

Fig. 8. Query the person with properties display in specified camera in range time.

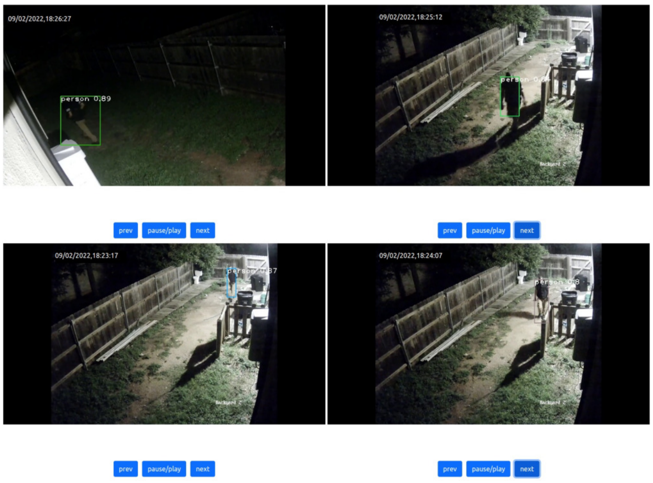


Fig. 9. The result of the searching in Fig. 8.

In the other case, we searched a man who wore a grey shirt who appear in all cameras and the all time, the result showed in Fig.10.

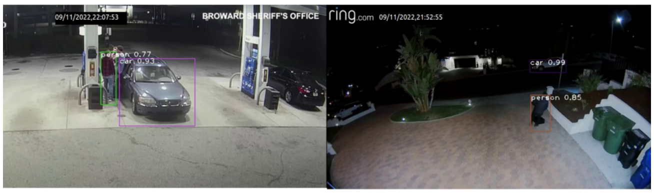
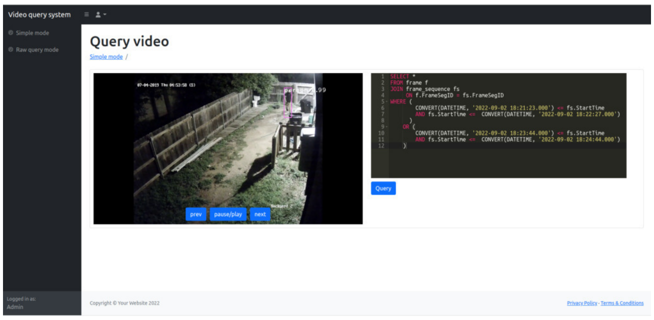


Fig. 10. The result of the searching.



```sql
SELECT *
FROM frame f
JOIN frame_sequence fs
     ON f.FrameSegID = fs.FrameSegID
WHERE (
        CONVERT(DATETIME, '2022-09-02 18:21:23.000') <= fs.StartTime
        AND fs.StartTime <=  CONVERT(DATETIME, '2022-09-02 18:22:27.000')
      )
    OR (
        CONVERT(DATETIME, '2022-09-02 18:23:44.000') <= fs.StartTime
        AND fs.StartTime <=  CONVERT(DATETIME, '2022-09-02 18:24:44.000')
    )
```

Fig. 11. User interface of Raw Query Mode.

2) Raw Query Mode: We built the raw query mode to enhance the query ability of the system. It has two sections, the first section displays the result as a list of images, and the second section is used to write SQL commands. The interface was designed as Fig.11. This mode can be used to query with a complex condition that the simple mode does not support. The example below demonstrated how to search with multiple cameras in multiple time ranges. To display the result as an image and display the time at which the frame is recorded, we need to join two tables: frame and frame_sequences. The script which we used to query, is shown in Listing 1 and the result of the query is illustrated in Fig.12.

```sql
SELECT * FROM frame f
JOIN frame_sequence fs
ON f.FrameSegID = fs.FrameSegID
JOIN video_segment vs
ON vs.VideoSegID = fs.VideoSegID
JOIN video v ON v.VideoID = vs.VideoID
WHERE
    VideoLocation = ’yard’
    OR VideoLocation = ’petrol station’
    AND (
        CONVERT(DATETIME, ’2022-09-11 21:35:23.000’)
     <= fs.StartTime
        AND fs.StartTime <= CONVERT(DATETIME, ’
    2022-09-11 21:50:27.000’)
    )
    OR (
        CONVERT(DATETIME, ’2022-09-11 22:35:23.000’)
     <= fs.StartTime
        AND fs.StartTime <= CONVERT(DATETIME, ’
    2022-09-11 22:50:27.000’)
    )
```

Listing 1. query images ordered by time

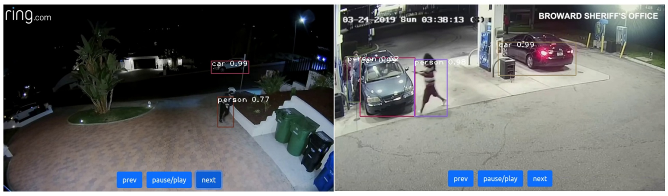


Fig. 12. The result of query command in Listing 1.

V. Conclusions

This paper proposed a design and implementation of a big data platform for real-time video surveillance. The proposed platform was designed with the aim that supporting user to manage their cameras system more efficiently by using the strong analytic power of big data tools integrating with a set of AI models. The proposed platform is able to collect both real-time video streams and historical video data by using Kafka and Spark Structured Streaming frameworks. It also provides an intelligent video processing module for object detection by using OpenCV, YOLO, and Keras libraries. Through the results of the initial video querying implementation, we show the viability of the proposed platform. In the future, we will work on optimizing AI models and integrating the platform with distributed search engines instead of using the SQL language in the relational database to improve search speed.

Acknowledgements

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2022-RS-2022-00156287) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

[1] Bunrong Leang et al. "Improvement of Kafka streaming using partition and multi-threading in big data environment". In: *Sensors* 19.1 (2019), p. 134.

[2] Ayae Ichinose et al. "A study of a video analysis framework using Kafka and spark streaming". In: *2017 IEEE International Conference on Big Data (Big Data)*. 2. IEEE. 2017, pp. 2396–2401.

[3] Kai Yu et al. "A large-scale distributed video parsing and evaluation platform". In: *Chinese Conference on Intelligent Visual Surveillance*. 3. Springer. 2016, pp. 37–43.

[4] Mark Hamilton et al. "Flexible and scalable deep learning with MMLSpark". In: *International Conference on Predictive Applications and APIs*. 4.

[5] Lei Huang et al. "Enabling versatile analysis of large scale traffic video data with deep learning and HiveQL". In: *2017 IEEE International Conference on Big Data (Big Data)*. 5. IEEE. 2017, pp. 1153–1162.

[6] Seda Kul et al. "Event-based microservices with Apache Kafka streams: A real-time vehicle detection system based on type, color, and speed attributes". In: *IEEE Access* 9.6 (2021), pp. 83137–83148.

[7] Md Azher Uddin et al. "SIAT: A distributed video analytics framework for intelligent video surveillance". In: *Symmetry* 11.7 (2019), p. 911.

[8] Sadettin Melenli and Aylin Topkaya. "Real-time maintaining of social distance in covid-19 environment using image processing and big data". In: *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*. 8. Springer. 2020, pp. 578–589.

[9] Lucy Linder et al. "Big building data-a big data platform for smart buildings". In: *Energy Procedia* 122.10 (2017), pp. 589–594.

[10] Tom Wilcox et al. "A Big Data platform for smart meter data analytics". In: *Computers in Industry* 105.11 (2019), pp. 250–259.

[11] Muhammad Syafrudin et al. "Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing". In: *Sensors* 18.12 (2018), p. 2946.

[12] Weishan Zhang et al. "A video cloud platform combing online and offline cloud computing technologies". In: *Personal and Ubiquitous Computing* 19.13 (2015), pp. 1099–1110.

[13] Jian Fu, Junwei Sun, and Kaiyuan Wang. "Spark–a big data processing platform for machine learning". In: *2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*. 16. IEEE. 2016, pp. 48–51.

[14] R Shyam et al. "Apache spark a big data analytics platform for smart grid". In: *Procedia Technology* 21.17 (2015), pp. 171–178.

[15] Baojun Zhou et al. "Online internet traffic monitoring system using spark streaming". In: *Big Data Mining and Analytics* 1.14 (2018), pp. 47–56.

[16] Jason Jinquan Dai et al. "Bigdl: A distributed deep learning framework for big data". In: *Proceedings of the ACM Symposium on Cloud Computing*. 15. 2019, pp. 50–60.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 103–108 DOI: 10.15439/2022R02 ISSN 2300-5963 ACSIS, Vol. 33

Comparative Study of Deep Learning Models for Predicting Stock Prices

Van-Thang Duong
Faculty of Information Technology
University of Transport in Ho Chi Minh City (UTH)
Ho Chi Minh, Vietnam
thangduongvan2000@gmail.com

Duc-Tuan-Anh Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
tuananh03@gmail.com

Thi-Thu-Hang Pham
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
thuhangpt249@gmail.com

Van-Hau Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education (UTEHY)
Hung Yen, Vietnam
haunv@utehy.edu.vn

Van-Quoc Anh Le
Faculty of Information Technology
University of Transport in Ho Chi Minh City (UTH)
Ho Chi Minh, Vietnam
anh@ut.edu.vn

***Abstract*—The stock market is volatile, dynamic, and nonlinear. Hence, predicting the stock prices has been a challenging task for any researcher in time series forecasting. Accurately predicting stock prices has been a hot topic for both financial and technical researchers. In this paper, we deploy six deep learning models (i.e., MLP, CNN, RNN, LSTM, GRU, and AE) to predict the closing price, one day ahead, of 20 different companies (i.e. 5 groups of 4) in the S&P 500 markets over the 7-years range (Jan 2015 - August 2022). The experimental results do not provide interesting insights, but also help us to deepen our understanding of how to use deep learning models in financial markets.**

***Index Terms*—stock price prediction, deep learning, machine learning.**

I. Introduction

The stock market plays an important role in the global economy today. Accurately predicting stock prices can avoid investment risk and lead to a hugely profitable investment. Recently, it has been a hot topic for both financial and technical researchers [1]–[3]. In fact, the stock market is volatile, dynamic, and nonlinear. Hence, predicting the stock prices has been a challenging task for any researcher in time series forecasting due to various factors, such as the global economy, political conditions, the company's financial performance, etc.

There are two approaches to predicting the stock markets. The first approach is qualitative (or fundamental analysis) [1], in which the intrinsic values are examined, such as market situation, financial factors, management effectiveness, consumer behaviors, and information from social media, economic analyst, etc.[1] The second approach is the technical analysis, in which the historical stock market activities are examined, for example, the price of opening, closing, maximum, minimum, adjusted closing prices and volume of a day.

Unlike the fundamental analysis that is useful for long-term investment, the technical analysis is easily influenced by short-term news. The second approach consists of two main methods, the traditional method and the machine learning method. To predict stock prices, the former widely uses classical techniques for time series models, like AR(Autoregressive), MA(Moving Average), ARMA(Autoregressive Moving Average), and ARIMA (Autoregressive Integrated Moving Average). In the latter, i.e. the machine learning method, while classical machine learning-based models have been extensively studied and obtained significant results due to their effectiveness despite working with a limited amount of data, deep learning-based models, nevertheless requiring a huge amount of data is gaining numerous interests because of its strength of learning complex patterns in unstructured data.

In this paper, we aim at taking advantage of deep learning models to predict stock prices. Our work has three contributions:

- We carry out experiments to evaluate 6 deep learning models.
- We conduct experiments on various stock prices (i.e. 5 five sectors of 4 stock prices) of S&P 500 databases.

[1] https://www.investopedia.com/terms/f/fundamentalanalysis.asp

- We contribute some insights that might help us to deepen our understanding on how to use deep learning models.

The paper is organized as follows. Section II provides the related works. Section III presents 6 deep learning models. Data and the measurements are presented in Section IV. Then, the experimental results are provided in Section V. Finally, we conclude in Section VI.

II. Related Works

The strength of deep learning models is able to find hidden features (or patterns) through a self-learning mechanism. The challenge of these models is to require a massive amount of data. However, we can collect lots of data on the stock markets easily. In this part, we will provide some works which are related to exploiting deep learning models for predicting the stock market.

To predict stock behavior, by collecting Google domestic trends as indicators. Xiong et al. [4] compare a traditional model for time series data, namely the GARCH (Generalized Autoregressive Conditional Heteroskedasticity) [5], and a state-of-the-art model for dealing with long-term dependencies, namely LSTM (Long Short-Term Memory) [6]. The results show that LSTM is superior to GARCH. Fischer and Krauss [7] deploy LSTM to the stocks of the S&P 500 from 1992 until 2015. Interestingly, LSTM performs better than a random forest, a standard deep neural net, and a logistic regression. Yu and Yan [8] combine LSTM with the time series phase-space reconstruction (PSR) method to predict stock prices. The experiments performed on six stock indices for various markets (S&P 500, DJIA, N 225, HSI, CSI 300, and ChiNext) demonstrate that the proposed model outperforms ARIMA, SVR (Support Vector Regressor), MLP (Multilayer Perception), and LSTM without PSR. Furthermore, Karmiani et al. [9] compare several predictive algorithms for the stock market. The results conducted on nine technology companies show that LSTM is a better choice compared to SVM, Backpropagation, and Kalman filter algorithms. To predict the index price of the stock market on the next day, Gao et al. [10] evaluate MLP, LSTM, CNN (Convolutional Neural Network), and attention-based neural network. To do so, S&P 500 index (the most developed market), CSI 300 index (the less developed market), and Nikkei 225 index (the developing market) are considered. The authors show that the attention-based model is insignificant better than the three other models.

It is reasonable to understand that LSTM has gained significant popularity in stock prediction because LSTM is a state-of-the-art model to deal with sequential data. Surprisingly, to the best of our knowledge, there are a few attempts to employ deep learning models, including LSTM, in financial markets. Hence, it inspires us to experimentally conduct the comparison between deep learning models in terms of predicting stock prices.

III. Models

Here we present 6 typical deep learning models which are used to conduct the experiment results in Section V.

A. Multilayer Perceptron (MLP)

An artificial Neural Network (ANN) is a computational model which imitates the way of information processing by neurons in the human brain by making the right connections among nodes. An ANN consists of three parts—a layer of input nodes, layers of hidden nodes, and finally a layer of output nodes (see Figure A1). Each layer consists of a group of multiple neurons/nodes which are connected to others via weighted links. A multilayer perceptron (MLP) is a fully connected network of ANN. The input layer processes the input data and passes it to the hidden layer, then the hidden layer handles the outcome from the previous layer and passes it to the next layer, finally, the output layer produces the result. The learning capability of MLPs takes place by way of adjusting weight values. Thanks to their well-organized structures, MLPs are able to successfully map nonlinear input to output by automatically extracting subtle patterns and multiple features from a large dataset through each layer (see Figure 1).

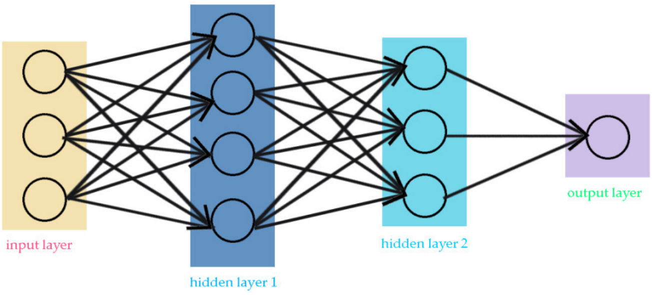


Fig. 1. An illustration of a simple MLP which consists of two consecutive hidden layers located between the input layer and the output layer. every neuron is interconnected and assigned weights (represented by arrows). Each neuron learns/adjusts its weights through its inputs and desired outputs.

B. Convolutional Neural Network (CNN)

The breakthrough in MLP happened when one variant of the models—Deep Convolutional Neural Networks—was ranked first in the ImageNet Large Scale Visual Recognition Challenge after approaching human performance in image classification [11]. Therefrom, Convolutional Neural Networks (CNN) and their descendants have been approaching superhuman performance in a wide range of domains, including pattern recognition, natural language processing, video processing, speech recognition, and time-series forecasting [12].

C. Recurrent Neural Network (RNN)

Notwithstanding the extreme success, CNNs face a big issue: they are not able to cope with time series and sequential data. In order to deal with this issue, Recurrent Neural Network [13]—one of the deep neural network models—was introduced. Due to the flexibility in architecture, computational power, and the rich inherent memory through feedback, RNNs have a wide range of applications in sequential data, including machine translation [14], speech recognition [15], time series anomaly detection [16], and time series forecasting [17].

TABLE I
List of twenty stocks divided into five groups. The symbol of stocks and their market cap are shown. T stands for trillion and B denotes billion.

Communication Services		Consumer Cyclical		Energy		Healthcare		Technology	
Stock	Market Cap	Stock	Market Cap	Stock	Market Cap	Stock	Market Cap	Stock	Market Cap
GOOG	1.397T	AMZN	1.285T	XOM	395.717B	UNH	484.229B	AAPL	2.483T
META	426.084B	TSLA	859.887B	CVX	307.551B	JNJ	429.03B	MSFT	1.889T
DIS	200.701B	HD	296.411B	SHEL	195.57B	LLY	291.533B	TSM	410.636B
VZ	172.608B	BABA	234.856B	TTE	132.514B	PFE	256.821B	NVDA	335.548B

D. Long Short-Term Memory (LSTM)

Nevertheless, RNNs have two serious problems: vanishing/exploding gradients and learning with long term dependencies [6]. First, like any CNNs, RNNs are able to learn by adjusting weight values. Technically, weight values are updated through the backpropagation algorithm. Unfortunately, in the case of RNNs with a large number of hidden layers, performing the backpropagation algorithm leads to vanishing gradient (i.e, exponential decrease) and exploding gradient (i.e., exponential growth) problems because a large number of derivatives have to be multiplied. Second, RNNs are only able to capture short-term dependencies in sequences. Therefore, we need a new type of architecture design to deal with the two above problems affecting RNNs. Fortunately, Long Short-Term Memory (LSTM) was introduced by Hochreiter and Schmidhuber [6]. LSTM is a type of RNN and is specifically designed to deal with longer dependencies in sequences [18] and reduce the exploding gradients. Unlike RNNs, instead of adding regular neural units (i.e., hidden layers), LSTM adds memory blocks. A common LSTM memory block consists of a cell state and three gates—an input gate, a forget gate, and an output gate.

E. Gated Recurrent Unit (GRU)

To deal with the vanishing gradient and exploding gradient problems, Cho et al. [19] introduced the gated recurrent unit (GRU) which is a variation of LSTM in terms of the architecture designed without the output gate. While LSTM consists of three gates (i.e., input, forget, and output gates), GRU comprises two gates: the reset gate and the update gate. Therefore, GRU has fewer parameters. The advantage of GRN over LSTM is better computation, although GRN obtains comparable results in many cases compared to LSTM [20].

F. Autoencoder (AE)

Autoencoder (AE) is a special artificial neural network [21]. AE consists of three parts: 1) encoder that converts the input into the bottleneck; 2) bottleneck that is a compressed representation keeping only the most important information; and 3) decoder that reconstructs the original input from the bottleneck. In fact, AE is able to handle data in which the features are correlations. Hence, dealing with noisy data is an interesting advantage of Autoencoder.

IV. Data and Evaluation

A. Data

In order to evaluate six deep learning models, we perform them on datasets with different characteristics. Particularly, there are five groups of stocks corresponding to five sectors where companies are working. They are Consumer Cyclical, Communication Services, Energy, Healthcare, and Technology. For each group, we select four stocks of influential companies in their sector, list of 20 stocks is shown in Table I.

The historical data of each stock is collect from January 2^{nd} 2015 to August 8^{th} 2022. There are six columns in one dataset: Date, Open, High, Low, Close, Volume. Figure 2 presents data of CVX stock from group Energy, Chervon Corporation engages in integrated energy and chemicals operations worldwide.

B. Evaluation

The performances of six models were evaluated using five statistical indices, i.e., Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Normalized Root Mean Squared Error (NRMSE) and Coefficient of Determination (R2), which can be expressed as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{2}$$

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}| \tag{3}$$

$$NRMSE = \frac{100 \times RMSE}{y_{max} - y_{min}} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{5}$$

where $\hat{y}_i$ is the i^{th} predicted daily closing price, y_i is the i^{th} objected daily closing price, $\bar{y}$ denote the mean observed daily closing price and n is the total number of data samples evaluated.

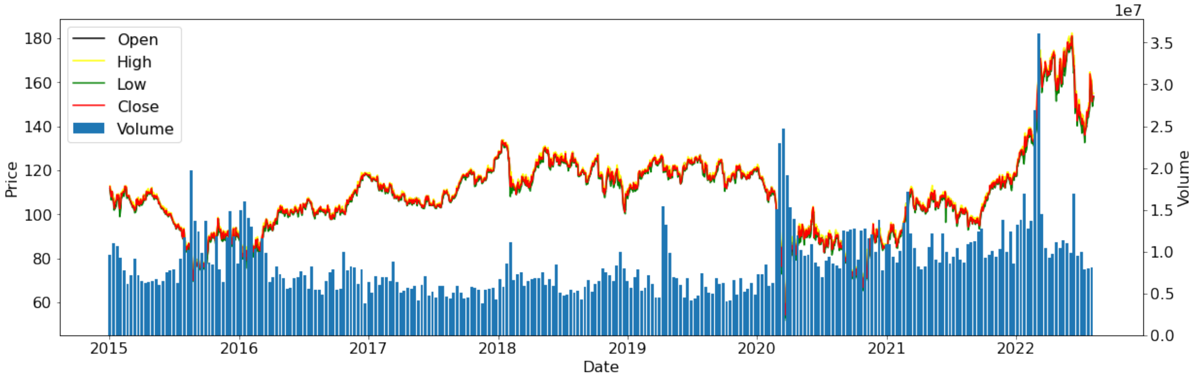


Fig. 2. Historical prices of CVX stock collected from January 2^{nd} 2015 to August 8^{th} 2022. The volumes of stock are presented by bar chart and four line charts respectively present Open, High, Low, Close prices of CVX stock.

V. Experimental Results

In this paper, we use Close price of the next day as target variable, while five features including Open, High, Low, Close, Volume are chosen as input. In forecasting stock problems, finding look-back window's size (lags) is an important part. However there isn't a right thesis about choosing the best lags of time for a forecasting problem. Therefore we run experiment to find out the best input size for forecasting stock problems. We run models in five cases with different input sizes consists of 3 days, 1 week, 2 weeks, 3 weeks and 1 month input. Overall, we run experiment for six deep learning models on twenty stocks, the result in one stock is summarized by five cases with different input sizes. The results of experiment allow us to compare performance of models and also select best input to predict stock prices.

TABLE II
Performance results of six models for forecasting closing prices of CVX stock using 3 days of input.

	MAE	RMSE	MAPE	NRMSE	R2
MLP	2.0789	2.8744	0.0158	2.9964	0.9878
RNN	**1.8899**	**2.6486**	**0.0146**	**2.7610**	**0.9897**
LSTM	2.1624	3.1518	0.0167	3.2856	0.9854
GRU	2.0294	*2.8022*	0.0156	*2.9211*	*0.9884*
CNN	2.1854	2.9568	0.0168	3.0823	0.9871
AE	*2.0041*	2.8204	*0.0153*	2.9401	0.9883

Table II presents the performance results between models for predicting CVX stock's price with lag time equal to 3 days. The table shows that forecast results of the RNN model are more accurate than other models. The RNN model has MAE of 1.899, RMSE of 2.649, MAPE of 0.015 and NRMSE of 2.761, which are smaller than all of the other models. For all of six models, the R2 indices have values approximately to 0.99 indicate a good correlation between the observed and forecasted values.

On a different dataset and a different input size, the results of models are changed. The forecast results of six models for UNH stock by using 1 week of input are shown in Table III.

TABLE III
Performance results of six models for forecasting closing prices of UNH stock using 1 week of input.

	MAE	RMSE	MAPE	NRMSE	R2
MLP	13.2710	15.9347	2.8767	7.1885	0.9204
RNN	6.7058	8.6229	1.4866	3.8900	0.9767
LSTM	7.0727	9.3069	1.5539	4.1985	0.9728
GRU	17.2761	19.4373	3.7737	8.7686	0.8816
CNN	**5.3518**	**6.9854**	**1.1976**	**3.1512**	**0.9847**
AE	10.7642	13.2524	2.3403	5.9784	0.9449

By comparing performance results of models, we can see that the CNN model is the best model in this case.

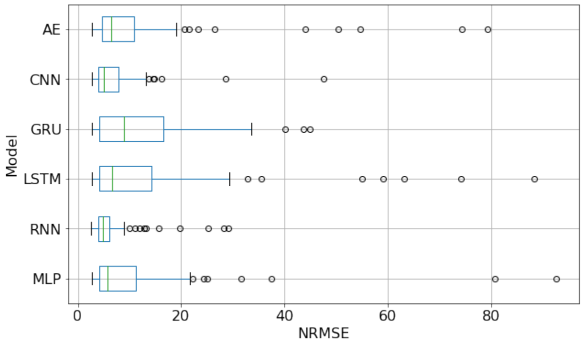


Fig. 3. Boxplot results of six models for predicting stock prices in all cases of experiment using NRMSE indices.

Looking at Table II and III, the difference between MAE and RMSE indices from these tables are quite big. The MAE indices in Table II range from 1.89 to 2.18 and in Table III they have values from 6.71 to 17.28. For RMSE indices, Table II shows results from 2.65 to 3.15 and Table III shows results from 6.99 to 15.93. Look at equation (1) and (2) we can know the reason of these differences. Two equations (1), (2) use the absolute error between the observed and forecasted values, while price of stocks are different, the results on two stocks are also different using MAE and RMSE indices. In order to summarize results from multiple stocks, we use three indices:

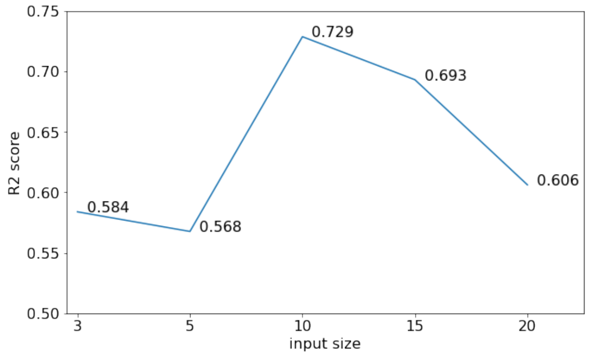


Fig. 4. Results of models in all cases based on the input size. The mean of R^2 indices are reported.

MAPE, NRMSE and R^2. These three indices calculate relative error between real and predicted values, so we can compare the results of models between stocks and discover in a bigger picture.

Figure 3 provides the boxplot of NRMSE results for six models in all cases of experiment. From the boxplot, we can compare the performance of models in forecasting stock problems. Through the median NRMSE of the RNN model is not significantly lower than the CNN model, the width of the box and the values of the right boundaries are significantly smaller in the RNN model than in the other five models, indicating that the RNN model provides an overall lower forecast error.

The R^2 or coefficient of determination, provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the models. The larger R^2 score, the better result and the best result appears when R^2 score equal to 1. In Figure 4, we don't compare the performance of models together but summarize the results from models and group them by each sizes of input. As we can see, changing input size from 3 days to 5 days doesn't improve the results too much. When the input size equal to 10 days e.g. 2 week, the R^2 score is 0.729 and is the largest score in 5 cases of input sizes. Continue increasing input size to 15 and 20 days make the forecast results become worse because the R^2 scores decrease.

From Figure 3 and 4, we can answer the question: which are the best model and the best input size for the predicting stock price problems? Figure 3 shows that the RNN model is the best model overall and Figure 4 tells us that 10 days is the best size of input for these problems. Now we compare models in a smaller case - group of stocks and find the best model for each group. Figure 5 presents the performance of models using MAPE indices. There are 5 groups: Communication Services, Consumer Cyclical, Energy, Healthcare and Technology, for each group we show results of six models. In Energy group, the performance of six models are all good and GRU model has smallest forecast errors. In the 4 other groups, the results of the RNN model and the CNN model outperform the results of the 4 other models. Particularly, the RNN model has best result and the second is the CNN model in Communication Services and Technology group. In Healthcare and Consumer Cyclical group, the RNN model is the second and the CNN model has the best performance.

VI. Conclusions

Through the paper, we have observed that the RNN model performs the most accurate results in terms of predicting stock prices. However, the choice of models depends on the sector we want to predict. If we want to predict the sector of Communication Services and Technology, the RNN model is the best choice. In the sector of Consumer Cyclical and Healthcare, the best model is the CNN model. And we should use the GRU model for predicting stocks in the Energy sector. Unlike many previous studies that demonstrated that the LSTM model is a state-of-the-art model to deal with financial time series data, our results show that the RNN model is the most suitable model for predicting stock prices because of the short temporal dependency of data.

In the future, we are going to exploit several more deep learning models (e.g., Variational Autoencoder - VAE, Generative Adversarial Network - GAN) on further companies.

References

[1] J. Hur, M. Raj, and Y. E. Riyanto, "Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade," *World Development*, vol. 34, no. 10, pp. 1728–1741, 2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305750X06001148

[2] L. Li, Y. Wu, Y. Ou, Q. Li, Y. Zhou, and D. Chen, "Research on machine learning algorithms and feature extraction for time series," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–5.

[3] "Forecasting directional movements of stock prices for intraday trading using lstm and random forests," *Finance Research Letters*, vol. 46, p. 102280, 2022.

[4] R. Xiong, E. P. Nichols, and Y. Shen, "Deep learning stock volatility with google domestic trends," 2015. [Online]. Available: https://arxiv.org/abs/1512.04916

[5] R. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982. [Online]. Available: https://EconPapers.repec.org/RePEc:ecm:emetrp:v:50:y:1982:i:4:p:987-1007

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[7] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221717310652

[8] P. Yu and X. Yan, "Stock price prediction based on deep neural networks," *Neural Comput. Appl.*, vol. 32, no. 6, pp. 1609–1628, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04212-x

[9] D. Karmiani, R. Kazi, A. Nambisan, A. Shah, and V. Kamble, "Comparison of predictive algorithms: Backpropagation, svm, lstm and kalman filter for stock market," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 2019, pp. 228–234.

[10] P. Gao, R. Zhang, and X. Yang, "The application of stock index price prediction with neural network," *Mathematical and Computational Applications*, vol. 25, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2297-8747/25/3/53

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Curran Associates, Inc.

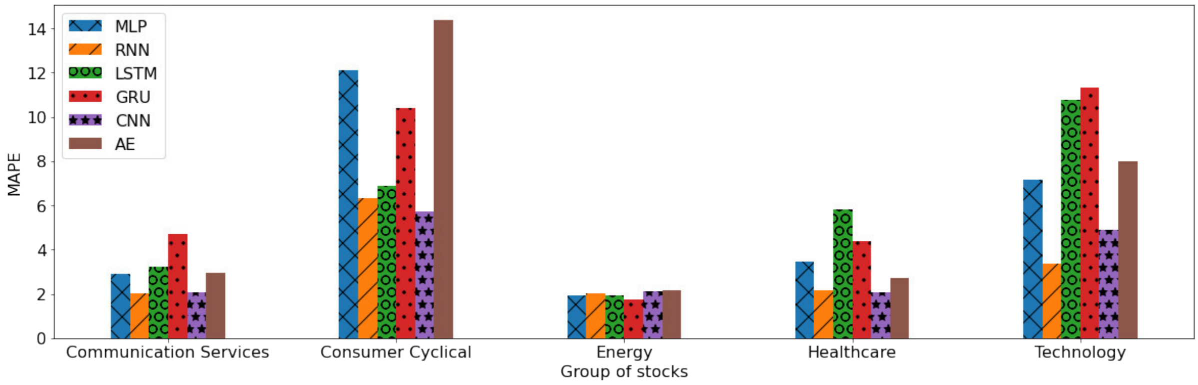


Fig. 5. Performance results of six models divided into five groups. The results are summarized by the mean of MAPE indices.

[12] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, vol. abs/1404.7828, 2014. [Online]. Available: http://arxiv.org/abs/1404.7828

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Representations by Back-Propagating Errors*. Cambridge, MA, USA: MIT Press, 1988, p. 696–699.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *CoRR*, vol. abs/1409.3215, 2014. [Online]. Available: http://arxiv.org/abs/1409.3215

[15] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005, iJCNN 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608005001206

[16] P. Malhotra, L. Vig, G. M. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *ESANN*, 2015.

[17] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207020300996

[18] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technical University Munich, 2008. [Online]. Available: https://d-nb.info/99115827X

[19] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *CoRR*, vol. abs/1409.1259, 2014. [Online]. Available: http://arxiv.org/abs/1409.1259

[20] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[21] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

DOI: 10.15439/2022R11
ISSN 2300-5963 ACSIS, Vol. 33

Utilize Deep learning to increase the performance of a Book recommender system using the Item-based Collaborative Filtering

Cu Nguyen Giap
Economic Information System and E-commerce
Thuongmai University
Hanoi, Vietnam
cunguyengiap@tmu.edu.vn

Le Thi Huyen Dieu
Faculty of Information Technology
FPT University -FPT Polytechnic
Hanoi, Vietnam
dieulth2@fpt.edu.com

Luong Thi Hong Lan
Faculty of Computer Science and Engineering
Thuyloi University
Hanoi, Vietnam
lanlhbk@tlu.edu.vn

Tran Thi Ngan, Tran Manh Tuan
Faculty of Computer Science and Engineering
Thuyloi University
Hanoi, Vietnam
{ngantt, tmtuan}@tlu.edu.vn

***Abstract*—Item-based Collaborative Filtering is a common and efficient approach for recommendation problems. In this study, we have investigated the power of deep learning in textual feature extraction and applied this advantage to a high-performance item-based collaborative filtering recommender system. The proposed approach has been experienced on book datasets added by texts collected from famous book review sites. The experiment proves that the proposed model has better performance thanks to the contribution of the new item profile process method based on Deep Learning.**

Index Terms—*Item-based Collaborative Filtering, Deep Learning, Textual embedded recommender system.*

I. Introduction

With the rapidly growing day-to-day data over the internet, item-based collaborative filtering (IBCF) is one of the best-known and most extensive recommendation system (RS) techniques. They provide accurate predictions when sufficient data is provided, as this technique is based on a computation of similarity among items and rating prediction using similar items.

One of the recommended application areas is the book market. Users are increasingly inclined to buy books on websites, and the role of book recommendation systems has become more important. In fact, books are a special product that differs from most of the products on e-commerce. Particularly, the book's content is often very long, and the user cannot grasp it in advance, but other information, such as title and type, etc., is not good enough to reflect the book. One of the successful RS algorithms is Item-based collaborative filtering which was introduced by Amazon and was widely applied to e-commerce [1-5]. This study wants to build an RS with book data in a more appropriate way by an increment of the IBCF algorithm.

In IBCF, several correlation-based similarity measures have been traditionally used to generate a top-k list of recommended items: Chigozirim Ajaegbu [6] combined three traditional similarity metrics, Cosine, Pearson, and Adjusted cosine, in an IBCF algorithm for Movielens RS. Pradeep Kumar Singh et al. [7] assume the similarity between users is essential to finding the similar neighbors of a target item. Monika Verma and Arpana Rawal [8] proposed an RS to predict popular books based on calculating the item-item likeness with the cosine/correlation coefficient in the attributes of the book database.

Although existing RS successfully produces decent suggestions, they still undergo some limitations, such as accuracy and scalability. In the last few years, a deep learning (DL)-based approach has been applied widely to enhance the quality of recommendations. Almaghrabi et al. [9] suggested a novel system based on DL-based augmentation for forecasting user ratings for various online databases: movies, music, and book collections. In [10-11], the authors supply a thorough review of DL-based recommendation approaches to clarify and suggest beginner researchers interested in the subject.

The RS uses an IBCF approach, for each unknown rated item i_t, it has to estimate the similarity of all items to identify the k-closest items of i_t, and then the rating of i_t is estimated from the rating of k-closest items. The common IBCF RS calculates the likeness of items based on an item profile that is normally a vector of binary, number, nominal or categorical elements.

However, in some domains, item descriptions are important to assess the similarity of products. These descriptions are textual variables, such as the book RS; assessing the book description as indispensable information is crucial. This problem carries out an important issue: extracting item textual profile features.

DL technology has been utilized in natural language processing problems for a long time, with many remarkable achievements. During the development of DL, this technology has shown a high potential application for textual feature extraction. This study was conducted to utilize the stacked denoise autoencoder and text summarize technique for textual feature extraction. And then, the extracted feature is used to improve the performance of a Book RS that uses IBCF.

To assist the proposed model, the data of the Book-crossing dataset is used to estimate the implementation of the proposed model and resemble it to the other algorithms' results. However, it must be noticed that the original Book-crossing dataset does not contain any description of the book; it has basic book information such as name, author, etc. Therefore, this dataset is added by book descriptions from famous book review websites.

This article is organized into four details. The first part briefly presents the introduction and motivation of the problem and our main contributions. The second part describes the typical approach of item-based collaborative filtering and the proposed increment of this algorithm. The next part introduces the way to collect relevant data and our experiments. Finally, we are concerned about the significant results of the proposed approach and future works.

II. RECOMMENDER SYSTEM AND PROPOSAL SOLUTION BASED ON DEEP LEARNING TECHNIQUES

A. *Item-based Collaborative Filtering*

Sarwar introduced IBCF in [12]; the IBCF model is depicted in Figure 1.

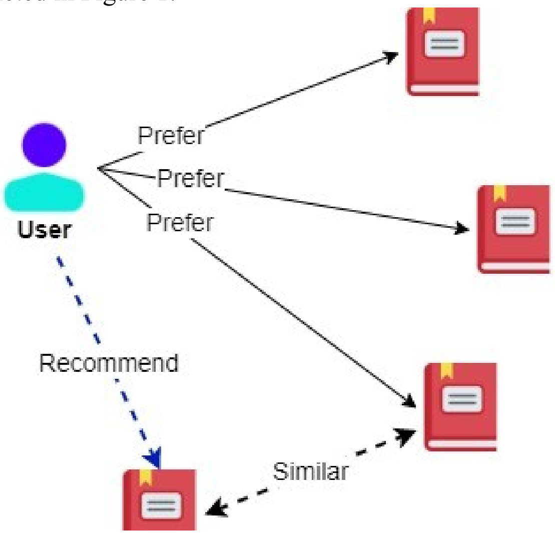


Fig. 1. Item-based collaborative filtering

In the IBCF, for each unknown rated item i_t, it has to estimate the similarity of this item with other rated items by the same user, and then a set of k-closest items of i_t is identified from the rated items. The IBCF predicts the rating of i_t by aggregating the rating of the k-closest items. The common Item-based collaborative filtering RS calculates the similarity of items based on an item profile that is normally a vector of binary, number, nominal or categorical elements. The common similarity estimators use Pearson similarity, Cosine similarity, and their adjustments.

Pearson similarity:

$$Pearson(i,j)=\frac{\sum_{u\in T_i\cap T_j}\left(r_{u,i}-\bar{r}_i\right)\left(r_{u,j}-\bar{r}_j\right)}{\sqrt{\left(\sum_{u\in T_i\cap T_j}\left(r_{u,i}-\bar{r}_i\right)^2\right)\left(\sum_{u\in T_i\cap T_j}\left(r_{u,j}-\bar{r}_j\right)^2\right)}} \quad (1)$$

Cosine similarity:

$$\text{Cosine}(i,j)=\frac{\sum_{u\in T_i\cap T_j} r_{u,i}\times r_{u,j}}{\sqrt{\left(\sum_{u\in T_i\cap T_j} r_{u,i}^2\right)\left(\sum_{u\in T_i\cap T_j} r_{u,j}^2\right)}} \quad (2)$$

The similarity of the items can be estimated by the rating of other users $\left(T_i \cap T_j\right)$ given on these items only, however, the item profile can supply more information to estimate better. This study concerns this problem and uses Deep learning technology to identify better similar items for a target item.

B. *Proposal model*

The purpose of the proposal is to utilize Deep learning techniques to extract features of the textual variables in item attributes. These features will be used in the item similarity estimator of the IBCF algorithm. Remarkably, the original data of books, including book titles are used to search and seek book reviews from public websites that contain interesting information about the book. It is added to the similarity estimator naturally.

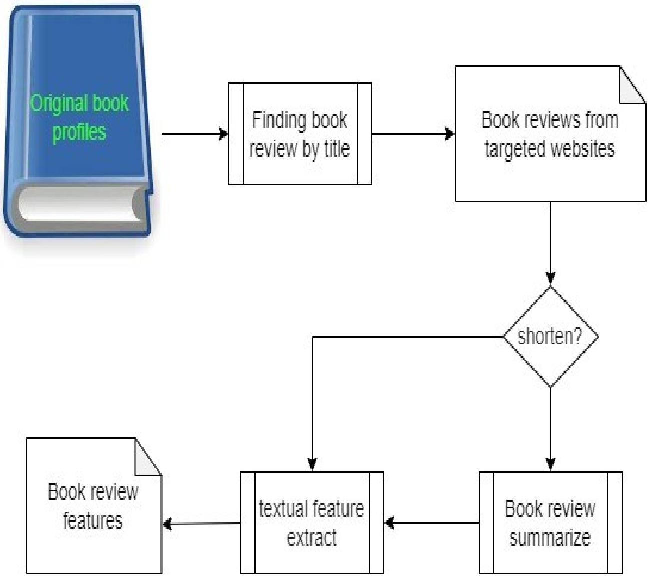


Fig. 2. The process of book review feature extraction

In fact, a book review contains a lot of exciting information that readers are concerned about and need when choosing a suitable book. Many websites publish book reviews, but in this study, we are concerned about the reviews presented in two critical aspects: literary reviews and commercial reviews. Therefore, several book-sale websites and some literary analysis websites will be targeted.

The proposed Textual embedding Item-based collaborative Filtering RS (TE_IbCF RS) consists of two phases. The detail of the proposed model is described as follows:

Phase 1: predict unknown item rating

a. Estimate item similarity to the rated item by users rating, extracted textual features, and selected original item attributes using Cosine similarity.

b. Select the k-nearest item set.

c. Predict unknown item rating by the following formula [13]:

$$r_{u,i}=\bar{r}_i+\frac{\sum_{j\in N_i}(r_{u,j}-\bar{r}_j)*sim(i,j)}{\sum_{j\in N_i} sim(i,j)} \quad (3)$$

Where $\bar{r}_i$ is the average rating of item i.

Phase 2: Recommendation by the top-N algorithm.

Notice that: in Phase 1, given these similarity measures are not enough to evaluate the appropriateness of the recommendations. The data is the most influential factor in finding the best suggestions; you should provide actual input data to get the accurate output. For book items in the knowledge base, we use the textual summary to describe the textual details, which usually gives the book's main topics.

C. Deep learning for textual embedding RS

To predict the interest of buyers and recommend books to them accordingly, two kinds of DL are applied for textual feature extraction.

- Text summarization technique
- Stacked denoising autoencoders

The text summarization technique is applied to shorten the review of the book if the user selects this option. The Stacked denoising autoencoders are used to extract textual features from book reviews. It has the function of dimensional reduction also.

<u>*Text summarization techniques*</u>:

The text summarization technique is used for massive applications that deal with a vast amount of strings and long text [14]. In the designed Book recommender system, the book description contains noise and irrelevant information due to the bias of the writer. Therefore a text summarization technique based on the Term frequency-inverse Document frequency (TF-IDF) has been applied to create a shortened version of the book description. This algorithm includes the following steps:

Step 1: word segmentation.

Step 2: remove stop-words

Step 3: word score estimation

Step 4: Sentence value calculation and filtering (using a user-defined threshold).

For example, a book description is "*Readers beware. The brilliant, breathtaking conclusion to J.K. Rowling's spellbinding series is not for the faint of heart--such revelations, battles, and betrayals await in Harry Potter and the Deathly Hallows that no fan will make it to the end unscathed. Luckily, Rowling has prepped loyal readers for the end of her series by doling out increasingly dark and dangerous tales of magic and mystery, shot through with lessons about honor and contempt, love and loss, and right and wrong. Fear not, you will find no spoilers in our review--to tell the plot would ruin the journey and Harry Potter and the Deathly Hallows is an odyssey the likes of which Rowling's fans have not yet seen and are not likely to forget. But we would be remiss if we did not offer one small suggestion before you embark on your final adventure with Harry--bring plenty of tissues*."

The above description is summarized as "*The brilliant, breathtaking conclusion to J.K. Rowling's spellbinding series is not for the faint of heart--such revelations, battles, and betrayals await in Harry Potter and the Deathly Hallows that no fan will make it to the end unscathed. Fear not, you will find no spoilers in our review--to tell the plot would ruin the journey and Harry Potter and the Deathly Hallows is an odyssey the likes of which Rowling's fans have not yet seen and are not likely to forget.*"

The advantage of the text summarization technique is that it maintains the major meaning of the original review but is present in a much shorter form. Therefore, the vector representation form of a book review will be shorter also, and it causes cheaper time costs for other processes followed.

<u>*Stacked Denoising Autoencoders*</u>:

A stacked denoising autoencoder (SDA) is a deep learning technology that can reduce the impact of noise in the input and extract the potential feature without the requirement of predefined labels [15-17]. In this study, a specific architecture of SDA was used to extract the useful part of the textual descriptions of books in a dataset. The SDA architecture is depicted in figure 3.

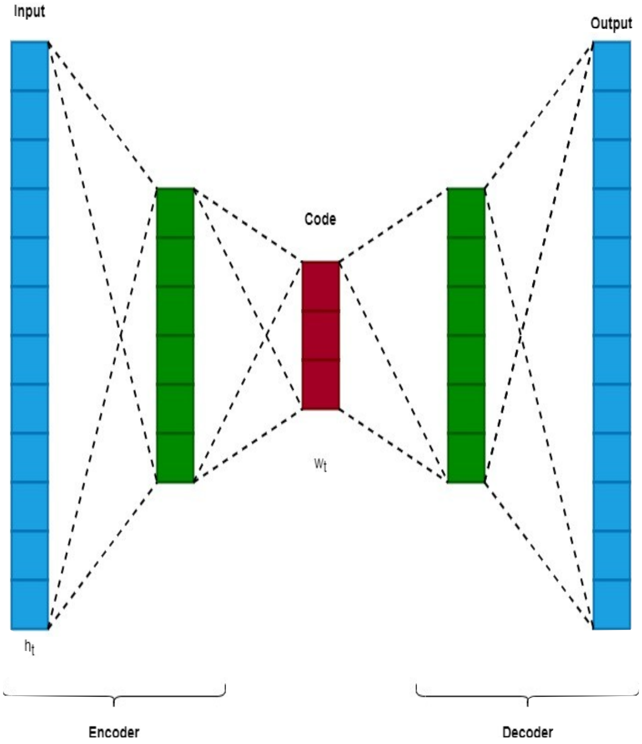


Fig. 3. Stacked denoising autoencoders

For textual feature extraction, the input layer is passed by the vector representation of text. This study uses the N-gram transformation of the original text. The code layer has a size suited to the expected reduction purpose. In this study, the standard size of the encoded feature is set at 10% of the input string. The number of neurons in the input and output layers is equal and depends on the book description's length.

For implementation, the book review is transformed by a 2-gram (or bi-gram) model that is suitable for the short and middle texts [18-19]. In short, texts are split into two-word sequences, and the vector representations of texts are conducted from the frequency of split strings. The detail of the bi-gram model was presented in the study of Shinsuke et al. [20].

Besides, when creating a vector representation of a text by bi-gram, the study has to apply a word segmentation algorithm and a stop-word removal algorithm. Because all book reviews are presented in English, this study uses the standard stop-word library integrated into the Sklearn library.

III. EXPERIMENTS

In this section, we estimate our proposed model for book RS scenarios. The practical outcomes indicate proof of meaningful advancement over competitive baselines.

A. Data construction

In order to experience the proposed model of RS, a well-known dataset of books, Book-crossing, is used. However, the public Book-crossing dataset does not contain any description of the book; therefore, it must be added by book reviews that are collected from two famous sites: amazon.com and goodread.com. The former has commercial reviews of books, and the latter supplies literary studies.

Within the time constraint, the testing dataset is a subset of Book-crossing data, in which 1000 books were selected randomly, and the dataset contains a relative of 2x1000 book reviews. Their reviews present the more interesting information about the book than the original data.

TABLE I. ATTRIBUTES OF BOOK IN THE DATA SET

Attribute	Type
book_id	Original attribute
best_book_id	Original attribute
work_id	Original attribute
books_count	Original attribute
authors	Original attribute
original_publication_year	Original attribute
original_title	Original attribute
title	Original attribute
language_code	Original attribute
average_rating	Original attribute
ratings_count	Original attribute
work_ratings_count	Original attribute
work_text_reviews_count	Original attribute
Amazon	Collected from amazon.com
rating_Amazon	Collected from amazon.com
Goodread	Collected from goodread.com
rating_Goodread	Collected from goodread.com

In fact, the book profile's quality is thickened and more useful for readers with the above extra reviews.

Besides, it should be noted that the books' rates are in the interval [0-10]. The data is really spare, and ratings of an item are highly dispersed. This is a challenge for any RS algorithm.

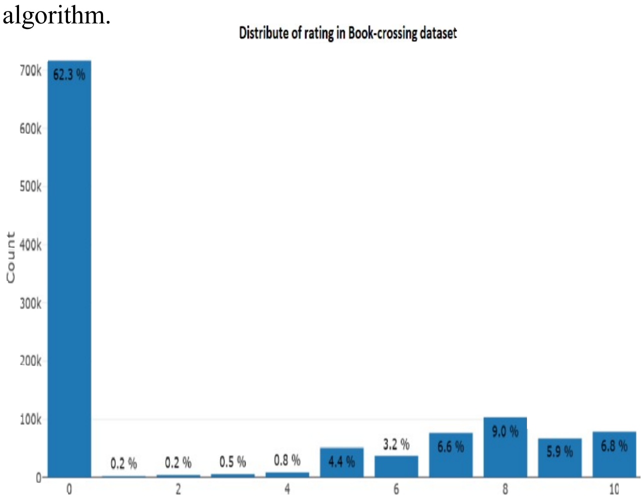


Fig. 4. Distribution of rating

B. *RS performance metrics*

In order to experiment, the Root Mean Square Error (RMSE) was utilized to estimate the usefulness of the proposed RS. The following formula calculates the measurement metric:

Error metric:

$$RMSE(I) = \sqrt{\frac{\sum_{i \subset I} \left(r_i - \hat{r}\right)^2}{|I|}} \qquad (4)$$

Where I is a set of items recommended to user u , and r_i, $\hat{r}_i$ are predicted user ratings and real ratings relatively.

RMSE is a typical measurement metric that is used to compare implemented algorithms in recommender systems. However, it should be noticed that the error metric is not all of the RS's performance. In the case of the textual embedded RS for books, it can rationally explain the recommendation based on book reviews, and this feature might be helpful in real applications indeed.

C. *Experimental results*

Algorithm performances

The Python programming language implemented the proposed model and used some valuable libraries. The stacked denoising autoencoder was implemented with the Theano platform and used the BaseEstimator from the "sklearn" library. Theano is a powerful open-source platform that can bring high performance to the implementation. The other task that conducted text transformation also used the "sklearn" library, mainly it used the CountVectorizer from the feature-extraction package and the standard stop-word library. The 2-gram model was chosen in the current stage of this study.

We have experienced the efficiency of the proposed models by running them with standard parameters picked from the literature review. These parameters might not be optimum for book review data; however, they were recommended for text processing in other studies. Finding the optimal set of parameters is planned for the next stage of this research.

Afterward, the RMSE results are shown in Table 2 respectively.

TABLE II. THE RMSE OF EXPERIENCED ALGORITHMS

Algorithm	RMSE
IF_CF	3.053
TE_IbCF without Text summarization	***2.717***
TE_IbCF with Text summarization	2.809

The proposed models using textual embedded features within the testing dataset can improve Book Recommender System performance. The **TE_IbCF** without Text summarization has the highest version, but the text summarization technique does not reduce the RS performance much. It means that the text summarization algorithm would be applied to increase the time performance; meanwhile, the recommender system can preserve its accuracy.

The book-crossing dataset has ratings of books that are remarkably dispersed. User reviews of books are very inconsistent; most books have the smallest rating of 0 and the highest rating of 10 (the minimum and maximum values in the measurement scale). Moreover, the bias of the 0 ratings is obvious. This point brings a complex challenge to building a proper automatic recommender system. Therefore, the improvement brought by the proposed textual embedded feature extraction model is valuable and should be applied in reality.

Finally, it should be noticed that the result in the sub-dataset of Book-crossing that includes 1000 books shows that the textual embedded feature conducted from book reviews can improve the performance of Item-based collaborative filtering. However, this testing data is not the full Book-crossing dataset that contains 10000 books; therefore, the above result can not be compared to other public studies on RS for books. The data set will be enriched in the future by adding reviews of the remaining books.

IV. CONCLUSIONS

In summary, this study has considered several improvements to the IBCF algorithm for RS in the book. The main contribution of this study is proposing a specific approach to the recommender system using embedded text. Adding the book reviews into the book similarity estimator process of an RS and using the deep learning techniques to extract meaningful textual features increases the performance of the IBCF recommender system. The stacked denoising autoencoder was implemented for textual feature extraction tasks and showed high potential. Although this research stage uses only the parameters suggested in the literature review, the results are also very encouraging.

The experiment was done on the well-known book dataset, the Book-crossing subset. The results gave excellent proof of the high efficiency of deep learning techniques, including text summarization technique, bigram algorithm, and stacked denoising autoencoders. These techniques extract and present meaningful features of book reviews, and therefore the book similarity can be estimated better. The new model also has a new advantage thanks to the rational recommendation given to the reader because it concerns the book reviews of both literary analysis websites and commercial websites.

Another minor contribution of this study is the construction of an extended dataset of the Book-crossing dataset with book reviews from literary and commercial websites. Currently, no systems allow a comprehensive and automatic collection of book reviews; thus, the book reviews are collected and matched manually. amazon.com and goodread.com are among the top five famous book review websites on the globe. It guarantees the quality of book reviews and book datasets also.

However, the narrow size of the testing dataset is a limitation of this study. The study will be expanded in the future by using a new vector representation of text generated by the Bag of Word algorithm. This approach was mentioned as a high-potential technique in text processing problems. The issue of figuring out the best parameters remains as future work also.

ACKNOWLEDGMENT

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2020.11.

REFERENCES

[1] Comparison of user-based and item-based collaborative filtering. [Online; accessed 17-August-2019]. https://medium.com/@wwwbbb8510/comparison-of-user-based-and-item-based/-collaborative-filtering-f58a1c8a3f1d

[2] Sarwar B, Karypis G, Konstan J, Riedl J (2001) Itembased collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, ACM, pp 285–295

[3] Yang Z, Wu B, Zheng K, Wang X, Lei L (2016) A survey of collaborative filtering-based recommender systems for mobile internet applications. IEEE Access 4:3273–3287

[4] Linden G, Jacobi J, Benson E (2001) Collaborative recommendations using item-to-item similarity mappings. [Google Patents]

[5] Deshpande M, Karypis G (2004) Item-based top-n recommendation algorithms. ACM Trans Inf Syst 22(1):143–177

[6] Ajaegbu, C. (2021). An optimized item-based collaborative filtering algorithm. Journal of ambient intelligence and humanized computing, 12(12), 10629-10636.

[7] Singh, P. K., Sinha, S., & Choudhury, P. (2022). An improved item-based collaborative filtering using a modified Bhattacharyya coefficient and user–user similarity as weight. Knowledge and Information Systems, 64(3), 665-701.

[8] Verma, M., & Rawal, A. (2022). An Enhanced Item-Based Collaborative Filtering Approach for Book Recommender System Design. ECS Transactions, 107(1), 15439.

[9] Almaghrabi, M., & Chetty, G. (2018, December). A deep learning based collaborative neural network framework for recommender system. In 2018 International Conference on Machine Learning and Data Engineering (iCMLDE) (pp. 121-127). IEEE.

[10] Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52(1), 1-37.

[11] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR), 52(1), 1-38.

[12] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

[13] Wu J, Chen L, Feng Y, Zheng Z, Zhou M, Wu Z (2013) Predicting quality of service for selection by neighborhoodbased collaborative filtering. IEEE Trans Systems, Man, and Cybernetics: Systems 43(2):428–439

[14] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268.

[15] Sagha, H., Cummins, N., & Schuller, B. (2017). Stacked denoising autoencoders for sentiment analysis: a review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 7(5), e1212.

[16] Liang, J., & Kelly, K. (2021). Training stacked denoising autoencoders for representation learning. *arXiv preprint arXiv:2102.08012*.

[17] Tong, H., Liu, B., & Wang, S. (2018). Software defect prediction using stacked denoising autoencoders and two-stage ensemble learning. *Information and Software Technology*, *96*, 94-111.

[18] Patil, A., & Mahalle, P. (2021). A Building Topical 2-Gram Model: Discovering and Visualizing the Topics Using Frequent Pattern Mining. In Proceeding of First Doctoral Symposium on Natural Computing Research (pp. 11-21). Springer, Singapore.

[19] Elghannam, F. (2021). Text representation and classification based on bi-gram alphabet. *Journal of King Saud University-Computer and Information Sciences*, *33*(2), 235-242.

[20] Mori, S., Nishimura, M., & Itoh, N. (1998). Word clustering for a word bi-gram model. In *ICSLP*.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 115–120 DOI: 10.15439/2022R15 ISSN 2300-5963 ACSIS, Vol. 33

On the Performance of Cellular Networks with Adaptive Modulation and Energy Harvesting—A Stochastic Geometry Approach

Nhut-Minh Ho
Post and Telecommunications Institute of Technology
Ho Chi Minh City, Vietnam
minhho@ptithcm.edu.vn

Sang Quang Nguyen (*)
Ho Chi Minh City University of Transport
Ho Chi Minh City, Vietnam
sang.nguyen@ut.edu.vn
(*) Corresponding author

Thanh-Toan Phan
Post and Telecommunications Institute of Technology
Ho Chi Minh City, Vietnam
phanthanhtoan@ptithcm.edu.vn

Lam-Thanh Tu
Ton Duc Thang University
Ho Chi Minh City, Vietnam
tulamthanh@tdtu.edu.vn

***Abstract*— The performance of ultra-dense cellular networks considering both adaptive discrete modulation (ADM) and energy harvesting (EH) is investigated. Particularly, mobile users (MUs) are charged its battery from all ambient radio frequency (RF) signals. Based on the amount of harvested energy as well as the channel conditions, MU will actively choose an appropriate modulation scheme that not only maximizes the rate but also satisfies the quality-of-service (QoS). Moreover, we consider the spatial-temporal correlation at the signal-to-interference-plus-noise ratios (SINRs) of base stations (BSs) which are totally different from work in the literature. Several important metrics are investigated such as, occurrence probabilities of different modulation schemes (Poc), coverage probability (Pcov), and achievable spectral efficiency (ASE). Finally, the results highlight the superiority of the proposed scheme compared to the conventional fixed modulation.**

***Index Terms*— Adaptive Modulation, Energy Harvesting, Performance Analysis, Stochastic Geometry**

I. Introduction

With the growing in the number of wirelessly connected devices, the demand for the power to feed such ultra-dense networks has attracted researchers [1]. Moreover, another issue of the ultra-dense networks is how to increase the spectral efficiency (SE) since the network is in the interference-limited regime rather than the noise-limited or neither of them. To overcome these unavoidable issues in the ultra-dense networks some advanced technologies are employed in the literature, for example, the cognitive radio networks (CRNs) [2], [3] that is proved to be an effective way to improve the SE, the satellite communications that can provide service at every corner of the earth without creating interference at the terrestrial [4], the interference alignment technique that takes the advantages of the multiple antennae at both transmitter and receiver to suppress the interference [5] and the adaptive discrete modulation (ADM) that significantly scales up the average rate thus improving the SE [6]. Particularly, a properly modulation scheme is chosen at the transmitter relying on the practical channel conditions to maximize the average rate thus facilitating the ASE. Nonetheless, these above techniques generally improve the SE of the wireless networks while the enhancement of energy efficiency (EE) is minor compared with the SE. Fortunately, another advanced technique called energy harvesting (EH) [7] has recently attracted many researchers since it allows the low energy devices (LEDs) to harvest energy from the surrounding radio frequency (RF) thus providing a solid response to the question *How to feed an ultra-dense network*. As a consequence, in the present work, we explore the performance of the ultra-dense cellular network by considering both ADM and EH techniques. Before going to discuss the novelties as well as the contributions of the considered networks, state-of-the-art of ADM, EH, and other advanced techniques are first visited.

A. State-of-the-art

The performance of the key metrics in modern wireless networks such as outage probability (OP), coverage probability (Pcov), ergodic capacity, spectral efficiency, and energy efficiency was studied extensively in [8]–[16]. The cooperative unmanned aerial vehicles (UAV) non-orthogonal multiple access (NOMA)-based in short packet communications (SPC) was studied in [8]. Particularly, the average end-to-end (e2e) block-error rates (BLERs) were derived in the closed-form expression. A novel expression of closed-form of the Pcov in the long-range (LoRa) networks was given in [9]. A tractable framework of both Pcov and ergodic capacity in cellular networks based on tools from stochastic geometry (SG) was derived by Andrews and other authors in [10]. Meanwhile, the closed-form expression of ergodic capacity in multi-hop decode and forward (DF) was given in [11].

Duy and others in [12] derived outage probability expressions in cognitive radio networks. Moreover, the transmit power of the secondary transmitter was computed in [13]. On the other hand, the performance of the combination of optical communications and wireless communications was conducted in [14] where RF signals are seamlessly transmitted through both fiber and mmWave. The impact of hardware impairment in UAV-NOMA-based systems was investigated in [15]. Their outcomes showed that the performance of the near user outperforms its counterpart under the influences of hardware impairments. A recently proposed definition of the Pcov that takes into account the correlation of the signal-to-interference ratios (SIRs) and signal-to-noise ratios (SNRs) at end-devices of the LoRa was comprehensively studied [16]. They also point out the influences of density of EDs on the performance of the coverage probability and spectral efficiency under different transmit power regions.

These previous works simply focus on the performance of wireless communications without considering adaptive discrete modulation and energy harvesting. These techniques, in fact, was studied separately in [17]–[24]. More precisely, authors in [17] investigated performance of the CRNs with EH relay assistance. They showed that their proposed EH relaying outperforms the conventional underlay CRN scheme in the OP. Thanh and others investigated the performance of the SWIPT-enabled networks [18], [19]. They proved that low energy devices can be operated without a battery provided that the number of transmit antennae at BS goes to infinity. The cooperative NOMA-based with SWIPT over Nakagami-m fading channels was addressed in [20]. The self-energy recycling (SER) scheme was studied in [21] under partial and full relay selection. To be more specific, they proposed to utilize the self-interference at the full-duplex relay to recharge its battery and forward information to the destination. Besides, Tung in [20] studied the combination of SWIPT and NOMA over Nakagami-m channels. The potential of the application of mmWave combined with SWIPT was thoroughly studied in [22]. Meanwhile, the adaptive multicast streaming service with ADM in cellular networks was addressed in [23] and the average and potential throughput of the adaptive modulation was derived in [24].

B. Principal novelties and contributions

Apart from works in the literature, we comprehensively investigate the performance of the mobile networks with the combination of adaptive discrete modulation and energy harvesting at the system level. More precisely, we summarize the core novelties and contributions as follows:

- We consider the Poisson point process (PPP) to model the randomness of mobile users (MUs) and base stations (BSs).
- We employ the power beacon scheme to charge the battery of MUs instead of using the SWIPT protocol and the bounded path-loss model is used as well.
- We adopt adaptive modulation to take the benefits of the channel state information (CSI) at MUs.
- Compared with state-of-the-art, we consider the spatial-temporal correlation at the signal-to-interference-plus-noise ratios (SINRs) which is extremely complicated and nontrivial.
- We investigate the performance of three key metrics i.e., achievable spectral efficiency, occurrence probabilities of modulation schemes, and coverage probability.
- Numerical results based on the Monte Carlo method is yield highlight the advantages of the considered networks compared with a fixed modulation scheme.

II. System Model

A. Cellular Networks Modeling

Considering a uplink cellular networks that both base stations and mobile users are followed by a homogeneous PPP (HPPP) denoted by Ξ_{BS} and Ξ_{MU} with corresponding densities υ_{BS} and υ_{MU}. Additionally, without loss of generality, we consider the fully-loaded scenario, i.e., $\upsilon_{\mathrm{MU}} \gg \upsilon_{\mathrm{BS}}$. It is noted that the most general case where the ratio of the density of BSs and MUs is a random number is studied in [25]. The performance is taken place at the typical cell where the serving BS denoted by BS_0 and the typical user denoted by MU_0 is situated at the origin of the 2-D plane. The results measure the link between BS_0 and MU_0 can be applied to all other transmission link thank to the Palm theory [26].

B. Transmission Procedure

The whole transmission is taken place in two phases. In the first phase, the BS_0 broadcasts high-power radio frequency signals to charge the battery of all MUs associated with it. In the second phase, the MU_0 which is selected to send data to the BS_0[1] using the energy harvesting in the first phase as well as on the instantaneous channel state information which is measured via the pilot signals in the first phase. It is emphasized that we do not consider the simultaneous information and power transmission since it scarifies parts of resources (time or frequency) for powering the battery. Moroever, the BS acts as the power beacon station both two phases. We consider the orthogonal resource allocation inside each cell. Hence, intra-interference in each cell is not taken into consideration, nonetheless, other-cell interference, obviously, is appeared.

C. Channel Modelling

Considering an arbitrary connection from a generic BS to a generic MU, it experiences small-scale fading and large-scale path loss. Shadowing is left for future work like work in the literature [27].

1) Small-scale fading: Considering c_m as the small-scale fading for a transmission between the BSs and MUs followed by a Rayleigh distribution. As a consequence, $|c_m|^2$ is an exponential distribution have mean $\lambda_m = 1, \forall m$ (w ithout loss of generality) denoting the channel gain.

[1] In the present paper, we do not focus on the user selection and leave it for future work.

2) Large-scale path-loss: The large-scale path-loss of an arbitrary link between BSs and MUs is evaluated as

$$W_m = W_0 \max\left\{d_m, 1\right\}^{\alpha}. \tag{1}$$

Here d_m is the transmission distance between BSs and MUs; $\alpha > 2$ and $W_0 = (4\pi f_c/c)^2$ are the path-loss and exponent. f_c (in Hz) denotes carrier frequency, $c = 3 \times 10^8$ (in meters per second) is the light speed.

Remark 1: Direct inspection (1), we can see that the adopted large-scale path-loss overcomes the unavoidable issue of the popular unbounded path-loss model that the received power approaches infinity when the transmission distance between indistinguishable [28].

D. Cell Association Criterion

Each MU is handled by a BS having the shortest distance to it. As the fully-loaded scenario is considered, all BSs are active. The serving BS, BS_0, is then formulated as

$$\mathrm{BS}_0 = \underset{m \in \Xi_{\mathrm{BS}}}{\operatorname{argmin}} \left\{W_m\right\}. \tag{2}$$

E. Adaptive Discrete Modulation

Adaptive discrete modulation is a mature technology to boost spectral efficiency especially when the channel is favorable. Particularly, based on the pilot signals at the first phase, the MU_0 exactly estimates the CSI between him and BS_0. Moreover, since we consider the fully-loaded scenario that is the worst case where all BSs are active, the MUs are then estimated at the interference at the BS_0. Based on the available CSI, the MU_0, then, selects the most appropriate modulation scheme that satisfies the quality-of-service (QoS) requirement. Let us first divide the whole range of SINR at BS_0 into $\mathcal{R} \in \mathbb{N}$ separate regions and the border of each interval is denoted by γ_{R}^{e}, $e \in \{0, \ldots, \mathcal{R}\}$, as follows:

$$0 = \gamma_{\mathrm{R}}^{0} < \gamma_{\mathrm{R}}^{1} < \ldots < \gamma_{\mathrm{R}}^{e} < \ldots < \gamma_{\mathrm{R}}^{\mathcal{R}} = +\infty \tag{3}$$

The popular rectangular M-QAM modulation is adopted in the present paper. More precisely, $M_o = 2^o$-QAM modulation is chosen provided that the SINR lies into the interval $\left[\gamma_{\mathrm{R}}^{o}, \gamma_{\mathrm{R}}^{o+1}\right)$, $o \in \{1, \ldots, \mathcal{R}-1\}$. Additionally, in case the SINR is too small or it is in the region $\left[\gamma_{\mathrm{R}}^{0} = 0, \gamma_{\mathrm{R}}^{1}\right)$, MU_0 will immediately halt the transmission and the outage event will appear.

In order to find out the border of all regions, we examine bit error rate (BER) as a measured metric. Particularly, BER of the M-QAM employing Gray coding over additive white Gaussian channel (AWGN) is deployed:

$$\begin{aligned} \mathrm{BER}_R =& \beta_o Q\left(\sqrt{\chi_o \gamma_{\mathrm{R}}^{o}}\right) \\ \Rightarrow \gamma_{\mathrm{R}}^{o} =& \frac{1}{\chi_o}\left[Q^{-1}\left(\frac{\mathrm{BER}_{\mathrm{R}}}{\beta_o}\right)\right]^2, \quad o \in \{1, \ldots, \mathcal{R}-1\}, \end{aligned} \tag{4}$$

where BER_R is the intended bit error rate threshold; $Q(.)$ is the Gaussian Q function and $Q^{-1}(.)$ is the inverse Gaussian Q function.

$$\beta_o = \begin{cases} 1 & l_o = 1,2 \\ 4/l_o & l_o \geq 3 \end{cases},$$
$$\chi_o = \begin{cases} 2/l_o & l_o = 1,2 \\ 3/\left(2^{l_o} - 1\right) & l_o \geq 3 \end{cases} \tag{5}$$

where $l_o = \log_2(M_o)$.

F. Transmit Power at MUs

In this work, the harvest-then-transmit protocol is employed at the MUs that harvested energy amount in the first phase denoted by $\mathcal{E}$ (in Joule) is computed as follows:

$$\mathcal{E} = \epsilon T\left(P_{\mathrm{tx}} \sum_{i \in \Xi_{\mathrm{BS}}} \frac{\left|c^{(i)}\right|^2}{W^{(i)}}\right)/2. \tag{6}$$

Here $\epsilon \in [0,1]$ denote the coefficient of energy conversion; T is the whole transmission block and is equal to 1 for simplicity; P_{tx} is the BSs transmit power. It should be noted that the AWGN noise in (6) is ignored since it is too tiny compared to power of interference. From (6), the MUs transmit power is computed as follows:

$$P_0 = \frac{\epsilon}{2}\left(P_{\mathrm{tx}} \sum_{i \in \Xi_{\mathrm{BS}}} \frac{\left|c^{(i)}\right|^2}{W^{(i)}}\right). \tag{7}$$

G. Signal-to-Interference-Plus-Noise Ratio

The SINR at BS_0 is given as

$$\gamma_0 = \frac{P_0 \frac{\left|c^{(0)}\right|^2}{W^{(0)}}}{P_{\mathrm{MU}} \sum\limits_{j \in \Xi_{\mathrm{MU}}} \frac{\left|c^{(j)}\right|^2}{W^{(j)}} + \sigma_0^2}, \tag{8}$$

where $\sigma_0^2 = -174 + \mathrm{NF} + 10\log_{10}(\mathrm{BW})$ (in dBm) is the BS_0 noise variance; NF (in [dB]) is the noise figure at the ED; BW is the bandwidth; $P_{\mathrm{MU}} = \mathbb{E}\left\{P_0\right\}$ is the transmit power of the MUs from other cell. In this work, for simplicity, we assume that P_{MU} is the average over spatial (MUs locations) and temporal (fading) of the whole networks. The spatial-temporary correlation at the transmit power of the MUs will be left for future work. $\mathbb{E}\{.\}$ is the expectation operator. $c^{(s)}$, $W^{(s)}$, $s \in \{0, j\}$, are the path-loss and fading between the BS_s and MU_0.

Remark 2: Inspecting (8), we observe that although the spatial-temporal correlation at P_{MU} do not take into consideration, these correlations at P_0 still hold. As a consequence, the considered networks are extremely complicated and novel compared with work in the literature [29].

III. Performance Metrics

We investigate three key metrics in the present work, i.e., the average achievable spectral efficiency, the coverage probability, and the occurrence probabilities of different modulation schemes (Poc). More precisely, Pcov is the probability that measures the number of outstanding transmissions out of the total transmission while Poc measures the percentage of each

TABLE I: Simulation parameters

Parameters [Unit]	Values
$R_{\text{BS}} = \frac{1}{\sqrt{\pi \upsilon_{\text{BS}}}}$ [m]	150
$R_{\text{MU}} = \frac{1}{\sqrt{\pi \upsilon_{\text{MU}}}}$ [m]	50
P_{tx} [dBm]	30
BER_R	10^{-3}
BW [kHz]	200
ϵ	0.5
$\mathcal{R}$	5
NF [dB]	5
α	2.3
f_c [GHz]	0.9

scheme appearance out of all transmissions. The ASE provides the average achievable spectral efficiency.

A. Coverage Probability

The Pcov under the considered network refers to the probability that the SINR of the BS_0 is larger than γ_{R}^1 and is computed as

$$\text{Pcov} = \Pr\left\{\gamma_{\Delta} \geq \gamma_{\text{R}}^1\right\}. \tag{9}$$

B. Occurrence probabilities of each modulation scheme

The probability that the MU_0 transmits at o modulation scheme defines as the occurrence probabilities denoted by Ψ_o and is computed as

$$\Psi_o = \Pr\left\{\gamma_{\text{R}}^o \leq \gamma_0 \leq \gamma_{\text{R}}^{o+1}\right\}, \quad o \in \{1, \ldots, \mathcal{R}-1\}. \tag{10}$$

It is obvious that the summation of Ψ_o, $o \in \{0, \ldots, \mathcal{R}-1\}$ is equal to one. In particular, we have

$$\sum_{o=0}^{\mathcal{R}-1} \Psi_o = 1. \tag{11}$$

C. Average Achievable Spectral Efficiency (ASE)

Under the adaptive modulation systems, the average achievable spectral efficiency (in bits/s/Hz) is computed by summing all the spectral efficiency of each region that is the multiplication of the Poc and its corresponding bit and is given as follows [6]:

$$\text{ASE} = \sum_{o=1}^{\mathcal{R}-1} l_o \Psi_o. \tag{12}$$

IV. Numerical Results

Numerical results via the Monte-Carlo simulation are employed in this section to evaluate the performance of the considered metrics, i.e., Pcov, Poc, and ASE. If there is no specific noticed in each figure, a set of simulation parameters is given in Table I. Here, R_{BS} and R_{MU} are the average cell radius of the BS and the average distance between MUs, respectively. Five levels of the M-QAM modulation are considered [6], specifically, no transmission, BPSK, QPSK, 16-QAM, and 64-QAM, respectively. It is certain that an arbitrary modulation level and/or different modulation schemes such as MPSK, differential modulation, etc. can also be effortlessly applied. Fig. 1 depicts the behaviors of the Pcov regarding the BER threshold. We observe that if the QoS decreases, the Pcov improves, and increasing P_{tx} is beneficial for the Pcov. Particularly, Pcov enhances over 0.1 if P_{tx} increases from 30 to 40 dBm at BER_R = 0.01.

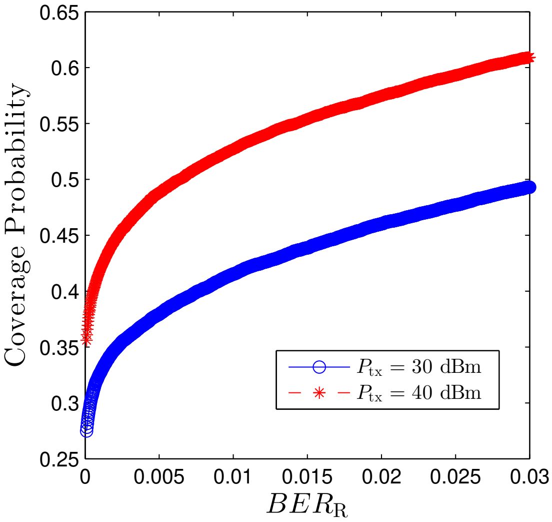


Fig. 1. Pcov as a function of BER threshold, BER_R with various values of P_{tx}.

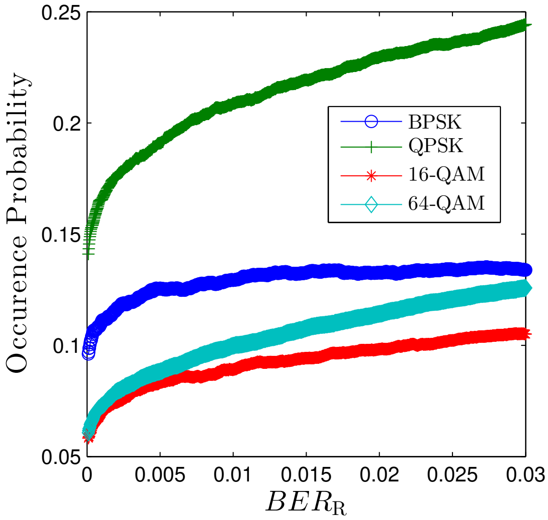


Fig. 2. Occurrence probability vs. BER threshold, BER_R.

Fig. 2 shows the behaviors of the occurrence probability as a function of BER_R. We see that under the current setup, the QPSK has the highest probability while the 16-QAM is the smallest one, and BPSK and 64-QAM are at the 2nd and 3rd position. Again, increasing BER_R will scale up the Poc like the Pcov. However, the increasing pace is different between these schemes. More precisely, the QPSK is again

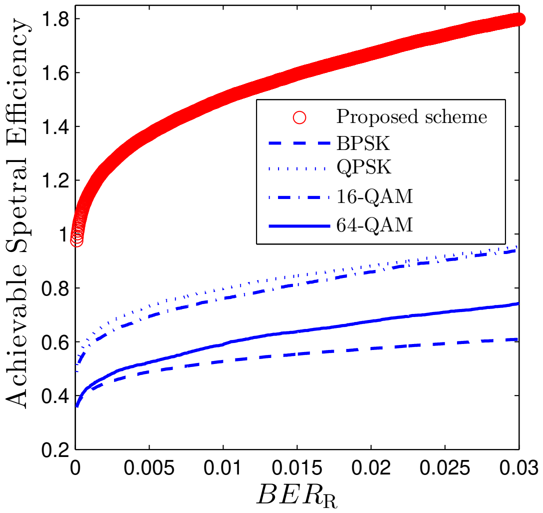


Fig. 3. ASE vs. BER threshold, BER_R.

favorable for the increase of BER_R while the BPSK is almost stable when $\text{BER}_\text{R} \geq 0.01$. The performance of the ASE with respect to BER_R is given in Fig. 3. It proves the superiority of the ADM compared with the fixed modulation. Particularly, we experience a major divergence between the proposed scheme vs. all other modulations. Among all fixed modulations, the QPSK achieves the best performance as like in Fig. 2. Nonetheless, different from Fig. 2, the 16-QAM is the 2nd best while in Fig. 2, the 2nd best is the BPSK modulation.

V. Conclusion

The performance of uplink cellular networks considering both adaptive modulation and energy harvesting was investigated in the present paper. Particularly, three vital metrics, i.e., Pcov, Poc, and ASE are addressed under the impact of both spatial and temporary correlation. Simulation results unveiled that adaptive modulation significantly outperforms fixed modulation schemes in terms of spectral efficiency. This work can be enhanced in many ways. One of these possible ways is to deploy diversity techniques at the BSs and/or MUs to further enhance the Pcov and ASE [30]. Additionally, the application of reconfigurable intelligent surfaces (RIS) and NOMA into the ultra-dense cellular networks is also promising [31]–[33]. The heterogeneous networks architecture where cellular networks co-exist with other networks such as cognitive radio networks, low power wide area networks (LoRa, SigFox), and device-to-device communications also scales up the system spectral efficiency [34], [35]. Facilitating the system performance by shortening the transmission distance such as multi-hop communications, and cooperative communications is a potential solution too. Finally, the advantages of machine learning and deep learning can not be ignored in order to significantly enhance the system performance as well [36], [37].

References

[1] Ericsson Mobility Report, Ericsson, 2017.

[2] N. Q. Sang *et. al.*, "Cognitive multihop cluster-based transmission under interference constraint," *2014 IEEE ISCE*, 2014, pp. 1-3.

[3] L. -T. Tu *et. al.*, "Performance Evaluation of Incremental Relaying in Underlay Cognitive Radio Networks with Imperfect CSI," *2020 IEEE ICCE*, 2021, pp. 472-477.

[4] T. N. Nguyen *et. al.*, "Outage Performance of Satellite Terrestrial Full-Duplex Relaying Networks with Co-Channel Interference," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1478-1482.

[5] J. Song *et. al.*, "On the feasibility of interference alignment in ultra-dense millimeter-wave cellular networks," *IEEE Asilomar 2016*, 2016, pp. 1176-1180.

[6] M. S. Alouini *et. al.*, "Adaptive modulation over nakagami fading channels," *Wireless Pers. Commun.*, vol. 13, pp. 119–143, May. 2000.

[7] J. Huang, C.-C. Xing, and C. Wang, "Simultaneous wireless information and power transfer: Technologies, applications, and research challenges," *IEEE Commun. Magazine*, vol. 55, pp. 26–32, Nov. 2017.

[8] P. N. Son *et. al.*, "Short packet communications for cooperative UAV-NOMA-based IoT systems with SIC imperfections," *Compt. Netw.*, 2022.

[9] L.-T. Tu *et. al.*, "A New Closed-Form Expression of the Coverage Probability for Different QoS in LoRa Networks," *IEE ICC 2020.*

[10] J. G. Andrews *et. al.*, "A Tractable Approach to Coverage and Rate in Cellular Networks," *IEEE Trans. Commun.*, vol. 59, no. 11, 2011.

[11] T. L. Thanh *et. al.*, "Capacity analysis of multi-hop decode-and-forward over Rician fading channels," *IEEE ComManTel 2014*, 2014, pp. 134-139.

[12] T. T. Duy *et. al.*, "Performance Enhancement for Multihop Cognitive DF and AF Relaying Protocols under Joint Impact of Interference and Hardware Noises: NOMA for Primary Network and Best-Path Selection for Secondary Network," *Wireless Commun. Mobile Comput.*, vol. 2021.

[13] L. -T. Tu *et. al.*, "Broadcasting in Cognitive Radio Networks: A Fountain Codes Approach," *IEEE Trans. Veh. Techno.*, *Early Access*, 2022.

[14] T. L. Thanh *et. al.*, "10-Gb/s wireless signal transmission over a seamless IM/DD fiber-MMW system at 92.5 GHz," *IEEE ICC 2015*, 2015, pp. 1364-1369.

[15] C. H. Duc *et. al.*, "Performance Evaluation of UAV-Based NOMA Networks with Hardware Impairment," *Electronics*, vol. 11, no. 1, p. 94, Dec. 2021.

[16] T. T. Lam *et. al.*, "On the Spectral Efficiency of LoRa Networks: Performance Analysis, Trends and Optimal Points of Operation," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2788-2804, April 2022

[17] P. N. Son *et. al.*, "Performance analysis of underlay cooperative cognitive full-duplex networks with energy-harvesting relay", *Compt. Netw.*, vol. 122, p. 9-19, Jun. 2018.

[18] L.-T. Tu *et. al.*, "System-level analysis of swipt mimo cellular networks," *IEEE Commun. Lett.*, vol. 20, pp. 2011–2014, Oct. 2016.

[19] L.-T. Tu *et. al.*, "System-level analysis of receiver diversity in swipt-enabled cellular networks," *IEEE/KICS J. Commun. Netw.*, vol. 18, pp. 926–937, Dec. 2016.

[20] T.-T. Nguyen *et. al.*, "Evaluation of Full-Duplex SWIPT Cooperative NOMA-Based IoT Relay Networks over Nakagami-m Fading Channels," *Sensors*, vol. 22, no. 5, p. 1974, Mar. 2022.

[21] T. N. Nguyen *et. al.*, "Partial and Full Relay Selection Algorithms for AF Multi-Relay Full-Duplex Networks With Self-Energy Recycling in Non-Identically Distributed Fading Channels," *IEEE Trans. Veh. Techno.*, vol. 71, no. 6, pp. 6173-6188, Jun. 2022.

[22] T. T. Lam *et. al.*, "Analysis of millimeter wave cellular networks with simultaneous wireless information and power transfer," *IEEE SigTelCom 2017*, 2017, pp. 39-43.

[23] M. Li *et. al.*, "Performance analysis of adaptive multicast streaming services in wireless cellular networks," *IEEE Trans. Mobile Comput.*, vol. 18, pp. 2616–2630, Nov. 2019.

[24] X. Qiu *et. al.*, "On the performance of adaptive modulation in cellular systems," *IEEE Trans. Commun.*, vol. 47, pp. 884–895, Jun. 1999.

[25] T. T. Lam *et. al.*, "On the Energy Efficiency of Heterogeneous Cellular Networks With Renewable Energy Sources—A Stochastic Geometry Framework," *IEEE Trans. Wireless Commun.*, 2020.

[26] F. Baccelli and B. Blaszczyszyn, Stochastic Geometry and Wireless Networks, Part I: Theory, Now Publishers, Sep. 2009.

[27] M. Di Renzo *et. al.*, "System-Level Modeling and Optimization of the Energy Efficiency in Cellular Networks – A Stochastic Geometry Framework", *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2539 - 2556, Apr. 2018.

[28] J. Song *et. al.*, "Bounded Path-Loss Model for UAV-to-UAV Communications," *ICWMC 2021*, pp. 20-21.

[29] H. H. Yang *et. al.*, "Spatio-Temporal Analysis for SINR Coverage in Small Cell Networks," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5520-5531, Aug. 2019.

[30] P. T. Tin *et. al.*, "Rateless Codes-Based Secure Communication Employing Transmit Antenna Selection and Harvest-To-Jam under Joint Effect of Interference and Hardware Impairments," *Entropy*, vol. 21, no. 7, p. 700, Jul. 2019.

[31] T. -T. T. Dao *et. al.*, "Performance Evaluation of Downlink Multiple Users NOMA-Enable UAV-Aided Communication Systems Over Nakagami-m Fading Environments," *IEEE Access*, vol. 9, pp. 151641-151653, 2021.

[32] T. V. Chien *et. al.*, "Coverage Probability and Ergodic Capacity of Intelligent Reflecting Surface-Enhanced Communication Systems," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 69-73, Jan. 2021.

[33] T. V. Chien *et. al.*, "Outage Probability Analysis of IRS-Assisted Systems Under Spatially Correlated Channels," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1815-1819, Aug. 2021.

[34] L. -T. Tu *et. al.*, "Energy Efficiency Analysis of LoRa Networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1881-1885, Sept. 2021

[35] L. -T. Tu *et. al.*, "Coverage Probability and Spectral Efficiency Analysis of Multi-Gateway Downlink LoRa Networks," *IEEE ICC 2022* 2022, pp. 1-6.

[36] L. -T. Tu *et. al.*, "Energy Efficiency Optimization in LoRa Networks—A Deep Learning Approach," *IEEE Trans. Intel. Transport. Syst.*, *Early Access*, 2022.

[37] A. Zappone *et. al.*, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization," *IEEE Veh. Technol. Mag.*, vol. 14, pp. 60–69, Sep. 2019.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 121–127 DOI: 10.15439/2022R32 ISSN 2300-5963 ACSIS, Vol. 33

Optimizing Reference Model for Disturbance Rejection Controller for 3-DoF Robot Manipulator

1st Duy Hoang
Intenational School
Vietnam National University
Hanoi, Vietnam
hoangduy@vnuis.edu.vn

2nd Dung Do Manh
Intenational School
Vietnam National University
Hanoi, Vietnam
dungdm@vnuis.edu.vn

3rd Thai Dinh Kim
Intenational School
Vietnam National University
Hanoi, Vietnam
thaikd@vnu.edu.vn

4th Hai Xuan Le
Intenational School
Vietnam National University
Hanoi, Vietnam
hailx@vnu.edu.vn

5th Thanh Ngoc Pham
Intenational School
Vietnam National University
Hanoi, Vietnam
pnthanh@vnu.edu.vn

6th Nguyen Huu Thai
Faculty of Electrical Engineering
Vinh University of Technology Education
Vinh, Vietnam
thainguyenktv@gmail.com

7th Long Mau Nguyen
Ha Tinh College of Technology
Hatinh, Vietnam
maulongskv@gmail.com

***Abstract*—In this article, an method is proposed combining optimal control for linear system and disturbances observer to control a 3 degree of freedom (3DoF) robot manipulator. By making the tracking error follow a given stable linear reference model through the observer, an optimal controller LQR will be designed to solve the optimization problem for the reference system, thereby leading to good control quality for the original system. The effectiveness of the method is shown through simulation results performed on Matlab/Simulink.**

***Index Terms*—Adaptive control, Optimal control, Observer, Disturbance rejection, Manipulator.**

I. Introduction

Robotic manipulators are widely used in industry and they play an important role in replacing humans in performing complex jobs that require high accuracy as well as high working frequency [1]. When the dynamic model of the system is known, the model based controller such as PD, PID and some improvements [2], [3], [4] are preferred approaches because of their simplicity and their capacity to apply in practice but the noise resistance of these controllers is not really good. For this reason, the nonlinear control methods including the sliding moed controller [5], [6], [7], the backstepping controller as well as the controllers that combine the nonlinear control method with the linear control method [8], [9], [10] was studied thereby improving the control quality under the influence of external disturbances. Although the stability of system is guaranteed by Lyapunov criterion, the control quality is still limited by the dependence of the control signal on the system model. In the case that the dynamic model of robot is not sufficiently exact, the controllers that rely on the model will no longer retain their effectiveness. Therefore, adaptive control methods have been proposed for the purpose of steer the states of robot to follow trajectory signal without information of system's dynamic model. The common utensils used for adaptive controller design include fuzzy systems [11] and neural networks [12] due to their property of being able to approximate any non-linear functions. By adjusting control parameters according to the change of system and working environment, the adaptive controller is capable of improving control quality in a variety of operating situations. The disturbance observer [13], [14], [15] is also an effective approach to eliminate the influence of external disturbances as well as system's uncertain parameters. The observer can be used just to estimate the disturbances [14] but if only the affect of disturbances is removed, the uncertain parameters can still persist and degrade the control quality. Therefore, by combining all the uncertainties of system and disturbances into an unique total uncertain component and remove it during the operation of system, the observers [13], [15] provide better capacity to deal with the change of uncertain parameters. Besides, the optimal control problem is also a requirement and there are many approaches to solve this problem [16], [17].

The optimal control problem for nonlinear systems in general is still a relatively complex and challenging topic. Algorithms that successfully solve the optimal control problem for linear systems do not seem to be directly applicable to nonlinear systems. For this reason, an approach to solve this problem can be mentioned is to make the tracking error of nominal nonlinear robot system to follow a reference linear model and then apply known optimal control algorithms to this reference model. Inspired by [13], the following paper will present an optimal control method for the robotic system based on the removal of non-linearity and make tracking error converged to zero according to the reference model. The proposed controller will be applied on 3-DoF robot manipulator which has been studied in [14]. However, unlike [14] where the observer is utilized to approximate the disturbances, our proposed observer will be used for linearization purpose by estimate and eliminate the total uncertain component. The selection of the reference model plays an important role to the control quality and an effective reference model can be determined through optimal control methods.

II. Optimal control problem for linear system

Consider a linear system:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{1}$$

where $\underline{x}$ is the state vector of system, $\underline{u}$ is the control signal vector, matrix A and B are constant matrices. The optimal control problem for a linear system is to design a linear state feedback controller $\underline{u} = -R\underline{x}$ so that the following cost function reaches the minimum value:

$$J = \frac{1}{2}\int_0^t \left(\underline{x}^T C\underline{x} + \underline{u}^T D\underline{u}\right) d\tau \tag{2}$$

with $C = C^T \geq 0$ and $D = D^T > 0$. From [18], the problem of finding the optimal control signal $\underline{u}$ is equivalent to finding a positive definite symmetry solution L^* of the Riccati equation

$$A^T L + LA + C - LBD^{-1}B^T L = 0 \tag{3}$$

Then, the coefficient matrix of the optimal controller R^* is calculated by

$$R^* = D^{-1}B^T L^* \tag{4}$$

However, finding the exact solution of (3) is relatively difficult and complicated. Therefore, the more commonly used method is approximating the solution of (3) instead of directly solving the Riccati equation. An approximate solution method proposed by Kleiman in [18] goes through the following steps:

- Determine R_0 to be the matrix of the state feedback controller so that the system is stable. If A has made the system stable, then R_0 can be chosen including all zero elements.
- Solving the Lyapunov equation

$$\left(A - BR_k\right)^T L_k + L_k\left(A - BR_k\right) = -C - R_k^T DR_k \tag{5}$$

To find the solution for L_k with $k = 0, 1, ...$

- Calculate R_{k+1} from L_k using the formula

$$R_{k+1} = D^{-1}B^T L_k \tag{6}$$

Repeat the second and third steps of algorithm until the error satisfies the condition $L_{k+1} - L_k < \epsilon$ for a given arbitrarily small ϵ. With Kleiman's algorithm, it can be proved that the larger the number of iterations, the closer the solution found from the algorithm is to the exact solution of (3), meaning that $\lim\limits_{k\to\infty} L_k = L^*$ then the coefficient matrix of the controller will as close to the optimal coefficient matrix $\lim\limits_{k\to\infty} R_k = R^*$. The proof has been presented in [18] and we obtain the parameter matrix for the optimal controller LQR.

III. Controller design

Since the model of robot manipulator in general and 3-DoF manipulator in particular is nonlinear, it is relatively difficult to directly apply the design of the optimal state feedback controller. Therefore, the Generalized Proportional Integral Observe is used to estimate all the disturbances affections along with the non-linearity of the system, thereby bringing the tracking errors to a linear reference model. From the obtained linear system, the optimal control algorithm will be applied to improve the control quality.

A. Model of 3-DoF Robot Manipulator

The model of 3-DoF manipulator is shown in Fig. 1 with 3 states are 3 rotating joint.

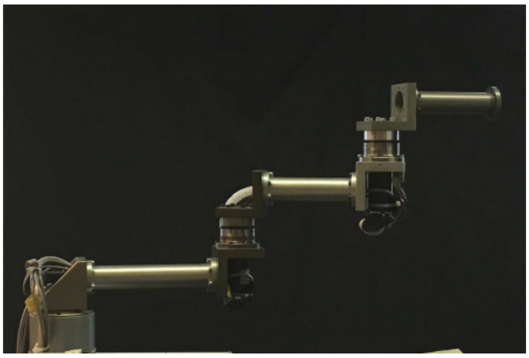


Fig. 1. Model of 3-DoF Robot Manipulator [14]

The dynamic model of 3-DoF robot can be described in Euler-Lagrange form with the following structure

$$M\left(\underline{q}\right)\ddot{\underline{q}} + \underline{\eta}\left(\underline{q}, \dot{\underline{q}}\right) = \underline{\tau} + \underline{\tau}_{ext} \tag{7}$$

where $\underline{q} \in \mathbb{R}^{3\times1}$ is the vector of movable joint variables, $\dot{\underline{q}}$ and $\ddot{\underline{q}}$ are respectively the first and the second order derivatives of $\underline{q}$, $\underline{\tau} \in \mathbb{R}^{3\times1}$ is the vector of the control signals and the input disturbances vector is denoted by $\underline{\tau}_{ext}$. Matrix $M \in \mathbb{R}^{3\times3}$ is the inertia matrix of the system (M is symmetric and positively definite), $\underline{\eta} \in \mathbb{R}^{3\times1}$ is the lumped vector of centripetal, coriolis and gravitational component. The elements of M and $\underline{\eta}$ will be detailed in Section IV.

B. Linearization Observer

The linearization process is performed based on the elimination of the affect of external disturbances and non-linearity in the system. First, the dynamic model of 3-DoF Robot is represented as

$$\ddot{\underline{q}} = M^{-1}(\underline{\tau} + \underline{\tau}_{ext} - \underline{\eta}) \tag{8}$$

with M and η are are shorthand for $M\left(\underline{q}, \dot{\underline{q}}\right)\ddot{\underline{q}}$ and $\underline{\eta}\left(\underline{q}, \dot{\underline{q}}\right)$. From (8), we have

$$\ddot{\underline{r}} - \ddot{\underline{q}} = -M^{-1}\underline{\tau} + \ddot{\underline{r}} - M^{-1}(\underline{\tau}_{ext} - \underline{\eta}) \tag{9}$$

which lead to

$$\ddot{\underline{\xi}} = K_1\underline{\xi} + K_2\dot{\underline{\xi}} - M^{-1}\underline{\tau} + \underline{f} \tag{10}$$

where $\ddot{\underline{r}}$ is the second order derivative of the reference signal $\underline{r}$, vector $\underline{\xi} = \underline{r} - \underline{q}$ is the tracking error, $\dot{\underline{\xi}}$ and $\ddot{\underline{\xi}}$ are respectively the first and the second order derivatives of ξ, K_1 and K_2 are two 3×3 square parameter matrix which are arbitrarily chosen. The influence of disturbances and non-linearity in the system is represented by total uncertain vector $\underline{f}$

$$\underline{f} = \ddot{\underline{r}} - M^{-1}(\underline{\tau}_{ext} - \underline{\eta}) - K_1\underline{\xi} - K_2\dot{\underline{\xi}} \tag{11}$$

The appearance of $\underline{f}$ in (10) is the cause of the difficulty in controller design because of its uncertainty. Therefore, the observer is proposed with purpose of approximating the value of $\underline{f}$ thereby eliminating the influence of this component on the system. Assume that $\underline{f}$ can be approximated by Taylor expansion with a sufficiently large number of degrees m with m is integer and $\underline{f}^m = \underline{0}$. Let $\underline{\xi}_1 = \underline{\xi}$, $\underline{\xi}_2 = \dot{\underline{\xi}}$, $\underline{\delta}_k = \underline{f}^{(k-1)}$ with $k = 1, 2, ..., m+1$ and denote the corresponding observed values are $\hat{\underline{\xi}}_1$, $\hat{\underline{\xi}}_2$, $\hat{\underline{\delta}}_k$, the structure of the observer is described as follow

$$\begin{cases} \dot{\hat{\underline{\xi}}}_1 = \hat{\underline{\xi}}_2 + \lambda_{m+1}(\underline{\xi}_1 - \hat{\underline{\xi}}_1) \\ \dot{\hat{\underline{\xi}}}_2 = K_1\hat{\underline{\xi}}_1 + K_2\hat{\underline{\xi}}_2 - M^{-1}\underline{\tau} + \hat{\underline{\delta}}_1 + \lambda_m(\underline{\xi}_1 - \hat{\underline{\xi}}_1) \\ \dot{\hat{\underline{\delta}}}_1 = \hat{\underline{\delta}}_2 + \lambda_{m-1}(\underline{\xi}_1 - \hat{\underline{\xi}}_1) \\ \qquad \vdots \\ \dot{\hat{\underline{\delta}}}_{m-1} = \hat{\underline{\delta}}_m + \lambda_1(\underline{\xi}_1 - \hat{\underline{\xi}}_1) \\ \dot{\hat{\underline{\delta}}}_m = \lambda_0(\underline{\xi}_1 - \hat{\underline{\xi}}_1) \end{cases} \tag{12}$$

with $\lambda_k \in \mathbb{R}^{3\times3}$ are diagonal matrices of which all the diagonals are positive. From (12) we deduce

$$\begin{cases} \hat{\underline{\xi}}_1^{(m+2)} = \hat{\underline{\xi}}_2^{(m+1)} + \lambda_{m+1}\tilde{\underline{\xi}}_1^{(m+1)} \\ \hat{\underline{\xi}}_2^{(m+1)} = \begin{pmatrix} K_1\hat{\underline{\xi}}_1^{(m)} + K_2\hat{\underline{\xi}}_2^{(m)} - (M^{-1}\underline{\tau})^{(m)} \\ +\hat{\underline{\delta}}_1^{(m)} + \lambda_m\tilde{\underline{\xi}}_1^{(m)} \end{pmatrix} \\ \hat{\underline{\delta}}_1^{(m)} = \hat{\underline{\delta}}_2^{(m-1)} + \lambda_{m-1}\tilde{\underline{\xi}}_1^{(m-1)} \\ \qquad \vdots \\ \ddot{\hat{\underline{\delta}}}_{m-1} = \dot{\hat{\underline{\delta}}}_m + \lambda_1\dot{\tilde{\underline{\xi}}}_1 \\ \dot{\hat{\underline{\delta}}}_m = \lambda_0\tilde{\underline{\xi}}_1 \end{cases} \tag{13}$$

where $\tilde{\underline{\xi}}_1 = \underline{\xi}_1 - \hat{\underline{\xi}}_1$ is observer error. Equation (13) lead to

$$\hat{\underline{\xi}}_1{}^{(m+2)} = \begin{pmatrix} \lambda_{m+1}\tilde{\underline{\xi}}_1^{(m+1)} + K_1\hat{\underline{\xi}}_1^{(m)} + K_2\hat{\underline{\xi}}_2^{(m)} \\ -M^{-1}\underline{\tau}^{(m)} + \lambda_m\tilde{\underline{\xi}}_1^{(m)} + \lambda_{m-1}\tilde{\underline{\xi}}_1^{(m-1)} \\ +... + \lambda_1\dot{\tilde{\underline{\xi}}}_1 + \lambda_0\tilde{\underline{\xi}}_1 \end{pmatrix} \tag{14}$$

From (10) we have

$$(M^{-1}\underline{\tau})^{(m)} = K_1\underline{\xi}_1^{(m)} + K_2\underline{\xi}_1^{(m+1)} + \underline{f}^{(m)} - \underline{\xi}_1^{(m+2)} \tag{15}$$

Substitute (15) into (14) we obtain

$$\underline{f}^{(m)} = \begin{pmatrix} \tilde{\underline{\xi}}_1^{(m+2)} + (\lambda_{m+1} - K_2)\tilde{\underline{\xi}}_1^{(m+1)} \\ +(\lambda_m - K_1 - K_2\lambda_{m+1})\tilde{\underline{\xi}}_1^{(m)} \\ +\lambda_{m-1}\tilde{\underline{\xi}}_1^{(m-1)} + ... + \lambda_1\dot{\tilde{\underline{\xi}}}_1 + \lambda_0\tilde{\underline{\xi}}_1 \end{pmatrix} \tag{16}$$

Since $\underline{f}^m \approx 0$, according to (16), if we can choose the parameter matrices so that the following polynomial $H(s)$ is Hurwitz

$$H(s) = \begin{pmatrix} s^{(m+2)}I_n + (\lambda_{m+1} - K_2)s^{(m+1)} \\ +(\lambda_m - K_1 - K_2\lambda_{m+1})s^{(m)} \\ +\lambda_{m-1}s^{(m-1)} + ... + \lambda_1\dot{s} + \lambda_0 s \end{pmatrix} \tag{17}$$

we can make the observation error $\tilde{\underline{\xi}}_1 \to 0$ when $t \to \infty$. When $\tilde{\underline{\xi}}_1 \to \underline{0}$, we also have $\hat{\underline{\xi}}_1 \to \underline{e}_1$ which lead to $\hat{\underline{\xi}}_2 \to \underline{\xi}_2$ and $\hat{\underline{\delta}}_1 \to \underline{\delta}_1$ is the total uncertain component $\underline{f}$ need to be approximated.

C. Control signal synthesis

From observer the estimation of total uncertainty component is obtained, we denote this value is $\hat{\underline{f}}$. Then the control signal $\underline{\tau}$ for 3-DoF robot manipulator will consist of 2 components satisfying

$$\underline{\tau} = M\left(-\underline{u} + \hat{\underline{f}}\right) \tag{18}$$

With control signal (18), system (10) becomes

$$\ddot{\underline{\xi}} = K_1\underline{\xi} + K_2\dot{\underline{\xi}} + \underline{u} - \hat{\underline{f}} + \underline{f} \tag{19}$$

and when $\hat{\underline{f}}$ approaches the real value $\underline{f}$ we obtain

$$\ddot{\underline{\xi}} = K_1\underline{\xi} + K_2\dot{\underline{\xi}} + \underline{u} \tag{20}$$

Let $\underline{x} = \begin{bmatrix} \underline{\xi} & \dot{\underline{\xi}} \end{bmatrix}^T$, $A = \begin{bmatrix} \Theta_3 & I_3 \\ K_1 & K_2 \end{bmatrix}$ and $B = \begin{bmatrix} \Theta_3 \\ I_3 \end{bmatrix}$ with Θ_3 and I_3 is the zero matrix and the identity matrix dimension of 3×3. System (20) will be re-expressed in linear form

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \tag{21}$$

With the linear system (21), we can use Kleiman's LQR optimal controller design algorithm that has been presented in section II to determine the parameter matrix of the optimal control signal $\underline{u}$. From there, combining the output from the observer and the optimal controller LQR, the complete control structure of the system is presented as follows

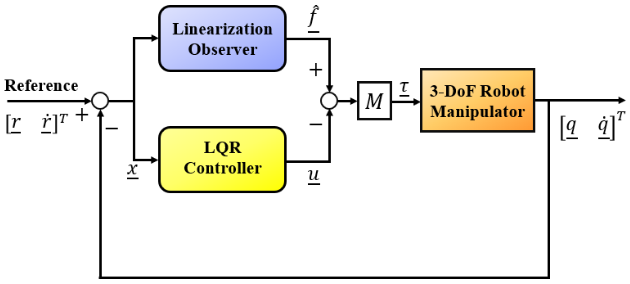


Fig. 2. Structure of controller

IV. Numerical Simulation

To verify the effectiveness of proposed controller, in this Section it will be applied on 3-DoF robot manipulator. First, the dynamic model of robot with the form (7)used for simulation will be presented based on [14], we have:

- The inertia matrix

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

  with

$$\begin{aligned} m_{11} &= \alpha_1 + 2\beta_1 c_{23} + 2\beta_2 c_2 + 2\beta_3 c_3 \\ m_{12} &= m_{21} = \alpha_2 + \beta_1 c_{23} + \beta_2 c_2 + 2\beta_3 c_3 \\ m_{13} &= m_{31} = \alpha_3 + \beta_1 c_{23} + \beta_3 c_3 \\ m_{23} &= m_{32} = \alpha_3 + \beta_3 c_3 \\ m_{22} &= \alpha_2 + \beta_3 c_3, \ m_{33} = \alpha_3 \end{aligned}$$

- The elements of lumped vector of the coriolis, centrifugal and gravitational components $\underline{\eta}\left(\underline{q}, \dot{\underline{q}}\right) = [n_1, n_2, n_3]^T$ in which

$$n_1 = \begin{pmatrix} \gamma_1 s_2 \dot{q}_1^2 + \gamma_2 s_{23} \dot{q}_1^2 + \gamma_3 s_2 (\dot{q}_1 + \dot{q}_2)^2 \\ +\gamma_4 s_3 \left(\dot{q}_1 + \dot{q}_2\right) + \gamma_5 s_{23} (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \\ +\gamma_6 s_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \end{pmatrix}$$

$$n_2 = \begin{pmatrix} \gamma_1 s_2 \dot{q}_1^2 + \gamma_2 s_{23} \dot{q}_1^2 + \gamma_4 s_3 (\dot{q}_1 + \dot{q}_2)^2 \\ +\gamma_6 s_3 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 \end{pmatrix}$$

$$n_3 = \gamma_2 s_{23} \dot{q}_1^2 + \gamma_4 s_3 (\dot{q}_1 + \dot{q}_2)^2$$

where $c_i = \cos\left(q_i\right)$, $s_i = \sin\left(q_i\right)$, $c_{ij} = \cos\left(q_i + q_j\right)$ and $s_{ij} = \sin\left(q_i + q_j\right)$ and the model parameters selected for simulation are given through Table I

TABLE I
The parameters of 3-DoF Robot

Parameter	Value	Parameter	Value	Parameter	Value
α_1	1.0425	β_2	0.1742	γ_3	-0.1742
α_2	0.4398	β_3	0.0281	γ_4	0.0281
α_3	0.1788	γ_1	0.1742	γ_5	-0.0405
β_1	0.0405	γ_2	0.0405	γ_6	-0.0281

The controller is designed with two components. The first component is the observer that converts nominal nonlinear system to linear model with the Taylor approximation order of $\underline{f}$ is $m = 1$, the parameter matrices of observer are

$$\begin{aligned} & K_1 = -diag\left([25; 49; 36]\right), \ K_2 = -diag\left([10; 14; 12]\right) \\ & \lambda_2 = 3W_0 + K_2, \ \lambda_1 = 3{W_0}^2 + \lambda_2 K_2 + K_1 \\ & \lambda_0 = {W_0}^3, \ W_0 = diag\left([80; 80; 80]\right) \end{aligned}$$

The second component is the LQR controller which is designed by the algorithm of Kleiman for linear system (1) with

$$A = \begin{bmatrix} \Theta_3 & I_3 \\ K_1 & K_2 \end{bmatrix}, \ B = \begin{bmatrix} \Theta_3 \\ I_3 \end{bmatrix}, \ C = I_6, \ D = I_3$$

and the parameter matrix for LQR optimal controller is obtained as

$$R = \begin{bmatrix} 1.93 & 0 & 0 & 4.28 & 0 & 0 \\ 0 & 1.01 & 0 & 0 & 3.26 & 0 \\ 0 & 0 & 1.36 & 0 & 0 & 3.71 \end{bmatrix}$$

Three scenarios for simulation will be performed as follows

- Scenario I: The system is not affected by disturbances and the reference is a trapezoidal signal given by

$$\underline{r}(t) = \begin{cases} 0.4t\underline{r}_t, & 0 < t < 2.5 \\ \underline{r}_t, & 2.5 \le t \le 7.5 \\ \left(1 - 0.4(t - 7.5)\right)\underline{r}_t, & t > 7.5 \end{cases} \quad (22)$$

  where $\underline{r}_t = [1; 1.2; 0.7]$

- Scenario II: The system is not affected by disturbances and the reference is a cyclic signal given by

$$\underline{r}(t) = \left(1 + \sin\frac{\pi}{2.5}t - \frac{\pi}{2}\right)\underline{r}_s \quad (23)$$

  with $\underline{r}_s = [0.5; 0.8; 0.2]$

- Scenario III: The system is affected by sinusoidal disturbances (see Fig. 3) and the reference is cyclic signal (23). The form of disturbances is designed according to [14] because it can perform the waveform of contact force in practice [14]

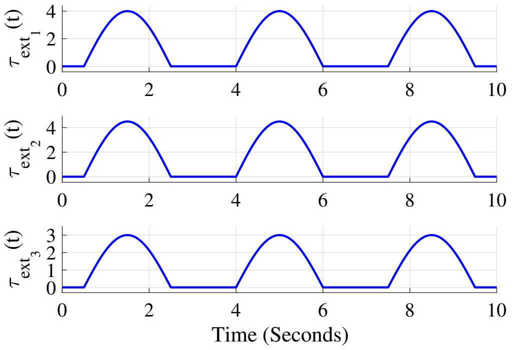


Fig. 3. The disturbances $\tau_{ext} = [\tau_{ext1}; \tau_{ext2}; \tau_{ext3}]$

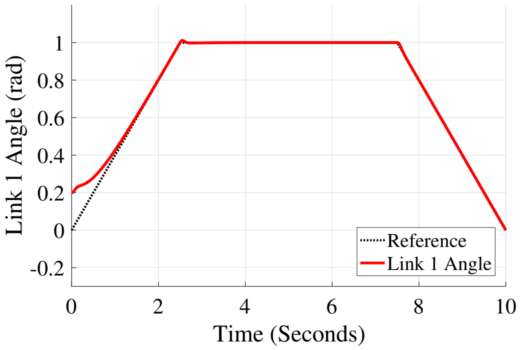


Fig. 4. State response of link 1, Scenario I

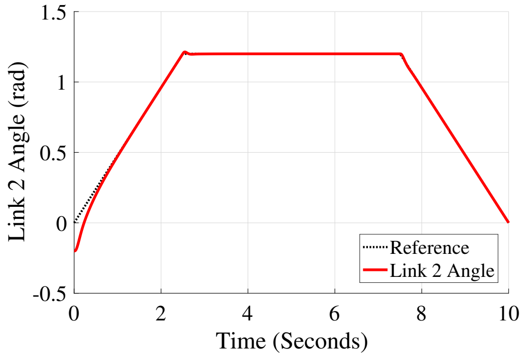


Fig. 5. State response of link 2, Scenario I

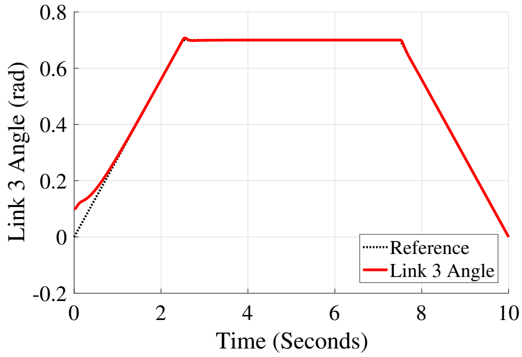


Fig. 6. State response of link 3, Scenario I

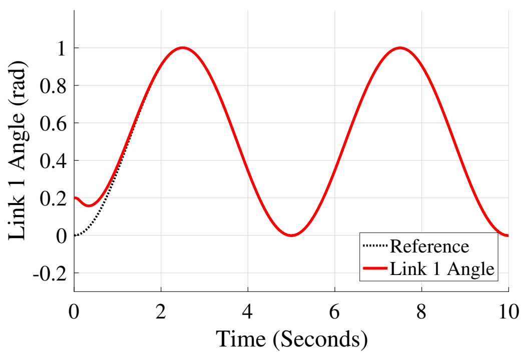


Fig. 7. State response of link 1, Scenario II

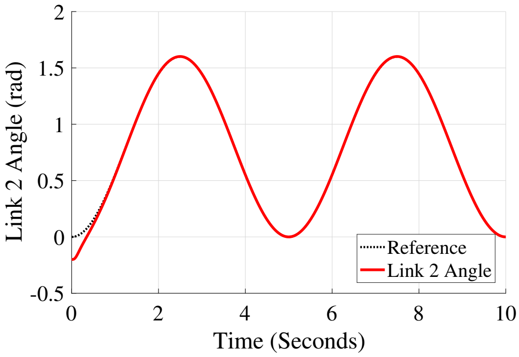


Fig. 8. State response of link 2, Scenario II

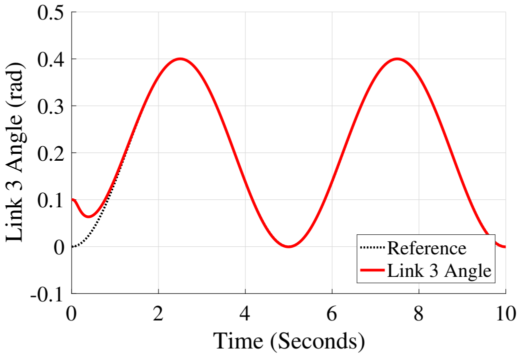


Fig. 9. State response of link 3, Scenario II

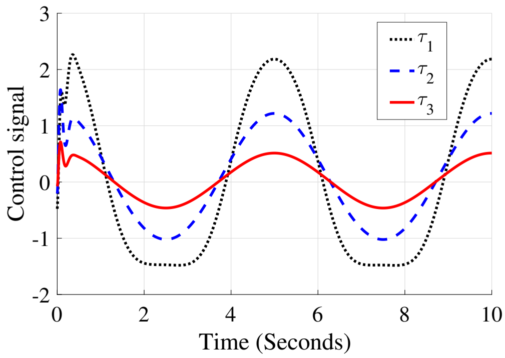


Fig. 10. Control signal $\tau = [\tau_1; \tau_2; \tau_3]$, Scenario II

In the absence of disturbances, the simulation results corresponding to Scenarios I (from Fig. 4 to Fig. 6) show that with desire trajectory is trapezoid, the proposed controller can quickly bring the states of the system to follow the trajectory signal. Furthermore, the proposed control strategy also shows efficiency when the reference is cyclic through the results in Scenarios II (from Fig. 7 to Fig. 10).

On the other hand, under the influence of disturbances, we can observe from Fig. 11 to Fig. 14 in Scenarios III that the proposed controller still retains the quality and efficiency. The angle of all 3 joints quickly track the trajectory signal and there is not too much difference compared to the control quality in Scenario II. Therefore, it can be concluded that the affect of disturbances has been removed by the linearization observer and the LQR optimal controller has completed the remaining job of optimally regulate the tracking error to converge to zero.

V. Conclusions

This paper presents an optimal control method to optimize the references model for an disturbances rejection controller based on linearization observer for 3-DoF robot manipulator. The simulation results have shown the effectiveness

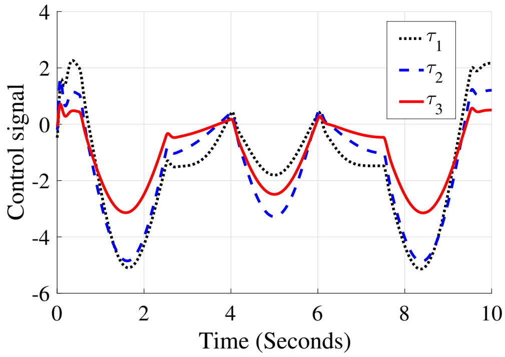


Fig. 14. Control signal $\tau = [\tau_1; \tau_2; \tau_3]$, Scenario III

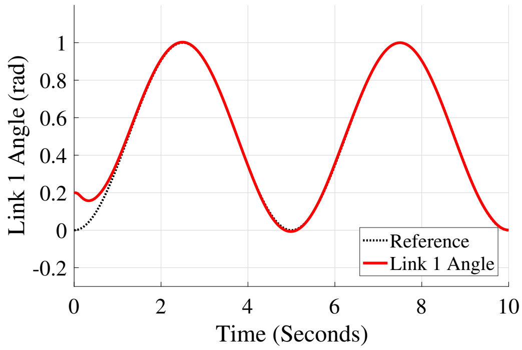


Fig. 11. State response of link 1, Scenario III

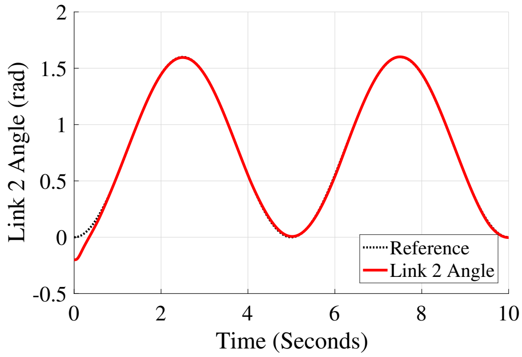


Fig. 12. State response of link 2, Scenario III

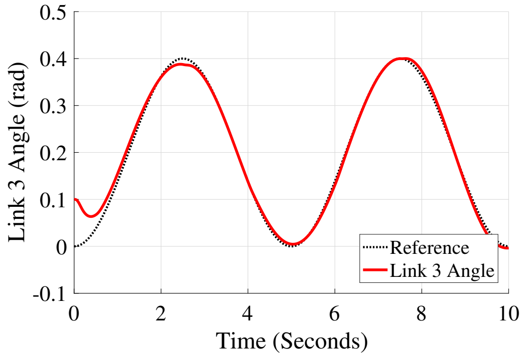


Fig. 13. State response of link 3, Scenario III

of proposed method even when the system is affected by disturbances. By combining all the non-linearity in the system together with the external disturbances into a total uncertain vector and remove it through observer, the tracking errors of nominal nonlinear model are regulated to follow a linear reference model. From there, the LQR optimal controller is designed to bring all the states of reference model to zero, lead to all the states of the original system track the references signal. The linear reference model holds the most important position determining the determining the efficiency of the controller. Therefore, determining the reference model based on the optimal control algorithm provide us an alternative approach instead of choosing arbitrarily.

REFERENCES

[1] M. M. Gao, X. Z. Jin, L. J. Ding, “Robust adaptive backstepping INTSM control for robotic manipulators based on ELM”, Neural Computing and Applications volume , vol. 34, pp. 5029–5039, 2022

[2] K. Jayaswal, D. K. Palwalia, S. Kumar, “Performance investigation of PID controller in trajectory control of two-link robotic manipulator in medical robots”, Journal of Interdisciplinary Mathematics, vol. 24, pp. 467-478, 2021.

[3] E. V. L. Nunes, L. Hsu, “Global Tracking for Robot Manipulators using a Simple PD Controller Plus Feedforward”. Robotica, vol. 28, no.1, pp. 23-34, 2010.

[4] J. S. Milind., G. Arunkumar, T.C. Manjunath, “PID Control of a Double Link (2-link) Flexible Robotic Manipulator (2-DOF) in the 3 DE Space”. 2018 4th International Conference for Convergence in Technology (I2CT), 2018

[5] S. A. Ajwad, R. U. Islam, M. R. Azam, M. I. Ullah., J. Iqbal, “Sliding mode control of rigid-link anthropomorphic robotic arm”, 2nd International Conference on Robotics and Artificial Intelligence (ICRAI), 2016

[6] S. Kasera, A. Kumar, L. B. Prasad, ”Trajectory Tracking of 3-DOF Industrial Robot Manipulator by Sliding Mode Control”, 4th IEEE Uttar Pradesh Section International Conference on Electrical Computer and Electronics (UPCON), 2017.

[7] M. H. Korayem, R. Shiri, S. R. Nekoo, “Sliding Mode Control Design with Adaptable Gains for Robotic Manipulators” 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), 2015.

[8] P.R. Ouyang P.R., J. Acob, V. Pano, “PD with sliding mode control for trajectory tracking of robotic system”, Robotics and Computer-Integrated Manufacturing, vol. 30, no. 2, pp. 189-200, 2014.

[9] P. R. Ouyang, J. Tang, W. H. Yue, S. Jayasinghe, “Adaptive PD plus Sliding Mode Control for Robotic Manipulator”, IEEE International Conference on Advanced Intelligent Mechatronics (AIM), 2016.

[10] U. Zakia, M. Moallem, C. Menon, “PID-SMC controller for a 2-DOF planar robot”, 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019.

[11] C. Urrea, J. Kern, J. Alvarado, “Design and Evaluation of a New Fuzzy Control Algorithm Applied to a Manipulator Robot”, Applied Sciences, vol. 10, no. 21, 2020.

[12] Q. Guo, Y. Zhang, B. G. Celler, S. W. Su, “Neural Adaptive Backstepping Control of a Robotic Manipulator With Prescribed Performance Constraint”, IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 12, pp. 3572 – 3583, 2019.

[13] M. Ramírez-Neria, R. Madonski, A. Luviano-Juárez, Z. Gao, H. Sira-Ramírez, “Design of ADRC for Second-Order Mechanical Systems without Time-Derivatives in the Tracking Controller”, 2020 American Control Conference (ACC), 2020.

[14] Z. Zhang, M. Leibold, D. Wollherr, “Integral Sliding-Mode Observer-Based Disturbance Estimation for Euler–Lagrangian Systems”, IEEE Transactions on Control Systems Technology, vol. 28, no. 6, pp. 2377-2389, 2019.

[15] S. M. Ahmadi, M. M. Fateh, "Task-space control of robots using an adaptive Taylor series uncertainty estimator", *International Journal of Control*, vol. 92, no. 9, pp. 2159-2169, 2018.

[16] E. Guechi, S. Bouzoualegh, Y. Zennir, S. Blažic, "MPC Control and LQ Optimal Control of A Two-Link Robot Arm: A Comparative Study", Machines, vol. 6, pp. 409-414, 2018.

[17] B. Zhao, Y. Li, "Model-free Adaptive Dynamic Programming Based Near-optimal Decentralized Tracking Control of Reconfigurable Manipulators", Int. J. Control Autom. Syst. 16, pp. 478–490, 2018.

[18] D. Kleinman, "On an iterative technique for riccati equation computations", IEEE Transactions on Automatic Control, vol. 13, no. 1, pp. 114-115, 1968.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 129–133 DOI: 10.15439/2022R12 ISSN 2300-5963 ACSIS, Vol. 33

A Study on Thyroid Nodule Image Classification System Using Small Amount of Training Samples

1st Quoc Tuan Hoang
Faculty of Mechanical Engineering
Hung Yen University of Technology and Education
Hung Yen, Vietnam
hoangquoctuan@utehy.edu.vn

2nd Xuan Hien Pham
Department of Mechanical Engineering
University of Transport and Communications
Hanoi, Vietnam
hienpx@utc.edu.vn

3rd Xuan Thang Trinh
Faculty of Mechanical Engineering
Hung Yen University of Technology and Education
Hung Yen, Vietnam
xttrinh@utehy.edu.vn

***Abstract*—To reduce errors caused by traditional diagnostic methods that rely heavily on physician experience, the diagnostic systems based on computer-aided have been researched and developed to assist physicians in diagnosing thyroid disease. Therefore, performance of the computer systems plays an important role to improve the quality of diagnostic processes. Although there has been a number of publish related to this issue, those studies still have limitations in which needing large data sets for training classification models is considered the most concerning limitation of previous studies. To solve this limitations, in this work, a classification method using artificial intelligence with a small amount of data to analyze thyroid ultrasound images was proposed. Thus we can save time and effort for data collection and the classification model processing time. Through baseline tests with an open data set, especially thyroid digital image database (TDID), the proposed method has improved the limitations of previous methods.**

***Index Terms*—malignant thyroid nodule, deep learning, artificial intelligence, ultrasound image**

I. Introduction

Cancer is an extremely dangerous disease, characterized by the ability to invade and spread to other parts of the body. If cancer is not detected and treated in time, the patient will die quickly. Cancer cells grow and divide rapidly and have the ability to invade and spread to other organs. As noted, several cancers have a high incidence today, such as Breast cancer in women is 24.5%, the highest among all cancers in women. Lung cancer is also common cancer, accounting for about 12% of all cancers worldwide. In Vietnam, lung cancer ranks first among the 10 most common cancers in both males and females and is the leading cause of death. Thyroid cancer is a common disease with no obvious symptoms. It occurs in both men and women. This kind of cancer is often found in the area of the head, face and neck. Thyroid cancer cells can metastasize to several parts of the body such as the liver, lungs, brain, and bones, thereby making a higher mortality rate.

On the human body, the thyroid gland is an important organ and it is located in the human neck. The thyroid gland produces and secretes two important hormones, including thyroxine and triiodothyronine. They are responsible for regulating metabolism in the human body.

For these reasons, it is absolutely necessary and important to diagnose and treat thyroid disease [2-6]. In previous reports, they indicated that the appearance of thyroid cancer nodules was a common serious problem in the thyroid region. Thyroid nodules are formed by abnormal growths appearing in the thyroid area of the human body. Many factors can cause thyroid nodules, for example overgrowth of thyroid tissue, iodine deficiency or thyroid cancer. Based on their characteristics, thyroid nodules are usually classified into two types to be benign cases (which are noncancerous nodules) and malignant cases (which can cause thyroid cancer) [7]. In both cases, benign and malignant, the presence of thyroid nodules can cause health effects on the patient. The thyroid gland may be affected with the appearance of nodules. Benign cases have little effect on the patient's health, but affect the patient's appearance, making it difficult for the patient to breathe and swallow. Malignant cases can cause thyroid cancer. The diagnosis and treatment of malignancies become very important.

The traditional diagnosis method is mainly based on the doctor's knowledge and experience when examining the external manifestations such as the appearance of tumors with hard characteristics; clear margins; smooth or rough surface; the presence of cervical lymphadenopathy; large and solid tumor which is fixed in front of the neck; hoarseness; possibly difficulty in breathing, difficulty in swallowing or entanglement due to tumor compression, infiltrated or ulcerated with bleeding neck’s skin. However, the major limitation of traditional diagnostic method is its performance is highly dependent on personal knowledge and the experience of the physician. For that reason, diagnostic performance varies and can take longer, these methods are laborious, invasive, and expensive.

In this century of growing technology, imaging-based diagnostic techniques have been widely used, especially computer-aided diagnostic (CAD) systems that have been developed to help physicians in the diagnostic and therapeutic process [8-10, 20]. In contrast, to the conventional thyroid diagnostic methods mentioned above, CAD methods used ultrasound images of thyroid nodules as input and provide information about nodules (benign or malignant) [2-6], thus limiting the unevenness of different doctor’s qualifications at different hospital levels and leading to faster initial diagnosis. Technological advances such as back propagation algorithms, neural networks and graphics processing units (GPUs), and techniques based on deep learning have recently been applied to solve many problems in medical image processing systems [7,8,9,10, 11,12,]. Regarding detecting/classifying thyroid nodules, the "deep learning" method has achieved much success. As the name implies, methods based on "deep learning", such as convolutional neural networks (CNN), automatically learn image features to train and produce results with high performance. Therefore, a deep learning-based approach can produce superior diagnostic results.

Deep Learning has high accuracy with a flexible neural network architecture, which can be easily changed to fit many different problems in the training process. High automation, self-adjusting, self-optimizing and the capability of performing parallel computing make deep learning-based methods produce a good performance and can handle large amounts of training data. However, there are still some limitations of previous studies. For instance, the large amount of training data is needed. Therefore, it takes longer time and more effort to collect these data. Additionally, the image acquisition devices are expensive and moreover, it requires co-operation and agreement between doctors and patients [7]. The training process consumes long time. The good computer's hardware is required to train model. Therefore, it is inapplicable to apply to some problems in which a large amount of training data is not available, such as a new kind of disease indicated in [13,14,15]. In this work, to solve the above-mentioned problem of previous studies, we propose a thyroid nodule classification method that can be trained on a small number of training data, we can save time and effort for both data collection and training classification models. Therefrom, we design a deep-learning network that can learn to not only recognize images of a class but also distinguish images between classes.

II. Proposed Method

A. Network architecture

The overview of proposed network architecture is described in Fig. 1. As mentioned in section I, we design a network that can distinguish images from different class labels, and recognize images of the same class. Therefore, our proposed network accepts two input images, as shown in Fig. 1.

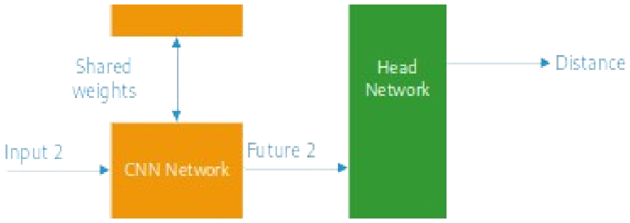


Fig 1. Overview of the proposed network.

In the training phase, our network receives two input images and uses a CNN network to extract the image feature vectors that are the best representation of the two input images. After that, the two feature vectors are transformed into another feature domain using a head network. The detailed descriptions of the CNN and head network are present in the next sections.

In the testing phase, a new input image is inputted to the network and our network will measure the distance from the input images to the training samples in the training dataset to find the best-matched samples with the input image. We used same network architecture with pre-trained weight for image feature extraction for both input 1 and input 2.

B. Related Work

Convolutional Neural Networks (CNNs), a type of artificial neural network, have become essential in various computer vision operations, and are receiving more attention across different published studies. For example, artificial intelligence association into brain magnetic resonance imaging (MRI) in cancer diagnosis [20], audiogram classification method [21], and cancerous region detection in the prostate [22]. The basic layers in a CNN include: convolution layer, pooling layer, fully connected layer, which are changed in number and arrangement to create suitable training models for different problems.

Convolutional Layer is the most important component in CNN, also the place to express the idea of building local connections instead of connecting all pixels. These local links are calculated by convolution between pixel values in a local image region with filters of small size. This filter is shifted through each image area in turn until it completes scanning the entire image, creating a new image that is less than or equal to the size of the input image. After feeding an image to the convolution layer, we get the output as a series of images corresponding to the filters used to perform the convolution. The weights of these filters are initialized randomly for the first time and will improve throughout training. The convolutional layer outputs a series of feature maps that enter the next convolutional or pooling layer.

Pooling Layer is another major computational component in CNN called Pooling, usually placed after the Convolution layer and ReLu layer to reduce the output image size while preserving the important information of the input image. Reducing the data size has the effect of reducing the number of parameters as well as increasing the computational efficiency. The sampling layer also uses a sliding window to scan all regions of the image similar to the Convolution layer and performs sampling instead of convolution, it mean we will choose to save a single representative value. for the entire information area. There are two commonly used Sampling methods, Max Pooling and Average Pooling. Thus, for each input image put through sampling, we obtain a corresponding output image, whose size is significantly reduced but still retains the necessary features for the later calculation process.

Fully-connected Layer is designed completely like a traditional Neuron network, it mean all pixels are fully connected to the Node in the next layer. Compared with the traditional neural network, the input image of this layer has been greatly reduced in size, while still ensuring important information for identification. Therefore, the recognition calculation using the feedforward model is no longer complicated and time consuming as in the traditional neural network. The final dense layer outputs the probabilities for classification by applying a softmax method.

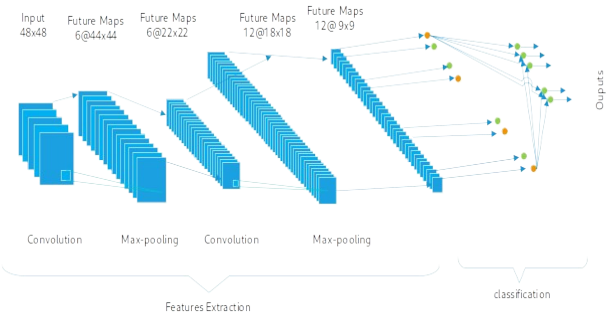


Fig. 2. Conventional CNN network.

Figure 2 shows the CNN network is used to extract image features for an input image. In our study, we use a pre-trained

CNN network trained on the ImageNet dataset to extract image features of input images. The reason is that the conventional CNN network contains a huge number of training parameters. Therefore, it is unable to train this kind of network using a small amount of training data. We called the CNN network in Figure 1 the black-bone in our study. The black-bone can be any CNN network that can be used for image feature extraction such as VGG, Residual, and Inception networks.

C. The head network

The head network based on an MLP network is shown in Fig. 3. The main part of our proposed network is the head network that has responsibility for learning the relationship between images within a class, as well as, among different classes. For that problem, we first transform the input image features to another domain using a conventional neural network. In the new image feature domain, the feature vectors of images that are in the same class are similar, while the feature vectors of different classes are made to be different. In order to obtain that purpose, we then measure the distance between two feature vectors from two input images (after CNN and domain transformation layers). That distance is forced to be 1 if the two input images are from the same class, and forced to be 0 if the input images are from different classes.

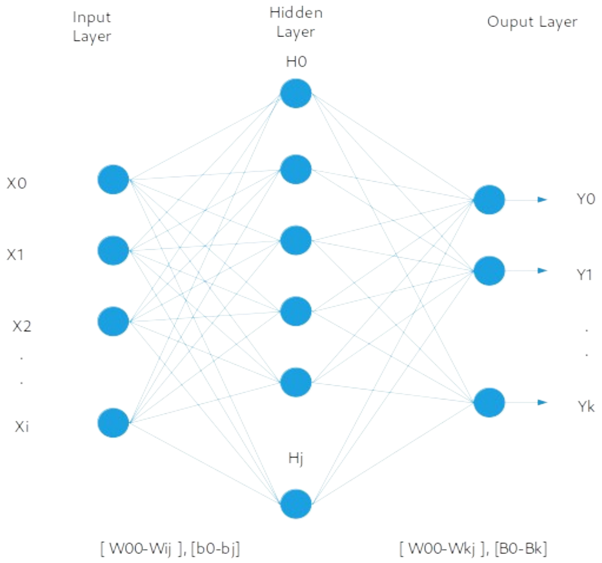


Fig 3. The head network is based on an MLP network.

The MLP network can contain several layers, each layer contains a number of neurons. In our experiments, we use three layers with number of neurons are 256 - 512 - 2. As shown in Figure 3.

III. Data Set

In this study, the realization of the proposed method is demonstrated using a thyroid ultrasound dataset that has been published under the name Thyroid Digital Imaging Database (TDID). In 2015, at the University of Colombia Nacional de Colombia, Pedraza et al collected and published this dataset [16]. The TDID dataset contains thyroid ultrasound images of 298 patients. Ultrasound images of each patient's thyroid region, which can be single or multiple images, were collected in RGB format with 560 pixels × 360 pixels in image size. As a result, a dataset consisting of 450 thyroid nodule images was extracted for our experiments.

Fig. 4 shows some example thyroid nodule images in the TDID dataset. As we can see that it is hard to distinguish the normal and disease thyroid image by human perception. That is why the computer aid diagnosis system is helpful. It is little difference between the two kinds of images, that is, the disease thyroid nodule images contain nodules with brighter nodule regions with the effect of the calcification phenomenon.

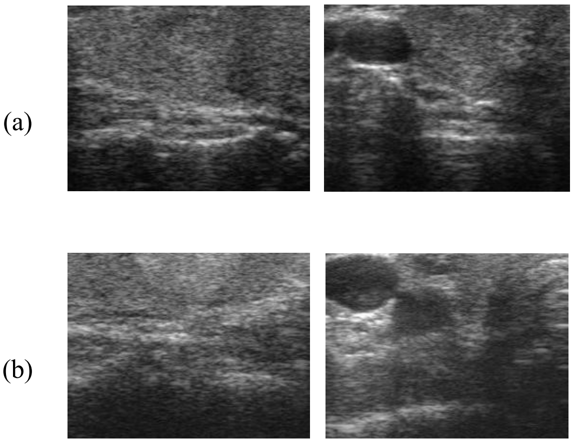


Fig. 4. Example of (a) normal, and (b) disease images in TDID dataset [16].

IV. Experimental Results

As mentioned in our previous section, we are dealing with the classification problem of the medical image that has a small amount of data for training. For our purpose, we divided the TDID dataset (mentioned in Section III), into five parts to perform a five-fold cross-validation method. However, we use data of a single part as the training data, while the images of the other four parts are used as the testing data. This procedure is different from previous studies that use data from four parts for training, and the data of the remaining part is used for testing. Because of different training and testing data, the classification performance of conventional CNN-based methods is different as shown in previous studies. For comparison purposes, we also evaluate the classification performance of two popular CNN-based classification networks, including the Inception and Residual network architectures [17].

In order to evaluate the classification accuracy of the proposed network and the conventional CNN network, we use the classification accuracy metric as shown in (1).

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

In (1), the TP is the measure of true positive samples that is the case when an image with the disease was correctly classified as a with-disease image. The TN indicates the true negative samples which is the case when the normal image was correctly classified as a normal one. The FP stands for false-positive, the FN stands for false-negative. For a fair comparison with the Residual and Inception networks, we also use the Residual and Inception network as the black-bone of our proposed method.

The experimental results are given in Table 1 and Table 2 for the cases of using a Residual network, and Inception network architecture.

TABLE I. Classification performance of our proposed network and previous CNN-based network using Residual network architecture (unit: %)

Method	Part 1	Part 2	Part 3	Part 4	Part 5	Average
Residual Network	79.619	84.091	75.414	70.055	78.177	77.472
Proposed Network with Residual Black-bone	83.152	84.091	85.912	81.319	79.282	82.751

TABLE II. Classification performance of our proposed network and previous CNN-based network using Inception network architecture (unit: %)

Method	Part 1	Part 2	Part 3	Part 4	Part 5	Average
Inception Network	85.054	82.102	85.635	84.066	84.254	84.222
Proposed Network with Inception Black-bone	86.141	84.375	86.188	84.890	85.083	85.335

For reference, we performed experiments with the residual and inception networks and performances are given in Table I and II. To measure the performance with the residual and inception network, we use fine-tuning the network by replaced the number of output neuron by 2 (original output neuron of these networks is 1000) and kept all convolution layers are same. [18-19].

As we can have observed from Table I, the Residual network produced an average classification accuracy of 77,47%. Using our proposed method, we enhanced that accuracy to 82,75%. This result indicates that our proposed network with the Residual network as black-bone outperforms the conventional residual network for the classification problem. A similar result was obtained in Table 2 using the Inception network architectures. As we can see in Table 2, our proposed network with the Inception architectures also outperform the conventional Inception network by producing an average classification accuracy of 85,33% that is higher than 84,22% produced by using the inception network.

As shown in Tables 1 and Tables 2, the classification performance of Inception Network–the based architecture was better than that of Residual Network-based architecture. The reason is that the Inception network-based architecture uses multiple convolution kernels to extract image features. As a result, it is more efficient to extract both large and small features than the Residual network-based architecture.

However, our proposed method outperforms all the two popular CNN networks (Residual and Inception network). It indicates that our proposed method is more efficient than the conventional CNN network when working with the lack of training data problem.

Although our proposed method produced higher classification accuracy than the convention networks network, it takes a longer processing time than the convention CNN network as it must measure the similarity from an input image to all images in the training set. However, it is acceptable with the medical image processing system in which accuracy plays a more important meaning than the processing time. In addition, with the support of graphical processing units (GPUs), the processing time is much reduced when we use strong computer specifications.

V. Conclusion

In this paper, we focus on solving a specific problem of medical image classification problem which is when a large amount of training data is not available. In order to solve this problem, we design a deep learning-based classification method based on an image-retrieval approach. In detail, we construct a system that has the ability to recognize images of the same class while distinguishing images between classes. Through experiments, we showed that our proposed method outperforms two conventional CNN-based methods (Residual-based and Inception-based CNN network architecture).

References

[1] TDID dataset Pedraza, L.; Vargas, C.; Narvaez, F.; Duran, O.; Munoz, E.; Romero, E. An open access thyroid ultrasoundimage database. In Proceedings of the 10th International Symposium on Medical Information Processing and Analysis, Cartagena de Indias, Colombia, 28 January 2015; pp. 1–6.

[2] Koundal, D.; Gupta, S.; Signh, S. Computer aided thyroid nodule detection system using medical ultrasound images. Biomed. Signal Process. Control 2018, 40, 117–130.

[3] Tessler, F.N.; Middleton, W.D.; Grant, E.G.; Hoang, J.K.; Berland, L.L.; Teefey, S.A.; Cronan, J.J.; Beland, M.D.; Desser, T.S.; Frates, M.C.; et al. ACR thyroid imaging, reporting and data system (TI-RADS): White paper of the ACR TI-RADS committee. J. Am. Coll. Radiol. 2017, 14, 587–595.

[4] Ma, J.; Wu, F.; Zhu, J.; Xu, D.; Kong, D. A pretrained convolutional neural network based method for thyroid nodule diagnosis. Ultrasonics 2017, 73, 221–230.

[5] Pedraza, L.; Vargas, C.; Narvaez, F.; Duran, O.; Munoz, E.; Romero, E. An open access thyroid ultrasoundimage database. In Proceedings of the 10th International Symposium on Medical Information Processing and Analysis, Cartagena de Indias, Colombia, 28 January 2015; pp. 1–6.

[6] Zhu, Y.; Fu, Z.; Fei, J. An image augmentation method using convolutional network for thyroid nodule classification by transfer learning. In Proceedings of the 3rd IEEE International Conference on Computer and Communication, Chengdu, China, 13–16 December 2017; pp. 1819–1823.

[7] Nguyen, D.T.; Pham, D.T.; Batchuluun, G.; Yoon, H.S.; Park, K.R. Artificial intelligence-based thyroid nodule classification using information from spatial and frequency domains. J. Clin. Med. 2019, 8, 1976.

[8] Vuong, Q.H.; Ho, M.T.; Vuong, T.T.; La, V.P.; Ho, M.T.; Nghiem, K.C.P.; Tran, B.X.; Giang, H.H.; Giang, T.V.; Latkin, C.; et al. Artificial intelligence vs. natural stupidity: Evaluating AI readiness for the Vietnamese medical information system. J. Clin. Med. 2019, 8, 168.

[9] Havaei, M.; Davy, A.; Warde-Farley, D.; Biard, A.; Courville, A.; Bengio, Y.; Pal, C.; Jodoin, P.-M.; Larochelle, H. Brain tumor segmentation with deep neural networks. Med. Image Anal. 2017, 35, 18–31.

[10] Bhandary, A. et al. Deep-learning framework to detect lung abnormality – A study with chest x-ray and lung CT scan images. Pattern Recogn. Lett. 2020, 129, 271–278.

[11] Kamnitsas, K.; Ledig, C.; Newcombe, V.F.J.; Simpson, J.P.; Kane, A.D.; Menon, D.K.; Rueckert, D.; Glocker, B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med. Image Anal. 2017, 36, 61–78.

[12] Zhu, Y.; Fu, Z.; Fei, J. An image augmentation method using convolutional network for thyroid nodule classification by transfer learning. In Proceedings of the 3rd IEEE International Conference on Computer and Communication, Chengdu, China, 13–16 December 2017; pp. 1819–1823.

[13] Vuong, Q.H.; Ho, M.T.; Vuong, T.T.; La, V.P.; Ho, M.T.; Nghiem, K.C.P.; Tran, B.X.; Giang, H.H.; Giang,T.V.; Latkin, C.; et al. Artificial intelligence vs. natural stupidity: Evaluating AI readiness for the Vietnamese medical information system. J. Clin. Med. 2019, 8, 168.Sensors 2020, 20, 1822 21 of 23

[14] Havaei, M.; Davy, A.; Warde-Farley, D.; Biard, A.; Courville, A.; Bengio, Y.; Pal, C.; Jodoin, P.-M.; Larochelle, H. Brain tumor segmentation with deep neural networks. Med. Image Anal. 2017, 35, 18–31.

[15] Bhandary, A. et al. Deep-learning framework to detect lung abnormality – A study with chest x-ray and lung CT scan images. Pattern Recogn. Lett. 2020, 129, 271–278.

[16] Pedraza, L.; Vargas, C.; Narvaez, F.; Duran, O.; Munoz, E.; Romero, E. An open access thyroid ultrasoundimage database. In Proceedings of the 10th International Symposium on Medical Information Processing and Analysis, Cartagena de Indias, Colombia, 28 January 2015; pp. 1–6.

[17] Introduction to Convolutional Neural Networks (CNN), Oct. 2022, [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/

[18] Nikhila, Ponugoti, et al. "Lightweight residual network for the classification of thyroid nodules." arXiv preprint arXiv:1911.08303 (2019).

[19] Guan, Qing, et al. "Deep learning based classification of ultrasound images for thyroid nodules: a large scale of pilot study." Annals of translational medicine 7.7 (2019).

[20] Duong-Trung, Nghia, Dung Ngoc Le Ha, and Hiep Xuan Huynh. "Classification-Segmentation Pipeline for MRI via Transfer Learning and Residual Networks." RICE. 2021.

[21] Kassjański, Michał, Marcin Kulawiak, and Tomasz Przewoźny. "Development of an AI-based audiogram classification method for patient referral." 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS). IEEE, 2022.

[22] Treigys, Povilas, et al. "Detecting Cancerous Regions in DCE MRI using Functional Data, XGboost and Neural Networks." Annals of Computer Science and Information Systems 32 (2022): 23-30.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 135–140 DOI: 10.15439/2022R19 ISSN 2300-5963 ACSIS, Vol. 33

Development of a Smart System for Early Detection of Forest Fires based on Unmanned Aerial Vehicles

Georgi Hristov
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
ghristov@uni-ruse.bg

Nikolay Zlatov
Institute of Mechanics. Bulgarian
Academy of Sciences
Sofia, Bulgaria
zlatovn@hotmail.com

Plamen Zahariev
Department of Telecommunications
University of Ruse
“Angel Kanchev” Ruse, Bulgaria
pzahariev@uni-ruse.bg

Chi Hieu Le
Faculty of Science & Engineering
University of Greenwich
Kent, United Kingdom
c.h.le@gre.ac.uk

Diyana Kinaneva
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
dkyuchukova@uni-ruse.bg

Georgi Georgiev
Department of Telecommunications
University of Ruse “Angel Kanchev”
Ruse, Bulgaria
gdgeorgiev@uni-ruse.bg

Yavor Yotov
Institute of Mechanics. Bulgarian
Academy of Sciences
Sofia, Bulgaria
javor.jotov@imbm.bas.bg

James Gao
Faculty of Science & Engineering
University of Greenwich
Kent, United Kingdom
j.gao@gre.ac.uk

Anh My Chu
Institute of Simulation Technology, Le
Quy Don Technical University
Hanoi, Vietnam
myca@lqdtu.edu.vn

Ho Quang Nguyen
Institute of Engineering & Technology
Thu Dau Mot University
Binh Duong, Vietnam
quangnh@tdmu.edu.vn

Le Minh Huynh
Department of Science & Technology
Saigon Hi-Tech Park
Ho Chi Minh City, Vietnam
hlminh.shtp@gmail.com

Trung Thanh Bui
Faculty of Mechanical Engineering
Hung Yen University of Technology
and Education. Hung Yen, Vietnam
buitrungthanh@gmail.com

Mohd Nor Azmi Ab Patar
College of Engineering, Universiti
Teknologi MARA, Shah Alam Selangor,
Malaysia
azmipatar@uitm.edu.my

Jamaluddin Mahmud
College of Engineering, Universiti
Teknologi MARA, Shah Alam
Selangor, Malaysia
jm@uitm.edu.my

Michael S. Packianather
School of Engineering
Cardiff University
Cardiff, United Kingdom
packianatherms@cf.ac.uk

***Abstract*—The naturally occurring wildfires and the people-related forest fires are events, which in many cases have significant impact on the environment, the wildlife and the human population. The most devastating among these events usually start in unpopulated remote areas, which are difficult to inspect or are not constantly being monitored or observed. This gives the local small-sized fires enough time to evolve into full-scale wide-area disasters, which in turn makes their suppression and extinguishing very difficult.**

In this paper, we present an autonomous system for early detection of forest fires, named THEASIS-M. The presented system represents a solution that is based on a combination of innovative technologies, including computer vision algorithms, artificial intelligence and unmanned aerial vehicles.

In the first part of the study, we provide an overview on the present applications of the UAVs in the forestry domain. The paper then introduces the general architecture of the THEASIS-M system and its components. The system itself is fully autonomous and is based on several different types of UAVs, including a fixed-wing drone, which provides the overall forest monitoring capabilities of the proposed solution, and a rotary-wing UAV that is used for confirmation and monitoring of the detected fire event. The widely used technologies for computer vision and image processing, which are used for the detection of fire and smoke in the real-time video streams sent from the UAVs to the ground control station, are highlighted in the next section of this study. Finally, the experimental tests and demonstrations of the proposed THEASIS-M system are presented and briefly discussed.

***Index Terms*—Unmanned aerial vehicles, computer vision, artificial intelligence, early detection of forest fires, autonomous system for fire detections**

I. Introduction

The unmanned aerial vehicles (UAVs), which are often also called drones, are advancing in many different application areas and the forestry domain is no exception. Activities, which previously were demanding or involving the use of human-operated agricultural machines or ground and aerial vehicles, can now be executed using drones at relatively lower costs and way easier. While the UAVs cannot yet fully replace the humans in all forestry-related activities, their capabilities and functionalities are constantly improving and these systems are being used for more and more complex and advanced tasks. The most trivial and widely-known application of the unmanned aerial vehicles in the agricultural and forestry domain is for observation purposes [1]. With the help of different UAVs and their on-board camera systems, local authorities, property owners and law-enforcing agencies can monitor any area of interest and can identify illegal activities, including unauthorised logging, poaching or harvesting [2]. The use of UAVs leads to the prevention of significant financial losses from the aforementioned and other criminal activities, but at the same time it is also leading to the saving of many endangered species, to the better management and maintenance of natural parks, to the preservation of trees and bushes and correspondingly to delays of the global warming processes [3].

The applications of the UAVs have been well-documented with their huge potentials, benefits and impacts [3-7, 14-18]. The data collection, remote measurements, estimation of the forest coverage, calculation of the biomass of the trees and evaluation of the different attributes of the forest vegetation are among the many examples of how the UAVs can be used for scientific activities, studies and protection of the forests.

The precision forestry and the modern forestry management processes involve interesting drone-related activities, including the mapping of the forests, the canopies and the canopy gaps [15], the development of ortho-photo maps and the creation of 3D models of the forest areas [16]. While many of these activities can be implemented using standard optical cameras, some of mentioned mapping and measuring activities require more advanced payloads, such as laser-based LiDAR systems [17], Global Positioning System (GPS) or inertial measurement units (IMUs) for estimation of the heights of the objects on the ground [18].

The majority of the above-mentioned examples can be categorized as passive applications of the UAVs in the forestry domain. These activities are not characterized by any direct impact or manipulation on the environment, the trees or the forests in general. With the recent advancement in the UAV-related technologies, it is now also possible to involve the UAVs in active applications in the forestry domain. The tree planting with UAVs is among these applications and represents the process of fast and efficient distribution of seeds or seedlings in the remote or inaccessible forest areas. Spraying or spreading fertilizers in the forest areas is another good example of an active application of the UAVs in the forestry domain [4].

It is clearly seen that UAVs have been used in many applications with well-documented benefits and impacts in the forestry domain, especially in modern forestry control, management, protection and conservation. This study investigates and considers the use of the UAVs for early detection and monitoring of forest fires, for actual fighting and extinguishing of wildfires and for post-fire damage assessment to be the applications with the highest significance and impact in the forestry domain. The THEASIS-M autonomous system for early detection of forest fires, which is presented in this paper (Fig. 1), was successfully developed and tested. The system is based on a combination of innovative technologies, including computer vision algorithms, artificial intelligence and UAVs.

The rest of the paper is organized as follows. Section II presents materials and methods with the focus on (i) the generalized model of the system for early detection of forest fires using UAVs; (ii) the neural networks, computer-vision algorithms and artificial intelligence technologies for detection of fire and smoke; and (iii) the selection of the functional components for development of the UAV-based fire detection and observation system. Section III presents the experimental demonstrations and results and finally, Section IV presents a summary, brief discussions and conclusions.

II. Materials and Methods

A. Generalized model of the system for early detection of forest fires using UAVs

The involvement of the UAVs in the antifire-related activities is relatively less expensive, compared to the use of manned aircrafts and satellites, but is also characterized by many disadvantages.

Generally, the UAVs are limited by the present state of the drone manufacturing technologies. Some of these limitations are related to the flight time, the payload capacity and the communication range of the drones. Another disadvantage of the use of the UAVs for firefighting applications is the requirement for proper training of the UAV pilots and operators. The UAVs are also characterised by specific limitations related to their type, size and propulsion systems. The fixed-wing UAVs for fire detection and monitoring are slowly gaining wider approval and acceptance, but still the majority of the used UAVs in this application domain are small-sized rotary-wing drones.

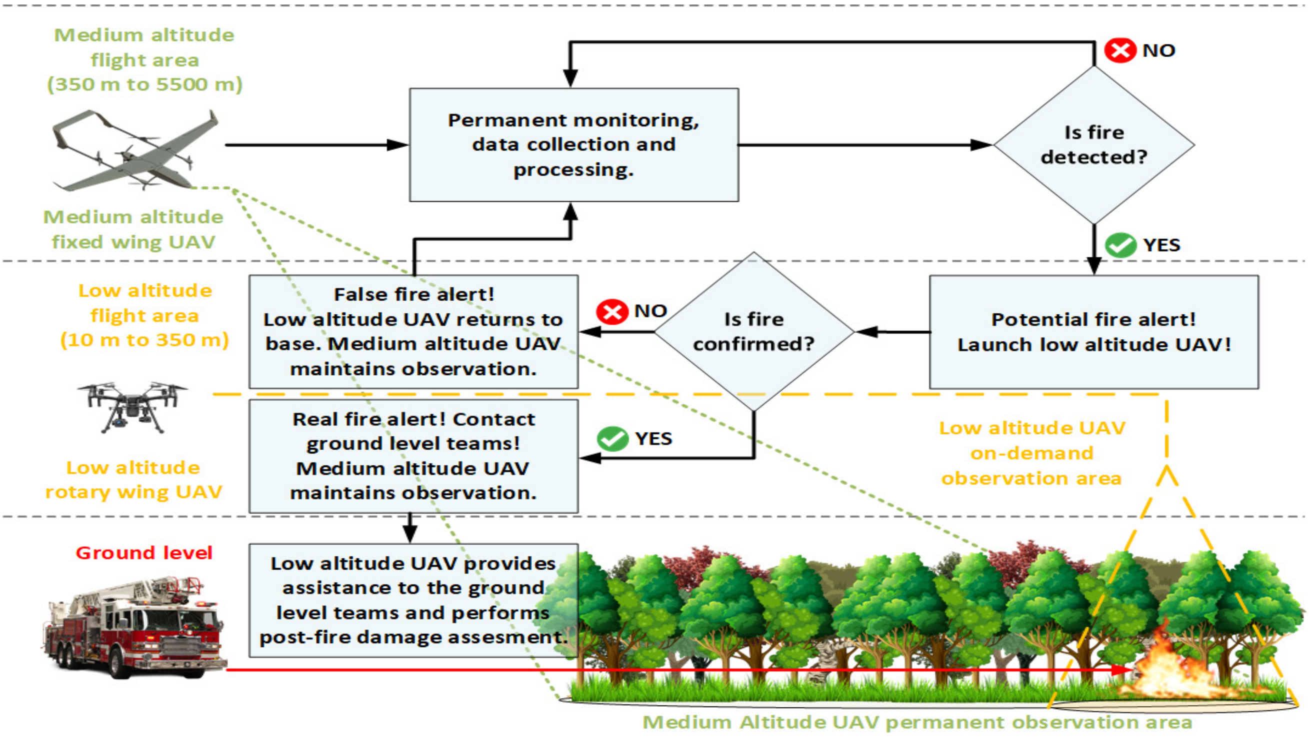

Medium altitude flight area (350 m to 5500 m)
Medium altitude fixed wing UAV
Permanent monitoring, data collection and processing.
Is fire detected?
NO
YES
Potential fire alert! Launch low altitude UAV!
Is fire confirmed?
NO
YES
Low altitude flight area (10 m to 350 m)
False fire alert! Low altitude UAV returns to base. Medium altitude UAV maintains observation.
Real fire alert! Contact ground level teams! Medium altitude UAV maintains observation.
Low altitude rotary wing UAV
Low altitude UAV on-demand observation area
Ground level
Low altitude UAV provides assistance to the ground level teams and performs post-fire damage assesment.
Medium Altitude UAV permanent observation area

Fig.1. A conceptual model of the proposed THEASIS-M system for early detection and observation of forest fires, which consists of a fixed-wing UAV, as a primary monitoring and observation unit and a secondary rotary-wing drone for confirmation and closer monitoring of the detected fire event and for post-fire damage assessment.

In addition, the majority of the forest fire detection and monitoring solutions, which are based on the fixed-wing UAVs, require complex and expensive launching and landing systems or even runways. This makes them unsuitable for many rapid response operations and activities, especially in remote and unpopulated areas, as this disadvantage introduces delays related to the preparation of the drone or the setup of the launching mechanisms.

Nowadays, the battery powered UAVs are the dominant technology on the market. The present technological state of the battery manufacturing processes is allowing the drones to stay in the air for up to 90 minutes. However, the flight duration is extremely dependant on the environmental conditions and on various drone characteristics, like the number of the motors, the wight of the drone, etc. While the available flight time is suitable for many different activities, the fire-related actions can demand longer operation periods.

By taking under consideration the majority of the presented and discussed disadvantages of the UAVs and the related to them technologies, we have developed a complex fire detection and observation system, which was named THEASIS-M (Fig.1). This system is based on previous research efforts and practical experiments conducted under the SFEDA transnational project, which was completed by several institutions from Greece, Cyprus and Bulgaria [19].

The primary UAV in the THEASIS-M system is a medium altitude fixed-wing UAV. This drone provides the overall forest monitoring capabilities of the system and is used for the initial detection of fire and smoke in the observed area. The detection can be accomplished with the use of both the optical or the thermal camera of the drone. To obtain a constant monitoring of the planned zone, the primary UAV is set to maintain its predefined initial flightpath, even if smoke or fire is detected mid-flight. In this way, the THEASIS-M system can report the detection of fire and smoke in multiple locations, instead on just focusing on the first one, which can also be false-positive.

Once the primary UAV reports the detection of fire or smoke, the estimated coordinates of the event are transmitted to a ground control station, logged and recorded in a database and then forwarded to a secondary UAV which is a low altitude UAV (Fig. 1). The purpose of using the secondary UAV is to conduct a closer inspection at the location of the reported event and then to confirm or deny the presence of fire or smoke. The secondary UAV can be a multi-rotor UAV with a high-resolution optical and thermal camera, but with reduced flight altitude and duration capabilities.

B. *Neural networks, Computer-Vision algorithms and Artificial Intelligence technologies for detections of fire and smoke*

With the rapid advancements in the area of the Computer Vision (CV) and Artificial Intelligence (AI), modern smart visual surveillance systems have been successfully developed for many applications, including for detection of fire and smoke, for real-time evaluation and optimal decision-making in firefighting and rescue operations, etc. [5-7]. Fundamentally, the ability of the machines to learn, gain experience, adjust to new inputs, make adaptive decisions and perform different tasks in a human-like manner is generally described under the popular term Artificial Intelligence. Machine learning (ML) is an AI technology that allows the computer systems to predict outcomes. For this purpose, the system uses historical data as input and predicts new output values. The ML methods, which are used in the development of an AI system, can vary. Some of the most widely known technologies are the symbolic regression, support vector machines, decision trees and random forests, as well as the algorithm for k-nearest neighbours.

The neural networks are probably the most popular among all of the machine learning technologies. There are several types of neural networks, but probably the most widely used for image detection and computer vision are the convolutional neural networks. Usually, they are formed by an input layer, several interconnected or fully connected hidden layers with multiple neurons, where the computation and decision making takes place, and an output layer, as shown in Fig. 2. The links between neurons in the network are parameterized with weights, which dictate the importance of the input value.

In order to develop an AI-solution for the fire and smoke detection, individual frames have to be periodically extracted from the real-time video stream from the UAV. These still images are then inputted into a pretrained neural network for analysis and decision-making. Although the input can be provided to an untrained neural network, the trained ones are outputting much better results. The algorithms designed to do object detection are categorized into two major groups, including one-stage and two-stage object detectors. The members of the first group are characterised with high inference speeds, while these from the second group have high localization and recognition accuracy.

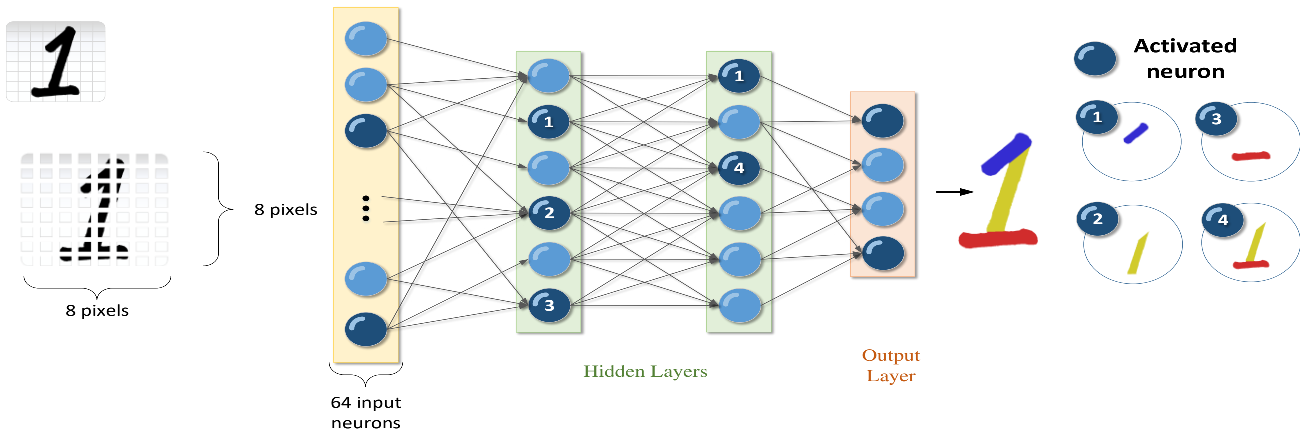


Fig.2. A general neural network model used in image detection, recognition and computer vision

In our study, we have used the Region-based Convolutional Neural Network (R-CNN) approach, which combines rectangular region proposals with convolutional neural network features [8, 9]. R-CNN is a two-stage detection algorithm, where the first stage identifies a subset of regions in an image that might contain an object and the second stage classifies the object in each region. The input data used for the training of the neural network consists of 1080 images and is divided in a set for training of the network (864 images or 80% of all used images) and a set for testing (the remaining 20% of all images or 216 images). If more images are used for the training set, the model could become more accurate. In this case, however, there is a trade-off between the model speed and the model accuracy, which must be taken into consideration.

C. *Selection of the functional components and the elements of the THEASIS-M system*

Prior to the testing and experimental evaluation of the proposed system for early detection of forest fires, the drone market was analysed and studied and the UAVs most suitable for the planned purposes were selected, purchased, modified and deployed.

The ALTi Transition-F vertical take-off and landing (VTOL) fixed-wing UAV [10], was selected as the primary aircraft for the discussed system. The ALTi Transition-F is one of the leading fixed-wing UAVs on the market. The Technical specifications of the drone are presented in Table I.

The ALTi Transition-F is ultra-compact, efficient and affordable system, which is able to take-off and land vertically in space-restricted locations using its four battery powered motors, with the following key technical specifications: (i) the flight endurance is up to 12 hours, which is achieved using an internal combustion engine, (ii) the dimensions of the wingspan, length and height are respectively 3000 mm, 2300 mm and 525 mm; (iii) the maximum take-off weight of the drone is 16 kg.

The main wings of the ALTi Transition-F are removable as shown in Fig. 3, which significantly reduces the UAV carry-size and allows for the rapid deployment, transport and storage.

TABLE I. SPECIFICATIONS OF THE UAV ALTI TRANSITION F [11]

Parameter	Description
Transportation Case	Pelican Air 1555 Travel Case
GCS Computer	Intel NUC 7i 3BNH + Logitech Keyboard & Mouse
GSC Display	ASUS 15.6" LED HD Monitor
Digital Data/Video Link	Microhard MIMO 2.4Ghz PMDDL24S0 ENC
Airside 2.4Ghz Antenna	Omni directional aircraft antenna
GSC 2.4Ghz Antenna	Omni directional GCS antenna
Aircraft Controller	Spektrum DX Controller
Long Range C2 Control Link	TBS Crossfire System
GCS Panel System	GCS enclosure with voltage monitor and connectors
Power Battery Pack	GCS power pack battery set

The UAV in our experiments was equipped with a NightHawk 2 EO/IR camera [11] with 20x zoom and thermal image resolution of 640x480 pixels. The camera weighs only 250 grams, which causes almost no effect on the drone performance and is not reducing its endurance significantly.

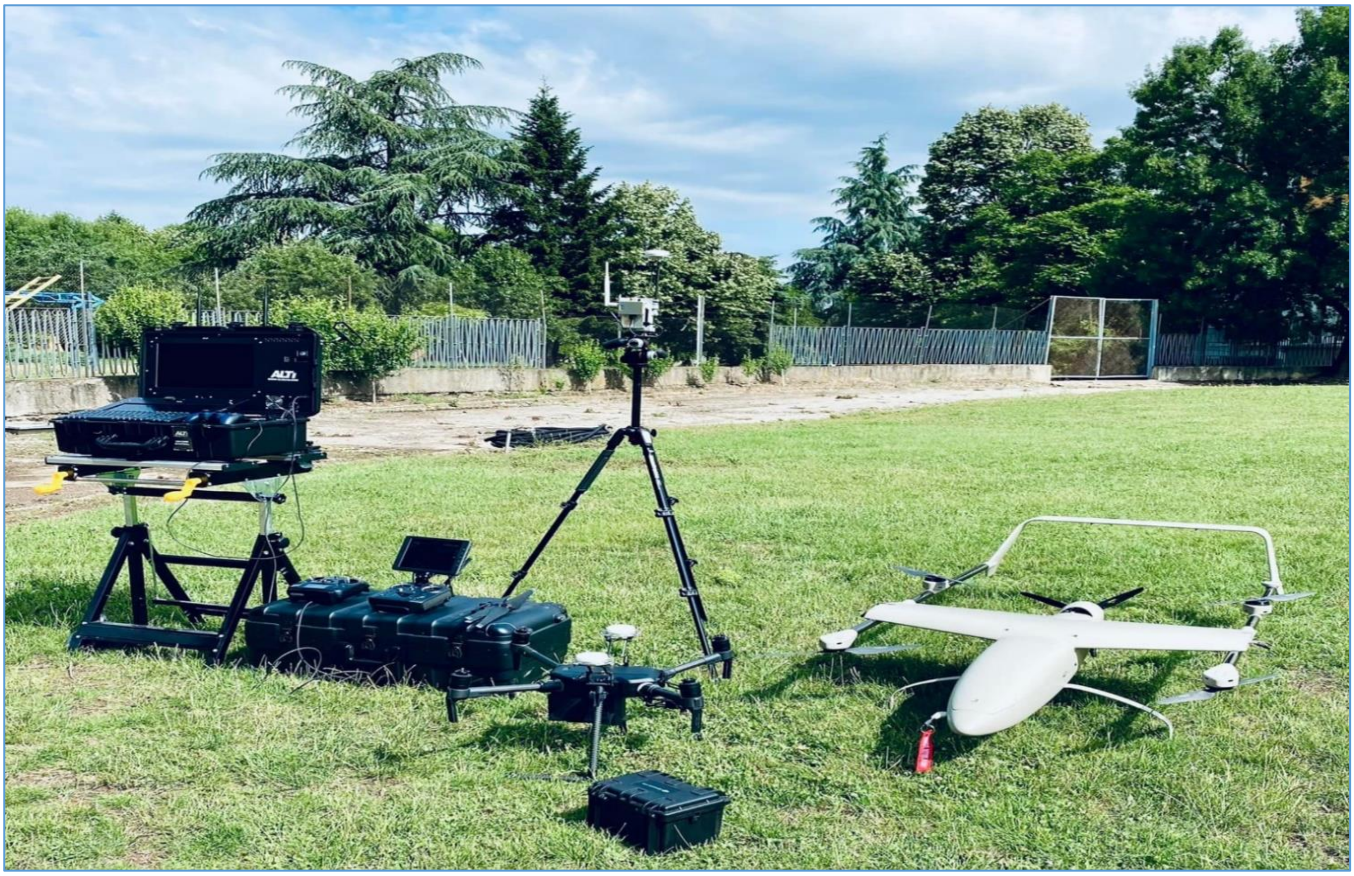


Fig.3. The UAVs used in the THEASIS-M system for early detection and monitoring of wildfires – (from left to right) the ground control station of the ALTi Transition F, the DJI Matrice M210 RTK drone with its case, remote controller and RTK station and the ALTi Transition F with its wings removed.

To confirm or deny the detection of the fire or smoke, the DJI Matrice 210 RTK drone [12] was selected as the secondary rotary-wing UAV in the THEASIS-M system, as shown in Fig. 3. The technical specifications of the DJI Matrice 210 RTK are presented in Table II.

TABLE II. SPECIFICATIONS OF THE DJI MATRICE M210 DRONE [12]

Parameter	Description
Package Dimensions	790×390×290mm
Dimensions (unfolded)	887×880×378 mm
Dimensions (folded)	716×220×236 mm
Folding Method	Folded Inward
Diagonal Wheelbase	643 mm
Number of Batteries	2
Weight (TB55)	Approx.4.57kg (with two standard batteries)
Max Take-off Weight	6.14KG
Max Payload (2 TB55)	Approx.1.57kg (with two standard batteries)
Folding Method	Folded Inward
Diagonal Wheelbase	643 mm
Max Wind Resistance	12 m/s
Operating Temperature	-4° to 113° F (-20° to 45° C)
IP Rating	IP43

The DJI Matrice 210 RTK drone is IP 43 certified, therefore, it can withstand humidity and can fly in the foggy or rainy conditions. In addition, the drone has a dual downward gimbal, which allows it to carry simultaneously two cameras. The DJI Matrice UAV, in our experiments, was equipped with an Zenmuse XT2 thermal camera [13], which integrates a high-resolution FLIR thermal sensor and a 4K visual camera with stabilization and processing technology for fast transformation of the aerial data into powerful insights.

III. EXPERIMENTAL DEMONSTRATIONS AND RESULTS

To evaluate the functionality of the developed THEASIS-M system and its efficiency, several experimental tests were carried out. Some of them were aimed at the testing and evaluation of the AI-based solution for the fire and smoke detection (Fig.4, left), while others were actual experiments with the full-scale system (Fig. 4, right).

The actual field experiments with the system were conducted on the territory of a national park in the Northern-central Region of Bulgaria as shown in Fig. 5. To test the efficiency of the system, several smoke traps and controlled fire sites were created.

The developed THEASIS-M system is a proof-of-concept that several UAVs can be used within one system and can provide the reliable and efficient detection of the forest fires. In all experiments, the detection algorithm of the primary UAV in the system was able to identify the smoke or fire locations and the coordinates of the sites were sent to the secondary UAV for the closer inspection and confirmation of the events.

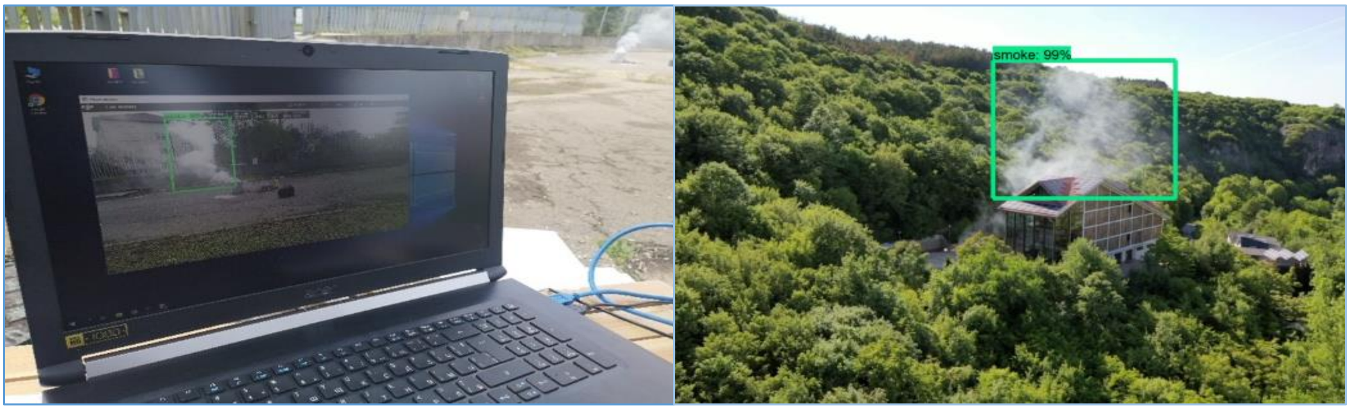


Fig.4. Evaluation of the AI-based solution for detection of fire and smoke during the ground-based tests (left) and real-time detection of smoke during the experimental tests at the nature park (right).

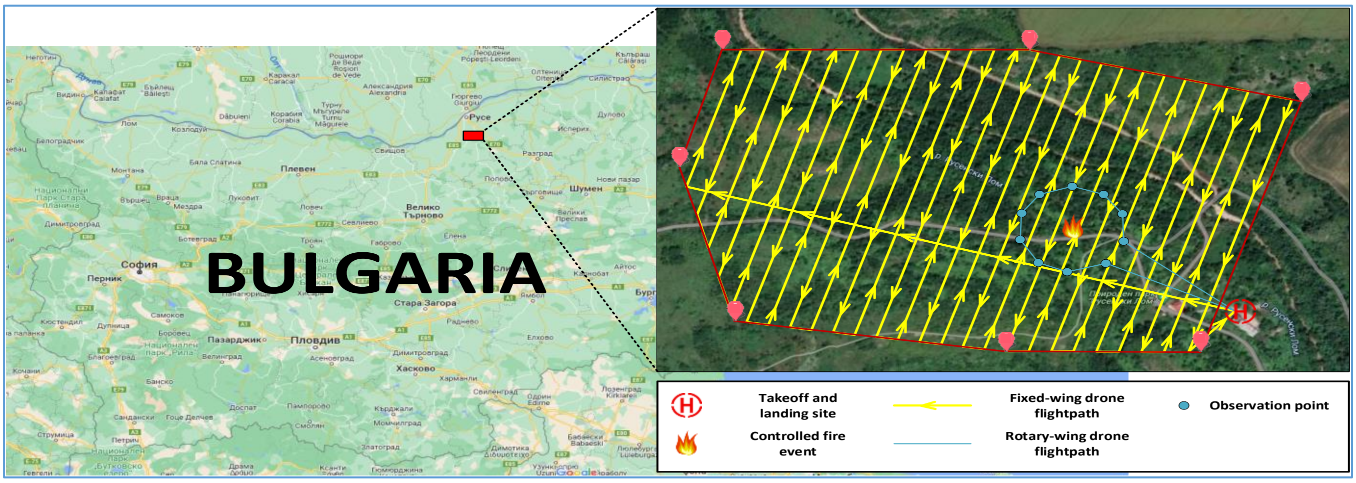


Fig.5. The location of the test site for the THEASIS-M system with the flightpath for the fixed-wing UAV, one of the controlled fire sites and the flightpath undertaken by the secondary UAV upon the report of the detected fire event.

IV. Summary and conclusions

The use of aerial systems for fire detection, monitoring and extinguishing is not a new concept. Prior to the wide acceptance of the UAV technologies, the fire detection, monitoring and fighting activities were implemented using manned aircrafts – special firefighting airplanes and helicopters, which have huge internal or external water tanks. The quick response time and the improved access to remote areas made the manned aircrafts crucial and highly demanded tools in the fight against wildfires. Nevertheless, firefighting is a very dangerous activity and is related to numerous risks, which have led to many incidents and the loss of firefighting aircrafts and crews. Therefore, there is an emerging need to look for a less risky solution to deal with the fire detection, monitoring and extinguishing.

The advances in the area of the micro and nano technologies and the spacecraft construction made the development of general-purpose non-military satellites possible. Naturally, it was a logical thing to develop solutions and systems for fire detection and monitoring from space. While this technology is characterized by zero direct risk to humans, it is unfortunately also very expensive to implement and very complex to manage and operate. Recently, it became possible to use the advancement in the UAV-related technologies in the forestry domain [1-9, 14-18], including for fire detection, monitoring and suppression.

This study presented the conceptual model of the THEASIS-M smart system for early detection of forest fires and a successfully developed and tested proof-of-concept prototype that is based on this model. The system includes a fixed-wing UAV, which provides the overall forest monitoring capabilities, and a rotary-wing UAV, which is used for confirmation and monitoring of the detected fire event. The THEASIS-M system can also be implemented using a set of two rotary-wing UAVs. The proposed solution can be subject to additional improvements and can be integrated with the available fire detection systems, including these based on stationary cameras or satellite images.

Acknowledgments

This work has been accomplished with the financial support from the grant number BG05M2OP001-1.002-0011-C02, financed by the Bulgarian Operational Programme "Science and Education for Smart Growth" (SESG) (2014-2020), and co-financed by the European Union through the European Structural and Investment Funds (ESIF).

This work was also supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Smart Crop Production” approved by Decision of the Ministry Council №866/26.11.2020.

This work has been partly supported for sustainable and collaborative research partnerships by and the Research Environment Links programme with the grant number 528085858, under the Newton Fund partnership funded by the UK Department for Business, Energy and Industrial Strategy and delivered by the British Council and by the research foundation of the Le Quy Don Technical University and the Thu Dau Mot University, Vietnam.

References

[1] S. Sudhakar *et al.* (2020) Unmanned Aerial Vehicle (UAV) based forest fire detection and monitoring for reducing false alarms in forest-fires. Computer Communications: 149, 2020, pp. 1-16. https://doi.org/10.1016/j.comcom.2019.10.007

[2] Thiel C. *et al.* (2020) Monitoring selective logging in a pine-dominated forest in central Germany with repeated drone flights utilizing a low cost RTK quadcopter. Drones 2020, 4, 11. https://doi.org/10.3390/drones4020011

[3] Sharifah M. S. M. D. *et al.* (2020) Applications of drone in disaster management: A scoping review. Science & Justice: 62 (1), 2022, pp. 30-42. https://doi.org/10.1016/j.scijus.2021.11.002

[4] Abderahman Rejeb *et al.* (2022) Drones in agriculture: A review and bibliometric analysis. Computers and Electronics in Agriculture: 198, 2022, 107017. https://doi.org/10.1016/j.compag.2022.107017

[5] Zilong Wang *et al.* (2022) Predicting transient building fire based on external smoke images and deep learning. Journal of Building Engineering: 47 (15), 2022, 103823. https://doi.org/10.1016/j.jobe.2021.103823

[6] Ali Hosseini, Mahdi Hashemzadeh and Nacer Farajzadeh (2022). UFS-Net: A unified flame and smoke detection method for early detection of fire in video surveillance applications using CNNs. Journal of Computational Science: 61, 2022, 101638. https://doi.org/10.1016/j.jocs.2022.101638

[7] Seong G.Kong *et al.* (2016) Fast fire flame detection in surveillance video using logistic regression and temporal smoothing. Fire Safety Journal: 79, 2016, pp. 37-43. https://doi.org/10.1016/j.firesaf.2015.11.015

[8] G. D. Georgiev *et al.* (2020) Forest Monitoring System for Early Fire Detection Based on Convolutional Neural Network and UAV imagery. In proceeding of 28th National Conference with International Participation TELECOM-2020, pp. 57-60. DOI: 10.1109/TELECOM50385.2020.9299566

[9] D. Kinaneva *et al.* (2020) An artificial intelligence approach to real-time automatic smoke detection by unmanned aerial vehicles and forest observation systems. 2020 International Conference on Biomedical Innovations and Applications (BIA): 2020, pp. 133-138, doi: 10.1109/BIA50171.2020.9244498

[10] ALTi Transition-F vertical take-off and landing (VTOL) fixed-wing UAV. Available at: www.altiuas.com [Access: 9/2022].

[11] NightHawk 2 EO/IR camera. Available at: www.nextvision-sys.com/nighthawk-2 [Access: 9/2022].

[12] The DJI Matrice 210 RTK unmanned aerial vehicle. Available at: www.dji.com [Access: 9/2022].

[13] The Zenmuse XT2 thermal camera. Available at: www.heliguy.com [Access: 9/2022].

[14] S.Karma *et al.* (2015) Use of unmanned vehicles in search and rescue operations in forest fires: Advantages and limitations observed in a field trial. International Journal of Disaster Risk Reduction: 13, 2015, pp. 307-312, https://doi.org/10.1016/j.ijdrr.2015.07.009

[15] M. Mohan, C.A. Silva, C. Klauberg, P. Jat, G. Catts, A. Cardil, A.T. Hudak, and M. Dia (2017) Individual Tree Detection from Unmanned Aerial Vehicle (UAV) Derived Canopy Height Model in an Open Canopy Mixed Conifer Forest. Forests 8(9):340, https://doi.org/10.3390/f8090340

[16] C. L. Scher, E. Griffoul and C. H. Cannon (2019) Drone-based photogrammetry for the construction of high-resolution models of individual trees. Trees vol. 33/2019, pp. 1385–1397. https://doi.org/10.1007/s00468-019-01866-x

[17] J. Li, B. Yang, Y. Cong, L. Cao, X. Fu and Z. Dong (2019) 3D Forest Mapping Using A Low-Cost UAV Laser Scanning System: Investigation and Comparison. Remote Sensing, 11(6):717 2019, https://doi.org/10.3390/rs11060717

[18] M. Pierzchała, P. Giguère, R. Astrup (2018) Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. Computers and Electronics in Agriculture, Volume 145, 2018, pp. 217-225, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2017.12.034

[19] Official webpage of the SFEDA transnational project. Available at: www.interreg-balkanmed.eu/approved-project/22/ [Access: 9/2022].

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 141–144
DOI: 10.15439/2022R17
ISSN 2300-5963 ACSIS, Vol. 33

Proof-of-Miner-Clustering-Authentication Consensus Method of Blockchain for IoT Networks

1st Tam T. Huynh
Faculty of Information Technology
Posts and Telecommunications
Institute of Technology Ho Chi Minh
City,
Vietnam tamht@ptithcm.edu.vn

2st Chinh N. Huynh
Faculty of Information Technology
Ho Chi Minh City University of
Technology and Education Ho Chi
Minh City, Vietnam
chinhhn@fit.hcmute.edu.vn

3st Thanh H. Nguyen
Faculty of Information Technology
Posts and Telecommunications
Institute of Technology
Ho Chi Minh City, Vietnam
thanhnh@ptithcm.edu.vn

***Abstract*—Using blockchain technology in the Internet of Things (IoT) security is a research trend in recent years. With large IoT networks, miners will have to verify a lot of transactions broadcast from IoT devices. This can cause a delay in saving valid transactions to the ledger. This paper proposes a proof-of-miner-clustering-authentication consensus method of blockchain for IoT networks. In the proposed method, miners in a blockchain network will be clustered, each miner in a cluster is responsible for verifying transactions from IoT devices in the area it manages. Mining of new blocks between clusters is done by the round robin method. Our consensus method can apply to private or consortium blockchain networks, helping to improve the transaction verification speed of miners.**

***Index Terms*—blockchain, IoT, consensus.**

I. Introduction

In today's rapidly developing digital technology era, the number and types of IoT devices being put into use are increasing day by day. The International Data Corporation forecasts the quantity of Internet-connected IoT devices reach 150 billion by 2025 [1]. Security for IoT networks are very important and urgent nowadays. With the current development trend of IoT, the use of a security platform based on blockchain for large IoT networks with high scalability needs is a suitable solution, because this technology has many advantages, such as decentralization, anonymity, and accountability [2-3].

In 2008, Satoshi Nakamoto introduced blockchain technology, which is a block-linked list [4]. Each block has a hash pointer that connects it to its parent block and stores the predecessor's hash value at a specific time. Genesis block is the name of the chain's initial block. A block structure includes a header which contains information management of the block, and a body containing valid transactions.

A blockchain network has two types of nodes: user nodes and miner nodes. Transactions can be carried out by User nodes, meanwhile, Miners hold the ledger that records a series of verified blocks. A consensus protocol is used in a blockchain network to synchronize ledger data between miners. Some consensus protocols include Proof-of-work (PoW), Proof-of-stake (PoS), Proof-of-activity (PoA) [5]. Blockchain is classified into three types: public blockchain, private blockchain, and consortium blockchain [6].

Regarding the management model of an IoT network, usually an IoT network is managed by one or several organizations. In the case of an IoT network managed by an organization, the organization can set up a private blockchain for the security platform. In case the IoT network is managed by several organizations, a consortium blockchain can be used for the security platform. Two important components in a blockchain network are miners and a consensus protocol. Miners need high computing performance and large enough storage capacity to verify transactions and store data for the entire network. The consensus mechanism is used in a blockchain network to synchronize data on the miners' ledger.

Normally, the data consensus process in a blockchain network is as follows: (1) when a node in a network performs a blockchain transaction, the transaction is broadcast to all network-connected miners; (2) These transactions will be saved in each miner's pool; (3) in each cycle of mining, a miner verifies and places valid transactions in a new block. The other miners will be informed about this new block, and this miner will also save this new block in its ledger; (4) after receiving this new block, the miners check the block's legitimacy, and if it is true, they add it to their ledger. Obviously, given the large size of IoT networks both in terms of the number of devices and geographic coverage, miners will receive and process a lot of transactions. This can cause delays in saving valid transactions to the blockchain ledger.

There have been many consensus methods applied to blockchain for IoT such as PoW, PoS, PoAh [10, 11, 16]. However, solutions using these consensus methods have not been mentioned to apply to large IoT networks. Therefore, this paper proposes a novel consensus approach based on the clustering of miners. The proposed method can be applied to large IoT networks both in terms of the number of devices and geographic coverage. In this method, miners in a private network or consortium blockchain will be grouped into multiple clusters, miners in each cluster will be responsible for verifying transactions from IoT devices in the area that the cluster manages. The proposed method helps to speed up the transaction verification and data consensus on the blockchain ledger.

The remainder of this paper is organized as follows. Section II reviews the related works. Section III describes the proposed method. Finally, our conclusion and future works are given in Section IV.

II. Related Works

Oscar Novo proposed a blockchain-based access control architecture for IoT. In this architecture, Management Hubs are used to manage IoT devices and act as a bridge between IoT devices and the blockchain network. The proposed architecture uses Ethereum for the private blockchain network [7]. Liu et al. [8] introduced a platform that ensures the integrity of IoT data stored on a cloud storage service, this secure platform uses the Ethereum blockchain. Panda et al. [9] proposed an authentication platform for IoT devices that uses the Ethereum blockchain. Sheron et al. [10] proposed a secure platform that provides a communication method that ensures privacy and integrity in the IoT environment, which uses the PoW consensus protocol.

The authors in [11] introduced a secure communication platform for IoT networks. The platform is implemented in the consortium blockchain network and uses a combined consensus algorithm. Overall, this consensus algorithm is similar to the PoS consensus protocol. Currently, the PoW consensus mechanism on the Ethereum blockchain is being replaced by the PoS consensus protocol since it uses quite a lot of electrical energy and processing resources[12]. Li Yang et al. [13] proposed a distributed consensus algorithm for blockchains on multi-hop IoT networks. The authors in [14] introduced a model that makes the consensus approach more energy efficient, utilizes less memory, and less processing time. Manal Mohamed Alhejazi et al. [15] presented Weighted Majority Consensus Algorithm for IoT systems.

The general architecture of the blockchain network of the solutions surveyed above is that all miners are not clustered, so all transactions from IoT devices propagate to all miners in the network. Therefore, when the IoT network size increases in both the number of connected devices and the geographic coverage, miners will have to process a large number of transactions in the network. This can greatly affect the processing performance of miners and can cause delays in saving transactions on the blockchain ledger. Moreover, blockchain networks that use the PoW consensus protocol, it is costly in terms of electrical energy consumed by miners and is not suitable for private networks or blockchain consortiums.

III. The Proposed Consensus Method

A. System Model

Figure 1 illustrates the architecture of a blockchain network for IoT, consisting of three clusters. Each of clusters has two miners and IoT devices, clustering can be based on the geographical location of miners in the network. The number of clusters and miners in each cluster is determined by the size of the IoT network. Administrator is possible to add IoT devices to the clusters.

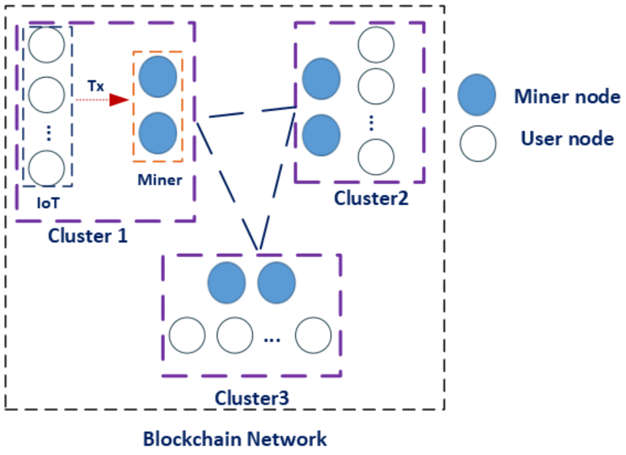


Fig. 1. The system model.

The system has two kinds of nodes:

- Miner Node: Miners in a cluster verify the transactions of IoT devices in that cluster, can create new blocks on the blockchain ledger.
- User Node: Each IoT device is a User Node in the blockchain network and can perform blockchain transactions on the network.

B. System Policies

The policies are implemented on miner nodes and user nodes as follows:

- Miner Node: Each miner maintains a list containing the communication addresses of other miners and a list of blockchain addresses of miners in the blockchain network. The administrator will configure this list on the miners.
- User Node: Each IoT device in a cluster knows the addresses of the miners in that cluster. When an IoT device in a cluster performs a transaction, the transaction is broadcast only to the miners in its cluster.

C. System Setup

In this work, it is denoted that $PCS(x, k)$ is a public-key cryptosystem with a message x and a key k. H is a cryptographic hash function. Symbol || is an operation of string concatenation. Let n be the number of miners in the network, and m be the number of IoT devices.

Each node in the system is generated a key pair by a public key cryptographic algorithm. Specifically, $PK_{m[i]}$ and $SK_{m[i]}$ are the public key and the corresponding private keys of the i-th miner, $0 < i \leq n$; $PK_{d[j]}$ and $SK_{d[j]}$ are the public key and the corresponding private key of the j-th IoT device, respectively, $0 < j \leq m$. In the blockchain network, miners and IoT devices use their private keys to create digital signatures and their public key as their blockchain address. The administrator also generates a public blockchain transaction address that is shared by the whole system, denoted SYS_Add. We denote $L = (PK_{m[1],}PK_{m[2]}, \dots, PK_{m[n]})$ as a list of miners' blockchain address in the network.

D. Consensus method

The consensus method includes three steps as follows:

- Step 1: an IoT device $d[j]$ performs a blockchain transaction, this device uses its private key to create a digital signature on that transaction, then this transaction along with the digital signature will be broadcast to the miners in its cluster.

  (i) The structure of a transaction as follows: *Tx={"Sender":<Blockchain_Address_of_Sender>, "Receiver":< <Blockchain_Address_of_Receiver>, "Content":<Content_of_Transaction>}.*

  (ii) Generating digital signature:

  $$h \leftarrow H(Tx)$$

  $$Sig \leftarrow PCS(h, SK_{d[j]})$$

  (iii) Broadcasting $(Tx||Sig)$ to miners in its cluster.

- Step 2: At a mining round, a selected miner $m[i]$ in the cluster will verify the signature on the received transactions (in its pool), if the digital signature is valid, the transaction will be considered valid. This miner puts valid transactions in a new block, then forms a digital signature on this new block and distributes it to other miners in the network along with the digital signature. Note that the number of transactions in a block is determined by the size of each transaction as well as the system policies for each specific application.

(i) Verifying Tx:

$$h \leftarrow PCS\left(Sig, PK_{d[j]}\right)$$

Where $PK_{d[j]}$ is the sender address field of Tx

$$h' \leftarrow H(Tx)$$

$$True/False \leftarrow (h' == h)$$

(ii) Creating a new block denoted b.

(iii) Generating digital signature on b

$$h\_b \leftarrow H(b)$$

$$Sig\_b \leftarrow PCS(h\_b, SK_{m[i]})$$

(iv) Broadcasting $(b||Sig\_b)$ to other miners

- Step 3: After receiving a new block along with a digital signature, the miners verify: (1) verify digital signature; and (2) Verify the miner who advertises this new block is on the list of miners in the network. If these two conditions are met, this new block will be added to the miners' ledger.

(i) Verifying digital signature:

$$h\_p \leftarrow PCS(Sig\_b, PK_{m[i]})$$

$$h'\_p \leftarrow H(b)$$

$$True/False \leftarrow (h\_p == h'\_p)$$

(ii) Verifying $PK_{m[i]}$ in L:

$$True/False \leftarrow (PK_{m[i]} \in L)$$

The proposed consensus process is shown in Fig. 2.

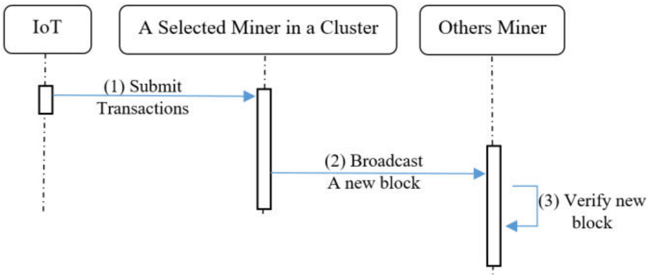


Fig. 2. The process of consensus method.

E. Mining round setup

As mentioned in Step 2 of the consensus method, at each mining round, a miner will be selected for a new block proposal. In this section, we will detail how to select miners at each cluster and when to start a mining round.

For the time to start a mining round: we assume that Δt is the time it takes for miners in a cluster to receive and verify a new block which broadcasts from a miner of the other cluster. we also assume that the computing power of each miner is the same. Proposing new blocks on the ledger will be done in the round robin method between clusters with the quantum time Δt. Specifically, the first block will be proposed by a miner in cluster 1. the second block will be done by a miner of cluster 2, the third block will be proposed by a miner in cluster 3, cluster 1 will propose the 4th block, and so on. New blocks will be proposed every Δt time. The round robin mining of clusters is shown in Figure 3.

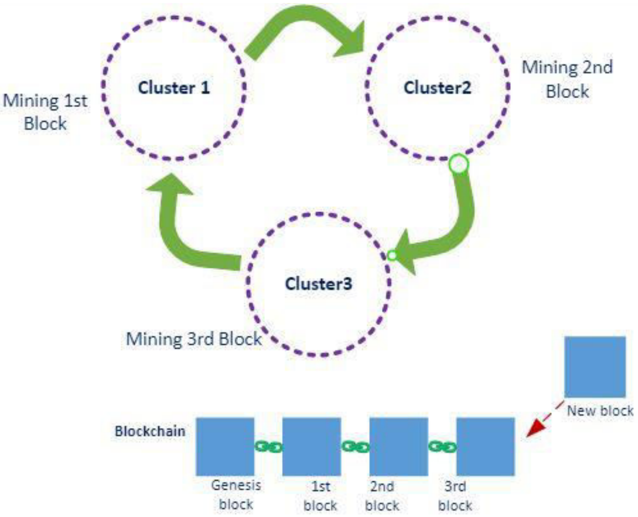


Fig. 3. The round robin mining of clusters.

The field "*Cluster_Number*" in the block header will be used to contain the cluster number that the block has mined, as shown in Figure 4.

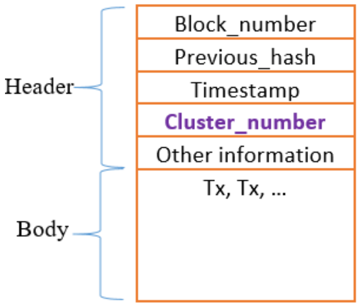


Fig. 4. The block structure.

For selecting a miner at each cluster: the administrator can use PoS consensus mechanism in the clusters, in which a fixed miner in each cluster will be selected to verify transactions and propose new blocks. In case something goes wrong with the selected miner, the administrator can transfer the mining role to another miner.

F. Discussion

Our consensus protocol is predicated on the authenticity of the event source. Specifically, IoT devices must sign on their transactions. Miners in the cluster verify the signature on that transactions; After mining a new block, the miner generates a signature on that block, then broadcast them to other miners in the network. The signature on the new block is used to verify the validation of the block.

Miners are divided into clusters by geographical area, each of which will manage the transactions of those central IoT devices. That improves mining speed in large IoT networks. However, all miners in the proposed blockchain architecture are honest miners. This means that it is very difficult for miners to be compromised by attackers and they also do not commit any fraud in the blockchain network.

IV. Conclusion and Future Work

This paper proposes a new consensus method of blockchain for IoT networks, it is named Proof-of-Miner-Clustering-Authentication Consensus. In the blockchain network architecture for IoT, we group miners into clusters. Each miner cluster is responsible for verifying transactions from devices in that cluster. Mining of new blocks is performed by the round robin method between clusters. The proposed consensus method can be used in private or consortium blockchain for IoT networks.

In our future work, we will apply the proposed consensus method to certain applications evaluate them, and subsequently improve the method.

References

[1] A. Patrizio, “IDC: Expect 175 zettabytes of data worldwide by 2025,” Network World, 2018.

[2] A. M. Antonopoulos, “Mastering Bitcoin: unlocking digital cryptocurrencies,” O’Reilly Me-dia, Inc, 2014.

[3] Z. Zheng, S. Xie, H. N. Dai, and H. Wang, “Blockchain challenges and opportunities: A survey,” in International Journal of Web and Grid Services, 14(4), pp. 352-375, 2018.

[4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system. Manubot,” 2008.

[5] T. T. Huynh, T. D. Nguyen, and H. Tan, “A Survey on Security and Privacy Issues of Blockchain Technology,” in 2019 International Conference on System Science and Engineering (ICSSE), IEEE, pp. 362-367, 2019.

[6] V. Buterin, “On Public and Private Blockchains. Ethereum Blog,” 2015. [Online]. Available: https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains.

[7] O. Novo, “Blockchain meets IoT: An architecture for scalable access management in IoT,” in IEEE internet of things journal, 5(2), pp. 1184-1195, 2018.

[8] B. Liu, X. L. Yu, S. Chen, X. Xu, and L. Zhu, “Blockchain based data integrity service framework for IoT data,” in 2017 IEEE International Conference on Web Services (ICWS), IEEE, pp. 468-475, 2017.

[9] S. S. Panda, U. Satapathy, B. K. Mohanta, D. Jena, and D. Gountia, “Blockchain Based Decentralized Authentication Framework for Resource Constrained IOT devices,” in 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, pp. 1-6, 2019.

[10] P. F. Sheron, K. P. Sridhar, S. Baskar, and P. M. Shakeel, “A decentralized scalable security framework for end-to-end authentication of future IoT communication,” Transactions on Emerging Telecommunications Technologies, 31(12), e3815, 2020.

[11] M. I. Khan, and I. A. Lawal, “Sec-IoT: A framework for secured decentralised IoT using blockchain-based technology,” in International Congress on Information and Communication Technology. Springer, Singapore, pp. 269-277, 2020.

[12] J. Li, N. Li, J. Peng, H. Cui, and Z. Wu, “Energy consumption of cryptocurrency Mining: A study of electricity consumption in Mining cryptocurrencies,” Energy, 168, pp. 160-168, 2019.

[13] L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu, and X. Cheng, “Distributed consensus for blockchains in internet-of-things networks,” Tsinghua Science and Technology, 27(5), pp. 817-831, 2022.

[14] S. Wadhwa, S. Rani, S. Verma, J. Shafi, and M. Wozniak, “Energy Efficient Consensus Approach of Blockchain for IoT Networks with Edge Computing,” Sensors, 22(10), 3733, 2022.

[15] M. M. Alhejazi, and R. M. A. Mohammad, “Enhancing the blockchain voting process in IoT using a novel blockchain Weighted Majority Consensus Algorithm (WMCA),” Information Security Journal: A Global Perspective, 31(2), pp. 125-143, 2022.

[16] D. Puthal, S. P. Mohanty, P., Nanda, E. Kougianos, and G. Das, “Proof-of-authentication for scalable blockchain in resource-constrained distributed systems,” in 2019 IEEE international conference on consumer electronics (ICCE), IEEE, pp. 1-5, 2019.

DOI: 10.15439/2022R30
ISSN 2300-5963 ACSIS, Vol. 33

Optimized Approach of Feature Selection Based on Binary Genetic Algorithm in Classification of Induction Motor Faults

Truong-An Le
Institute Of Engineering Technology
Thu Dau Mot University
Binh Duong, Vietnam
anlt@tdmu.edu.vn

***Abstract*—In this paper, an effective model for detection and classification of multiple faults in induction motors is presented. Signal analysis method S-transform is applied to analyze and extract features from the current signals of four test motor states including three fault states (bearing fault, broken rotor bar, stator winding short-circuit) and one normal state. The feature set is extracted based on signal spectrum. With strong exploration capabilities in the search space, binary genetic algorithm (BGA) is proposed to select the optimal feature subset. As the classifier, the backpropagation neural network and support vector machine are used. The simulation results showed that the average accuracy of 100 trails is 98.3% and the optimal feature subset equal to 36% of total original features. As such, 64% of irrelevant features have been removed. In conclusion, the proposed model combined with BGA reached highly effective in the classification of induction motor.**

***Index Terms*—S-transform, binary genetic algorithm, SVM, BPNN, feature selection, fault detection.**

I. Introduction

Induction motors are widely used in many fields, especially in industry. How to make the electric motor work stably, that can detect faults early to avoid serious damage is a worth considering. The priority tasks to ensure the longevity of electric motors are maintenance and fault diagnosis [1]. In fact, many incidents have occurred leading to serious losses in production [2]. Therefore, the main topics mentioned include condition monitoring and fault diagnosis of induction motors in healthy and faulty conditions; detect broken bars fault, bearing damage, and inter-turn short circuit fault in induction motors [3]. There have been many studies focusing on data analysis methods such as Fast Fourier transform (FFT) [4], Wavelet transform (WT) [5], short-time Fourier transform (STFT) [6], S-transform (ST), and Hilbert-Huang transform (HHT) [7] which are combined with the classification models use supervised learning algorithms such as k-nearest neighbor (k-NN), support vector machines (SVMs) and artificial neural networks (ANNs). However, signal analysis methods are used to extract characteristics of signals in the time-frequency domain. And then intelligent classification models were applied to solve the classification problem, which does not achieve high efficiency. Because the extracted features are originally based on the human experience. In which there are features that carry very little information, they are not effective for the classification process. Therefore, the feature selection techniques are used to select the most important features, increasing classification efficiency and reducing computation time [8-11].

In recent years, there has been a lot of research using optimization algorithms for the feature selection process to remove irrelevant and redundant features from datasets to improve the performance of the machine learning algorithms [12]. There are many optimization algorithms used such as ant colony optimization (ACO), particle swarm optimization (PSO), and their variants. One of them is the genetic algorithm (GA). The main idea of GA is to combine different solution generation after generation to extract the best genes (candidates) from each. By doing so, it creates new and more fitted individuals. Besides, one of GA's outstanding abilities is exploring the search space. However, its limitation is the high computational cost. Based on the advantages of GA, this paper proposed a method to optimize the feature set extracted from the S-transform method based on the binary genetic algorithm to detect induction motor faults. Signal analysis method S-transform is applied to analyze and extract features from the current signals of four test motor states including three fault states (bearing fault, broken rotor bar, stator winding short-circuit) and one normal state. The feature extraction process of the current signals based on spectrum analysis is also implemented. Binary genetic algorithm (BGA) is used to select the most important feature to improve classification efficiency. This is a variant of GA that works in the binary region. Two well-known classification methods, back propagation neural network (BPNN) and SVM, are applied separately to compare the performance of the two models and select the best classifier.

In this study, the model of diagnosis and classification of motor faults works in three phases:

i. Feature extraction: S-transform method is adopted is adopted to analyze the current signals from the test motors. The information-carrying features of the signal are extracted in the time and frequency domain from the spectrum of the signal.

ii. Feature selection: BGA is applied to remove redundant and irrelevant features and find the optimal feature subset. The number of features in the optimal feature subset will be smaller than the original feature set, which means reducing the size of the dataset for the next classification process.

iii. Classification: Two common classifiers (BPNN and SVM) are applied. The efficiency of each classifier is evaluated and compared separately based on the input dataset which is the optimal feature subset of the BGA. Finally, the best fault detection model is found.

II. Feature Extraction

The S-transform is a generic form of the short-time Fourier transform (STFT) with a feature that can change the width of the window function. This can overcome other algorithms with fixed window functions in the frequency domain [13]. The S-transform spectrum contains important characteristics that are used to identify motor failures.

The STFT of signal is defined as (1)

$$STFT(\tau,f)=\int_{-\infty}^{\infty} h(t)g(\tau-t)e^{-j2\pi ft}dt \qquad (1)$$

where τ is spectral localization time, f is Fourier frequency and $g(t)$ is window function.

The S-Transform is obtained by defining Gaussian window function, shown as

$$g(t)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-t^2}{2\sigma^2}} \qquad (2)$$

The dilation (window width) σ is made proportional to the inverse of frequency, shown as

$$\sigma(f)=\frac{1}{a+b\left|f\right|} \qquad (3)$$

If $a=0$, $\sigma(f)$ means S-Transform and $\sigma(f)$ denotes STFT for $b=0$ [14].

Substituting (2) and (3) in (1), the S-Transform of $h(t)$ is obtained as

$$S(\tau,f)=\int_{-\infty}^{\infty} h(t)g(\tau-t,f)e^{-j2\pi ft}dt \qquad (4)$$

After the signal is analyzed by ST method, based on time-frequency matrix, feature extraction process is implemented as shown in Fig.1, including the following steps:

Step 1: The selection of features for the classification process is an important step to achieve high efficiency. In the time domain, five characteristics is selected including maximum (Max), minimum (Min), Mean, mean square error (Mse) and standard deviation (Std), to form five characteristic curves named Tmax, Tmin, Tmean, Tmse, and Tstd. Similarly, in the frequency domain, five same properties are selected and obtained the following five characteristic curves, named Fmax, Fmin, Fmean, Fmse, and Fstd.

Step 2: Each characteristic curve in the time and frequency domain will have five values extracted (Max, Min, Mean, Mse, and Std). Thus, 50 features are extracted from ten curves.

Step 3: The values of the characteristics are quite different. Therefore, we need to normalize the data before taking the next steps. A proposed simple normalized method is rescaled the feature values between 0 and 1.

Step 4: Perform feature selection method.

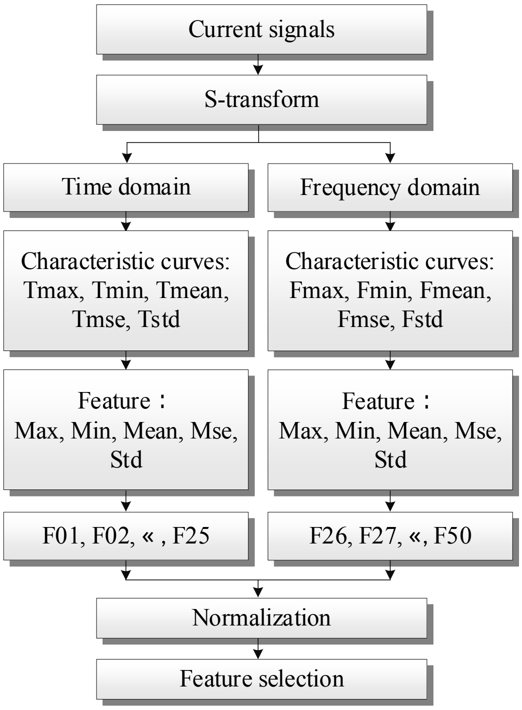

Current signals
S-transform
Time domain
Frequency domain
Characteristic curves: Tmax, Tmin, Tmean, Tmse, Tstd
Characteristic curves: Fmax, Fmin, Fmean, Fmse, Fstd
Feature：Max, Min, Mean, Mse, Std
Feature：Max, Min, Mean, Mse, Std
F01, F02, « , F25
F26, F27, «, F50
Normalization
Feature selection

Fig. 1. Feature extraction process

III. Feature Selection

GA is one of the evolutionary algorithms. It is inspired by natural selection. Genetic algorithms use operators inspired by biological processes such as mutation, crossover, and selection to solve search and optimization problems [15]. In a genetic algorithm, there are five main stages: coding of chromosomes, initialization of the population, evaluation of each chromosome, generation of new chromosomes based on genetic operations such as selection, crossover, and mutation. The final condition is the end of the evaluation and genetics process.

Currently, there are many studies applying GA to solve optimization problems, find optimal results on real number search space [16]. That approach is also known as continuous GA [17]. Unlike continuous GA, binary GA operates on binary search space. Then the chromosome is a bit string, also known as a binary chromosome. Inside the binary chromosomes are genes that carry the value 0 or 1. In this study, the binary chromosomes are feature subsets. In which, gene 0 represents the unselected feature with the corresponding position in the binary chromosome and gene 1 represents the selected feature. The initial population is a randomly generated set of chromosomes. Chromosomes are evaluated by a fitness function. In this study, the k-NN approach was used as a fitness function for chromosome evaluation because k-NN is an efficient classifier and especially does not cost much to compute. Fitness value is the classification error rate by k-NN. The classification error rate is calculated based on the number of false predictions out of the total number of prediction samples. Next is the process of generating new chromosomes (new individuals) based on the genetic inspiration in nature that good parents will produce good children. First the evaluated chromosomes will be ranked, the dominant individuals will be kept and passed on to the next generation. The rest of the population is used to create new individuals through crossover and mutation. Based on these mechanisms, new individuals are generated that have the good traits of the previous generation and are potentially characterized through crossover and mutation operations. It can be said that BGA in particular or GA in general have the ability to find excellent individuals in the local area and probe potential individuals in the global area. The setting parameters for the BGA are shown in Table 1. A flowchart of the feature selection approach of the BGA is shown in Fig. 2 and the step-by-step process is described as follows [18]:

Step 1: Randomly generate a population of n chromosomes. Note that the number of chromosomes in the population greatly affects the computation time of the algorithm.

Step 2: The chromosomes in the population were evaluated by the fitness function. The value of the chromosomes is expressed through the fitness value.

Step 3: The next generation is created through the following three operations:

Selection: Based on the fitness value ranking, parental chromosome pairs are selected.

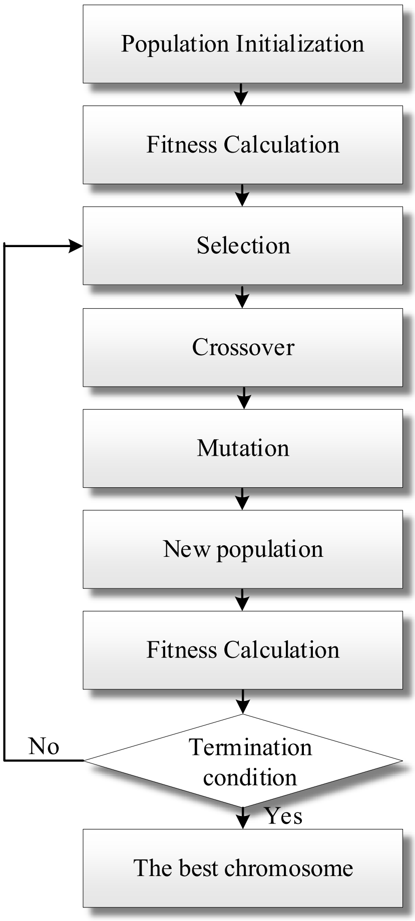


Fig. 2. Flowchart of BGA

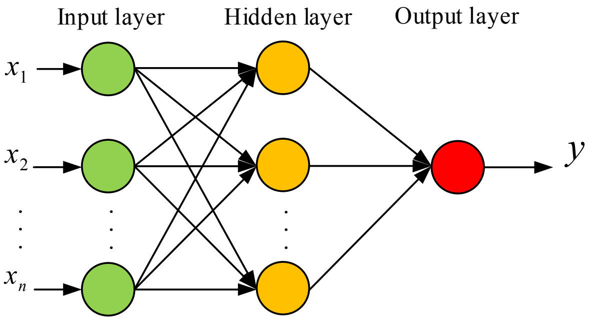


Fig. 3. Schematic diagram of BPNN

Crossover: Crossed between parents to produce a new individual with many good traits from the parents with a predetermined crossover probability.

Mutation: With a predetermined mutation probability, this operation causes mutations at several positions in the chromosome.

Step 4: The new generation is evaluated by fitness function. The value of the best individuals through each generation is recorded.

Step 5: If the end condition is satisfied, stop, and return the best solution in current population.

Step 6: Go to step 3 if the stop condition is not met.

IV. Classification

A. *Backpropagation Neural Network (BPNN)*

In the field of motor fault detection, artificial neural networks are showing superiority. The backpropagation algorithm belongs to a supervised learning group, which is the most common group of machine learning algorithms. In a BPNN the input layer is connected to the hidden layer and output layer by means of interconnection weights. This algorithm helps calculate these interconnection weights from the output layer to the input layer. The output layer is calculated first because it is closer to the expected output and loss function [19]. The BPNN using in this study includes three layers, The number of input neurons is determined by the optimal number of feature sets after being selected by BGA, 20 hidden units in the hidden layer, 1 output units in the output layer. The schematic of BPNN structure is shown in Fig.3

TABLE I. PARAMETERS USED IN BGA

BGA Parameters	Value
Population size	100
Sample lenght	50
Chromosome type	bitstrings
Fitness function	k-NN
Number of generations	100
Crossover	Two points crossovers
Mutation	Uniform
Mutation probability	0.1
Selection tournament	2
Elite count	2
Number of stall generations	50

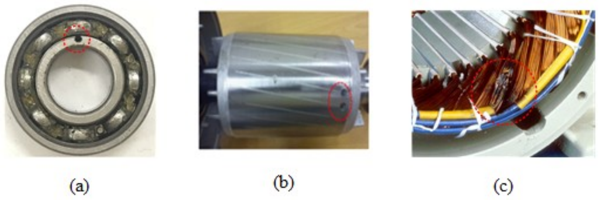


Fig. 4. Types of faults a) Bearing damage; (b) Broken rotor bar; (c) Stator winding short-circuit

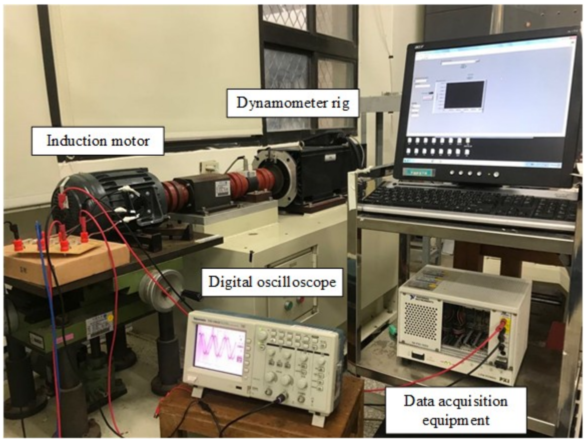


Fig. 5. Experimental setup

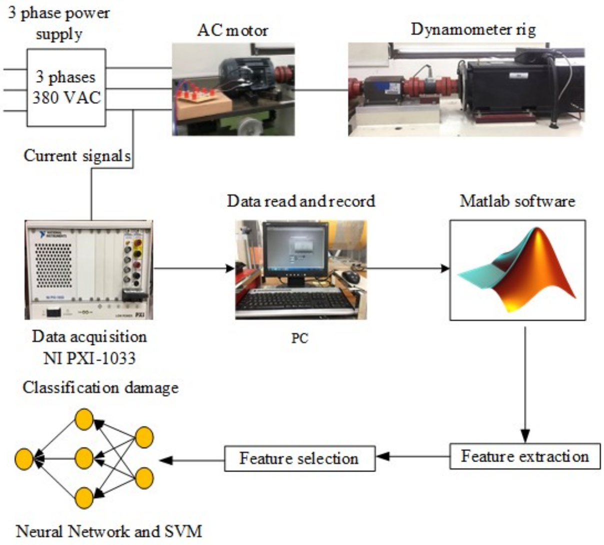


Fig. 6. Experiment process

where are $x_1, x_2, \ldots, x_n$ input units, and y are output units.

B. Support vector machine (SVM)

(SVM) is a supervised learning algorithm that can be used for binary classification or regression. In recent years, this method is commonly used in the diagnosis and detection of motor faults. SVM used the margin of separation between the two classes in the data is maximized. In this paper, SVM is used to classify motor failures. Each motor failure case is treated as a separate class corresponding to a separate label.

Given the multi-class training ability of SVM as well as its popularity in the field of pattern recognition, the selection of SVM in this study is appropriate. Besides, One vs All (OVA) technique is one of the important techniques established in SVM. OVA allows splitting multiple classes into two classes. This means that each classified object will be treated as a positive class and the rest of them as a negative class. The radial basis function is considered a kernel function.

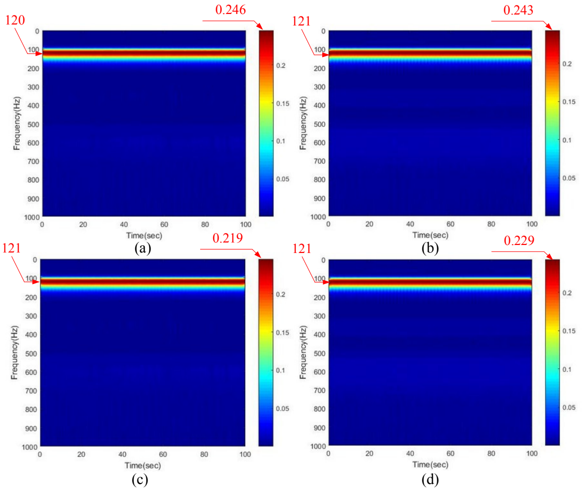


Fig. 7. S-transform spectrum
(a) S-Transform spectrum of healthy motor with a maximum amplitude of 0.246 occurs at 120Hz;
(b) S-Transform spectrum of bearing damage motor with a maximum amplitude of 0.243 occurs at 121Hz;
c) S-Transform spectrum of broken rotor bar motor with a maximum amplitude of 0.219 occurs at 121Hz;
(d) S-Transform spectrum of stator winding short-circuit motor with a maximum amplitude of 0.229 occurs at 121Hz.

V. Experiment and Results

A. Experimental setup

In this study, four different fault cases of induction motors were analyzed. These include three fault states (bearing fault, broken rotor bar, stator winding short-circuit) and one normal state as shown in Fig. 4. Bearing damage is created by drilling a hole through the inner race. Two small 2.0 mm diameter holes were created on the cage bars to simulate broken rotor bar fault. The stator winding short-circuit fault is created by damaging the insulation of the windings on the stator. And Fig. 5 describes the hardware connection of the experiment. We used a dynamometer rig (69Hz/11kW/2000rpm) and torque sensor which are considered as a dummy load of the induction motor to perform the experiment. The induction motor used in this experiment is the three phases squirrel-cage motor, 4 poles, 2 HP, 380VAC and 60Hz.

The voltage signal is measured on a phase of the test motors by the data acquisition device, NI PXI-1033, and record data on personal computer (PC). The sampling rate is 1000Hz, the measurement time is 100 seconds, measuring 50 signals for each type of motor faults. This study uses the MATLAB program to compile and analyze signals. The analysis method is S-transform. The experiment process is shown in Fig.6.

B. Simulation Results

Similar to the introduction section, the results are presented in three phases:

TABLE II. The classification accuracy under noise condition

SNR (dB)	Accuracy (%)					
Methods	Normal	30dB	25dB	20dB	15dB	10dB
SVM	**98.3**	**97.6**	**85.4**	**64.1**	43.9	30.9
BPNN	96.7	88.0	80.5	67.3	**59.3**	**49.6**

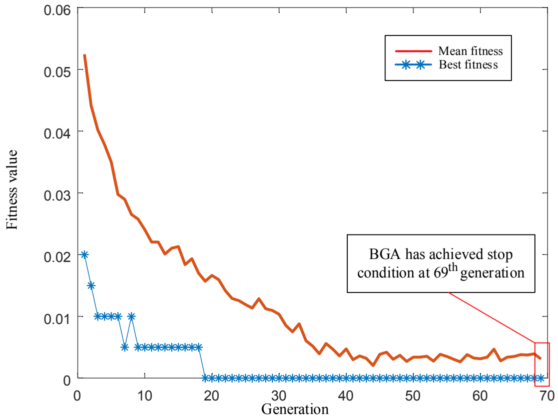


Fig. 8. BGA simulation

a) Feature extraction

After analyzing the current signal by S-transform, we obtained the spectrum diagram with the horizontal axis denotes the time domain and the vertical axis denotes the frequency domain. The S-transform spectrum of each type of motor faults in the no-load condition are shown in Fig. 7. Then 50 features are extracted from the 10 characteristic curves. The important features in the feature dataset are selected by BGA to increase the accuracy and decrease the running time of the classifier.

b) Feature selection using BGA

Based on the BGA configuration in Table 1, the following results were obtained. The purpose of the BGA is to minimize classification error based on the fitness function. The mean fitness value of each population converges to a minimum value. The stall generation is the number of generations counted by the program since the last upgrade of the fitness value, that is, the mean relative change in the value of the best fitness function over generations is less than the setting value 10^{-6}. The BGA terminates at generation 69th. The optimal feature subset obtained 18 features in the original feature set. The particular feature indexes are 5, 6, 8, 10, 11, 15, 16, 20, 26, 27, 28, 29, 38, 39, 42, 43, 48, 49. Thus, the optimal feature subset is 36%. That means the number of redundant features removed is 64%. The simulation results are shown in Figure 8, where Mean fitness curve is the mean fitness value of a population and Best fitness curve is the best fitness value of an individual.

c) Classification

The results of classifying motor faults based on BPNN and SVM are shown in Table 2. One of the limitations of this study is the selection of two classifiers (BPNN and SVM) based on personal experience. To ensure the stability of the training model, the training dataset uses 80% of the input dataset. In normal conditions, the classification accuracy of both models (SVM, BPNN) used in this study achieves the average of 100 tests, 98.3% and 96.7% respectively. In order to increase the reliability of the identification model, we simulated interference from the environment into the feature dataset by adding SNR=30dB, SNR=25dB, SNR=20dB, SNR=15dB, SNR=10dB of white noise to the original signal with increasing levels. That is, the smaller the SNR, the greater the noise level. In table 2, the results showed that SVM model is better than BPNN model in noise conditions. However, at high noise level (SNR = 15dB, 10dB), SVM proved less effective.

VI. Conclusion

In this study, BGA is proposed for feature selection and the results achieved are very positive and reliable. The proposed model has the ability to identify types of common faults of induction motors with the accuracy is 98.3% in normal condition. However, the feature selection technique using basic BGA, the convergence speed of BGA has not been evaluated in this study. In the future, we will investigate the hybrid method between BGA and another optimization method to improve the model's performance.

References

[1] A. Choudhary, D. Goyal, S.L. Shimi, *et al.*, "Condition Monitoring and Fault Diagnosis of Induction Motors: A Review," *Arch. Computat. Methods. Eng.*, vol. 26, no. 4, pp. 1221–1238, Sep. 2019.

[2] H. Henao, G. Capolino, M. Fernandez-Cabanas, F. Filippetti, C. Bruzzese, E. Strangas, S. Hedayati-Kia, "Trends in Fault Diagnosis for Electrical Machines: A Review of Diagnostic Techniques," *IEEE Ind. Electron. Mag.*, vol. 8, no. 2, pp. 31-42, Jun. 2014.

[3] T. Yang, H. Pen, Z. Wang, "Feature Knowledge Based Fault Detection of Induction Motors Through the Analysis of Stator Current Data," *IEEE Trans. Instrum. Meas*., vol. 65, no. 3, pp. 549-558, Jan. 2016.

[4] K. Li, P. Chen, H. Wang, "Intelligent Diagnosis Method for Rotating Machinery Using Wavelet Transform and Ant Colony Optimization," *IEEE Sens. J.*, vol. 12, no. 7, pp. 2474-2484, Apr. 2012.

[5] C.Y. Lee and T.A. Le, "Intelligence bearing fault diagnosis model using multiple feature extraction and binary particle swarm optimization with extended memory," *IEEE Access*, vol. 8, pp. 198343-198356, Nov. 2020.

[6] LH. Wang, XP. Zhao, JX. Wu, *et al.*, "Motor Fault Diagnosis Based on Short-time Fourier Transform and Convolutional Neural Network," *Chin. J. Mech. Eng.*, vol. 30, no. 6, pp. 1357–1368, Nov. 2017.

[7] R. Mythily and W. Aisha Banu, "Feature Selection for Optimization Algorithms: Literature Survey," *J. Eng. Appl. Sci.*, vol. 12, no.1, pp. 5735-5739, 2017.

[8] B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, Y. Xiao, "Bio-inspired feature selection: An improved binary particle swarm optimization approach," *IEEE Access*, vol. 8, pp. 85989-86002, May 2020.

[9] J. Li, H. Kang, G. Sun, T. Feng, W. Li, W. Zhang, B. Ji, "IBDA: improved binary dragonfly algorithm with evolutionary population dynamics and adaptive crossover for feature selection," *IEEE Access*, vol. 8, pp. 108032-108051, Jun. 2020.

[10] C. Y. Lee, T. A. Le, Y. T. Lin, “A Feature Selection Approach Hybrid Grey Wolf and Heap-Based Optimizer Applied in Bearing Fault Diagnosis,” *IEEE Access*, vol. 10, pp. 56691-56705, May 2022.

[11] C. Y. Lee and T. A. Le, “An enhanced binary particle swarm optimization for optimal feature selection in bearing fault diagnosis of electrical machines,” *IEEE Access*, vol. 9, pp. 102671-102686, Jul. 2021.

[12] Z. Huang, C. Yang, X. Zhou, T. Huang, “A hybrid feature selection method based on binary state transition algorithm and ReliefF,” *IEEE J. Biomed. Health Inform.*, vol. 23, no.5, pp. 1888-1898, Sep. 2018.

[13] T. Zhong, S. Zhang, G. Cai, Y. Li, B. Yang and Y. Chen, “Power Quality Disturbance Recognition Based on Multiresolution S-Transform and Decision Tree,” *IEEE Access*, vol. 7, pp. 88380-88392, Jun. 2018.

[14] M. V. Chilukuri and P. K. Dash, “Multiresolution S-transform-based fuzzy recognition system for power quality events,” *IEEE Trans. Power Deliv.*, vol. 19, pp. 323-330, Jan. 2005.

[15] Mitchell, M., “Genetic algorithms” In *Encyclopedia of Computer Science*, pp. 747-748, Jan. 2003.

[16] Y. J. Gong, J. J. Li, Y. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, J. Zhang, “Genetic learning particle swarm optimization,” *IEEE Trans. Cybern.*,” vol. 46, no. 10, pp. 2277-2290, Sep. 2016.

[17] P. Ignaciuk and Ł. Wieczorek, “Continuous genetic algorithms in the optimization of logistic networks: Applicability assessment and tuning,” *Applied Sciences*, vol. 10, no. 21, p. 7851, Nov. 2021.

[18] L. Haldurai, T. Madhubala, R. Rajalakshmi, “A Study on Genetic Algorithm and its Applications,” *IOSR J. Comput. Eng.*, vol. 4, no. 10, pp. 139-143, Oct. 2016.

[19] V. Singh, P. Gangsar, R. Porwal, A. Atulkar, “Artificial intelligence application in fault diagnostics of rotating industrial mac

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 151–156 ISSN 2300-5963 ACSIS, Vol. 33 DOI: 10.15439/2022R26

Improving Logical Structure Analysis of Visually Structured Documents with Textual Features

Huu-Loi Le*, Nghia Luu Trong†, Huyen Ngo Thanh‡
*Hung Yen University of Technology and Education, Hung Yen, Vietnam
Email: lehuuloi.cs@gmail.com
†Hanoi University of Science and Technology, Hanoi, Vietnam
Email: nghia.lt204888@sis.hust.edu.vn
‡Hung Yen University of Technology and Education, Hung Yen, Vietnam
Email: nthuyen@utehy.edu.vn
‡Corresponding author

***Abstract*—This paper introduces a new model to improve the quality of logical structure analysis of visually structured documents. To do that, we extend the model of Koreeda and Manning [1]. In order to enhance textual features, we define a new feature that uses the font size of texts as an indicator. As our observation, the font size is an important indicator that can be used to represent the structure of a document. The new font size feature is combined with visual, textual, and semantic features for training an analyzer. Experimental results on four legal datasets show that the new font size feature contributes to the model and helps to improve the F-scores. The ablation study also shows the contribution of each feature in our model.**

***Index Terms*—Logical structure analysis, VSDs, feature engineering, information extraction.**

I. Introduction

A lot of natural language processing (NLP) models, tasks, and pipelines usually require clean texts for training and processing. However, in real applications, raw data may be not clean and well organized. For example, legal documents (e.g., contracts and legal codes) are not so clean and many documents use visually structured documents (VSDs) such as PDFs [1]. As pointed out by Obermaier et al. [2], among 7.3 million documents found in Panama papers, approximately 30% were PDFs. Therefore, a good VSDs reader is required to facilitate NLP tasks in actual applications.

Compared to text data (e.g., news), VSDs contain richer information. For instance, Fig. 1 shows an example of the difference between a VSD and the raw text. While the raw text without logical structure analysis contains a sequence of tokens, the VSDsincludes both text and its structure. By reading the VSDs, we know which paragraphs are parents and which paragraphs are children. In addition, characteristics of the VSDs (e.g., font size, bold text, the position of a paragraph) can be taken into account as good indicators for logical structure analysis. We argue that an information extraction (IE) should be aware of the structure of a document to output high-quality extracted information, especially in specific domains (e.g., legal or business documents) [1], [3].

So far, there are two main directions for VSDs processing. The first direction uses rules for extracting the structure of a VSD [4]. The second direction is to analyze the structure of a VSD for logical structure parsing [5], [6] by using machine learning. While the rule-based approach can achieve high accuracy but it suffers from rule definition and management. On the other hand, the machine learning approach can generalize to new VSD types. Xu et al. introduced LayoutLM which can analyze the structure of a VSD [7], [8]. LayoutLM uses the Transformer architecture [9] to train a language model for VSDs. More recently, Koreeda and Manning introduce a model for capturing the logical structure of VSDs by using feature engineering [1]. The authors combined visual features, textual features, and semantic features for training a parser. The proposed model achieved good results on four legal datasets.

This paper improves the quality of logical structure analysis by using feature engineering. We extend the work of [1] to take into account visual, textual, and semantic features. For improvement, we also define a new feature that takes advantage of font size for training the model. Experimental on four types of datasets in the legal domain show that the proposed model obtains promising results compared to strong baselines. This paper makes two main contributions.

- It improves the quality of logical structure analysis by introducing a new feature that takes advantage of font size from texts. As our observation, the font size is an important indicator that can be used to represent the structure of a document. For example in Fig. 1, a title of a section has a larger font size than the title of its subsections. The new feature is combined with visual, textual, and semantic features to train the analyzer.
- It validates the contribution of the new feature and the proposed model on four datasets. Experimental results show that the model achieves promising results. The ablation study also shows the contribution of each feature that facilitates the next studies of logical structure analysis.

II. Related work

Hatsutori et al. [4] introduced a system that is based on the rule that fully depends on numberings. However, our idea and the result in Section IV can define that the system incorporating textual and semantic cues performs more effectively than their method. In contrast, *Sporleder and Lapata* [10]

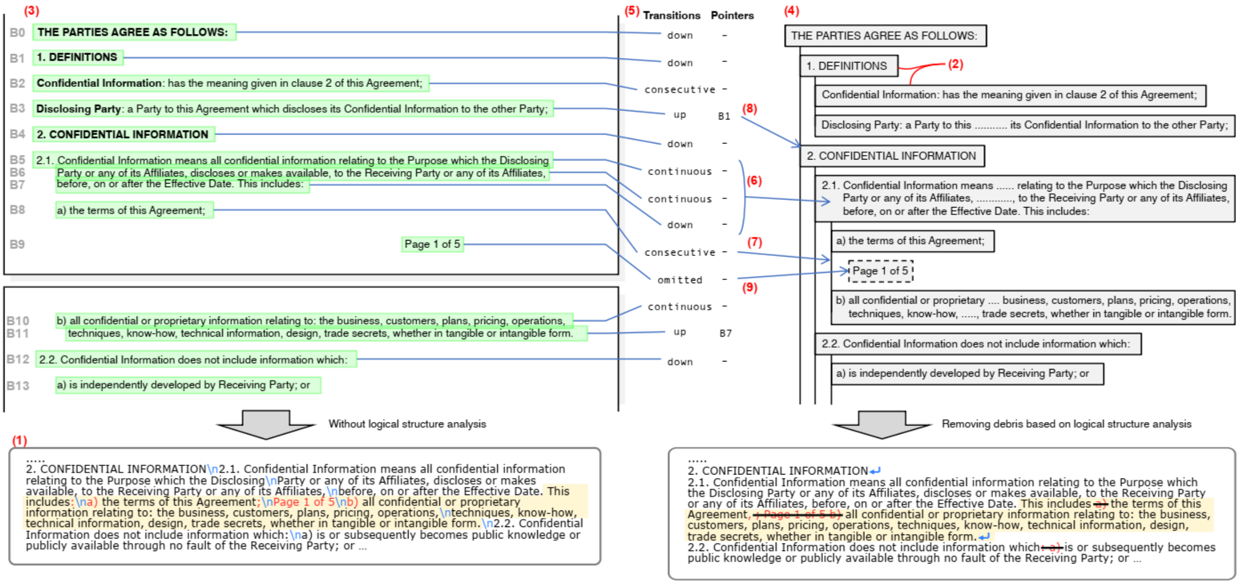

(3)
B0 THE PARTIES AGREE AS FOLLOWS:
B1 1. DEFINITIONS
B2 Confidential Information: has the meaning given in clause 2 of this Agreement;
B3 Disclosing Party: a Party to this Agreement which discloses its Confidential Information to the other Party;
B4 2. CONFIDENTIAL INFORMATION
B5 2.1. Confidential Information means all confidential information relating to the Purpose which the Disclosing
B6 Party or any of its Affiliates, discloses or makes available, to the Receiving Party or any of its Affiliates,
B7 before, on or after the Effective Date. This includes:
B8 a) the terms of this Agreement;
B9 Page 1 of 5
B10 b) all confidential or proprietary information relating to: the business, customers, plans, pricing, operations,
B11 techniques, know-how, technical information, design, trade secrets, whether in tangible or intangible form.
B12 2.2. Confidential Information does not include information which:
B13 a) is independently developed by Receiving Party; or
(5) Transitions Pointers
down -
down -
consecutive -
up B1
down -
continuous -
continuous -
down -
consecutive -
omitted -
continuous -
up B7
down -
(4)
(2)
(8)
(6)
(7)
(9)
THE PARTIES AGREE AS FOLLOWS:
1. DEFINITIONS
Confidential Information: has the meaning given in clause 2 of this Agreement;
Disclosing Party: a Party to this .... its Confidential Information to the other Party;
2. CONFIDENTIAL INFORMATION
2.1. Confidential Information means ...... relating to the Purpose which the Disclosing Party or any of its Affiliates, ..., to the Receiving Party or any of its Affiliates, before, on or after the Effective Date. This includes:
a) the terms of this Agreement;
Page 1 of 5
b) all confidential or proprietary .... business, customers, plans, pricing, operations, techniques, know-how, ..., trade secrets, whether in tangible or intangible form.
2.2. Confidential Information does not include information which:
a) is independently developed by Receiving Party; or
Without logical structure analysis
Removing debris based on logical structure analysis
(1)
.....
2. CONFIDENTIAL INFORMATION\n2.1. Confidential Information means all confidential information relating to the Purpose which the Disclosing\nParty or any of its Affiliates, discloses or makes available, to the Receiving Party or any of its Affiliates,\nbefore, on or after the Effective Date. This includes:\na) the terms of this Agreement;\nPage 1 of 5\nb) all confidential or proprietary information relating to: the business, customers, plans, pricing, operations,\ntechniques, know-how, technical information, design, trade secrets, whether in tangible or intangible form.\n2.2. Confidential Information does not include information which:\na) is or subsequently becomes public knowledge or publicly available through no fault of the Receiving Party; or ...
.....
2. CONFIDENTIAL INFORMATION↵
2.1. Confidential Information means all confidential information relating to the Purpose which the Disclosing Party or any of its Affiliates, discloses or makes available, to the Receiving Party or any of its Affiliates, before, on or after the Effective Date. This includes ~~a)~~ the terms of this Agreement, ~~Page 1 of 5 b)~~ all confidential or proprietary information relating to: the business, customers, plans, pricing, operations, techniques, know-how, technical information, design, trade secrets, whether in tangible or intangible form.↵
2.2. Confidential Information does not include information which ~~a)~~ is or subsequently becomes public knowledge or publicly available through no fault of the Receiving Party; or ...

Fig. 1: Sample of visualization of the logical structure analysis for VSDs [1].

proposed a system that fully depends on textual and semantic cues to detect a paragraph boundary for plain texts. Although their method is not focused on dealing with VSDs, we can incorporate their ideas as an additional feature of our system.

Abreu et al. [11] and *Ferrés et al.* [12] had their works in analysis of the logical structure by identifying some special structures in VSDs like subheading, ... These studies, however, are not handling in the paragraph-level logical structure and too coarse-grained. Therefore, those studies cannot satisfy our demands and fulfill our needs. *Xu et al.* [8] implemented more detailed and included extracted list items. Despite its improvement, it is still not suitable for our study because analysis of logical structures is not the center of this work.

Despite the difference in the goal, *Gillick* [13] proposed a sentence boundary detection system that has some similar textual features to our technique. But we apply richer textual and visual features that they do not employ to reach our goal which is to predict structures together with boundaries as precisely as possible.

A system proposed by *Koreeda et al.* [1] incorporates a combination of textual, visual, and semantic cues to analyze the logical structures in VSDs with a machine learning classifier. This work fully meets all our needs so we apply the same strategy for our research. In the proposed work, we improve classification performance by extracting and adding the font size of the text as a new feature.

III. Approach

A. Problem setting and formulation

In this work, we focus on the logical structure analysis of VSDs. The input is a document that contains a series of blocks extracted by the available layout analysis tool. Our goal is to extricate paragraphs from the document and recognize their relationships. To deal with this problem, we make a tree with each block as a node. To generate this tree, we identify the transition label between every two consecutive blocks that describe their relationships in this tree. As in [1], we also use five transition labels tran$_i$ between b_i (here before the i-th block) and b_{i+1} such as:

- *continuous*: b_i and b_{i+1} are continuous if they are both in a paragraph (Fig. 1(6))
- *consecutive*: b_i and b_{i+1} are consecutive if they are in two paragraphs at the same level (Fig. 1(7))
- *down*: b_{i+1} start a new paragraph its level is lower than the level of the paragraph that b_i belongs to (Fig. 1(6))
- *up*: b_{i+1} start a new paragraph its level is higher than the level of the paragraph that b_i belongs to (Fig. 1(8))
- *omitted*: b_1 is debris and omitted. (Fig. 1(9)) Now $trans_{i-1}$ is the relationship between b_{i-1} and b_{i+1}

While *down* here is well-defined, *up* is not clear and must be considered what level we should reach. To deal with this, a pointer is used for each up block, which b_j is denoted as a level that b_i belongs to ($pnt_i = b_j$ where $j < i$) (The example of detailed implementation is found in [1])

B. Logical structure analysis system

In this work, our logical structure analysis system is built based on a machine learning classifier and several handcrafted features. A machine learning model is more suitable for our research than a deep learning model because we can include textual, visual, and semantic cues in the model and it also needs less training data than a deep learning one.

To consider each block, our system extracts features from the paragraph with a group of four blocks and applies multi-class classification over five transition labels. Because *omitted* makes the target of transition changed, we delete *omitted* blocks in features extraction. With non-omitted transition, we extract features from $[b_{i-1}; b_i; b_{i+1}; bi + 2]$. Because we need to know the appearance of omitted blocks, we identify them by running the first prediction and using this information to identify other labels.

Our system can be changed flexibly to suitable several types of documents. In order to do that, we need to modify the document's features, so we build a list of features for each type by having a visual inspection of the training dataset (Table I). Some features are explained in detail in the study [1]. Since there are some different characters between TXT files and PDFs so we consider space characters as horizontal spacing and blank lines as vertical spacing. Therefore we can apply a system for TXT files like PDFs.

C. *Pointer recognition system*

We use a machine learning classifier to implement the pointer recognition system with handcrafted features. Because a down label is called, it creates a new level of the block so we need to point to this level when the up label is called. We extract all pair $[b_j, b_i]$ that $trans_j = down$ and $trans_i = up$, then we use features extracted from those pair to train a classifier to predict the pointer $(pnt_i = b_j)$

When the pointer at the block with down label (b_j), some features of the beginning block in the paragraph that contain b_j (we denote this as $b_{\text{begin}(j)}$) are so important. Therefore, we use $b_{\text{begin}(j)}$ to extract features from the pair $[b_j, b_i]$ such as:

- Consecutive numbering: Consider a number in b_i is contiguous to it in b_j and $b_{\text{begin}(j)}$ or not: Boolean features.
- Indentation: Consider the relationship about indentation of $[b_j, b_i]$ and $[b_{\text{begin}(j)}, b_{i+1}]$: Categorical features in [larger, smaller, stays the same].
- Left aligned: Consider $b_j, b_{i+1}, b_{\text{begin}(j)}$ are left aligned or not: Binary features.
- Transition counts: The number of blocks between b_i and b_j with down or with up, respectively: Numerical features.

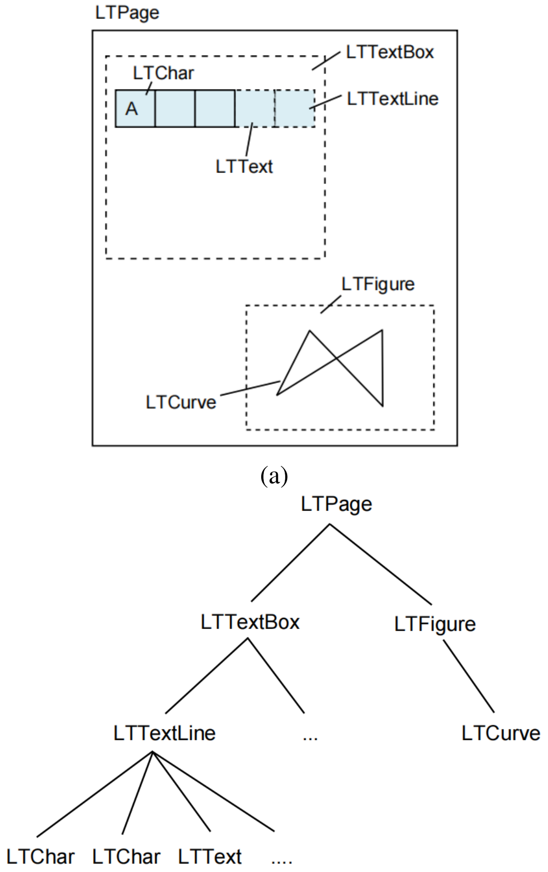
We use those features for all document types despite the customizable pointer features. We used this strategy because of the successful implementation in those studies [1], [14].

D. *Fontsize Extraction*

In this section, we detail our workaround and algorithm for extracting font size. We experimented with four types of visually structured documents (VSDs) in different file formats and languages.

- **Contract**$^{\textbf{pdf}}_{\textbf{en}}$: English NDAs in PDF format.
- **Law**$^{\textbf{pdf}}_{\textbf{en}}$: English executive orders from local authorities.
- **Contract**$^{\textbf{txt}}_{\textbf{en}}$: English NDAs in visually structured plain text format.
- **Contract**$^{\textbf{pdf}}_{\textbf{ja}}$: Japanese NDAs in PDF format.

For PDFs, we use *PDFMiner*[1] and extract each *LTTextLine*, roughly corresponding to each line of text, as a block. We have merged multiple *LTTextLines* where the *LTTextLines* overlap vertically (Fig. 2). For plain text, we just need to use each non-blank line of plain text as a block.



(a)

(b)

Fig. 2: Layout objects (a) and its tree structure (b)

The focus of this research was to suggest more information for each block extracted from the PDF file, which is the font size. So why font size? In VSDs, we can easily see that the title is often larger than the content within it. Therefore, with the desire to increase the accuracy of the transition labels, we add to each block information about its font size.

In the following, we will show how we extract font size from PDF files. First, we perform layout analysis for the PDF datasets. Parsing a PDF file is generally time and memory-consuming because a PDF file has such a large and complex structure. However, we do not use all parts for most PDF processing tasks, instead only a few needed parts are used. Therefore *PDFMiner* only parses the content when it is necessary, also known as the lazy parsing strategy. You need to use at least two classes: *PDFPaser* and *PDFDocument* to parse PDF files. These two objects are associated with each other. *PDFParser* fetches data from a file, and *PDFDocument* stores it. In addition to the two classes mentioned above, You will also need *PDFPageInterpreter* to process the page contents and *PDFPageAggregator* extract the deceive to page

[1] https://euske.github.io/pdfminer/

TABLE I: List of features for feature extraction.

Relationship		Blocks	Document type		
			Contract$^{\text{pdf}}_{\text{en}}$/Law$^{\text{pdf}}_{\text{en}}$	Contract$^{\text{txt}}_{\text{en}}$	Contract$^{\text{pdf}}_{\text{ja}}$
Visual features					
V1	Indentation	1-2,2-3	v	v	v
V2	Indentation after erasing numbering	1-2,2-3		v	v
V3	Centered	2,3	v	v	v
V4	Line break before right margin	1,2	v	v	v
V5	Page change	1-2,2-3	v		
V6	Within top 15% of a page	2	v		v
V7	Within bottom 15% of a page	2	v		v
V8	Larger line spacing	1-2,2-3	v	v	v
V9	Justified with spaces in middle	2,3	v	v	v
V10	Similar text in a similar position	2	v		v
V11	Emphasis by spaces between characters	1,2			v
V12	Emphasis by parentheses	1,2			v
Textual features					
T1	Numbering transition	2	v	v	v
T2	Punctuated	1,2		v	v
T3	List start (/[-;:,]$/)	1,2	v	v	v
T4	List elements (/(;|, |and|or)$/)	2	v	v	
T5	Page number (strict)	1,2,3	v	v	v
T6	Page number (tolerant)	1,2,3	v	v	v
T7	Starts with "whereas"	3	v	v	
T8	Starts with "now, therefore"	3	v	v	
T9	Dictionary-like (includes ":" & not V4)	2,3	v	v	
T10	All capital	2,3	v	v	
T11	Contiguous blank field (underbars)	1-2,2-3	v	v	v
T12	Horizontal line ("*-=#%_+" only)	1,2,3		v	
T13	**Font-size (our feature)**	2	v		v
Semantic features					
S1	Language model coherence	1-2-3	v	v	v

The "Blocks" columns list blocks used to extract features for $trans_2$ (e.g. "1-2, 2-3" means $[b_{i-1}; b_i]$ and $[b_i; b_{i+1}]$ are used to extract two sets of features). Features with similar intended functionality are assigned the same feature name and implementations may vary for different document types.

aggregator to get LT object elements. *PDFResourceManager* is used to store shared resources such as fonts or images. A layout analyzer returns a *LTPage* object for each page in the PDF document. This object contains child objects within the page, forming a tree structure (Fig. 2). After we have performed the layout analysis, we use these *layouts* to extract the necessary information. We use **For** loop to traverse objects inside *layouts*. We set it as lt_obj. If lt_obj is an instance of *LTTextLine* then extract the text content. Continue the for loop to go up one more level in the structure tree (Fig. 2(b)). i.e. *character* traversal in lt_obj. **If** *character* is an instance of *LTChar*, then extract the font size. After extracting the font size, we check the condition that our text length is greater than 0 or not. **If** the text we extracted above has a length greater than zero, then return text and font size. Going back to the first branch condition, **else if** lt_obj is an instance of *LTTextBox* or *LTFigure*, we can not extract anything, so we use recursion to continue. At this point, the input is no longer the layout, the new input is lt_obj.

IV. Experiments and results

A. Evaluation metrics

In experiments, we use F-score for the task that identifies relationships between the pairs of blocks including (1) the same paragraph, (2) sibling, and (3) ancestor-descendant relationships, respectively.

The number of K-folds we use in the evaluation process are customized individually for each dataset. Specifically, for Contract$^{pdf}_{en}$ and Law$^{pdf}_{en}$, we used five-folds cross-validation. But, we used twelve for Contract$^{txt}_{en}$ and fifteen for Contract$^{pdf}_{ja}$.

B. Baselines

We compared our system against the following baselines:

a) Visual: This baseline is purely based on visual cues; i.e. indentation and line spacing. For each consecutive pair of blocks, this baseline outputs (1) *continuous* when indentation is unchanged and line spacing is normal, (2) *consecutive* when indent is unchanged change and line spacing is larger than normal, (3) *down* for larger indents and (4) *up* for smaller indents. On *up*, it points back to the closest block with the same indentation.

b) NB: This baseline presents a method for preprocessing unstructured documents in general to estimate document structure. The method consists of three algorithms and the recommendation follows a rule-based approach. The three algorithms are: (1) one is based on style information, such as bold font; (2) another is based on numbered objects, such as sections; and (3) the other is based on a document's Table of Contents, which summarizes the document's structure. We focus on algorithm 2 because our implementation is the same as the feature numbering transition (T13) (Table I) [4].

TABLE II: Results for evaluation on IE perspective.

Relationship			Contract$_{en}^{pdf}$				Law$_{en}^{pdf}$				Contract$_{en}^{txt}$				Contract$_{ja}^{pdf}$			
			Visual	NB	MTP	Ours	Visual	NB	MTP	Ours	Visual	NB	MTP	Ours	Visual	NB	MTP	Ours
Same paragraph	Micro	P	**0.982**	0.484	0.944	0.947	**0.891**	0.219	0.858	0.835	**0.993**	0.540	0.983	0.988	0.446	0.402	**0.973**	0.968
		R	0.683	0.947	0.951	**0.952**	0.681	**0.969**	0.957	0.954	0.708	0.917	**0.978**	0.971	0.552	**0.985**	0.966	0.967
		F	0.806	0.641	0.947	**0.948**	0.772	0.357	**0.905**	0.890	0.826	0.680	**0.980**	0.979	0.494	0.571	**0.969**	0.968
	Macro	P	**0.980**	0.644	0.955	0.956	0.906	0.328	**0.936**	0.944	**0.990**	0.595	0.969	0.976	0.481	0.478	**0.971**	0.959
		R	0.670	**0.966**	0.951	0.951	0.634	**0.974**	0.951	0.944	0.746	0.934	**0.976**	0.973	0.527	**0.985**	0.956	0.952
		F	0.782	0.736	0.948	**0.949**	0.731	0.452	**0.933**	0.929	0.847	0.687	0.971	**0.973**	0.450	0.617	**0.955**	0.947
Siblings	Micro	P	0.332	0.677	**0.841**	0.808	0.430	0.647	**0.849**	0.828	0.397	0.780	0.784	**0.849**	0.106	0.151	0.699	**0.770**
		R	0.323	0.765	0.736	**0.779**	0.283	0.504	0.712	**0.793**	0.481	**0.763**	0.723	0.725	0.506	0.571	0.691	**0.754**
		F	0.328	0.718	0.785	**0.793**	0.341	0.567	0.774	**0.810**	0.435	0.772	0.752	**0.782**	0.176	0.238	0.695	**0.762**
	Macro	P	0.443	0.678	**0.791**	0.779	0.598	0.493	**0.793**	0.797	0.482	0.677	**0.814**	0.803	0.347	0.237	0.719	**0.769**
		R	0.427	0.691	0.751	**0.781**	0.417	0.379	0.696	**0.720**	0.557	0.603	**0.758**	0.701	0.506	0.536	0.663	**0.740**
		F	0.337	0.650	0.748	**0.750**	0.410	0.385	0.724	**0.734**	0.435	0.605	**0.754**	0.729	0.292	0.283	0.671	**0.738**
Descendants	Micro	P	0.381	0.184	0.502	**0.596**	**0.717**	0.132	0.456	0.535	0.239	0.190	0.541	**0.664**	0.536	0.125	0.577	**0.788**
		R	0.123	**0.879**	0.807	0.831	0.303	**0.881**	0.858	0.855	0.048	**0.888**	0.771	0.836	0.340	0.580	**0.826**	0.811
		F	0.186	0.304	0.619	**0.694**	0.409	0.229	0.596	**0.658**	0.080	0.313	0.635	**0.740**	0.416	0.205	0.679	**0.799**
	Macro	P	0.295	0.242	0.655	**0.678**	0.438	0.173	0.581	**0.608**	0.193	0.269	0.639	**0.699**	0.462	0.122	0.737	**0.832**
		R	0.194	**0.848**	0.798	0.822	0.314	0.764	0.837	**0.855**	0.072	**0.859**	0.735	0.758	0.358	0.519	**0.834**	0.819
		F	0.203	0.340	0.641	**0.681**	0.327	0.230	0.617	**0.651**	0.096	0.367	0.625	**0.673**	0.372	0.195	0.739	**0.803**
Accuracy	Micro		0.772	0.778	0.914	**0.921**	0.827	0.685	0.908	**0.922**	0.587	0.674	0.828	**0.863**	0.618	0.623	0.940	**0.958**
	Macro		0.686	0.679	0.889	**0.891**	0.732	0.427	0.840	**0.853**	0.571	0.580	0.841	**0.852**	0.623	0.492	0.899	**0.916**
Average F1	Micro		0.440	0.555	0.784	**0.812**	0.507	0.384	0.758	**0.786**	0.447	0.588	0.789	**0.834**	0.362	0.338	0.781	**0.843**
	Macro		0.441	0.576	0.779	**0.793**	0.489	0.356	0.758	**0.771**	0.459	0.553	0.783	**0.792**	0.372	0.365	0.788	**0.829**

”Micro”: Micro-average, ”Macro”: Macro-average, ”P”: Precision, ”R”: Recall, ”F”: F1 score

c) MTP: [1] This baseline is a combination of multimedia markers. i.e. visual (such as indentation and line spacing), textual (such as section numbering and punctuation), and semantics (such as language model coherence) cues. The formula used here is a transition parser that predicts a transition label between each consecutive pair of text fragments.

C. Implementation Details

Our system makes use of a modular and customizable design and is implemented in Python. Users can put in force a brand new function extractor clearly through writing a new function extractor class wherein every function is applied as its class function. For example, $@single\_input\_feature([1])$ denotes that the following function should be carried out to the second block of every context. Likewise, the functions for pointer identity may be carried out with the aid of using marking a function with $@pointer\_feature()$, which takes a candidate block b_j $(tb1)$, a goal block b_i $(tb2)$, the block subsequent to the goal block b_{i+1} $(tb3)$ and $b_{head(j)}$ $(head\_tb)$ as an input. For each document, a feature extractor object is initialized. In which all features are automatically gathered to become a feature vector. To facilitate implementation, a new feature extractor can inherit from an existing feature extractor.

After establishing the feature that we propose, we conduct experiments on datasets where for each dataset we evaluate the contribution of features using random forest (Fig. 3). It can be observed that our feature (font size) is one of the features that has contributed a lot. In addition, we found that the *page_like2* and *mask_continuation* features made little or no contribution. Therefore, we have considered removing these features.

Finally, we used Random Forest and Decision Tree as the transition and pointer classifiers respectively. The settings for both models are described in Table III and IV.

TABLE III: Set of hyperparameters for each dataset for transition classifier

Document Type	RandomForestClassifier		
	n_estimators	min_samples_split	max_features
Contract$_{en}^{pdf}$	400	5	'log2'
Law$_{en}^{pdf}$	300	5	'sqrt'
Contract$_{en}^{txt}$	300	5	'sqrt'
Contract$_{ja}^{pdf}$	300	2	'sqrt'

TABLE IV: Set of hyperparameters for each dataset for pointer classifier

Document Type	DecisionTreeClassifier		
	max_leaf_nodes	criterion	random_state
Contract$_{en}^{txt}$	2	'entropy'	None
Contract$_{ja}^{pdf}$	2	'gini'	0
	RandomForestClassifier		
	n_estimators	min_samples_split	max_features
Contract$_{en}^{pdf}$	400	5	'log2'
Law$_{en}^{pdf}$	100	2	'sqrt'

D. Results and Discussion

1) F-score comparison: Structure evaluation is shown on Table II. Our system obtained micro-average structure prediction accuracy of 0.921 for Contract$_{en}^{pdf}$, 0.922 for Law$_{en}^{pdf}$, 0.863 for Contract$_{en}^{txt}$ and 0.958 for Contract$_{ja}^{pdf}$, significantly outperforming the best baselines with 0.914, 0.908, 0.828 and 0.940, respectively. Micro-average transition label prediction accuracies are 0.948 (Contract$_{en}^{pdf}$), 0.936 (Law$_{en}^{pdf}$), 0.959 (Contract$_{en}^{txt}$) and 0.929 (Contract$_{ja}^{pdf}$).

For *Average F1*, our model is superior to the baselines. Compared to MTP [1], our system shows larger gaps of 6.2% for Contract$_{ja}^{pdf}$, 4.5% for Contract$_{en}^{txt}$, 2.8% for both Contract$_{en}^{pdf}$ and Law$_{en}^{pdf}$. For relationships, our model also

TABLE V: Results for feature sets on IE perspective.

		Feature Set 1	Feature Set 2	Feature Set 3
Accuracy	Micro	**0.895**	0.892	0.860
	Macro	**0.862**	0.822	0.801
Average F1	Micro	**0.760**	0.728	0.676
	Macro	**0.761**	0.701	0.678

outweighs the baseline in some scenarios, for example, 6.7% and 18% in the Siblings and Descendants respectively. These experimental results verify the effectiveness of the proposed work by extracting and adding font size as a new feature.

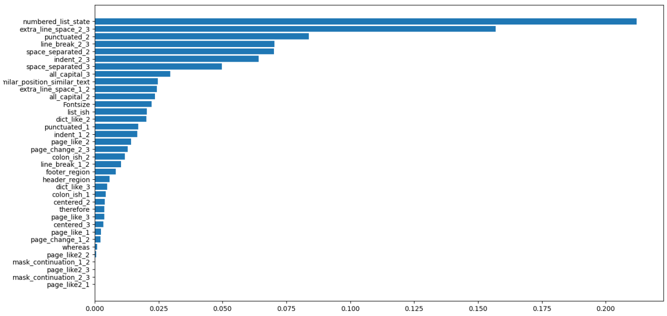


Fig. 3: Importance measurement of features (Contract$_{en}^{pdf}$)

2) Feature contribution: In this section, we go deeper into assessing the contribution of the remaining features. To do that, we evaluate the proposed model under three feature sets based on measuring feature importance (as shown in Fig. 3). In particular, we divide features into 3 subsets: Feature set 1 includes features from *numbered_list_state* to *footer_region* in Fig. 3; Feature set 2 includes *numbered_list_state* to *Fontsize*; Feature set 3 is the top 5 features with the highest contribution.

The experimental results are presented in Table V. As shown in the table, adding more features results in higher performance. This means that all selected features contribute to the performance of classification.

V. Conclusion

This paper introduces a new model to improve the logical analysis of visually structured documents. In order to do that, we introduce a new feature named font size that takes advantage of textual aspects to distinguish the structure of a VSD. The new feature based on the observation that the font size is an important indicator that can be used to represent the structure of a document. The new feature is combined with other visual features, textual features, and semantic features for training an analyzer. Experimental results on four legal datasets show that the new feature contributes to improving the performance of the model. The ablation study also shows the contribution of each feature.

Future work will investigate new features to improve the quality of the analysis. The model should be also tested on other genres and datasets.

Acknowledgement

We would like to thank Nguyen Hong Son, Huy-The Vu, and Minh-Tien Nguyen for their useful comments and supervision.

References

[1] Y. Koreeda and C. Manning, "Capturing logical structure of visually structured documents with multimodal transition parser," in *Proceedings of the Natural Legal Language Processing Workshop 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 144–154. [Online]. Available: https://aclanthology.org/2021.nllp-1.15

[2] V. W. Frederik Obermaier, Bastian Obermayer and W. Jaschensky, "About the panama papers," in *Süddeutsche Zeitung*, 2016.

[3] M.-T. Nguyen, D. T. Le, and L. Le, "Transformers-based information extraction with limited data for domain-specific business documents," *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104100, 2021.

[4] Y. Hatsutori, K. Yoshikawa, and H. Imai, "Estimating legal document structure by considering style information and table of contents," in *New Frontiers in Artificial Intelligence*, S. Kurahashi, Y. Ohta, S. Arai, K. Satoh, and D. Bekki, Eds. Cham: Springer International Publishing, 2017, pp. 270–283.

[5] C. G. Stahl, S. R. Young, D. Herrmannova, R. M. Patton, and J. C. Wells, "Deeppdf: A deep learning approach to extracting text from pdfs," Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), Tech. Rep., 2018.

[6] C. Soto and S. Yoo, "Visual detection with context for document layout analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3464–3470.

[7] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1192–1200.

[8] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. A. F. Florêncio, C. Zhang, W. Che, M. Zhang, and L. Zhou, "Layoutlmv2: Multi-modal pre-training for visually-rich document understanding," *CoRR*, vol. abs/2012.14740, 2020. [Online]. Available: https://arxiv.org/abs/2012.14740

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] C. Sporleder and M. Lapata, "Automatic paragraph identification: A study across languages and domains," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004, pp. 72–79.

[11] C. Abreu, H. Cardoso, and E. Oliveira, "FinDSE@FinTOC-2019 shared task," in *Proceedings of the Second Financial Narrative Processing Workshop (FNP 2019)*. Turku, Finland: Linköping University Electronic Press, Sep. 2019, pp. 69–73. [Online]. Available: https://aclanthology.org/W19-6410

[12] D. Ferrés, H. Saggion, F. Ronzano, and À. Bravo, "Pdfdigest: an adaptable layout-aware pdf-to-xml textual content extractor for scientific articles," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[13] M. Ostendorf, M. Collins, S. Narayanan, D. W. Oard, and L. Vanderwende, "Proceedings of human language technologies: The 2009 annual conference of the north american chapter of the association for computational linguistics," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

[14] S. Zhang, X. Ma, K. Duh, and B. V. Durme, "AMR parsing as sequence-to-graph transduction," *CoRR*, vol. abs/1905.08704, 2019. [Online]. Available: http://arxiv.org/abs/1905.08704

DOI: 10.15439/2022R18 ISSN 2300-5963 ACSIS, Vol. 33

A Flexible Approach for Automatic Door Lock Using Face Recognition

Van-Dat Ly, Hong-Son Vu
Faculty of Electrical and Electronics Engineering, Hung Yen University of Technology and Education
Khoai Chau, Hung Yen, Vietnam
First author: lydat.ute@gmail.com
Corresponding author: hongson.ute@gmail.com

***Abstract*—The model of smart door lock using face recognition based on hardware is the Jetson TX2 embedded computer proposed in this paper. In order to recognize the faces, face detection is a very important step. This paper studies and evaluates two methods of face detection, namely Histograms of Oriented Gradients (HOG) method which represents the approach using facial features and Multi-task Cascaded Convolutional Neural Networks method (MTCNN) represents using of deep learning and neural networks. To evaluate these two methods, the experimental model is used to verify the hardware platform, which is the Jetson TX2 embedded computer. The face angle parameter is used to rate the detection level and accuracy for each method. In addition, the experimental model also evaluates the speed of face detection from the camera of these methods. Experimental results show that the average time for face detection by HOG and MTCNN method are respectively 0.16s and 0.58s. For face-to-face frames, both methods detect very well with an accuracy rate of 100%. However, with various face angles of 30o, 60o, 90o, the MTCNN method gives more accurate results, which is also consistent with published studies. The smart door lock model uses the MTCNN face detection method combined with the Facenet algorithm along with a data set of 200 images for 1 face with accuracy of 99%.**

***Index Terms*—Face detection, Face recognition, HOG method, MTCNN method, Jetson TX2 embedded computer, deep learning.**

I. Introduction

With the development of embedded computers with increasingly faster processing cores, face recognition problems are becoming easier. To detect a face on an image or a video frame, it is necessary to first detect the area of the image containing the human face. This can be considered as the most important step, deciding the accuracy of the identification problems. The face detection can base on algorithms, encoding features and relationships between face features, can also use face standard samples to describe faces or face features. The methods can be mentioned as Haarlike, Adaboost, HOG methods. Recently, methods using deep learning, machine learning, neural networks have been mentioned more and more. These method use models or patterns that will be learned from a set of training images that exhibit typical of the appearance of a human face in the image.

Some studies on the problems of face detection [1-4] have used the HOG method to detect people and recognize facial features. The effect of each calculation stage on the recognition efficiency was studied. Researches indicated that for good recognition results, small scale gradients, well oriented plots, high quality local contrast in overlapping descriptors were required.

There are some researches about face detection method using deep learning such as S3FD (Single Shot Scale-Invariant Face Detector) by S. Zhang et al. [5], the MTCNN method of Kaipeng Zhang et al. [6], the Faster R-CNN face detection method of S. Ren et al. [7]. Guangyong Zheng et al. [8] solved the difficult problem of training the deep neural networks by using ResNet to control the disappearance of gradient when face detection through video sequences. These methods were considered to be very efficient.

In this paper, the level of face detection at different angles and the same aspect ratio from the camera of two methods HOG and MTCNN based on the hardware platform Jetson TX2 are evaluated. Thereby helping to choose a face detection method for later recognition problems on embedded computers.

II. HOG method

The essence of the HOG method [1] is that the object shape is described by two matrices including the gradient magnitude matrix and the gradient direction matrix. To create these two matrices, the image is first divided into a grid of squares and on which many adjacent or overlapping local regions are defined. A local area consists of many local cells measuring 8x8 pixels. Then, a graph of gradient magnitude statistics is calculated on each local plot. The HOG descriptor is created by concatenating the histogram vectors corresponding to each cell into a composite vector. To improve accuracy, each value of the histogram vector over the local area will be normalized to a first-order or quadric order normal. This normalization is intended to provide better invariance to changes in illumination and shade.

The HOG method for object detection consists of 5 steps:

Preprocessing step: In this step, the frames will be changed to the same general size, to facilitate the equal division of the image into blocks, cells and feature calculation in the next steps.

$$I_x = I \cdot D_x ; I_y = I \cdot D_y \tag{1}$$

$$D_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} ; D_y = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}^T \tag{2}$$

Where: T is the transpose matrix

Gradient magnitude:

$$G = \sqrt{I_x^2 + I_y^2} \tag{3}$$

Gradient calculation step: Performed by two convolutions of the original image with 2 vector dimensions respectively in two directions Ox and Oy.

The gradient direction:

$$\theta = \arctan\left(\frac{I_y}{I_x}\right) \tag{4}$$

Calculate feature vector for each cell step: Based on the slope, calculate the feature vectors in the direction on each cell. In which, the direction space is divided into p-dimensional feature vector of the cell. Then the tilt angle at each discrete pixel is transformed into the feature vector dimensions of the cell.

Block normalization step: By calculating an intensity threshold in a block and using that value to normalize all cells in the block, local histograms are contrast normalized for enhanced recognition form. The result after the normalization step will be a feature vector that is scaled to light, so it is more invariant to changes in lighting conditions.

Calculate HOG vector step: The feature vector of the block is calculated by concatenating the feature vector of each element in the cell together. The number of components of the feature vector at each block is calculated by the formula:

$$K = n_c \cdot K_1 \tag{5}$$

Where: K is the feature in a cell, n_c is the number of elements in a cell, and K_1 is the number of features in an element.

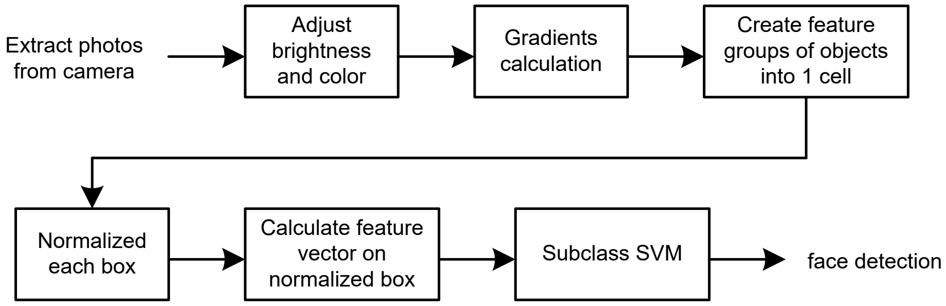

Extract photos from camera
Adjust brightness and color
Gradients calculation
Create feature groups of objects into 1 cell
Normalized each box
Calculate feature vector on normalized box
Subclass SVM
face detection

Fig. 1. Face detection procedure by HOG method

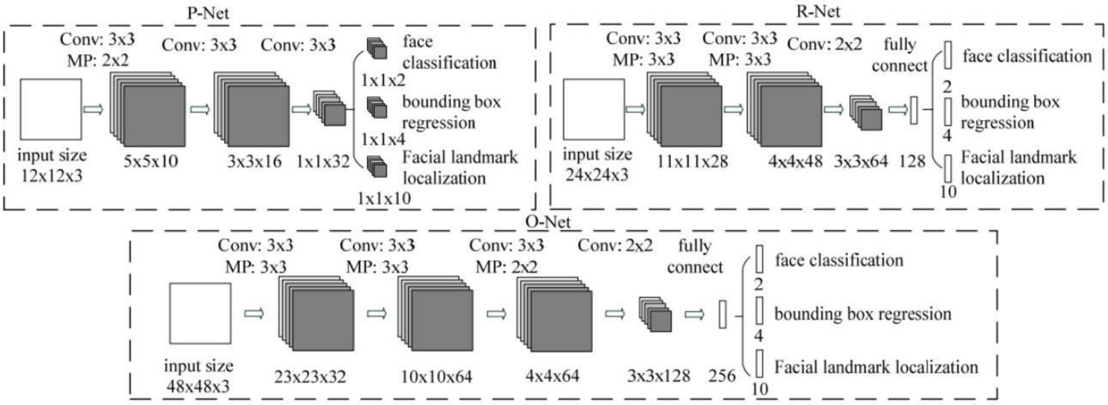


Fig. 2. MTCNN network architecture [6]

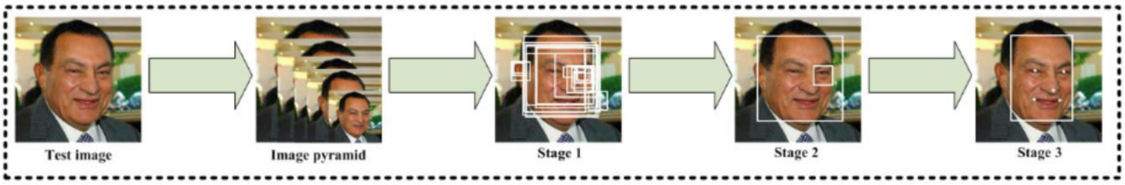


Fig. 3. Steps of image processing in MTCCN method [6]

III. MTCNN METHOD

MTCNN is a face detection method according to the cascade model [9, 10] using 3 neural networks namely P-Net, R-Net, O-Net as shown in Figure 2 to detect faces in 3 steps as follows:

Step 1: The input image will be resized, to create many copies from the original image with different sizes, from large to small, forming 1 Image Pyramid with each copy used 1 core 12x12 pixels and step of 2 to go through the entire image, detecting faces. Then, the above cores are transmitted over the P-Net (Proposal Network). The lattice results in a series of bounding boxes, each bounding box will contain four corner coordinates to determine the location in its containing core.

Step 2: The image continues through the R-Net (Refine Network) where the images will be inserted with zero-pixels in the missing parts of the bounding box if the bounding box exceeds the boundary of the image. All bounding boxes will now be resized to 24x24. The following result is also the new coordinates of the remaining boxes and is fed into the next network, the O-Net (Output Network).

Step 3: O-Net (Output Network), do the same as in R-Net, the images are resized to 48x48. But the output of the network will now return 3 values including: 4 coordinates of the bounding box (out [0]), coordinates of 5 landmarks on the face, including 2 eyes, 1 nose, 2 wings lips (out [1]) and confidence score of each box (out [2]). All will be saved into a library with the above 3 points. These steps are depicted in Figure 3.

IV. Model to evaluate the face recognition level of HOG and MTCNN method

The hardware is an embedded computer Jetson TX2 of NVIDIA which is developed with powerful configuration with dual core NVIDIA Denver2 2GHz, 4 cores ARM Cortex-A57 2GHz, 8GB of memory and integrated 256-Core Pascal GPU, built-in camera as shown in Fig. 4. With this configuration, Jetson TX2 is widely used for computer vision problems, artificial intelligence, deep learning and machine learning problems.

Fig. 4. KIT Jetson TX2

To evaluate the above two methods, Python language and Open source computer vision library (OpenCV) tool are used. OpenCV is a repository of open source code used to process images, develop graphical applications in real time. OpenCV allows to improve the speed of the CPU when performing real-time operations.

The input image from the camera is processed by two methods, the detection time is calculated from the start of recognition until the face is detected for each method. The input images for recognition are front view, 30^o, 60^o and 90^o inclined view. When the face is identified, the HOG method will draw a red rectangle, the MTCNN method will draw a blue rectangle around the recognized face.

In addition, the accuracy of the methods was evaluated over 300 trials for each face sample. Accuracy will be calculated using the formula:

$$A(\%) = \frac{A_1}{300} \times 100 \tag{6}$$

Where: A is the accuracy, A_1 is the number of times the face was detected.

The algorithm for image processing is shown in Figure 5. In which, the time for face detection for each method is calculated from the start of calling the detection function until the face is detected. Because the recognition image is taken from the camera, the face detection using both methods above using a program will ensure objectivity when using the same input image source at the same time.

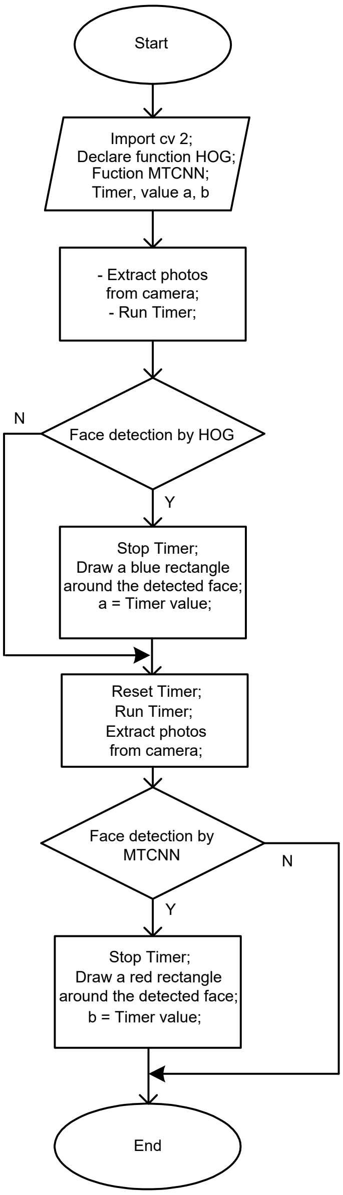
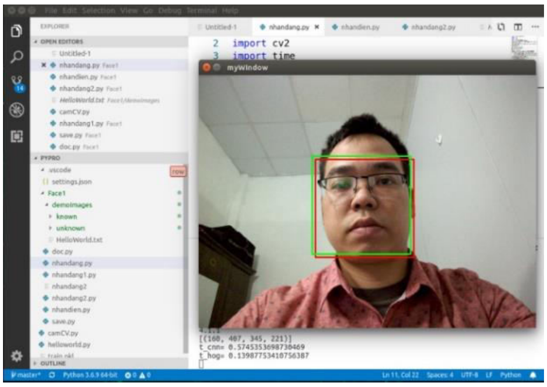


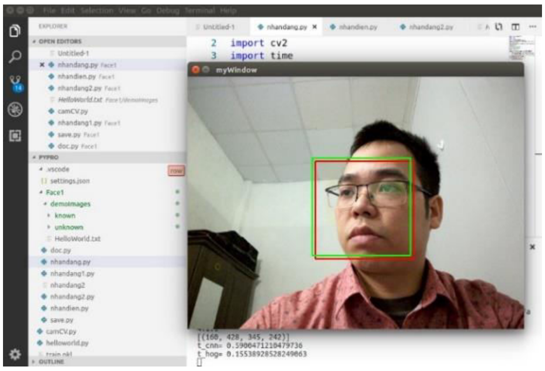
Fig. 5. Processing algorithm of face recognition



```
import cv2
import time
```

```
[(160, 407, 345, 221)]
t_cnn= 0.5745353698730469
t_hog= 0.13987753410756387
```

Fig. 6. Face detection in the front view



```
import cv2
import time
```

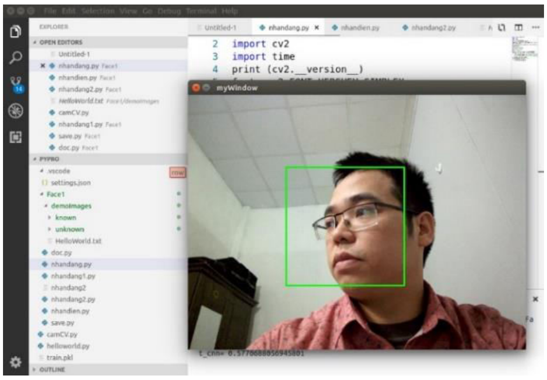
```
[(160, 428, 345, 242)]
t_cnn= 0.5900471210479736
t_hog= 0.15538928528249063
```

Fig. 7. Face detection with face anlge of 30^o

The results of face detection in the front view are presented as shown in Figure 6. The results show that both HOG and MTCNN methods can recognize the face well in front view.

Figure 7 presents face detection result with face angle of 30^o. The result show that HOG and MTCNN method have detected the face in this case.

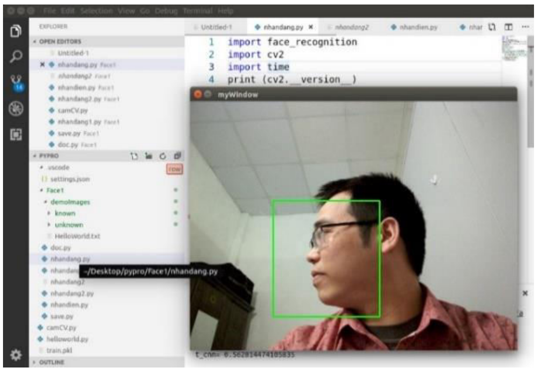
Figure 8 and Figure 9 respectively describe the face detection results with various face angles of 60^o and 90^o. The results show that in this case, only the MTCNN method can detect the face, while the HOG method cannot detect it.



```
import cv2
import time
print (cv2.__version__)
```

```
t_cnn= 0.5770688056945801
```

Fig. 8. Face detection with face anlge of 60^o



```
import face_recognition
import cv2
import time
print (cv2.__version__)
```

```
t_cnn= 0.562814474105835
```

Fig. 9. Face detection with face anlge of 90^o

Synthesized results through 300 trials to identify faces by HOG method takes an average of 0.16s and MTCCN method takes about 0.58s. The accuracy over 300 trials are summarized in Table 1 and Figure 10.

The above results show that the processing time when running on the platform of Jetson TX2 for images with only 1 face of both methods gives very fast results. In which, the HOG method has a fast processing time of only 0.16s, but it is only accurate with direct angles and small tilt angles. The detecting time of the MTCNN method is slower than HOG method, but the average time is 0.58s, showing the processing power of the GPU core, this time also responds very well to the recognition problems. Besides, the MTCCN method gives very high accuracy at face tilt angles of 60^o and 90^o.

TABLE I. RESULT OF IMAGE DETECTION WITH DIFFERENT FACE ANGLE IN 300 TRIALS

Face angle	**Detection level**	
	HOG method	***MTCNN method***
Direct direction	100%	100%
30^o	91.3%	98.3%
60^o	22.6%	91.6%
90^o	0%	30.6%

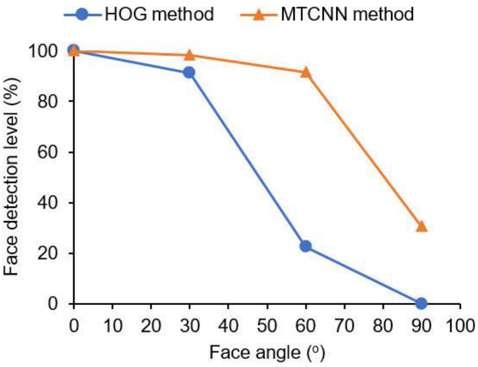


Fig. 10. Face detection with various face angle in 300 trials

V. FACE RECOGNITION AND APPLICATION TO UNLOCKING PROBLEM

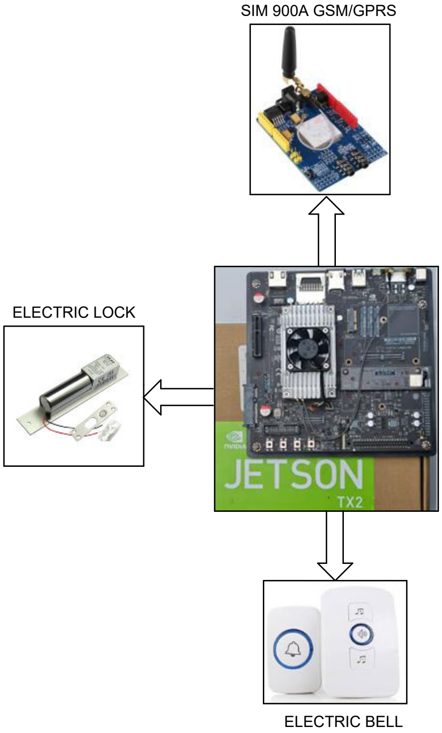


Fig. 11. Hardware structure of the system

From the evaluation results of the above two face detection methods, a model applies the face recognition algorithm to automatically unlock the door is built. The model uses the MTCCN method that combines the Facenet algorithm and the training image set of 200 images for one face. The SQLite application is used to create a dataset where the images of the dataset are taken directly from camera.

The system's hardware structure is designed with compact requirements that can be applied in real life. The face recognition application in opening and closing the door automatically based on the principle of identifying faces appearing in the camera frame. Thereby, the door is automatic unlocking if the face appearing in the camera frame matches with the database.

The main blocks of the system are depicted in Figure 11. The main equipment includes: Jetson TX2 embedded computer on the computer that has built-in camera. The actuator includes an electric lock, an electric bell and a 900A Sim module that sends messages to the owner's phone number. The camera has the function of collecting images, sending frames to an embedded computer for processing and executing image processing algorithms including face detecting and face recognition through the OpenCV library.

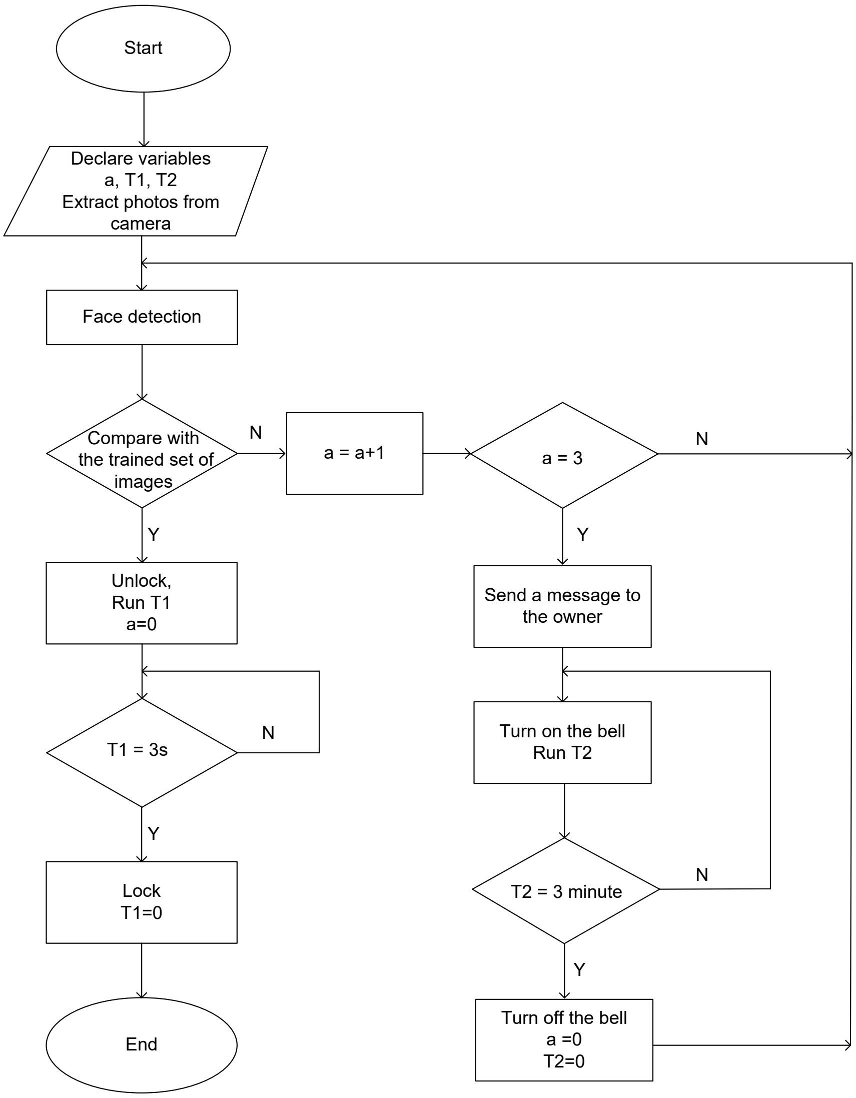


Fig. 12. Flowchart of face recognition algorithm for automatic door unlocking problem

The face detected from the camera frame is compared with the training set, if it matches, the person's name is given and the key is automatically unlocked. In case of 5 invalid face unlock times, the system will alarm with a buzzer for 3 minutes and send a message to the pre-set phone number via the 900A SIM module. The algorithm flowchart is shown in Figure 12. Through testing, it shows that with a frontal view, the average time to process the unlock is 0.7s and the accuracy reaches over 99%.

VI. CONCLUSIONS

In this paper, two models for face detection on the hardware platform Jetson TX2 have been deployed. Both methods showed good results for the front view. For the problem that needs to be identified at the inclined angle, the MTCNN method gave better efficiency. The MTCNN method and the Facenet algorithm were selected to apply to the automatic door unlocking problem. The system was compactly built and had high accuracy. In addition, the experimental research results can be applied to other problems such as detection, identification, warning or automatic time-keeping problems in corporate companies. Besides, the problem of reporting people doesn't wear facemasks during the complicated situation of the Covid-19 epidemic is also applied in this algorithm.

REFERENCES

[1] N. Dalal, B. Triggs, “Histograms of Oriented Gradients for Human Detection”. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005

[2] S K Eng* , H Ali, A Y Cheah, Y F Chong, Facial expression recognition in JAFFE and KDEF Datasets using histogram of oriented gradients and support vector machine, IOP Conf. Series: Materials Science and Engineering 705 (2019) 012031, doi:10.1088/1757-899X/705/1/012031

[3] S Yallamandaia, N Purnachand., “A novel face recognition technique using Convolutional Neural Network, HOG, and histogram of LBP features”, IEEE, 2022.

[4] D.Lakshmi, R.Ponnusamy, “Facial emotion recognition using modified HOG and LBP features with deep stacked autoencoders”, Microprocessors and Microsystems, vol. 82, 2021.

[5] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang and S. Z. Li, "Sˆ 3FD: Single Shot Scale-Invariant Face Detector," in Computer Vision (ICCV), 2017 IEEE International Conference on, 2017

[6] Kaipeng Zhang1 Zhanpeng Zhang2 Zhifeng Li1 Yu Qiao1, Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. IEEE Signal Processing Letters (SPL), vol. 23, no. 10, pp. 1499-1503, 2016

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015

[8] Guangyong Zheng, Yuming Xu, “Efficient face detection and tracking in video sequences based on deep learning”, Information Sciences, vol. 568, pp. 265-285, 2021.

[9] P. Viola and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, pp. 137-154, 2004

[10] Chunming Wu, Ying Zhang, “MTCNN and FACENET Based Access Control System for Face Detection and Recognition”, Automatic Control and Computer Sciences, vol. 55, pp. 102-112, 2021

DOI: 10.15439/2022R21 ISSN 2300-5963 ACSIS, Vol. 33

Outage Probability and Intercept Probability Of Cognitive IoTs Networks With Relay Selection, Passive Eavesdropper and Hardware Noises

Pham Xuan Minh, Nguyen Van Hien,
Pham Quoc Hop, Tran Trung Duy, Le Quang Phu
Posts and Telecommunications Institute of Technology
HoChiMinh city campus, Viet Nam
{minhpx, nvhien, pqhop, trantrungduy, phulq}
@ptithcm.edu.vn

Nguyen Trung Hieu
Department of Electrical Engineering
Posts and Telecommunications
Institute of Technology
Ha Noi, Viet Nam
hieunt@ptit.edu.vn

***Abstract*—This paper measures outage probability (OP) and intercept probability (IP) of a secondary secure network in a relaying underlay spectrum-sharing cognitive radio (USS-CR) system. In this scheme, one of successful secondary relays is chosen to obtain the highest signal-to-noise ratio (SNR) at a secondary destination. This paper considers the case where channel state information (CSI) of the eavesdropping links is not available. The OP and IP performance of the considered scheme is calculated via analysis and simulations, under impact of hardware noises, over Rayleigh fading channels**

***Index Terms*—Underlay spectrum-sharing cognitive radio, physical-layer security, outage probability, intercept probability, hardware impairments.**

I. Introduction

Recently, physical-layer security (PLS) has been reported in many literature (i.e., [1]-[4]) as an efficient approach to obtain security for wireless communication systems. For the secure communication in PLS over fading channels, the data links should be better than the eavesdropping ones. In [5]-[8], diversity-based transmitting/ receiving techniques in multi-input multi-output (MIMO) networks were proposed to increase SNR of the data channels. In [9]-[11], various cooperative jamming approaches were proposed to reduce SNR of the eavesdropping channels. However, implementation of the approaches in [9]-[11] is a difficult work because they require a high synchronization and a perfect interference cancellation. When wireless devices, e.g., sensors, are only equipped with a single antenna, cooperative relaying [12]-[15] can be efficiently applied into the PLS networks. Moreover, relay selection algorithms such as partial and full relay selection were employed to increase the e2e SNR of the data links, which also improved the secrecy performance such as secrecy outage probability (SOP) and IP/OP trade-off. Published works [15]-[16] proposed joint relay selection and cooperative jamming strategies for secured communication cooperative relaying scenarios. In [17]-[18], the authors introduced intelligent reflecting surface (IRS)-based PLS schemes, where the source-destination communication was assisted by IRS, instead of the cooperative relays.

In the USS-CR networks [19]-[20], secondary users must optimally change their transmit power to protect quality of service of a primary network. Particularly, CSI of the secondary transmitter -the primary receiver links are used to calculate the instantaneous transmit power. Different from [19]-[20], published work [21] introduced a new transmit power adaptation method, where average transmit power of the secondary devices was appropriately adjusted so that the OP performance of the primary network did not exceed a pre-designed value. Published works [22]-[26] concerned with PLS in the USS-CR networks. In particular, the authors in [22]-[23] evaluated the IP-OP trade-off of the PLS USS-CR networks. The authors [24] measured SOP and average secrecy capacity (ASC) for USS-CR over Nakagami-m channels. Reference [25] proposed a new PLS Internet of Things (IoTs) model using NOMA and short packet communication. The authors in [26] derived secrecy outage probability of unmanned aerial vehicle-aided USS-CR networks adopting non-orthogonal multiple access (NOMA).

Recently, topic of evaluating performance of the PLS schemes under impact of hardware imperfection has gained much attention. In [27], the authors calculated probability of positive secrecy capacity for multi-hop PLS schemes with hardware noises. In [28], the authors proposed a secure amplify-and-forward (AF) scheme, with a multi-antenna eavesdropper and hardware impairments. The authors in [29] studied the OP-IP trade-off for NOMA-aided PLS models, under effect of IQI (in-phase and quadrature-phase imbalance). Reference [30] investigated joint impact of co-channel interference and hardware imperfection on the IP and OP performance of sensor networks. The authors of [31]-[32] studied performance of hybrid satellite-terrestrial relaying PLS schemes with hardware impairments.

This paper evaluates OP and IP of the secure USS-CR networks, under effect of hardware impairments. Particularly, we focus on the cooperative phase in the secondary network, where successful relays transmit the data to a destination. Follows operation principle of USS-CR [19]-[20], these relays must change their transmit power to satisfy the interference constraint. With presence of a passive eavesdropper, one of the successful relays is selected to provide the highest instantaneous SNR for the relay-destination communication. We then derive exact formulas of OP and IP for the proposed PLS USS-CR scheme, and realize computer simulations to validate them.

A. *Related Works*

At first, different with [27]-[30], this paper considers the PLS USS-CR networks. Next, although the related works [22]-[26] and [33]-[37] also study the IP/OP trade-off performance for the PLS USS-CR networks, the main difference can be listed as follows:

- Published work [22] considered full-duplex jamming with assistance of IRS in device-to-device communication networks; the authors in [23] applied radio-frequency energy harvesting technique into the PLS USS-CR networks; and reference [24] evaluated secrecy outage performance. However, the works [22]-[24] did not consider impact of hardware noises on the secrecy performance.

- Reference [25] applied NOMA and short packet communication for the PLS USS-CR networks, while published work [26] studied UAV-aided NOMA MIMO systems. However, the techniques adopted in [25]-[26] are out of scope of this paper.

- Although the authors in [33] also studied PLS in the cooperative phase, the main difference is that [33] adopted jammer and relay selection strategies for maximizing instantaneous secrecy capacity. Next, effect of hardware noises was not studied in [33]. Finally, this paper considers the PLS scheme with presence of the passive eavesdropper, i.e., CSI of the eavesdropping link is not available at the legitimate transmitters and receivers.

- Different with relay selection methods studied in [34], the relay selection method in this paper is performed to maximize SNR obtained at the secondary destination.

- Different with this paper, reference [35] concerns with SOP evaluation of the multi-hop PLS scheme in the USS-CR with hardware imperfection. In addition, this paper does not study radio frequency energy harvesting as in [35].

- Main difference between this work and [36] can be listed as: i) the secondary transmitters in [36] must change transmit power, follows a joint constraint of minimum OP value of the primary network and maximum IP value at the secondary eavesdropper; ii) this paper does not consider presence of the interference from the primary network.

- Different with published work [37], this paper does not consider Fountain codes and the cooperative jamming technique.

Finally, we note that different with the previous works, this paper uses mathematical tools to derive new OP and IP formulas for the proposed scheme over Rayleigh fading channel. Moreover, this paper also analyzes impact of hardware noises and number of relay nodes on the IP-OP trade-off.

The remainder of this paper is given as follows. Operation principle and performance of the considered scheme are shown in Section II and III, respectively. Both the computer simulations and the theoretical results are given in Section IV. Finally, final discussion and recommendation are given in Section V.

II. System Model

Figure 1 presents the cooperative transmission in the considered USS-CR IoTs scheme, where in the secondary network, M successful relays (SR) want to send the source data to the destination (SD) with presence of the eavesdropper (SE). For ease of presentation, the broadcast phase, the transmission between the secondary source and the secondary relays, is not illustrated in this paper (see [33]). Then, one of these relays is selected to forward the source data to SD. Moreover, the selected relay must change transmit power to satisfy the interference threshold (I_{th}) given by the primary receiver (PR). All the SR, SD, SE and PR nodes are assumed to have only one antenna.

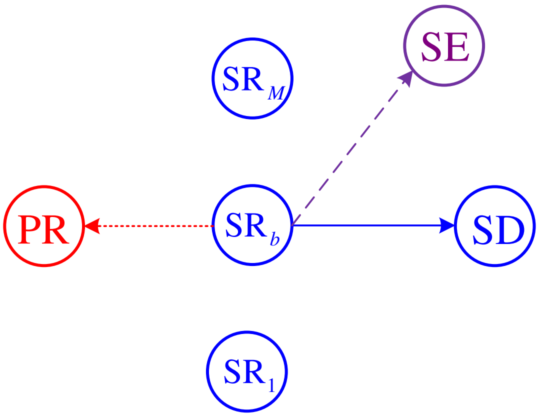


Fig. 1. The considered USS-CR IoTs scheme.

Considering the A → B Rayleigh fading channel between the transmitter A $(\text{A} \in \{\text{SR}_m\})$ and the receiver B $(\text{B} \in \{\text{PR,SD,SE}\})$, with $m \in \{1,2,...,M\}$. Let $g_{\text{A,B}}$ denote channel gain between A and B. Therefore, cumulative distribution function (CDF) and probability density function (PDF) of $g_{\text{A,B}}$ are expressed, respectively as

$$F_{g_{\text{A,B}}}(x) = 1 - \exp(-\lambda_{\text{A,B}} x),$$
$$f_{g_{\text{A,B}}}(x) = \lambda_{\text{A,B}} \exp(-\lambda_{\text{A,B}} x). \tag{1}$$

We assume that the random variables $g_{\text{SR}_m,\text{B}}$ are idependent and identical, i.e., $\lambda_{\text{SR}_m,\text{B}} = \lambda_{\text{SR,B}}$ for all m and B. Similar to [35]-[36], the transmit power of SR_m can be obtained as

$$P_{\text{SR}_m} = \frac{I_{\text{th}}}{(1+\kappa_{\text{P}}^2) g_{\text{SR}_m,\text{PR}}} = \frac{Q_{\text{th}}}{g_{\text{SR}_m,\text{PR}}}, \tag{2}$$

where κ_{P}^2 is hardware-noise level of the $\text{SR}_m \to \text{PR}$ link (see [35]-[36]), for all m, and $Q_{\text{th}} = I_{\text{th}} / (1+\kappa_{\text{P}}^2)$.

From (2), we can write SNR of the $\text{SR}_m \to \text{SD}$ link as

$$\gamma_{\text{SR}_m,\text{SD}} = \frac{P_{\text{SR}_m} g_{\text{SR}_m,\text{SD}}}{\kappa_{\text{D}}^2 P_{\text{SR}_m} g_{\text{SR}_m,\text{SD}} + N_0}, \tag{3}$$

where κ_{D}^2 is hardware-noise level of the $\text{SR}_m \to \text{SD}$ links, for all m, and N_0 is variance of additive noise at SD.

Substituting (2) into (3), which yields

$$\gamma_{\mathrm{SR}_m,\mathrm{SD}} = \frac{\frac{Q_{\mathrm{th}}}{g_{\mathrm{SR}_m,\mathrm{PR}}} g_{\mathrm{SR}_m,\mathrm{SD}}}{\kappa_{\mathrm{D}}^2 \frac{Q_{\mathrm{th}}}{g_{\mathrm{SR}_m,\mathrm{PR}}} g_{\mathrm{SR}_m,\mathrm{SD}} + N_0}. \tag{4}$$

From (4), the best relay is selected as follows:

$$\begin{aligned} \mathrm{SR}_b : \gamma_{\mathrm{SR}_b,\mathrm{SD}} &= \max_{m=1,2,\dots,M} \left(\gamma_{\mathrm{SR}_m,\mathrm{SD}}\right) \\ \leftrightarrow \frac{g_{\mathrm{SR}_b,\mathrm{SD}}}{g_{\mathrm{SR}_b,\mathrm{PR}}} &= \max_{m=1,2,\dots,M} \left(\frac{g_{\mathrm{SR}_m,\mathrm{SD}}}{g_{\mathrm{SR}_m,\mathrm{PR}}}\right), \end{aligned} \tag{5}$$

where $b \in \{1,2,\dots,M\}$. Equation (5) implies that SR_b provides the highest SNR obtained at SD (or the ratio $g_{\mathrm{SR}_b,\mathrm{SD}}/g_{\mathrm{SR}_b,\mathrm{PR}}$ is highest).

Then, the instantaneous SNR obtained at SE can be obtained as

$$\begin{aligned} \gamma_{\mathrm{SR}_b,\mathrm{SE}} &= \frac{P_{\mathrm{SR}_b} g_{\mathrm{SR}_b,\mathrm{SE}}}{\kappa_{\mathrm{E}}^2 P_{\mathrm{SR}_b} g_{\mathrm{SR}_b,\mathrm{SE}} + N_0}, \\ &= \frac{\frac{Q_{\mathrm{th}}}{g_{\mathrm{SR}_b,\mathrm{PR}}} g_{\mathrm{SR}_b,\mathrm{SE}}}{\kappa_{\mathrm{E}}^2 \frac{Q_{\mathrm{th}}}{g_{\mathrm{SR}_b,\mathrm{PR}}} g_{\mathrm{SR}_b,\mathrm{SE}} + N_0}, \end{aligned} \tag{6}$$

where κ_{E}^2 is hardware-noise level of the eavesdropping links, for all *m*, and N_0 also denotes variance of the noise at SE.

Next, OP at SD and IP at SE are defined, respectively as

$$\begin{aligned} \mathrm{OP} &= \Pr\left(\gamma_{\mathrm{SR}_b,\mathrm{SD}} < \gamma_{\mathrm{th}}\right), \\ \mathrm{IP} &= \Pr\left(\gamma_{\mathrm{SR}_b,\mathrm{SE}} \ge \gamma_{\mathrm{th}}\right), \end{aligned} \tag{7}$$

where γ_{th} is a pre-determined value.

III. Performance Analysis

Combining (4) and (7), we rewrite OP as follows:

$$\mathrm{OP} = \Pr\left(\left(1-\kappa_{\mathrm{D}}^2\gamma_{\mathrm{th}}\right) X_b < \frac{N_0\gamma_{\mathrm{th}}}{Q_{\mathrm{th}}}\right), \tag{8}$$

where $X_m = g_{\mathrm{SR}_m,\mathrm{SD}}/g_{\mathrm{SR}_m,\mathrm{PR}}\ (\forall m)$. We observe that if $1-\kappa_{\mathrm{D}}^2\gamma_{\mathrm{th}} \le 0$ then $\mathrm{OP}=1$. Considering the case of $1-\kappa_{\mathrm{D}}^2\gamma_{\mathrm{th}} > 0$, we can rewrite (8) as

$$\mathrm{OP} = \Pr\left(X_b < \theta_{\mathrm{D,th}}\right) = F_{X_b}\left(\theta_{\mathrm{D,th}}\right), \tag{9}$$

where

$$\theta_{\mathrm{D,th}} = \frac{N_0\gamma_{\mathrm{th}}}{\left(1-\kappa_{\mathrm{D}}^2\gamma_{\mathrm{th}}\right)Q_{\mathrm{th}}}. \tag{10}$$

To find CDF $F_{X_b}(x)$, we first find CDF $F_{X_m}(x)$ which can be formulated as

$$\begin{aligned} F_{X_m}(x) &= \Pr\left(g_{\mathrm{SR}_m,\mathrm{SD}} < x g_{\mathrm{SR}_m,\mathrm{PR}}\right) \\ &= \int_0^{+\infty} F_{g_{\mathrm{SR}_m,\mathrm{SD}}}(xy) f_{g_{\mathrm{SR}_m,\mathrm{PR}}}(y)\,dy. \end{aligned} \tag{11}$$

Substituting (1) into (11), after calculating the integral, we obtain (12) as

$$F_{X_m}(x) = \frac{\lambda_{\mathrm{SR,SD}}x}{\lambda_{\mathrm{SR,PR}} + \lambda_{\mathrm{SR,SD}}x} = \frac{x}{x+\Omega_{\mathrm{D}}}, \tag{12}$$

where $\Omega_{\mathrm{D}} = \lambda_{\mathrm{SR,PR}}/\lambda_{\mathrm{SR,SD}}$. Then, PDF of X_m is obtained as

$$f_{X_m}(x) = \frac{\Omega_{\mathrm{D}}}{\left(x+\Omega_{\mathrm{D}}\right)^2}. \tag{13}$$

Therefore, the distribution functions of X_b are given as

$$F_{X_b}(x) = \left[F_{X_m}(x)\right]^M = \left(\frac{x}{x+\Omega_{\mathrm{D}}}\right)^M, \tag{14}$$

$$f_{X_b}(x) = \frac{M\Omega_{\mathrm{D}}x^{M-1}}{\left(x+\Omega_{\mathrm{D}}\right)^{M+1}}. \tag{15}$$

Substituting (14) into (9), we obtain OP as given in (16) below:

$$\mathrm{OP} = \left(\frac{\theta_{\mathrm{D,th}}}{\theta_{\mathrm{D,th}}+\Omega_{\mathrm{D}}}\right)^M. \tag{16}$$

For the IP performance; combining (6) and (7), we can obtain (17) as

$$\begin{aligned} \mathrm{IP} &= \Pr\left(\frac{g_{\mathrm{SR}_b,\mathrm{SE}}}{g_{\mathrm{SR}_b,\mathrm{PR}}} \ge \theta_{\mathrm{E,th}}\right) \\ &= \int_0^{+\infty} F_{g_{\mathrm{SR}_b,\mathrm{PR}}}\left(\frac{x}{\theta_{\mathrm{E,th}}}\right) f_{g_{\mathrm{SR}_b,\mathrm{SE}}}(x)\,dx. \end{aligned} \tag{17}$$

where

$$\theta_{\mathrm{E,th}} = \frac{N_0\gamma_{\mathrm{th}}}{\left(1-\kappa_{\mathrm{E}}^2\gamma_{\mathrm{th}}\right)Q_{\mathrm{th}}}. \tag{18}$$

It is noted from (17)-(18) that we only consider the case where $1-\kappa_{\mathrm{E}}^2\gamma_{\mathrm{th}} > 0$. Indeed, if $1-\kappa_{\mathrm{E}}^2\gamma_{\mathrm{th}} \le 0$, IP always equals to zero.

Moreover, to calculate IP in (17), our next objective is to find CDF of $g_{\mathrm{SR}_b,\mathrm{PR}}$. Using derivation methods used in [38]-[40], we can formulate $F_{g_{\mathrm{SR}_b,\mathrm{PR}}}(x)$ as

$$\begin{aligned} &F_{g_{\mathrm{SR}_b,\mathrm{PR}}}(x) = \\ &\int_0^{+\infty} \frac{\partial \Pr\left(g_{\mathrm{SR}_m,\mathrm{PR}} < x, X_m < z\right)}{\partial z} \frac{f_{X_b}(z)}{f_{X_m}(z)}\,dz. \end{aligned} \tag{19}$$

As marked in (19), the probability $\Pr\left(g_{\mathrm{SR}_m,\mathrm{PR}} < x, X_m < z\right)$ is computed as

$$\begin{aligned}
&\Pr\left(g_{\mathrm{SR}_m,\mathrm{PR}} < x, X_m < z\right)\\
&= \Pr\left(g_{\mathrm{SR}_m,\mathrm{PR}} < x, g_{\mathrm{SR}_m,\mathrm{SD}} < g_{\mathrm{SR}_m,\mathrm{PR}} z\right)\\
&= \int_0^x f_{g_{\mathrm{SR}_m,\mathrm{PR}}}(u)\left[\int_0^{uz} f_{g_{\mathrm{SR}_m,\mathrm{SD}}}(v)\,dv\right]du \qquad (20)\\
&= \frac{z}{z+\Omega} - \exp\left(-\lambda_{\mathrm{SR,PR}}x\right)\\
&+ \frac{\Omega}{z+\Omega}\exp\left(-\lambda_{\mathrm{SR,PR}}x\right)\exp\left(-\lambda_{\mathrm{SR,SD}}xz\right).
\end{aligned}$$

Plugging (13), (15), (19) and (20) together, after some careful calculation, we obtain

$$\begin{aligned}
F_{g_{\mathrm{SR}_b,\mathrm{PR}}}(x) = 1 - \sum_{n=0}^{M-1}(-1)^n C_{M-1}^n M\\
\times\left\{\begin{array}{l}
(-1)^n \dfrac{n}{(n+1)!}\left(\lambda_{\mathrm{SR,PR}}\right)^{n+1} x^{n+1} E_1\left(\lambda_{\mathrm{SR,PR}}x\right) + \\
\exp\left(-\lambda_{\mathrm{SR,PR}}x\right)\left[\begin{array}{l}
\displaystyle\sum_{k=0}^{n}(-1)^{k+2}\frac{\left(\lambda_{\mathrm{SR,PR}}\right)^k}{(n+1)n\ldots(n+1-k)}x^k\\
\displaystyle+\sum_{k=0}^{n-1}(-1)^{k+2}\frac{\left(\lambda_{\mathrm{SR,PR}}\right)^{k+1}}{n(n-1)\ldots(n-k)}x^{k+1}
\end{array}\right]
\end{array}\right\}, \qquad (21)
\end{aligned}$$

where $E_1(.)$ is exponential integral function [41].

Substituting (21) and $f_{g_{\mathrm{SR}_b,\mathrm{SE}}}(x) = \lambda_{\mathrm{SR,SE}}\exp\left(-\lambda_{\mathrm{SR,SE}}x\right)$ into (17); using [41, eq. (6.228.2)] and $E_1(x) = -\mathrm{E}i(-z)$, we finally obtain an exact formula of IP as follows:

$$\begin{aligned}
\mathrm{IP} = 1 - \sum_{n=0}^{M-1}(-1)^n C_{M-1}^n M\\
\times\left\{\begin{array}{l}
(-1)^n \dfrac{n\left(\Omega_{\mathrm{E}}\right)^{n+1}\theta_{\mathrm{E,th}}}{(n+2)\left(\theta_{\mathrm{E,th}}+\Omega_{\mathrm{E}}\right)^{n+2}}\\
\times {}_2F_1\left(1, n+2; n+3, \dfrac{\theta_{\mathrm{E,th}}}{\Omega_{\mathrm{E}}+\theta_{\mathrm{E,th}}}\right)\\
\displaystyle+\sum_{k=0}^{n}\frac{(-1)^{k+2}k!}{(n+1)n\ldots(n+1-k)}\frac{\left(\Omega_{\mathrm{E}}\right)^k\theta_{\mathrm{E,th}}}{\left(\theta_{\mathrm{E,th}}+\Omega_{\mathrm{E}}\right)^{k+1}}\\
\displaystyle+\sum_{k=0}^{n-1}\frac{(-1)^{k+2}(k+1)!}{n(n-1)\ldots(n-k)}\frac{\left(\Omega_{\mathrm{E}}\right)^{k+1}\theta_{\mathrm{E,th}}}{\left(\theta_{\mathrm{E,th}}+\Omega_{\mathrm{E}}\right)^{k+2}}
\end{array}\right\}, \qquad (22)
\end{aligned}$$

where $\Omega_{\mathrm{E}} = \lambda_{\mathrm{SR,PR}}/\lambda_{\mathrm{SR,SE}}$, and ${}_2F_1(.)$ is Gaussian hypergeometric function [41].

IV. Simulation and Theoretical Results

This section presents both simulation and theoretical results of OP and IP, which are in an excellent agreement in all figures below. To focus on evaluating impact of the number of relays (*M*) and hardware noises, we fix the other parameters as follows: $\kappa_{\mathrm{P}}^2 = 0$, $\lambda_{\mathrm{SR,SD}} = 1$, $\lambda_{\mathrm{SR,SE}} = 50$, $\lambda_{\mathrm{SR,PR}} = 5$, and $\gamma_{\mathrm{th}} = 5$.

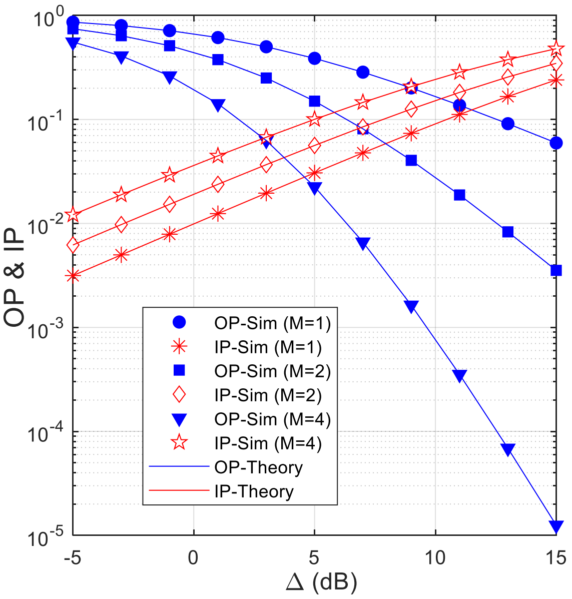


Fig. 2. OP and IP as a function of Δ (dB) when $\kappa_{\mathrm{D}}^2 = \kappa_{\mathrm{E}}^2 = 0.1$.

Fig. 2 presents the considered performance metrics IP and OP, follows $\Delta\left(\Delta = Q_{\mathrm{th}}/N_0\right)$ in dB, where $\kappa_{\mathrm{D}}^2 = \kappa_{\mathrm{E}}^2 = 0.1$ and the number of secondary relays (*M*) changes from 1 to 4. As we can see, the OP value decreases as increasing Δ because increasing Δ (or Q_{th}) also increases transmit power of the secondary relays. However, with higher value of Δ, IP at SE also increases. Therefore, Fig. 2 shows the OP-IP trade-off with the changing of Δ. It is also seen that OP significantly decreases as increasing *M*. However, with higher value of *M*, transmit power of the selected relay also increases, which leads to the increasing of IP. Hence, the value of *M* also impacts on the IP-OP trade-off.

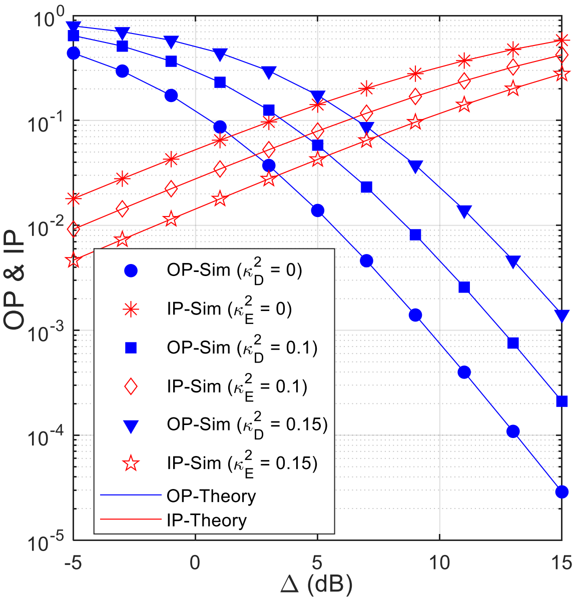


Fig. 3. OP and IP as a function of Δ (dB) with $M = 3$.

Figure 3 shows negative effect of hardware-impairment level on the OP and IP performance. In Fig. 3, the number of secondary relays is fixed by 3, while values of κ_{D}^2 and κ_{E}^2 are assigned by 0, 0.1 and 0.15. It is worth noting that $\kappa_{\mathrm{D}}^2=0$ (or κ_{E}^2=0) means that the transceiver hardware of SR and SD (or SR and SE) is perfect. As expected, OP (IP) increases as increasing (decreasing) $\kappa_{\mathrm{D}}^2\left(\kappa_{\mathrm{E}}^2\right)$. As a result, if the SD and SE nodes have the same transceiver hardware, i.e., $\kappa_{\mathrm{D}}^2=\kappa_{\mathrm{E}}^2$, there also exists the OP-IP trade-off. Particularly, the outage performance can be improved when SR and SD are equipped with better transceiver hardware (i.e., κ_{D}^2 and κ_{E}^2 are lower), but the IP performance is worse.

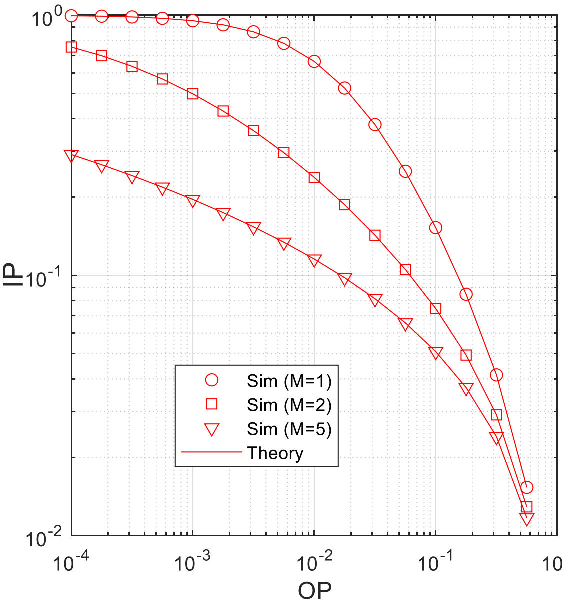


Fig. 4. IP versus OP when $\kappa_{\mathrm{D}}^2=\kappa_{\mathrm{E}}^2=0.05$.

Fig. 4 illustrates the IP/OP trade-off, where we first fix OP by pre-determined values (denoted by $\varepsilon_{\mathrm{OP}}$). Then, by solving $\mathrm{OP}=\varepsilon_{\mathrm{OP}}$, we find values of Δ, by using (16), as

$$\Delta=\frac{\gamma_{\mathrm{th}}\left(\left(\varepsilon_{\mathrm{OP}}\right)^{-\frac{1}{M}}-1\right)}{\Omega_{\mathrm{D}}\left(1-\kappa_{\mathrm{D}}^2\gamma_{\mathrm{th}}\right)}. \tag{23}$$

Next, substituting Δ into (22), we obtain the corresponding values of IP. As we can see from Fig. 4, to obtain lower OP value, the considered scheme gets higher IP one, and vice verse. Moreover, with the same OP; IP decreases as increasing *M*. Therefore, increasing the number of relays also enhances the OP-IP trade-off.

V. Conclusion

This paper derived new formulas of OP and IP for the proposed PLS USS-CR networks with the opportunistic relay selection and under impact of hardware imperfection. The results showed the OP-IP trade-off, which could be significantly enhanced by increasing the number of secondary relays.

In future, we will develop our scheme to the multi-eavesdropper ones as well as evaluate the OP-IP trade-off of the proposed schemes over Nakagami-*m* or Rician fading channels.

References

[1] N. Nguyen, et al, "Secure Massive MIMO With the Artificial Noise-Aided Downlink Training," IEEE Journal on Selected Areas in Communications, vol. 36, no. 4, pp. 802–816, Apr. 2018.

[2] B. M. ElHalawany, A. A. A. El-Banna and K. Wu, "Physical-Layer Security and Privacy for Vehicle-to-Everything," IEEE Communications Magazine, vol. 57, no. 10, pp. 84-90, Oct. 2019.

[3] K. N. Le and V. W. Y. Tam, "Wireless Secrecy Under Multivariate Correlated Nakagami-m Fading," IEEE Access, vol. 8, pp. 33223-33236, Jan. 2020.

[4] Z. Wei, et al, "Energy- and Cost-Efficient Physical Layer Security in the Era of IoT: The Role of Interference," IEEE Communications Magazine, vol. 58, no. 4, pp. 81-87, April 2020.

[5] N. Yang, et al, "Transmit Antenna Selection for Security Enhancement in MIMO Wiretap Channels," IEEE Transactions on Communications, vol. 61, no. 1, pp. 144-154, Jan. 2013.

[6] N. Yang, H. A. Suraweera, I. B. Collings and C. Yuen, "Physical Layer Security of TAS/MRC With Antenna Correlation," IEEE Transactions on Information Forensics and Security, vol. 8, no. 1, pp. 254-259, Jan. 2013.

[7] L. Yang, M. O. Hasna and I. S. Ansari, "Physical Layer Security for TAS/MRC Systems With and Without Co-Channel Interference Over $\eta-\mu$ Fading Channels," EEE Transactions on Vehicular Technology, vol. 67, no. 12, pp. 12421-12426, Dec. 2018.

[8] D. T. Hung, et al, "Performance Comparison between Fountain Codes-Based Secure MIMO Protocols with and without Using Non-Orthogonal Multiple Access," Entropy, vol. 21, no. 10, pp. 1-23, Oct. 2019.

[9] T. M. Hoang, T. Q. Duong, N. -S. Vo and C. Kundu, "Physical Layer Security in Cooperative Energy Harvesting Networks With a Friendly Jammer," IEEE Wireless Communications Letters, vol. 6, no. 2, pp. 174-177, April 2017.

[10] H. D. Hung, T. T. Duy and M. Voznak, "Secrecy Outage Performance of Multi-hop LEACH Networks using Power Beacon Aided Cooperative Jamming with Jammer Selection Methods," AEU-International Journal of Electronics and Communications, vol. 124, ID 153357, pp. 1-25, Sept. 2020.

[11] P. M. Nam, H. D. Hung, T. T. Duy and L. T. Thuong, "Security-Reliability Tradeoff of MIMO TAS/SC Networks using Harvest-to-Jam Cooperative Jamming Methods With Random Jammer Location," ICT Express, 2022. Doi: 10.1016/j.icte.2021.11.003

[12] P. N. Son and H. Y. Kong, "Exact Outage Probability of Two-Way Decode-and-Forward Scheme with Opportunistic Relay Selection Under Physical Layer Security," Wireless Personal Communications, vol. 77, no. 4, pp. 2889-2917, Aug. 2014.

[13] P. T. Tin and T. T. Duy, "Power Allocation Strategies For Dual-Hop Relay Protocols With Best Relay Selection Under Constraint Of Intercept Probability," ICT Express, vol. 5, no. 1, pp. 52-55, Mar. 2019.

[14] H. Li, et al, "Secrecy Outage Probability of Relay Selection Based Cooperative NOMA for IoT Networks," IEEE Access, vol. 9, pp. 1655-1665, Dec. 2020.

[15] D. -H. Ha, et al, "Security-Reliability Trade-Off Analysis for Rateless Codes-Based Relaying Protocols Using NOMA, Cooperative Jamming and Partial Relay Selection," IEEE Access, vol. 9, pp. 131087-131108, Sept. 2021.

[16] X. Ding, et al, "Security-Reliability Tradeoff Analysis of Artificial Noise Aided Two-Way Opportunistic Relay Selection," IEEE Transactions on Vehicular Technology, vol. 66, no. 5, pp. 3930-3941, May 2017.

[17] W. Khalid, et al, "RIS-Aided Physical Layer Security With Full-Duplex Jamming in Underlay D2D Networks," IEEE Access, vol. 9, pp. 99667-99679, Jul. 2021.

[18] L. -T. Tu and A. Bradai, "On the Performance of Physical Layer Security of RIS-aided Communications," in Proc. of 2021 IEEE Conference on Antenna Measurements & Applications (CAMA), 2021, pp. 570-574.

[19] T. D. Hieu, T. T. Duy and S. G. Choi, "Performance Enhancement for Harvest-to-Transmit Cognitive Multi-hop Networks With Best Path

Selection Method Under Presence Of Eavesdropper," in Proc. of ICACT 2018, Feb. 2018, pp. 323-328

[20] P. T, Tin, et al, "Outage Analysis of the Power Splitting Based Underlay Cooperative Cognitive Radio Networks," Sensors, vol. 21, no. 22, 7653, Nov. 2021.

[21] T. L. Thanh, et al, "Broadcasting in Cognitive Radio Networks: A Fountain Codes Approach," IEEE Transactions on Vehicular Technology, 2022. Doi: 10.1109/TVT.2022.3188969

[22] W. Khalid, et al, "RIS-Aided Physical Layer Security With Full-Duplex Jamming in Underlay D2D Networks," IEEE Access, vol. 9, pp. 99667-99679, Jul. 2021.

[23] P. Yan, Y. Zou, X. Ding and J. Zhu, "Energy-Aware Relay Selection Improves Security-Reliability Tradeoff in Energy Harvesting Cooperative Cognitive Radio Systems," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5115-5128, May 2020.

[24] P. Chakraborty and S. Prakriya, "Secrecy Outage Performance of a Cooperative Cognitive Relay Network," IEEE Communications Letters, vol. 21, no. 2, pp. 326-329, Feb. 2017.

[25] Z. Xiang, et al, "NOMA-Assisted Secure Short-Packet Communications in IoT," IEEE Wireless Communications, vol. 27, no. 4, pp. 8-15, Aug. 2020.

[26] X. Zheng, J. Zhang and G. Pan, "On Secrecy Analysis of Underlay Cognitive UAV-Aided NOMA Systems with TAS/MRC," IEEE Internet of Things Journal, (2022). Doi: 10.1109/JIOT.2022.3181826.

[27] P. T. Tin, D. T. Hung, T. T. Duy and M. Voznak, "Analysis of Probability of Non-zero Secrecy Capacity for Multi-hop Networks in Presence of Hardware Impairments over Nakagami-m Fading Channels," RadioEngineering, vol. 25, no. 4, pp. 774-782, Dec. 2016.

[28] M. Li, et al., "Effects of Residual Hardware Impairments on Secure NOMA-Based Cooperative Systems," IEEE Access, vol. 8, pp. 2524-2536, Nov. 2020.

[29] X. Li, et al, "Secrecy Analysis of Ambient Backscatter NOMA Systems Under I/Q Imbalance," IEEE Transactions on Vehicular Technology, vol. 69, no. 10, pp. 12286-12290, Oct. 2020.

[30] B. Li, Y. Zou, J. Zhu and W. Cao, "Impact of Hardware Impairment and Co-Channel Interference on Security-Reliability Trade-Off for Wireless Sensor Networks," IEEE Transactions on Wireless Communications, vol. 20, no. 11, pp. 7011-7025, Nov. 2021.

[31] K. Guo, et al, "On the Secrecy Performance of NOMA-Based Integrated Satellite Multiple-Terrestrial Relay Networks With Hardware Impairments," IEEE Transactions on Vehicular Technology, vol. 70, no. 4, pp. 3661-3676, April 2021.

[32] K. Guo, C. Dong and K. An, "NOMA-Based Cognitive Satellite Terrestrial Relay Network: Secrecy Performance Under Channel Estimation Errors and Hardware Impairments," IEEE Internet of Things Journal, vol. 9, no. 18, pp. 17334-17347, Sept. 2022.

[33] Y. Liu, et al, "Relay Selection for Security Enhancement in Cognitive Relay Networks," IEEE Wireless Communications Letters, vol. 4, no. 1, pp. 46-49, Feb. 2015.

[34] Y. Zou, B. Champagne, W. -P. Zhu and L. Hanzo, "Relay-Selection Improves the Security-Reliability Trade-Off in Cognitive Radio Systems," IEEE Transactions on Communications, vol. 63, no. 1, pp. 215-228, Jan. 2015.

[35] P. T. Tin, et al, "Secrecy Performance of TAS/SC-based Multi-hop Harvest-to-Transmit Cognitive WSNs under Joint Constraint of Interference and Hardware Imperfection," Sensors MDPI, vol. 19, no. 5, (1160), March 2019.

[36] P. T. D. Ngoc, T. T. Duy and H. V. Khuong, "Outage Performance of Cooperative Cognitive Radio Networks under Joint Constraints of Co-Channel Interference, Intercept Probability and Hardware Imperfection," EAI Transactions on Industrial Networks and Intelligent Systems, vol. 6, no. 19, pp. 1-8, Jun. 2019.

[37] T. T. Duy, L. C. Khan, N. T. Binh and N. L. Nhat, "Intercept Probability Analysis of Cooperative Cognitive Networks Using Fountain Codes and Cooperative Jamming," EAI Transactions on Industrial Networks and Intelligent Systems, vol. 8, no. 26, pp. 1-9, Apr. 2021.

[38] K. Tourki, H. -C. Yang and M. -S. Alouini, "Accurate Outage Analysis of Incremental Decode-and-Forward Opportunistic Relaying," IEEE Transactions on Wireless Communications, vol. 10, no. 4, pp. 1021-1025, Apr. 2011.

[39] K. Tourki, K. A. Qaraqe and M. -S. Alouini, "Outage Analysis for Underlay Cognitive Networks Using Incremental Regenerative Relaying," IEEE Transactions on Vehicular Technology, vol. 62, no. 2, pp. 721-734, Feb. 2013.

[40] K. Tourki, F. A. Khan, K. A. Qaraqe, H. -C. Yang and M. -S. Alouini, "Exact Performance Analysis of MIMO Cognitive Radio Systems Using Transmit Antenna Selection," IEEE Journal on Selected Areas in Communications, vol. 32, no. 3, pp. 425-438, Mar. 2014.

[41] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, 2014.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 171–176 DOI: 10.15439/2022R29 ISSN 2300-5963 ACSIS, Vol. 33

A hybrid method based MPP tracking strategy for solar power systems

Sy Ngo
Faculty of Technology and Engineering
Thu Dau Mot University
Binh Duong City, Vietnam
syn@tdmu.edu.vn

Thanh-Dong Ngo*
Faculty of Technology and Engineering
Thu Dau Mot University
Binh Duong City, Vietnam
dongnt@tdmu.edu.vn

Cao-Tri Nguyen
Faculty of Technology and Engineering
Thu Dau Mot University
Binh Duong City, Vietnam
trinc@tdmu.edu.vn

***Abstract*—One of the key factors used to measure the efficiency of a solar power system is the convergence time during the MPP tracking. In other words, the number of duty ratio samples should be as low as possible. In particular, the MPP tracking takes a long time and easily falls into the local MPP when the PV system is partially shaded, which reduces the operational efficiency of the PV system. The incremental conductance (In-Cond) algorithm and the improved grey wolf optimization (GWO) method have been combined to provide a novel approach for a standalone PV power system to overcome this drawback. In the proposed methodology, the global MPP is searched using the hybrid method, which is integrated the improved GWO with the In-Cond algorithm. To demonstrate the feasibility of the proposed method, MATLAB simulations the PV system are provided. The global MPP is not only obtained under uniform irradiance, but also under partial shading influences.**

***Index Terms*—Partial shading influence, photovoltaic (PV), maximum power point (MPP), boost converter, direct current (DC), grey wolf optimization (GWO), incremental conductance (In-Cond) algorithm.**

I. Introduction

Renewable energy sources including solar energy, wind energy, tidal energy, and hydrogen energy are considered clean energy sources. Among them, the solar energy is not only universal but also non-polluting and inexhaustible, it is one of the most popular energy sources. To enhance the solar energy conversion efficiency, many authors have proposed solutions to find the global MPP by using the perturbation and observation (P&O) method [1], the In-Cond algorithm [2], etc. The P&O and In-Cond algorithms are most commonly used in MPP tracking because they are easy to program, simple, and quickly reach the MPP. But the demerit of the P&O method is easy to oscillate around the MPP which reduces the performance of the PV system. Furthermore, under partial shading influences, these traditional methods are easily fallen into the local MPPs. To overcome these failures, researchers have addressed them through a number of innovative modifications [3]. However, an optimal solution for the PV system to operate in the real environment has not to be found, if these innovative solutions operate well in a homogeneous environment, they will be difficult with complex environmental events, and vice versa. Some solutions have performed relatively well, but design constraints and the computational complexity are large when the implemented processors are cheap and have low reliability.

The P-V curve has only one peak under the uniform distribution of PV irradiance, the above methods can easily catch this peak. However, when PV power panels are affected by environmental factors (e.g. clouds and obstacles), they produce not only global MPP but also many local MPPs. In these situations, the above traditional methods cannot find the global MPP of the entire operating area. Therefore, an innovative method is needed to track the global maximum power area under all environmental conditions. Many improved methods have been listed [4, 5], which can be classified into two kinds: hardware based maximum power tracking methods and soft computing based maximum power tracking methods. The MPP tracking methods based on soft computing included metaheuristic methods [6], improved metaheuristic methods [7], and hybrid methods [8-10]. On the contrast, hardware based methods included PV power system reconfigurations [11], etc. However, each method has its advantages. The general goal of these methods is to search the global MPP quickly, the output power of the PV system has low oscillation and simple calculations.

Based on the results of the above studies, this paper has designed a PV maximum power controller and achieved the expected results. Inspired by the grey wolf optimization [12] and the hybrid method combining the GWO and P&O methods [13], a new hybrid method is proposed to effectively search the global MPP. In this hybrid method, the improved GWO method is applied to search the global region quickly, and move to the In-Cond algorithm to find the MPP in this global area. This solution not only reduces the calculation time and calculation manipulation but also reduces the ripple amplitude of the output power. A stand-alone PV power conversion system including an MPP tracking controller, a DC boost converter [14], and a load has implemented [15]. With different duty ratio values, the system will receive different power values from the PV power panels. Improved GWO method uses these duty ratio values to find the global power area even considering partial shading influences as well as uniform irradiances.

II. Standalone solar power system

A. Model of the solar power panel

The PV cells are connected in series and parallel to form the PV power array ($N_s * N_p$) as seen in [16]. The output current I_{pv} and output power P_{pv} of the PV panels are calculated using the following equations:

$$I_{pv} = I_{ph} * N_p - I_{sa} * N_p * (e^{\frac{k_o * V_{pv}}{N_s}} - 1) \tag{1}$$

$$P_{pv} = V_{pv} * I_{pv} = V_{pv} * I_{ph} * N_p - V_{pv} * I_{sa} * N_p * (e^{\frac{k_o * V_{pv}}{N_s}} - 1) \tag{2}$$

where I_{sa} is a reverse saturation current, $k_o = q/(\mathrm{n} * k * T)$, q is the amount of charge of a single electron ($1.6 * 10^{-19} c$), k is Boltzmann constant ($1.38 * 10^{-23} J/{}^oK$), T is the operating temperature of the PV power panel (${}^oK$), n is a factor of PV cell, V_{pv} is the output voltage of the PV power array. The current source I_{ph} effected by the PV irradiance intensity is calculated as follows:

$$I_{ph} = (I_{sc} + K_{sc} * (T - T_{ref})) * \lambda / 100 \tag{3}$$

where I_{sc} is the short-circuit current at the reference temperature $T_{ref}({}^oK)$ and PV irradiance condition 100 mW/cm^2, K_{sc} is the temperature coefficient ($mA/{}^oC$) of the short-circuit current, λ is the PV irradiance intensity (mW/cm^2). Different environmental conditions will produce different P-V curves. When the same PV irradiance intensity, the output power of the PV system decreases if ambient temperature increases. Conversely, when considered at the same ambient temperature, the output power of the PV panel will increase if the PV irradiance intensity increases. In real-world environments, panels can be affected by both different irradiance intensity and temperature, and finding the MPP is more complicated.

B. Partial shading influences

In order to increase the output voltage, the PV power panels are connected in series. A diode is connected in parallel to the PV panel to isolate this panel in the worst situation. Fig. 1 shows an example of three-series PV power panels. When the irradiance intensity of each PV power panel is the same, they will generate the same current values. If one PV power panel is shaded by clouding etc., the output current and voltage of the shaded PV power panel will decrease. In the worst situation, the shaded PV power panel generated the current is equal to zero, the parallel diode with this PV power panel will turn on and the shaded PV power panel is isolated to avoid the minimum output current. For example, the PV power panels are set the PV irradiance intensities as shown in Fig. 1 (from top to bottom). Allow us denote I_1, I_2 and I_3 as the short-circuit currents for the three PV irradiance intensities, respectively (i.e., $I_1 > I_2 > I_3$). if the output current of the measured PV array is $I_{pv} < I_3$, this current is limited by PV panels with the smallest PV irradiance intensity, which means these PV panels are working normally. The output voltage of the PV array is $V_{pv} = v_{pv1} + v_{pv2} + v_{pv3}$. When the output current of the PV array is limited by $I_2 < I_{pv} < I_1$, the third PV power panel has a current equal to zero, and the diode D_3 conducts the current of the PV array. That means the third PV power panel is isolated, i.e., $I_{pv} = i_{pv1} = i_{pv2}$, $i_{pv3} = 0$, the output voltage of the PV array is $V_{pv} = v_{pv1} + v_{pv2}$. From (1), the output voltage of the PV power panel is calculated as follows:

$$v_{pv(j)} = \frac{N_s}{k_o} \ln\left(\frac{I_{sa} * N_p + I_{ph(j)} * N_p - i_{pv(j)}}{I_{sa} * N_p}\right); j = 1, 2, ..., n \tag{4}$$

where $v_{pv(j)}$ and $i_{pv(j)}$ are the output voltage and output current of the j^{th} PV power panel. $I_{ph(j)}$ depends on the PV irradiance intensity.

In addition, to illustrate the partial shading influences, the PV irradiance intensities have been set to get the P-V curves as seen in Fig. 2. When the PV system under partial shading influences, three power peaks are generated, one peak has the highest power, the remaining two peaks have the lower power. More PV irradiance intensity changes will create more local peaks. Because many power peaks are generated, the traditional method easily falls into the local MPP area. A new MPP search method is needed to effectively tracking the global MPP of the PV array.

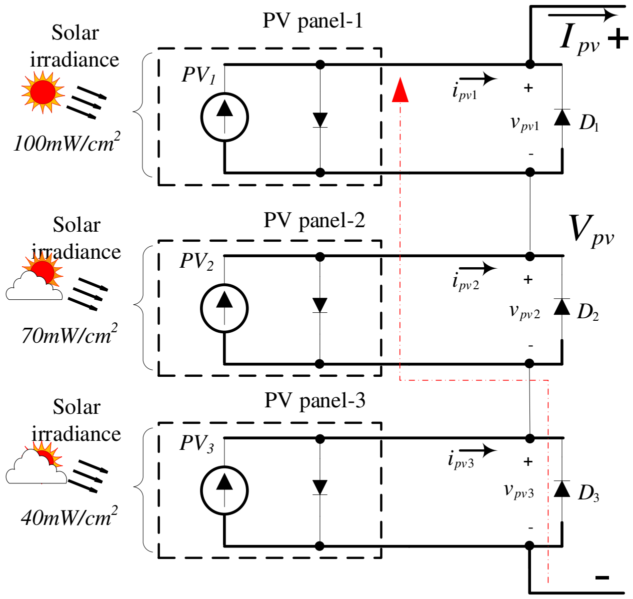


Fig. 1.Model of one partial shading situation of the PV string.

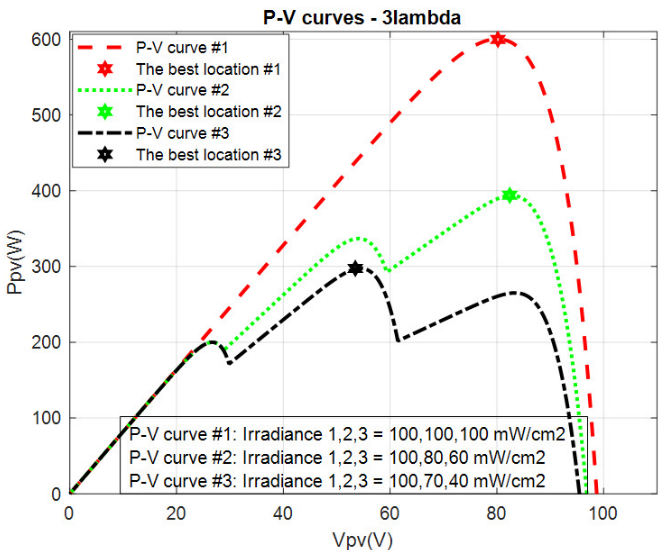


Fig. 2. P-V curves under partial shading influences.

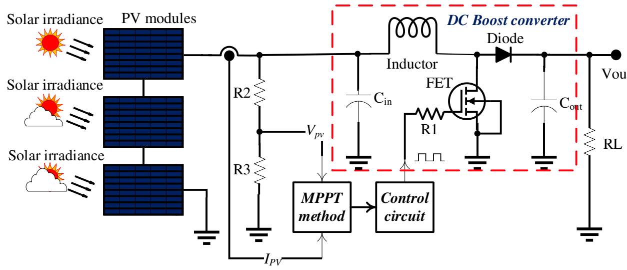


Fig. 3.Basic controller and boost converter for the PV panels.

III. MPP TRACKING STRATEGY OF THE SOLAR POWER SYSTEM

A. *Circuit design for the stand-alone solar power system*

For efficient use of PV panels, a stand-alone PV energy conversion system is illustrated in Fig. 3. This system includes PV power panels, the MPPT method, a control circuit, a DC boost converter, and a DC load. When the PV power panels receive solar irradiance energy, they will generate the current and voltage. The proposed hybrid method is used to track the global MPP. From the searched result, the control circuit is applied to control the duty ratio of the boost converter, and this boost converter is used to transfer the global maximum power value into the load. When the system operates, the initial duty ratio (D) value is chosen based on the assumption that the working power is greater than or equal to 80% of the maximum PV power. The solar irradiance intensity on the PV power panels is assessed by both the measured voltage and current. The measured current after processing is connected to the control circuit. The measured voltage value after using a voltage divider with a resistor bridge to get the appropriate voltage value is also connected to the control circuit for processing. The control circuit uses these two signals with the proposed hybrid method to find the best MPP. In this proposed method, the location of the grey wolf has replaced by the duty ratio and the fitness F_i has also replaced by the output power (P_{pv}) of the PV panels.

B. *Proposed MPP tracking method for solar power systems*

As mentioned above, when PV power panels are shaded by the environment conditions, many power peaks are generated, the conventional MPP search methods will be easy to fall into the local MPP. The power conversion efficiency of the PV energy array will reduce. Therefore, this paper was proposed a combined solution, which is a hybrid method between the improved GWO method and the traditional In-Cond method. The improved GWO method is used not only to avoid local areas but also to quickly find the global maximum energy area. When the global area is found, it will be switched to the In-Cond algorithm to achieve the global MPP. This hybrid method can effectively prevent the local trapped area, and it is classified into two stages:

1) Phase 1. Grey wolf optimization method

The grey wolf optimization (GWO) method is used to solve optimization problems in the global area search. The GWO method is known as one of the most modern heuristic optimization algorithms that first introduced in [17]. This algorithm is inspired by the lifestyle of the grey wolves in the process of living and hunting prey in wildlife. Mostly, the grey wolves like to live in a group 5 to 10 on average. The social dominance hierarchy of the grey wolves is very strict, their leadership is *classified* into four levels with ascending dominance from top to bottom [18]. The behavior of grey wolves has been mimicked in the optimization field with the grey wolf algorithm. The leadership hierarchy of the grey wolves is done by assuming the leader levels as follows: the leading wolves are called the level 1 ($L1$), subleaders are called the level 2 ($L2$), the lower rank wolves are called the level 3 ($L3$), and the lowest rank wolves are called the leader level 4 ($L4$). When the grey wolves hunt prey, they encircle prey during the hunt. The behavior of encircling prey is modeled with mathematical equations [19]. In the first phase of the hybrid method, the improved GWO method was proposed for a new approach by dividing N areas with incremental values from $D_{\min}$ to $D_{\max}$, and each wolf will randomly work within its range. In this way, the improved GWO method found the global area more quickly than the solution using random wolves without arrangement. Each grey wolf will randomly move within its area as follows:

$$D_n = \left[(D_{\min} + (n-1)*\Delta D) \quad (D_{\min} + n*\Delta D)\right] \tag{5}$$

where $\Delta D = (\mathrm{D}_{\max} - \mathrm{D}_{\min}) / N)$, N is the number of grey wolves in the herd; $D_{\min}$ and $D_{\max}$ are the minimum, maximum values of duty ratio. Each D_n location will have a corresponding fitness value ($P_n = P(D_n)$). Among these N locations, the best fitness location is called D_{best} (or D_{L1}) corresponding to the wolf $L1$, one location is called D_{L2} which the fitness is smaller the fitness of wolf $L1$, and one location is called D_{L3} which the fitness is smaller the fitness of wolf $L2$. After selecting the three best grey wolves, the three grey wolves will continue to prey in their areas. The new location of these grey wolves are updated according to the following equations:

$$D_{Ln\_update} = D_{Ln} \pm \Delta D / k; n = 1,2,3 \tag{6}$$

where k is the number of iterations. Since the three best values are used to update the new locations, this solution significantly reduces the number of test samples in unnecessary locations. Therefore, this new approach greatly reduces the time to find the global area. The grey wolves will finish the hunt when (7) is satisfied. This means that the global area is found.

$$\left|D_{best} - D_{Ln-update}\right| \le \varepsilon \cdot \Delta D \tag{7}$$

where ε is a custom number, it has a value less than 1. Depending on the number of grey wolves, this value decreases as the number of grey wolves increases and vice versa.

2) Phase 2. Incremental conductance algorithm

When the distance between the grey wolves and the prey is satisfied (7), this means that the grey wolves hunted or approached the prey, phase 1 with the improved GWO method will finish. In other words, the improved GWO method has reached global maximum power value or was close to it. To get the perfect solution, the global maximum power value is always achieved in all situations, the In-Cond algorithm is included to find this global maximum power value is always achieved in all situations, the In-Cond algorithm is included to find this global maximum power value. The In-Cond algorithm uses the dP_{pv} / dV_{pv} slope to determine the direction of voltage regulation. The condition to obtain the MPP must be satisfied by the following equation:

$$\frac{dP_{pv}}{dV_{pv}} = \frac{d(V_{pv} * I_{pv})}{dV_{pv}} = I_{pv} + V_{pv} * \frac{dI_{pv}}{dV_{pv}} = 0 \qquad (8)$$

The equation above is completed to yield the following equation:

$$\frac{dI_{pv}}{dV_{pv}} = -\frac{I_{pv}}{V_{pv}} \qquad (9)$$

where dV_{pv} , dI_{pv} are the value differences of the voltage and current measured before and after the increase. By comparing the instantaneous conductivity value (I_{pv} / V_{pv}) with the incremental conductivity value dI_{pv} / dV_{pv}), this method will find the operating point with the maximum power. When the condition (9) is satisfied, the In-Cond algorithm will skip the adjustment process of the duty ratio. The flow chart of the proposed hybrid method is seen in Fig. 4, which outlines each step in the process of finding and achieving a global MPP.

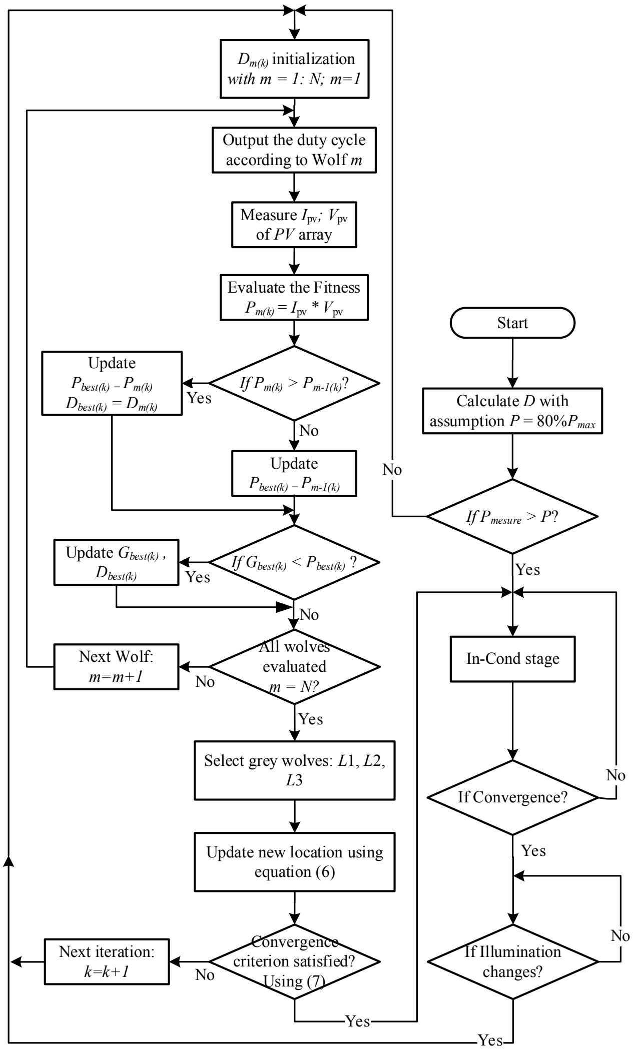

$D_{m(k)}$ initialization with $m = 1: N; m=1$
Output the duty cycle according to Wolf m
Measure I_{pv}; V_{pv} of PV array
Evaluate the Fitness $P_{m(k)} = I_{pv} * V_{pv}$
Update $P_{best(k)} = P_{m(k)}$ $D_{best(k)} = D_{m(k)}$
If $P_{m(k)} > P_{m-1(k)}$?
Yes
No
Update $P_{best(k)} = P_{m-1(k)}$
Update $G_{best(k)}$, $D_{best(k)}$
If $G_{best(k)} < P_{best(k)}$?
Yes
No
Next Wolf: $m=m+1$
All wolves evaluated $m = N$?
No
Yes
Select grey wolves: $L1$, $L2$, $L3$
Update new location using equation (6)
Next iteration: $k=k+1$
Convergence criterion satisfied? Using (7)
No
Yes
Start
Calculate D with assumption $P = 80\%P_{max}$
If $P_{mesure} > P$?
No
Yes
In-Cond stage
If Convergence?
No
Yes
If Illumination changes?
No
Yes

Fig. 4. Flow chart of the proposed hybrid method.

When the system begins to operate, the initial D value is set based on the assumption that operating power = 80% of the maximum power value. This condition is satisfied when the irradiance intensity on the panels is uniform and maximum. The In-Cond algorithm is applied to minimize the MPP tracking time without performing the sequence from stage 1 to stage 2. This initial D value selection is based on the simulated P-V output curve of the PV power system.

IV. Simulation results

A. Setup

MATLAB simulations have displayed the results of the different methods. The specifications of the PV power panel are P_{max}=200W, N_s=54pcs, N_p=1, V_{oc}=32.9V, V_{max}=26.3V, I_{sc}=8.21A, I_{max}=7.61A. For the improved GWO method, the number of grey wolves is 8. The hybrid method is switched to the In-Cond algorithm after satisfying the condition (7).

The differences among the In-Cond algorithm, the conventional GWO method, the PSO, and the proposed method were simulated for considering the partial shading influences as shown in Fig. 2. The GWO method named "conventional GWO" used the equation in [19] to update the locations of the wolves. The methods were simulated in all 3 situations, and each situation was simulated in 6.0 seconds.

B. Simulation results

The PV power system is connected as shown in Figure 1(b). This case is also established for the three situations with the irradiance intensities of 100, 100, 100 mW/cm^2; 100, 80, 60 mW/cm^2; and 100, 70, 40 mW/cm^2, respectively (see Fig. 2). The simulation results are displayed in Fig. 5 to Fig. 8.

In circumstances of the partial shading influence, the In-Cond algorithm also falls into the local MPP (situations 2 and 3). Similarly, The PSO method takes a longer time than the hybrid method to get the global MPP [6]. The conventional GWO method not only takes a long time to find the global MPP but also sometimes fails to reach this global MPP. With the combination of the improved GWO method and the In-Cond algorithm, the global MPP is easy achieved. In this paper, the improved GWO method has found the

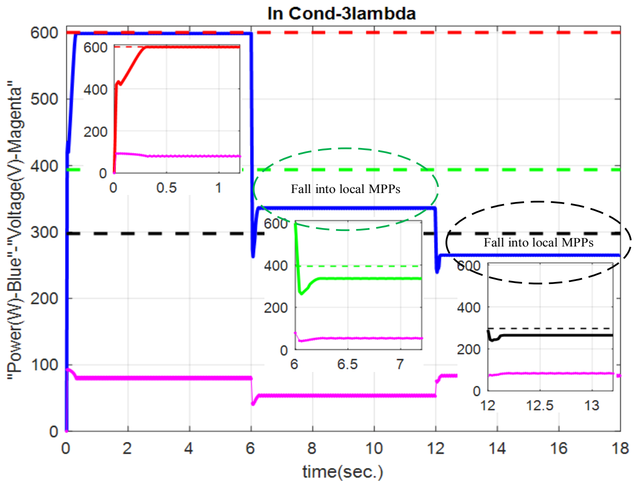


Fig. 5. In-Cond algorithm under partial shading influences.

global area quickly. As a result, the proposed hybrid method is successful in searching and tracking the global MPP in all 3 situations.

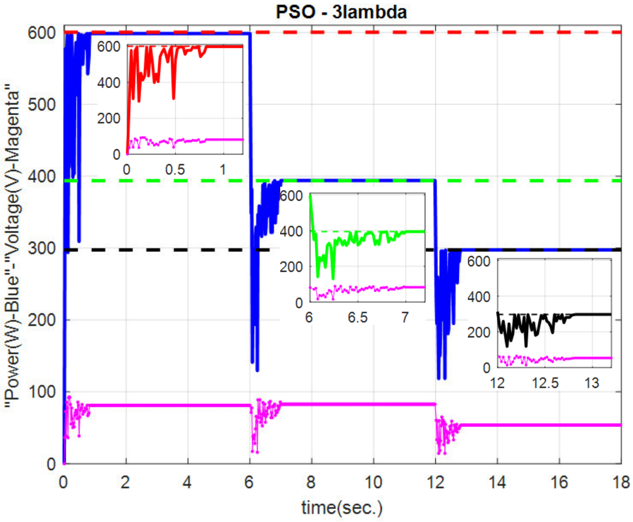


Fig. 6. PSO method under partial shading influences.

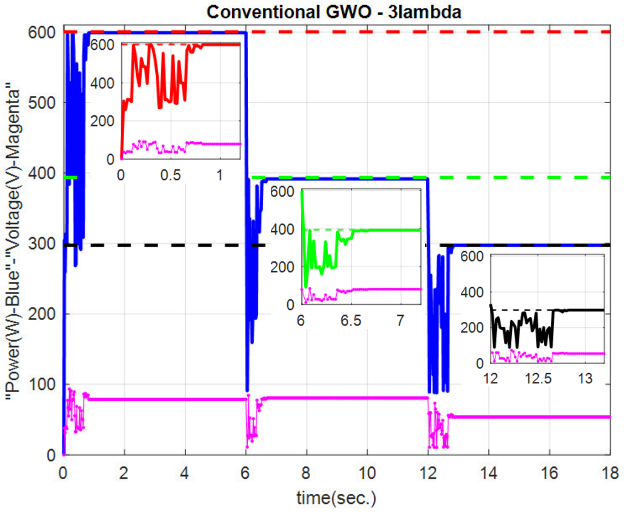


Fig. 7. Conventional GWO method under partial shading influences.

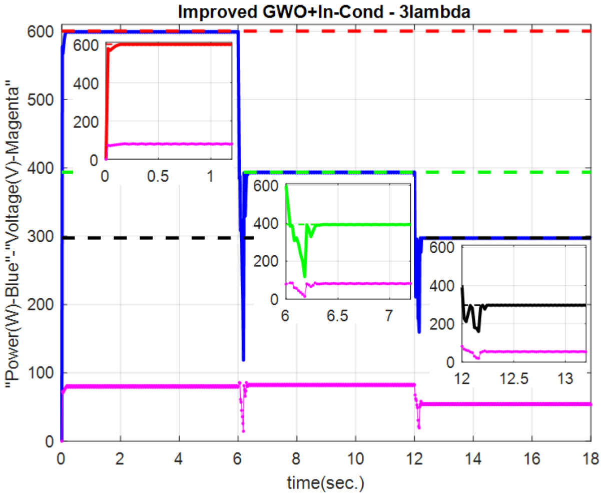


Fig. 8. Proposed hybrid method under partial shading influences.

Since the three best grey wolves were selected to update the new locations, the proposed hybrid method is reduced the execution time as mentioned above. Besides, the convergence of this method is faster than the conventional GWO method [13, 18]. The power values tend to gradually approach the maximum power value as shown in Fig. 8.

V. Conclusion

This paper was used a stand-alone PV power system with the proposed method. With the arrangement of the initial duty ratio values combined with the update of the selected duty ratio values, the proposed hybrid method has achieved the expected results. This proposed method has improved the disadvantages of conventional MPP tracking methods under partial shading influences. The proposed method is effective not only to avoid the local MPP but also to quickly achieve the best PV power of the energy conversion process. Besides, the ability to quickly converge, the simple way of updating values are also the highlights of this paper compared to the conventional GWO, PSO methods, and In-Cond algorithm.

References

[1] V. R. Kota, and M. N. Bhukya, “A novel linear tangents based P&O scheme for MPPT of a PV system,” *Renewable Sustainable Energy Reviews,* vol. 71, pp. 257-267, 2017.

[2] C. Li, Y. Chen, D. Zhou, J. Liu, and J. Zeng, “A High-Performance Adaptive Incremental Conductance MPPT Algorithm for Photovoltaic Systems,” *Energies,* vol. 9, no. 4, pp. 288, 2016.

[3] J. Ahmed, and Z. Salam, “An enhanced adaptive P&O MPPT for fast and efficient tracking under varying environmental conditions,” *IEEE Transactions on Industrial Informatics,* vol. 9, no. 3, pp. 1487-1496, 2018.

[4] C.-S. Chiu, and S. Ngo, “A Novel Algorithm-based MPPT Strategy for PV Power Systems under Partial Shading Conditions,” *Elektronika ir Elektrotechnika,* vol. 28, no. 1, pp. 42-51, 2022.

[5] A. Ashouri-Zadeh, M. Toulabi, A. S. Dobakhshari, S. Taghipour-Broujeni, and A. M. Ranjbar, “A novel technique to extract the maximum power of photovoltaic array in partial shading conditions,” *International Journal of Electrical Power & Energy Systems,* vol. 101, pp. 500-512, 2018.

[6] S. Veerapen, H. Wen, X. Li, Y. Du, Y. Yang, Y. Wang, and W. Xiao, “A novel global maximum power point tracking algorithm for photovoltaic system with variable perturbation frequency and zero oscillation,” *Solar Energy,* vol. 181, pp. 345-356, 2019.

[7] N. H. Saad, A. A. El-Sattar, and A. E.-A. M. Mansour, “Improved particle swarm optimization for photovoltaic system connected to the grid with low voltage ride through capability,” *Renewable Energy,* vol. 85, pp. 181-194, 2016.

[8] A. M. Eltamaly, and H. M. H. Farh, “Dynamic global maximum power point tracking of the PV systems under variant partial shading using hybrid GWO-FLC,” *Solar Energy,* vol. 177, pp. 306-316, 2019.

[9] C.-S. Chiu, and S. Ngo, “Hybrid SFLA MPPT design for multi-module partial shading photovoltaic energy systems,” *International Journal of Electronics*, pp. 1-22, 2022.

[10] S. Ngo, C.-S. Chiu, and P.-T. Nguyen, "MPPT Design Using the Hybrid Method for the PV System Under Partial Shading Conditions." pp. 77-87.

[11] D. S. Pillai, J. P. Ram, M. S. S. Nihanth, and N. Rajasekar, “A simple, sensorless and fixed reconfiguration scheme for maximum power enhancement in PV systems,” *Energy Conversion and Management,* vol. 172, pp. 402-417, 2018.

[12] A. Tjahjono, D. O. Anggriawan, M. N. Habibi, and E. Prasetyono, “Modified Grey Wolf Optimization for Maximum Power Point Tracking in Photovoltaic System under Partial Shading Conditions,” *International Journal on Electrical Engineering Informatics,* vol. 12, no. 1, pp. 94-104, 2020.

[13] S. Mohanty, B. Subudhi, and P. K. Ray, “A grey wolf-assisted perturb & observe MPPT algorithm for a PV system,” *IEEE Transactions on Energy Conversion,* vol. 32, no. 1, pp. 340-347, 2017.

[14] I. Korotyeyev, “Steady-state analysis of DC converter using Galerkin’s method,” *The international journal for computation mathematics in electrical electronic engineering*, 2019.

[15] M. Malekzadeh, A. Khosravi, and M. Tavan, “Observer based control scheme for DC-DC boost converter using sigma–delta modulator,” *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering,* vol. 37, no. 2, pp. 784-798, 2018.

[16] S. Ngo, and C. S. Chiu, "Simulation Implementation of MPPT Design under Partial Shading Effect of PV Panels." pp. 1-6.

[17] S. Mirjalili, and S. M. Mirjalili, “Grey wolf optimizer,” *Journal Advances in engineering software,* vol. 69, pp. 46-61, 2014.

[18] S. K. Cherukuri, and S. R. Rayapudi, “Enhanced grey wolf optimizer based MPPT algorithm of PV system under partial shaded condition,” *International Journal of Renewable Energy Development,* vol. 6, no. 3, pp. 203, 2017.

[19] F.-S. Pai, and P.-S. Tseng, “An efficient GWO MPPT for a PV system using impedance information acceleration,” *International Journal of Electronics,* vol. 106, no. 4, pp. 648-661, 2019/04/03, 2019.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 177–182 DOI: 10.15439/2022R34 ISSN 2300-5963 ACSIS, Vol. 33

From A Proposed CNN Model to A Real-World Application in Rice Disease Classification

Hoang Long Nguyen
Hong Duc University
Thanh Hoa, Viet Nam
longlcf0311@gmail.com

Thi Ha Tran
HUST
Ha Noi, Viet Nam
hattran.set@gmail.com

Hong Ha Le Thi
Hong Duc University
Thanh Hoa, Viet Nam
lethihongha@hdu.edu.vn

Dinh Cong Nguyen
Hong Duc University
Thanh Hoa, Viet Nam
nguyendinhcong@hdu.edu.vn

***Abstract*—A compact and precise application of rice disease classification is helpful to assist farmers in their work for treatment on the plants and therefore could be quick and accurate to measure and eliminate the effects of diseases more profitably. In the past, the works were completed by naked-eye observation and basically relied on the experiences. Even so, the results are quite subjective and heuristic. In this paper, a mobile application to automatically classify several kinds of rice diseases from rice plant images and then to accurately recommend the uses of pesticides or chemicals. To do so, a proposed convolutional neural network (CNN) model is given. The results show that the proposed CNN model achieves the performance with the best trade-off between accuracy and time efficiency in comparison with the state-of-the-art models in our dataset. This model could be easily embedded into a mobile application to process in near real-time processing.**

***Index Terms*—Rice diseases, Mobile app, CNN.**

I. Introduction

In Vietnam, the agricultural sector still plays a key role in the economy. Based on the report of the general statistics office of Vietnam, the sector donated 23.52%[1] to the total economic growth in 2021, proving its role as the important pillar of Vietnam's economy. Among agricultural growth, the rice sector is known as one of the most crucial foundations, it contributes 30%[2] to the overall agricultural production values.

Rice has been grown in Vietnam for about two thousands of years. However, the application of advances in high technology in cultivation is limited and has not kept pace with the world. In many regions, some tasks rely on the manual works with experiences, shown in Figure 1.

In fact, there are several kinds of diseases that directly resulted in the yield losses, sometimes standing at about 15% on average [1]. Therefore, farmers need to detect the diseases and to have effective solutions to eliminate their effects. In the past, these tasks were mostly based on manual inspections, and then to extirpate by the experiences. The works are sometimes not really powerful enough and to be a waste of money. Therefore, the need of automatic rice disease detection and then to recommend optimal solutions from experts are required, which could help farmers make less time and effort, and be more effective than the visual ways of detecting[2].

Fig. 1. Visual examples of how farmers are taking care of their crops.

Thanks to technology development, mobile devices nowadays have become more common in our life. Hence, the images captured from the devices become easier than ever. A mobile application (app) using the images to detect and classify rice diseases would be a great approach. To do so, an accurate and efficient detection machine embedded inside the app has gained lots of attention from the literature, especially with convolutional neural networks (CNNs). Over the past decade, CNNs have led the main trend in research activities over the world with object detection and recognition in particular since the first milestone with the AlexNet system in 2012 [3]. Following that the CNN models have been improved time by time and achieved state-of-the-art results, particularly the agricultural sector [4]. However, these deep architectures and systems are time-consuming and little compatible with a low-level energy consumption architectures [22]. Therefore, proposing an accurate model while adapting well with time constraint, is an open question in literature up to this time.

In this paper, our contributions could be listed.

- A new CNN model is proposed to classify four popular kinds of rice diseases in Vietnam. The model is competitive with the top accurate models of the literature while guaranteeing the best trade-off between accuracy and time efficiency.
- A mobile application (app) is designed where the pro-

[1] https://en.nhandan.vn/business/economy/item/10567002

[2] https://www.statista.com/topics/5653/agriculture-in-vietnam/

posed CNN model is embedded inside. Users can be straightforward to install and use the app. Four kinds of rice diseases are classified and then the app returns the guides about the particular disease and the advice from experts on how to kill it.

The remainder of the paper is organized as follows. Section II discusses the related works in the agricultural domain. Then, section III presents our model and app. Next, section IV shows the experimental results. Finally, section V concludes the topic and gives the future works.

II. Related works

Deep learning techniques have a high performance in image classification recently. Various approaches have been used for recognizing and detecting plant diseases with artificial neural networks (ANNs) [6], Support Vector Machine (SVM) for sugar beet diseases [7] based on stages of disease. Some mango leaf diseases were detected in [8], authors combined low level features (shape and color) and CNN features together, and then applied the SVM to classify diseases. Authors in [9] worked on detecting tea leaf diseases by combining feature extraction [10] and neural network ensemble (NNE).

The DNN-JOA method was proposed in [11]. K-means clustering method was applied for the segmentation process. A feedback loop was employed between the classification and segmentation stages to improve the quality of the classification process. At the end, the proposed approach achieved a high overall accuracy about more than 90% for classifying four kinds of paddy leaf diseases.

Different CNN architectures were implemented in two different training methods to detect four stages of apple black rot from images in the PlantVillage dataset [12]. The first method was created and trained models from scratch while the second method was based on the pretrained models such as VGG16, Inceptionv3, and Resnet50. Finally, fine-tuned VGG16 model played as the best candidate with 90.4% of accuracy on the test set.

It is true that, most of the plant diseases are detected on their leaves. Therefore, authors in [13] provided 79,265 leaf images from various weather conditions, angles, and daylight hours. After that, they proposed a two-stage architecture called PlantDiseaseNet. PlantDiseaseNet combined the AlexNet model as the feature extractor and Yolov3 as the classifier to propose the regions of interest in the first stage. Following that, an architecture like the ResNet architecture was applied to classify plant diseases. The model was achieved an accuracy of 93.67% in the whole dataset.

Besides training model from scratch, the transfer learning technique was commonly used in identifying plant leaf symptoms. The pre-trained VGGNet on the ImageNet dataset and the Inception module for the feature extractor were employed and a new CNN model named the INC-VGGN model was proposed for the class prediction in [14]. This model attained an accuracy of 92% on the their own dataset. Similarly to [14], authors in [15] trained fine-tuned AlexNet and GoogleNet to detect nine tomato diseases from 14,828 raw images of tomato leaves.

In the experiment [16], they compared three famous CNN architectures such as the SVM, ResNet, and VGG models to classify four apple leaf diseases. They conducted that ResNet18 with fewer layers of ResNet achieved better recognition performance.

Many models with efficient techniques were used in those researches mentioned above. They focused on accurate plant disease classification by using various types of deep learning models, especially CNN architecture such as AlexNet, GoogleNet, ResNet, and so on. Some of those used neural network ensemble, while others used SVM, ANN. These studies contributed to accurate classification of plant diseases. In addition, there are recent and strong models proposed for an accurate detection and classification like [21], [22]. Nevertheless, most of them focus on modifying the technique to achieve high accuracy and do not consider the effect of a number of parameters on deploying onto mobile applications.

One of our purposes is to deploy a lightweight mobile application so we need a small-parameter model that can run fast on any platform. Additionally, in some remote areas, the internet connectivity is limited or has slow speed. Hence a memory CNN model is necessary for plant diseases recognition and detection. On the other hand, the reduction of the number of parameters in the CNN model limits the capability of learning, therefore, balancing between memory elements and classification accuracy is significant.

To deal with this problem, we proposed a new CNN model that achieved high accuracy despite its tiny number of parameters. We performed many experiments on the MobileNetv3 model [20] with a special attention on the MobileNetv3-small version, the DenseNet201 model [17] and the SimpleCNN model in [18]. Finally, the results show that our proposed CNN architecture surpasses the three models for classifying four rice plant disease classes.

III. The proposed system

In the paper, we propose an end-to-end system. The main purpose is to classify the rice diseases from the app. For convenience, Fig 2 shows our approach. Our contribution here is to design a new CNN model which guarantees the best trade-off between the accuracy and time efficiency. This makes the model reasonable to embed inside the app. Our approach is inspired from the Simple CNN model [18]. To avoid underfitting, we increased model complexity by adding one convolutional layer.

For the illustrative purpose, Figure 3 presents the model with six convolutional layers, followed by Batch normalization and MaxPooling2D layers to extract features. For classifying the rice diseases, we employ the three layers for fully-connected components.

As the second contribution, the mobile application is designed to embed the proposed CNN model inside. When the trained model is obtained by the Keras framework[3],

[3] https://keras.io/

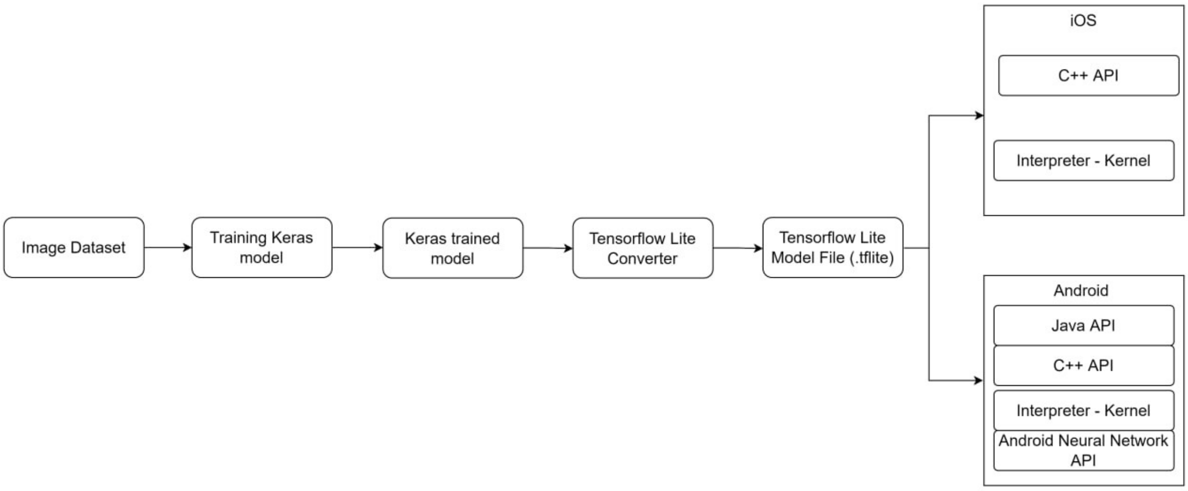


Fig. 2. Our proposed system was developed to classify rice diseases.

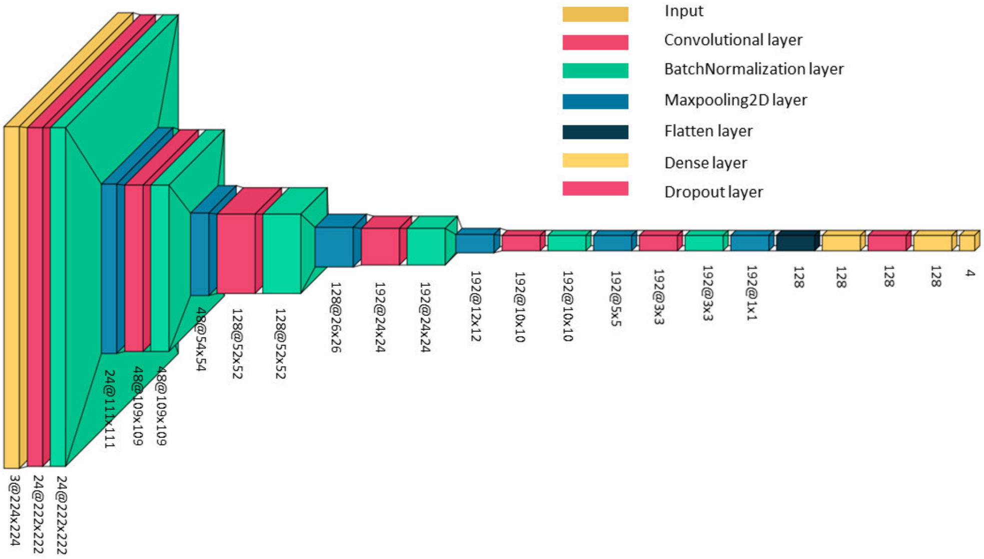


Fig. 3. The proposed CNN model for classification of rice diseases.

it is converted into the TFLite model[4] before deployed in mobile devices. In order to identify and clarify the system requirements, we present here a general use-case diagram for our mobile application in Figure 4.

User could upload or take directly images from smartphones. The app will predict the images and then return the results. User can look at the results of prediction and take a deeper view on the related information and how to treat the corresponding disease. With the extended action, as soon as classified objects are obtained, the correct results will be uploaded to the server. By doing so, it tends to enrich the dataset, known as a key factor to upgrade the proposed model in the next time. In addition, as soon as users take or upload images from smartphones, the timestamp and location will be recorded by the system and saved to the database with the result of the prediction in order that they can accurately handle the disease and track the progress of the treatment. As one of the advantages, it could be a reference for the experts to give their advice on the progress of treatment.

Administrator chooses the best model which suits well with data while updating the information related to the kinds of

[4] https://www.tensorflow.org/lite/guide

diseases. Furthermore, administrator also could retrain the model during the period of time to update with the new samples of data.

With the assumption that most of users are farmers and they may have less knowledge about new techniques. Therefore, should they retrain or select a model without understanding about images, they may get unexpected results. Therefore, we have restricted their right to retrain or update the model. In the next perspective, we could add more features such as a social community so that users can interact with each other for further information.

Moreover, user could interact with user for further information about the specific rice disease, and then give their advice if necessary. In addition, expert also update information or suggest new techniques to treat the diseases.

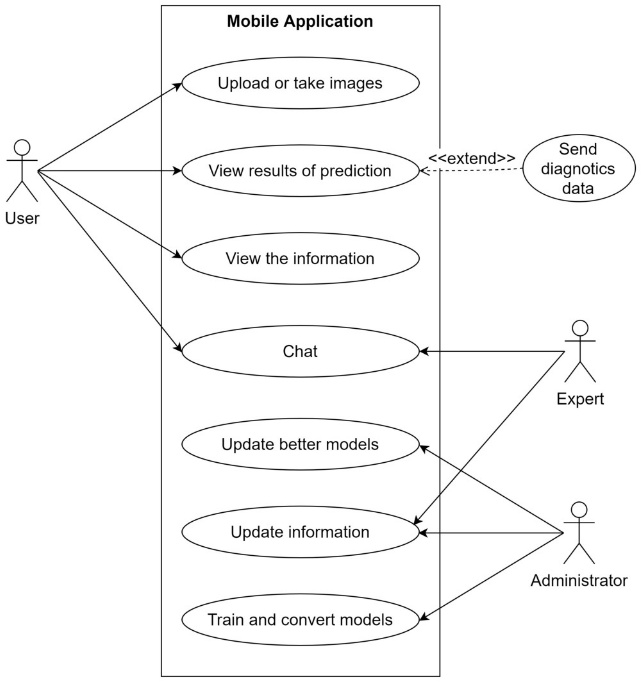


Fig. 4. The general mobile application use-case diagram.

IV. Experimental Results

In this section, we will evaluate the performance of our proposed model against some well-known models. To do so, section IV-A shows the standard dataset while section IV-B presents the environment setups for experiments and the parameter configuration. At last, section IV-C discusses the results.

A. Dataset

For evaluating the proposed model, we use the dataset in [19] which includes 5932 images of four kinds of rice diseases: Bacterial blight, Blast, Brown Spot and Tungro, shown in Table I. Some images from the dataset are given in Figure 5. Based on our experiences, this dataset is large enough to evaluate the performance of the proposed model. In the real deploying process, the proposed model could be periodically retrained to fit well with each region. The dataset is divided into 3 sets with 60% for the training set, 20% for the validation set, 20% for the test set. This division is resulted of several training and testing experiments

TABLE I
Rice Leaf Disease Image Samples.

Class name	Number of images
BacterialBlight	1584
Blast	1440
Brownspot	1600
Tungro	1308

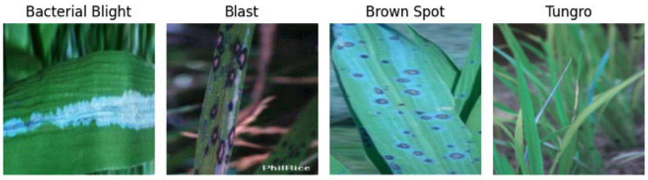


Fig. 5. Images are extracted from the dataset [19].

B. Protocols

To make the performance evaluations are fair enough, we employ the MobileNetv3-small [20], DenseNet201 [17] and SimpleCNN in [18] models in the experiments. All models were trained by baseline learning from scratch. To be more detailed, some configurations of the training process could be listed as optimization algorithm: Adam; learning rate: 0.001; the number of epochs: 120; the loss function: categorical-crossentropy; data augmentation techniques: Rotations, Shifts, Flips, Brightness, Zoom.

For metrics of evaluation process, we employ accuracy and a confusion matrix as the standard metrics. Accuracy measures the number of true classifications over the total number of classifications. A confusion matrix presents the performance of the model where the model is embarrassing when it gives predictions.

Thanks to Google colab notebook[5], all of our experiments for the training and testing of the models were implemented here.

C. Results

For the training set, Figure 6 indicates how well the models have suited with the dataset. As illustrated, the Simple CNN does not perform well with the dataset while the DenseNet201, MobileNetv3-small, proposed models tend to converge.

In the test set, Table II shows the results of the proposed model against other models with the accuracy metric. The DenseNet201 model is the best classifier with 99.65% followed it by the proposed model with 99.21%, the MobileNetv3-small model with 91.3%. However, it is worth

[5] https://colab.research.google.com/

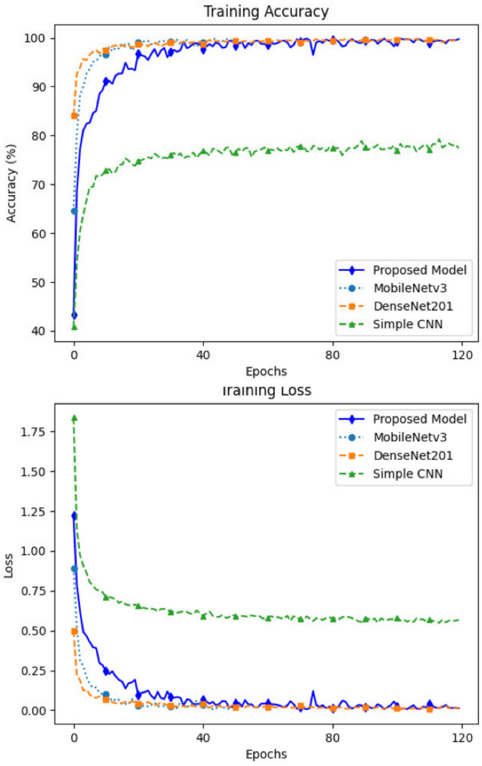


Fig. 6. The training accuracy and training loss among models on the selected dataset.

noting here that the number of parameters of the DenseNet201 model is about 25 times as much as the proposed model. The SimpleCNN model is quite slight, however, it performs a bit poorly in this dataset.

TABLE II
THE PERFORMANCE EVALUATION OF THE PROPOSED MODEL AGAINST OTHER MODELS.

CNN architecture	Number of Parameters	Accuracy
DenseNet201	20 millions	99.65%
MobileNetv3-small	3 millions	91.3%
SimpleCNN	0.276 millions	75.1%
Proposed Model	0.8 millions	99.21%

Figure 7 shows the ability to classify four kinds of the rice diseases. The results prove that the proposed model fits completely with the dataset.

Within the mobile app, we accomplished to deploy the proposed model in both operating systems such as: iOS and Android. Figure 8 shows the returned results from the app. For the average processing time, we measured on Redmi Note 9S[6]. In total with one image, it is about 0.11 seconds with 0.03 seconds in the image processing step and 0.07 seconds in the image classification step. It demonstrates that the app could respond mostly in near real-time processing.

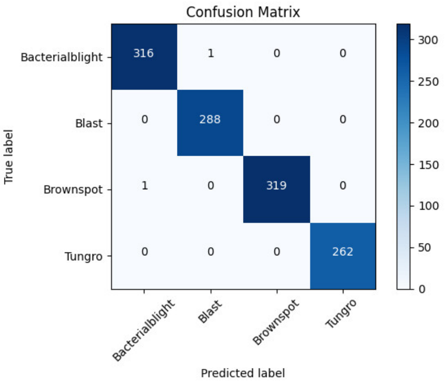


Fig. 7. The confusion matrix of the proposed model on the test set.

V. Conclusion and future works

The main contribution of the paper is to present a novel CNN model for rice disease detection with the best trade-off between accuracy and time-efficiency. Our proposed model is competitive with some state-of-the-art models when obtained about 99.21% in accuracy on the dataset of rice leaf diseases [19]. Moreover, with near 0.8 millions of parameters, our model appears as the top slight architecture. It guarantees the ability to deploy into low-cost hardware architectures. As a case study, we successfully embedded the model into the mobile app within iOS and Android operating systems. It allows users to automatically detect and give the detailed information for each rice leaf disease without the Internet.

As perspectives, the proposed model could be updated in order to detect more kinds of diseases corresponding to each specific cycle of diseases. In addition, the app would allow users to directly retrain the model by themselves from uploaded images. Due to some administrative procedures, the complete mobile app will be available on Google Play Store and App Store soon.

References

[1] J Kihoro, NJ Bosco, H Murage, E Ateka, D Makihara, "Investigating the impact of rice blast disease on the livelihood of the local farmers in greater Mwea region of Kenya". Springerplus 2, no. 1: 1-13, 2013

[2] V. Singh , A.K.Misrab, "Detection of plant leaf diseases using image segmentation and soft computing techniques", Information Processing in Agriculture, 4, 41-49, 2016

[3] A. Krizhevsky, .I Sutskever, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems 25, 2012.

[4] S Sladojevic, M Arsenovic, A Anderla, "Deep neural networks based recognition of plant diseases by leaf image classification", Computational intelligence and neuroscience, 2016

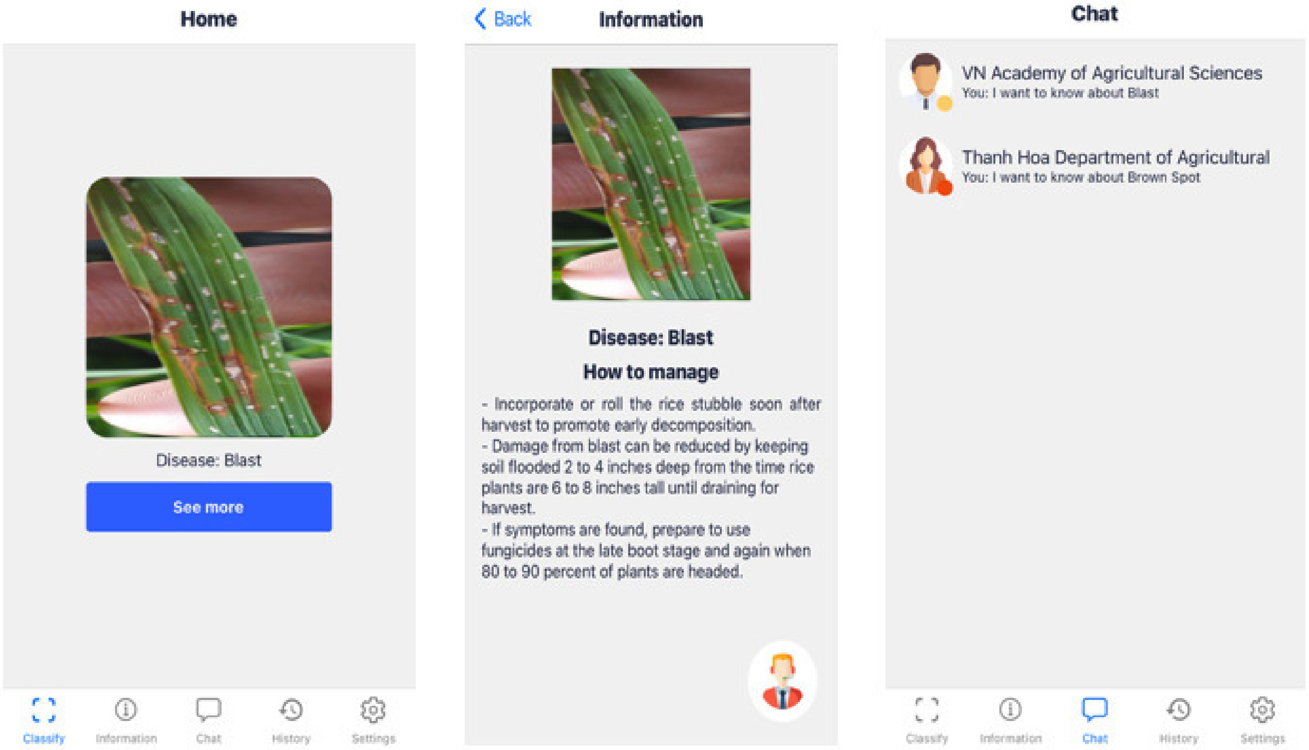
[5] Zhao, R., Niu, X., Wu, Y., Luk, W., Liu, Q. " Optimizing CNN- based object detection algorithms on embedded FPGA platforms". ISARC (2017).

[6] H. Cartwright, Ed., Artificial Neural Networks, Humana Press, 2015

[7] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance," Computers and Electronics in Agriculture, vol. 74, no. 1, pp. 91–99, 2010.

[8] Md. Rasel Mia, Sujit Roy, Subrata Kumar Das* Mango Leaf Diseases Recognition Using Neural Network and Support Vector Machine

[6] https://www.mi.com/global/redmi-note-9s/specs



Disease: Blast

How to manage

- Incorporate or roll the rice stubble soon after harvest to promote early decomposition.
- Damage from blast can be reduced by keeping soil flooded 2 to 4 inches deep from the time rice plants are 6 to 8 inches tall until draining for harvest.
- If symptoms are found, prepare to use fungicides at the late boot stage and again when 80 to 90 percent of plants are headed.

VN Academy of Agricultural Sciences
You: I want to know about Blast

Thanh Hoa Department of Agricultural
You: I want to know about Brown Spot

Fig. 8. Visual examples of the mobile app for classification and then to show the detailed information, or to contact with the expert.

[9] B. C. Karmokar, M. S. Ullah, Md. K. Siddiquee, and K. Md. R. Alam, “Tea leaf diseases recognition using neural network ensemble,” International Journal of Computer Applications, vol. 114, no. 17, pp. 27–30, 2015.

[10] I.Guyon and A. Elisseeff , “An Introduction to Feature Extraction”, Series Studies in Fuzziness and Soft Computing, Physica-Verlag, Springer, 2006.

[11] S. Ramesh, D. Vydeki, “Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm”, Information Processing in Agriculture, Vol. 7, Issue 2, 2020, Pages 249-260, ISSN 2214-3173

[12] Wang, G., Sun, Y., Wang, J. , “Automatic image-based plant disease severity estimation using deep learning”, Computational intelligence and neuroscience, 2017.

[13] Arsenovic M, Karanovic M, Sladojevic S, Anderla A, Stefanovic D, “Solving Current Limitations of Deep Learning Based Approaches for Plant Disease Detection,” Symmetry, 2019

[14] Junde Chen, Jinxiu Chen, Defu Zhang, Yuandong Sun, Y.A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” Computers and Electronics in Agriculture, Volume 173, 2020

[15] B. Mohammed; B.Kamel; M.Abdelouahab, “ Deep learning for tomato diseases: classification and symptoms visualization,” Applied Artificial Intelligence, 2017.

[16] X. Li and L. Rai, ”Apple Leaf Disease Identification and Classification using ResNet Models,” 2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT), 2020, pp. 738-742

[17] G. Huang, Z.Liu “Densely connected convolutional networks,” Proceedings of the IEEE conference on computer vision and pattern recognition, p. 7, 2017.

[18] R.Chowdhury R., et al. “Identification and recognition of rice diseases and pests using convolutional neural networks,” Biosystems Engineering, 2020.

[19] S.Kumar (2020), “Rice Leaf Disease Image Samples”, Mendeley Data.

[20] H.Andrew, et al. ”Searching for mobilenetv3,” Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.

[21] SK Upadhyay, ”Deep Transfer Learning-Based Rice Leaves Disease Diagnosis and Classification model using InceptionV3”, International Conference on Computational Intelligence and Sustainable Engineering Solutions, pp. 493–499, 2022

[22] MH Tunio, L Jianping, MHF Butt, ” Identification and Classification of Rice Plant Disease Using Hybrid Transfer Learning”, ICCWAMTIP, pp. 525–529, (2021)

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 183–188
DOI: 10.15439/2022R31
ISSN 2300-5963 ACSIS, Vol. 33

Material parameter identification for clinching process simulation using neural network metamodels

Duc Vinh NGUYEN
Institute of Southeast Vietnamese Studies
Thu Dau Mot University
Thu Dau Mot City, Vietnam
nguyenducvinh@tdmu.edu.vn

Pai-Chen LIN
Department of Mechanical Engineering
National Chung-Cheng University
Chiayi 62102, Taiwan
imepcl@ccu.edu.tw

Minh Chien NGUYEN
Institute of Southeast Vietnamese Studies
Thu Dau Mot University
Thu Dau Mot City, Vietnam
nguyenminhchien@tdmu.edu.vn

Yang-Jiu WU
National Chung-Cheng University
Chiayi 62102, Taiwan

Hoang Son TRAN
Research Unit, MSM division
University of Liège
Liège, Belgium
hstran@uliege.be

Xuan Van TRAN
Institute of Southeast Vietnamese Studies
Thu Dau Mot University
Thu Dau Mot City, Vietnam
xuantv@tdmu.edu.vn

***Abstract*—Clinching is a mechanical joining method in which two sheet workpieces are clamped and locked together using a punch and a die. Process parameters for such joint elements are often designed based on numeric simulation. Before this step, the identification of good material parameters is crucial to get validated computational results. In this paper, neural network metamodels are used for this specific task as a means to deal with large computation time. The identified material parameters reduce significantly the error between computational results and experimental results.**

***Keywords*—neural network metamodel, material parameters, finite element analysis, optimization, clinching**

I. Introduction

In automotive industry, compared to conventional joining processes such as spot welding, mechanical joining offers many advantages such as less preparatory work, simpler and cheaper equipment, good environmental behavior, no thermal degradation and suitable for joining dissimilar materials [1, 2]. Clinching, also known as mechanical press joining, is a brand of mechanical joining.

During clinching, two sheets of same or different materials and thicknesses are clamped together by an impact extrusion between a punch and a female die. Additional elements such as rivets are not necessary. The two layers are locked as the upper layer is spread into the lower layer inside the gap at the bottom of the die called the die groove (Fig. 1).

Numerical simulation is often used for designing clinching process parameters to reduce the costs for the experiments. Simulation of such joining process has been performed by different authors using the finite element method [3, 4, 5]. One of the main difficulties is about the choice of material parameters, *i.e.* stress-strain relation at high plastic strains (up to 200%) and friction related parameters. For the former one, the tensile curve at low strain (< 10 %) is often fitted into a strain hardening law, such as power law model which in turn is used to extend the stress-strain curve to a wider range of plastic deformation. This extrapolation method does not ensure a correct material behavior at high strain. For the latter one, the coulomb friction model is assumed in most cases. The friction coefficients between different parts are often selected within the logical range in a rather arbitrary way without solid experimental supports for the exact materials and configurations. Shear stress limits, if applied, remains also a subjective choice.

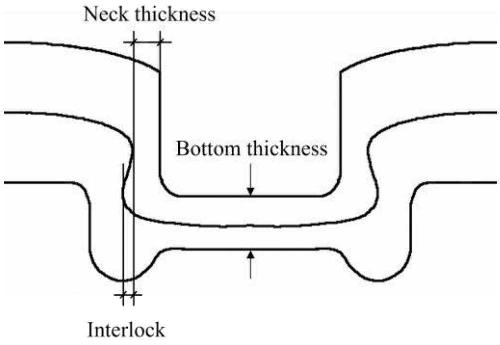


Fig. 1. Typical clinching joint geometry.

In order to select an acceptable combination of parameters which produces comparable results to the experiments, one must run a large amount of simulations. This routine quickly becomes intractable when the number of variable parameters increases and if the computation time is at the range of few hours which is the case for clinching process simulation.

In this study, the application of neural network metamodels is examined to tackle the cumbersome task mentioned above. Specifically, the metamodel replaces computationally expensive simulations in the optimization loop. It will be shown that a neural network can be used to reproduce the effect of material parameters on the punching force and the final joint geometry. It also gives an inside perspective into the unknown material behaviors.

In this paper, the experiments and the simulation model are described in Section 2. The metamodel construction is detailed in Section 3. Its use for the identification of material parameters is then discussed in Section 4.

II. Numerical Simulation

A. Materials and experimental process

In this study, two identical AA5052 aluminium alloy sheets with a thickness of 1.6 mm are joined. During the process, the die is fixed. The punch is driven by a hydraulic machine. At the initial stage, the punch is supported by a solid polyurethane (PU) holder in the form of a ring. The rubber-like holder is deformed as the punch moves downward and guarantees the position of the metal sheets. During the process, the maximum punching force is registered.

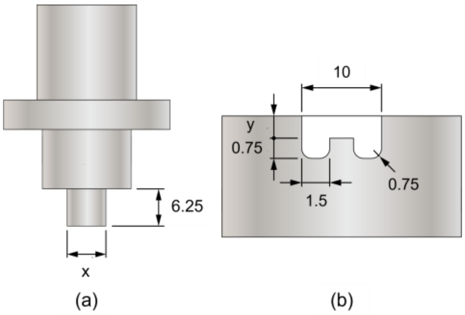


Fig. 2. Punch (a) and die (b) geometry where punch diameter and die depth are denoted as x and y respectively.

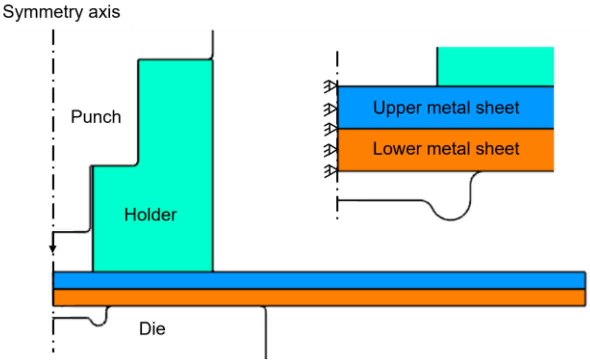


Fig. 3. Illustration of the initial configuration of the FEM model.

The formed clinching joint is then cut into halves so that the cross section dimensions can be measured. The geometry of the punch and the die is illustrated on Fig. 2. A typical joint geometry can be found on Fig. 1.

Three experiments have been conducted with three different configurations of process parameters. The punch diameter, the die depth and the final bottom thickness of the joint are varied. The details of each experiment and the produced joint geometry dimensions are provided in Table 1.

In this study, The tests A and B will be used to determine the material parameters. The test C is reserved for the validation. The idea is that an unique combination of material parameters should be able to give good numerical results for all experiments.

B. *Numerical model*

A finite element model is built by the help of the commercial software ABAQUS in order to simulate clinching process. Fig. 3 shows the schematic of the initial configuration. Assuming the axisymmetric nature of the problem, a two-dimensional axisymmetric model is adopted. Thus, only half of the structure is simulated. The punch and die are defined as analytical rigid.

The two metal sheets and the holder are meshed using the quadrilateral bilinear axisymmetric element with reduced integration and hourglass control (CAX4R). There are 5000 and 3277 elements in each metal sheet and in blank holder respectively. Remeshing is used to prevent excessive element distortions.

The axisymmetric boundary condition is imposed on the center line of the two metal sheets. The loading is applied by imposing the punch linear motion along the symmetry axis. The initial gap between the punch's lowest surface and the upper surface of the upper metal sheet is 3.75 mm. The maximum displacement of the punch is chosen in order to reproduce the experimental final bottom thickness of the joint. The entire loading time is fixed within 1 second.

An isotropic elasto-plastic model is used to simulate the mechanical behavior law of the two sheets. The data are derived from an uniaxial tensile test of a reference sample. The Young's modulus and Poisson's ratio are 69 GPa and 0.33 respectively. A power law hardening model is chosen to extend the stress-strain curve to a higher plastic strain :

$$R(p) = R_0 * (p + p_0)^n \qquad (1)$$

where R ang p are the true flow stress and true plastic strain respectively. Using least square fitting, the material constants are determined as R_0 = 330.5 MPa, p_0 = 0.478E-3 and n = 0.128. The fitting results are shown on Fig. 4. For simplicity purposes, the PU holder is modelled as elasto-plastic materials whose data are also derived from an uniaxial tensile test.

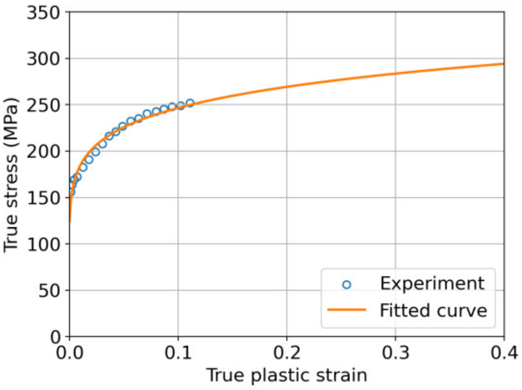


Fig. 4. Fitted stress-strain curve based on experimental data.

TABLE 1. PROCESS PARAMETERS OF THREE REFERENCE EXPERIMENTS AND COMPARISON TO SIMULATION RESULTS

Configurations		Punch diameter (mm)	Die depth (mm)	Bottom thickness (mm)	Punching force (N)	Interlock (mm)	Neck thickness (mm)	Lower sheet bottom thickness (mm)
A	Experiment	7	1.1	0.8	64 400	0.32	0.68	0.56
	Simulation				67 530	0.47	0.59	0.56
	Error				5%	47%	13%	0%
B	Experiment	6.5	1.1	1.0	50 600	0.12	0.75	0.66
	Simulation				53 680	0.31	0.65	0.67
	Error				6%	158%	13%	2%
C	Experiment	7	0.7	1.0	51 800	0.03	0.88	0.66
	Simulation				59 660	0.22	0.78	0.68
	Error				15%	633%	11%	3%

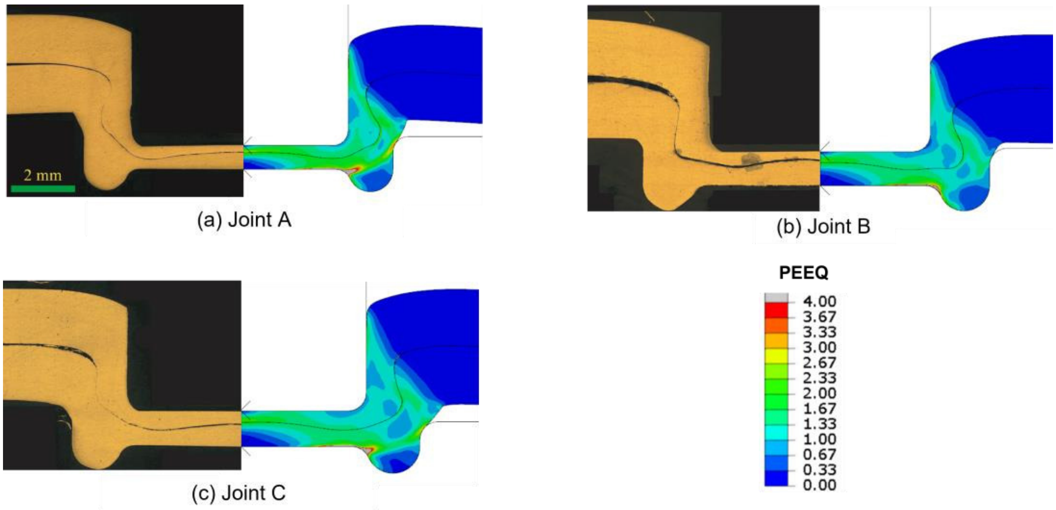


Fig. 5. Comparison between experimental and numerical cross section of clinching joints. The field color represents the equivalent cumulated plastic strain in two metal sheets.

All contacts are defined as surface-to-surface contact pairs. No penetration is explicitly allowed (hard contact). This contact constraint is enforced by the kinematic contact algorithm. The rigid body (tools) surface is always the master surface. A balanced master-slave contact pair is used for the contact between the two metal sheets.

The Coulomb friction model is assumed to model the tangential behavior. The friction coefficient between the different parts is chosen as follows: 1.0 between two metal sheets, 0.25 between sheet and punch/die and 0.1 between sheet and black holder. Furthermore, a shear stress limit of 75 MPa which is roughly half the yield stress of the aluminium alloy is applied to the friction between the punch and the metal sheet. One must note that even if the friction parameters taken in the model are within the logical range based on the literature, they are taken in a rather arbitrary way.

The explicit solver is chosen to solve this problem. Fig. 5 shows the simulated geometry of the joints and the experimental cross sections. The dimensions are detailed in Table. 1. Even if the overall shape of the simulated joints is rather good, the exact values of important features, *i.e.* interlock, neck thickness, punching force, do not answer to our satisfaction. Notes that the equivalent plastic strain in two metal sheets can reach up to 200%.

III. Neural network metamodels

One can simply try to simulate on a large number of possible values of material parameters, in order to find a good combination which is able to approximate the experimental results of all experiments. However, one FEM simulation takes about one hour on our computer. This level of computation time makes this routine intractable. Instead, a metamodel can be constructed in order to reproduce the relation between the material parameters and the results as predicted by the simulations, within a reasonable amount of simulation runs.

This section described the construction of a such metamodel on the configuration A. The same applies for the metamodel on the configuration B which is not detailed here for repetition.

A. *Data generation*

The inputs of the model should represent the material parameters we want to determine which are the stress-strain curve at high strain level and the friction behavior between contact surfaces.

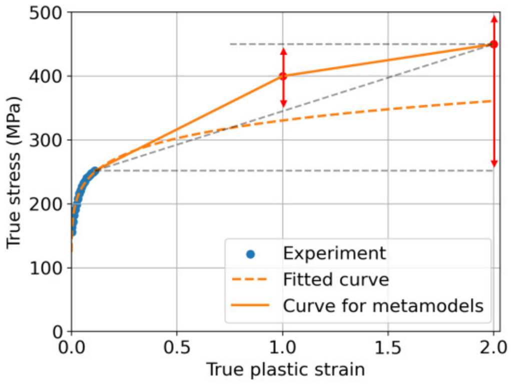


Fig. 6. Stress-strain curve used for metamodels. The arrows illustrate the range and the constraints on the two input parameters.

The following contacts are considered : between punch and upper metal sheet, between die and lower metal sheet and between two sheets. For each contact, two parameters of the Coulomb model are considered : the friction coefficient and the shear stress limit. In this study, the range of the friction coefficients is from 0 and 1. The shear stress limits can range from a fifth to the “full” yield stress limit of the aluminium alloy 88 MPa. Their value is normalized by the latest before fed into the metamodel.

It is decided to model the extrapolation of the stress-strain curve using two values: the stress at 100% plastic strain and the stress at 200% plastic strain (Fig. 6). The curve is the linear interpolation between the last experimental data point and these two points. Despite its oversimplification, this representation offers flexibility of the curve and avoids the limitation of using a specific work hardening law. However, some constraints must be applied to ensure physical common sense. First, the stress at 200% strain must be greater than the stress at 100% strain which in turn must be greater than the last experimental measured stress. Second, as a stress-strain

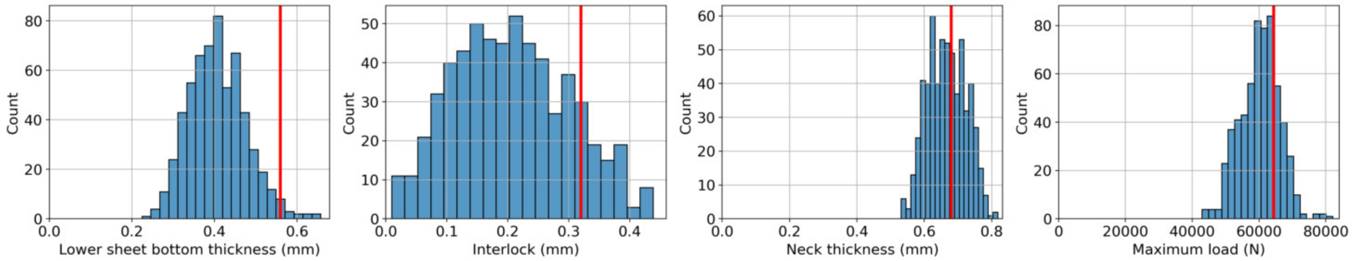


Fig. 7. Histograms of outputs in the dataset. The red lines represent the experimental results.

curve must be convex, the stress at 100% strain must be greater than the linear interpolation between the stress at 200% strain and the last experimental measured stress. In this study, the stress at 200% strain is considered up to 500 MPa.

Within the mentioned ranges, input samples are generated pseudo-randomly using Halton sequence, which is one of Quasi-Monte Carlo methods, in order to maximize the coverage of the parameter space. This method offers an advantage over other methods like Latin Hypercube sampling that one can incrementally add more points to the data set without forming clusters.

Once input samples are generated, they are fitted into the FEM simulations which return outputs. Four outputs are considered in this study : maximum punching force, interlock, neck thickness and lower sheet bottom thickness. The computation routine and the extraction of interested features are executed automatically using Python scripts and the Abaqus Scripting Interface. Fig. 7 shows the histogram of the four outputs. The experimental results are within the covered domains.

In order to speed up learning, the simulation outputs are scaled using their minimum and maximum values such that all feature values are in the range (0, 1). The last 2 input (stresses) features are also scaled by using two limits 250 MPa and 500 MPa. The first 6 input features are naturally within this range.

B. *Neural network architecture*

A fully connected neural network with only one hidden layer is chosen for the metamodel. Its architecture is illustrated in Fig. 8. There are 8 (hidden) nodes in the hidden layer (and one bias node). This architecture choice is later explained in Section III.C.

Each hidden node represents a neuron of the network. It is connected to every node in the previous layer. Each connection is assigned with a weight which is a trainable parameter of the model. A hidden node calculates the weighted sum of the values from the previous nodes and passes it to the sigmoid activation function which outputs values in between 0 and 1. No activation function is used for the output nodes.

C. *Training*

The first hundred samples in the Halton sequence are selected for the validation set which is used to evaluate the

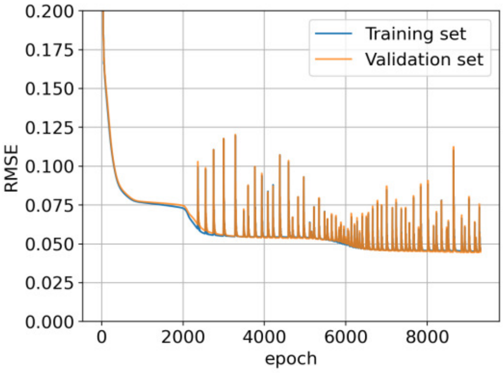


Fig. 9. Root mean squared errors (RMSE) on the training set and the validation set against the number of epochs during the training.

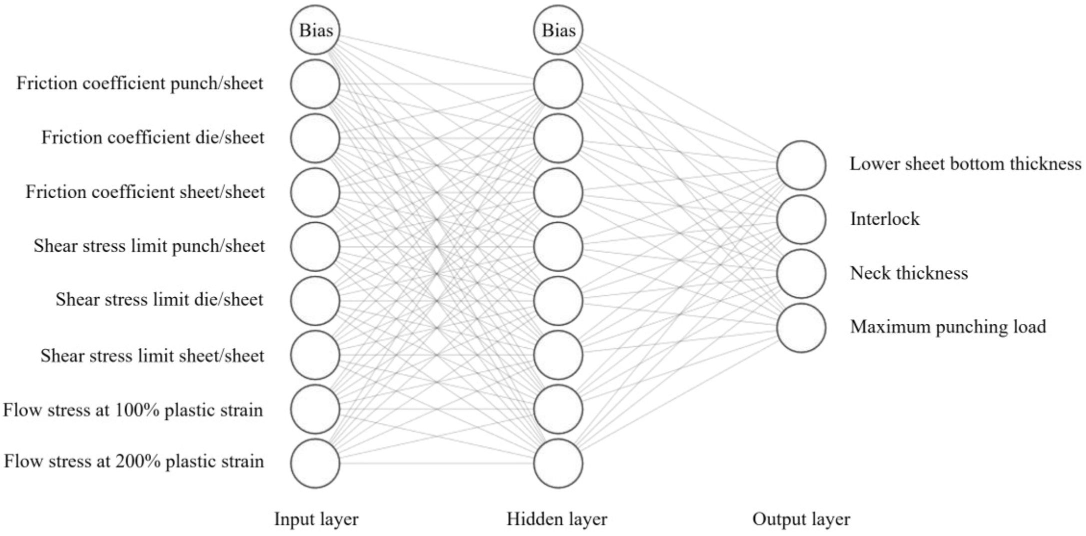

Bias
Bias
Friction coefficient punch/sheet
Friction coefficient die/sheet
Friction coefficient sheet/sheet
Shear stress limit punch/sheet
Shear stress limit die/sheet
Shear stress limit sheet/sheet
Flow stress at 100% plastic strain
Flow stress at 200% plastic strain
Lower sheet bottom thickness
Interlock
Neck thickness
Maximum punching load
Input layer
Hidden layer
Output layer

Fig. 8. Neural network architecture used for metamodels

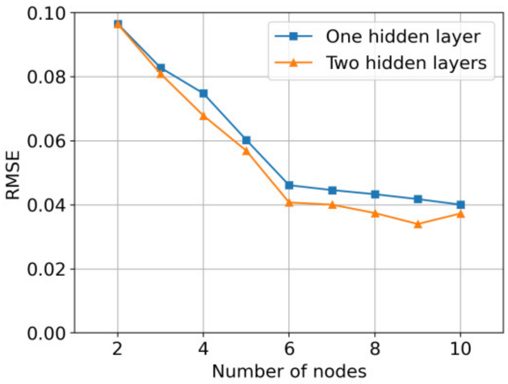


Fig. 10. Root mean squared errors on the validation set as a function of the number of nodes in each hidden layer.

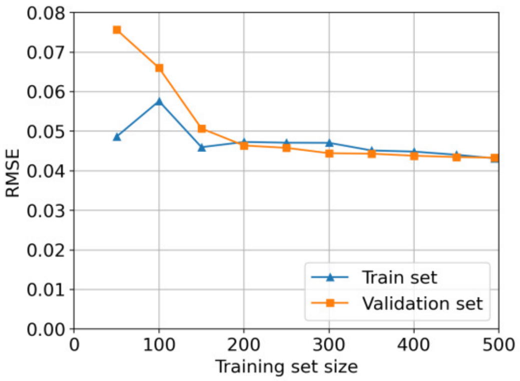


Fig. 11. Root mean squared errors on the validation set as a function of the training set size.

model's performance on unseen data. The next 400 samples are used as the training set.

For the training, the loss function is the mean squared error (MSE) averaged over 4 outputs. Backpropagation learning is performed with the Adam algorithm that involves gradient descent with momentum and adaptive learning rate. An epoch is comprised of only one batch which contains the full training set. In other words, the training is not stochastic. Early stopping regularization is used to stop the training before overfitting with the validation loss as the monitored metric. The model weights are initialized randomly from a normal distribution with mean 0 and variance 1. In order to get a good local minimum, different seeds are tested for weight initialization. Then the seed giving the lowest validation loss is chosen for the final results. The training history is shown in Fig. 9. The oscillation near the convergence is a typical feature of the Adam algorithm related to the adaptive momentum.

The necessary code was implemented in Python using Keras library with TensorFlow backend.

D. *Hyperparameters tuning*

The number of hidden layers and the number of nodes in each hidden layers are two important hyperparameters to be tuned for each problem. An oversimple architecture neural network cannot predict with good accuracy. A complex model is difficult to train and can suffer from overfitting. One must find a compromise between these two problems.

Fig. 10 plots the evolution of the error (loss) evaluated on the validation set in function of the number of hidden nodes for two cases: one hidden layer and two hidden layers. For simplicity, the numbers of nodes in two hidden layers are constrained to be equal. It is shown that a neural network with more than 8 hidden nodes does not offer a higher prediction accuracy. In addition, adding another hidden layer also does not significantly improve the metamodel performance.

E. *Effect of the training set size*

Fig. 11 shows the metamodel accuracy in function of the size of the training set. It can be seen that adding more data over 300 samples does not help to improve the prediction accuracy remarkably. Moreover, it is a good indication that the error on the validation set is close to the error on the training set.

IV. Material Parameter Identification

Two metamodels have been constructed for two process configurations A and B in order to predict simulation results from 8 material parameters. The material parameters therefore can be optimized in order to minimize the difference between the simulation results and the experimental results. The loss is chosen to be the average of the mean squared error of the four outputs of the two metamodels. The optimization is performed using a gradient based algorithm. The gradient of the loss is calculated by automatic differentiation thanks to GradientTape API provided by TensorFlow.

TABLE 2. Optimized Material Parameters

Solutions	**Friction coefficient punch/sheet**	**Friction coefficient die/sheet**	**Friction coefficient sheet/sheet**	**Shear stress limit punch/sheet (normalized)**	**Shear stress limit die/sheet (normalized)**	**Shear stress limit sheet/sheet (normalized)**	**Flow stress at 100% strain (MPa)**	**Flow stress at 200% strain (MPa)**
Global optimum	1.0000	1.0000	1.0000	0.2000	1.0000	0.3575	415.8707	391.7660
1	0.7410	0.5162	0.4858	0.3386	1.0000	0.3611	394.5972	372.5730
2	0.5881	0.4398	0.7545	0.3160	1.0000	0.2994	399.2369	371.1448
3	0.5959	0.7602	0.5231	0.3577	1.0000	0.2826	441.4521	351.3855
4	0.7469	0.4927	0.4655	0.3442	1.0000	0.3248	415.6395	360.2948
5	0.5983	0.5205	1.0000	0.2138	1.0000	0.3340	440.7345	382.0971
6	0.7800	0.5051	0.8045	0.2543	1.0000	0.3671	397.4937	386.2823
7	0.8225	0.6115	0.8042	0.3382	1.0000	0.2777	416.7320	361.1949
8	0.2781	0.6865	0.4518	0.3875	1.0000	0.2846	404.0742	355.6252
Mean	0.64	0.57	0.66	0.32	1.0	0.32	414	368

TABLE 3. SIMULATION RESULTS WITH OPTIMIZED MATERIAL PARAMETERS

Configurations		Punching force (N)	Interlock (mm)	Neck thickness (mm)	Lower sheet bottom thickness (mm)
A	Experiment	64 400	0.32	0.68	0.56
	Simulation	75 960	0.19	0.77	0.52
	Error	18%	40%	13%	7%
B	Experiment	50 600	0.12	0.75	0.66
	Simulation	49 400	0.12	0.75	0.60
	Error	2%	< 8%	< 1%	9%
C	Experiment	51 800	0.03	0.88	0.66
	Simulation	57 050	0.01	0.90	0.66
	Error	10%	67%	2%	< 2%

The Table 2 shows some material parameter combinations as suggested by the optimization which give approximately the same level of deviation from the experimental results as predicted by two metamodels (less than 10% than the variation range, see Fig. 7). Each solution results from a different initialization of the input variables at the beginning of the optimization process.

It is worth to mention that the global optimum in this problem stays at the border of the searched domain. We opted for slightly lower accuracy for more physical meanings. However, the shear stress limit at the contact between die and sheet still remains at its maximum which is the yield stress limit of the aluminium sheet. There may be a physical meaning behind this fact, but it is quite surprising that the contact between punch and sheet (basically the same materials) does not exhibit the same behavior. All other material parameters seem to fluctuate around and close to some means. However, a combination of these means does not give results comparable with the experiments.

The simulation results using the first solution are shown in Table 2. Even if the error in the interlock on the configuration A is still non negligible, the error levels are significantly smaller and more spread out over all outputs compared to the results obtained with non-optimized material parameters in Table 1. The observed high error indicates that the optimization algorithm has a hard time to satisfy two experimental results simultaneously, despite a rather high number (8) of variables.

Interestingly, even if no metamodel is constructed for the configuration C and their experimental results are not used to tune material parameters, we get a good comparison. The simulated joint geometry is shown in Fig. 12. The vanishing of the interlock is well simulated. This shows the generalizability of the determined material parameters on others process configurations.

V. CONCLUSION AND PERSPECTIVES

In this paper, the authors explore the applicability of machine learning based metamodels to tackle the high computation time problem encountered during the material parameter tuning task in the context of clinching process simulation. It is shown that simple neural network metamodels are able to learn with high precision the relation between material parameters and joint geometry as predicted by FEM numerical simulations. They then are used to replace simulations inside the optimization loop.

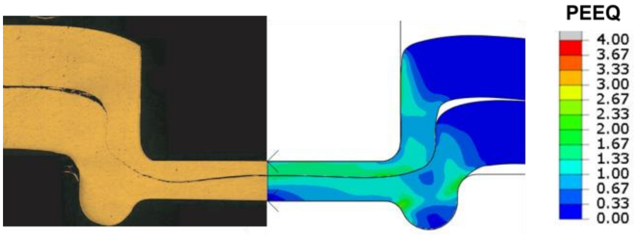


Fig. 12. Simulation results on the configuration C with optimized material parameters.

The obtained results are promising. The optimized material parameters help to reduce significantly the error between simulated results and experimental results, not only in the configurations where metamodels are constructed but also on a new simulation. This methodology shows its value in the cases where material parameters are not available or difficult to measure. On top of that, it may help to understand more about physical phenomena behind the numbers.

In order to increase the reliability of the determined material parameters, it is worth to conduct more experiments. The applicability of others machine learning algorithms can also be assessed.

ACKNOWLEDGMENT

This research is funded by Thu Dau Mot University, Binh Duong Province, Vietnam under grant number NNC. 21.2.012.

REFERENCES

[1] S. Gao, B. Lothar, “Mechanism of mechanical press”, Int J Mach Tools Manufact 34, 1994, pp 641–657.

[2] N. Nong, O. Keju, Y. Zu, Q. Zhiyuan, T. Changcheng, L. Feipeng, “Research on press joining technology for automotive metallic sheets”, J Mater Proc Techn, vol. 137, 2003, pp. 159-163.

[3] F. Lambiase, A. Di Ilio, “Optimization of the clinching tools by means of integrated FE modeling and artificial intelligence techniques”, Procedia CIRP, vol. 12, 2013, pp. 163-168.

[4] E. Roux, P.O. Bouchard, “Kriging metamodel global optimization of clinching joining processes accounting for ductile damage”, Journal of Materials Processing Technology, vol. 213, 2013, pp. 1038-1047.

[5] M. Eshtayeh, M. Hrairi, “Multi objective optimization of clinching joints quality using Grey-based Taguchi method”, Int J Adv Manuf Technol, vol. 87, 2016, pp. 233–249.

Efficient Backoff Priority-based Medium Access Control Mechanism for IoT Sensor Networks

Thu-Hang T. Nguyen
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
hangntt@ptit.edu.vn

Hai-Chau Le
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
chaulh@ptit.edu.vn

Trong-Minh Hoang
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
minhht@ptit.edu.vn

Trinh Nguyen Chien
Posts and Telecommunications Institute of Technology
Hanoi, Vietnam
trinhnc@ptit.edu.vn

***Abstract*—Recent rapid penetration of Internet of Things (IoT) in various fields such as smart homes, healthcare, and industrial applications has raised new challenges on the QoS requirements including data prioritization and energy saving. In IoT networks, data is heterogeneous and varies in a wide range of categories and urgency. More critical data must be served more quickly and reliably than regular data. In order to deal with crucial issues effectively and improve the performance of wireless sensor networks in IoT, we propose an efficient Backoff Priority-based Medium Access Control (BoP-MAC) scheme that supports multiple priority data and exploits the use of backoff mechanism. In our proposed solution, data priority is utilized to properly resize the backoff window at the MAC layer to ensure that high-priority data are transferred earlier and more reliably. Numerical simulations are used on OMNeT++ to verify the efficiency of our proposed BoP-MAC protocol in comparison with that of a notably conventional MAC protocol called Timeout Multi-priority-based MAC (TMPQ-MAC) protocol. The attained experimental results demonstrate that our developed BoP-MAC protocol outperforms the comparable conventional one and becomes more efficient for large-scale wireless sensor networks. It can effectively cope with various data priorities and enhance significantly the overall performance, in terms of latency, energy consumption, and packet success ratio, of the network.**

***Index Terms*—Backoff window, Internet of things, MAC protocol, Wireless sensor networks.**

I. Introduction

Nowadays, Internet of Things (IoT) has been emerging as one of the key digital transformation technologies and predicted to influence the global economy with an estimated $4 trillion to $11 trillion and 75.4 billion connected devices by 2025 [1]. IoT has been one of hot research topics in a wide variety of academic and industrial disciplines [2-6]. Many researches have been introduced in order to cope with IoT challenges and issues including QoS flexibility [2, 3], energy efficiency [4-6] and particularly various priority data provision [7-9]. In general, conventional works consider separately or simultaneously the requirements of data priority and energy usage, and their methods can be divided into three main categories that are MAC layer, routing and queue priority in network layer, or application layer [6]. However, each method category has its own limitations. The application layer and priority-queue or routing approaches could theoretically prioritize a wide variety of traffic and data types, but they have a high complexity that is not suitable for the fact that, in IoT, sensors normally have restricted memory and energy [8, 9]. Conversely, the MAC layer approach capable of reducing energy consumption while ensuring a sufficient communication quality is more widely used [10-15]. The reason is that MAC protocol directly controls transceivers, which are the most energy consumed elements. Hence, the development of energy-efficient, QoS-guaranteed and data priority-based MAC protocols is essential.

Up-to-date MAC protocols developed, to the best of our knowledge, hardly fulfill the critical issues of modern IoTs, particularly for provisioning concurrently multiple-priority-level data services [11-15]. In the work of [11], a MAC protocol has been introduced to deal with two priority levels (high or low) of the data packet and high priority data packet is preferentially handled. However, using a fixed timeout timer causes the latency to increase significantly. To improve on this, an advanced MAC protocol which is Timeout Multi-Priority-Based MAC (denoted as TMPQ-MAC) [12] is a receiver-initiated protocol that is able to provide a synchronous way and take into account four separate packet priority levels, where the timeout timer stops early on receiving the highest priority data transmission request in order to decrease the end-to-end delay and extend the lifetime of network. Moreover, in [13], the authors assigned the packet priority by considering the residual energy, rather than using the data emergency and the work did not guarantee the small average latency of packets. In addition, [14] only considered very few levels of data priority. On the other hand, different scheduling algorithms were developed to enhance the media access control protocol performance, specially by dynamically adapting the size of contention windows [14, 15]. It is showed that adaptively controlling the contention window size plays a major role in improving the network efficiency. Indeed, in order to enhance the network performance, the authors in [15] also introduced an adaptive contention backwindow scheme by calibrating the waiting time but unfortunately, it was only applied for un-narrow band and not restricted powered WLAN. On the other hand, several MAC protocols that consider the backoff mechanism have been introduced [16, 17], however, they did not consider the data priority and have not been designed for low-speed IoT environments.

In this paper, to overcome the shortcomings of the above studies, we target an efficient media access control protocol by taking the advantages of a data priority-based collision avoidance approach. Our developed MAC scheme is able to exploit the use of duty cycle and active/sleep periods by applying RTS/CTS handshaking mechanism like traditional MAC protocols while provisioning multi-priority data services and controlling the backoff contention window of data transmission considering the data priority for IoT wireless sensor networks and so, called backoff priority-based

MAC protocol (BoP-MAC). Numerical simulations using OMNeT++ are employed to evaluate the performance of the proposed BoP-MAC solution. We also compare our developed solution to the notable traditional WSN MAC protocol, that is TMPQ MAC protocol, under various network conditions. The obtained results prove that our proposed BoP-MAC solution remarkably gains better performance than the TMPQ-MAC. It offers significantly lower average delay, consumes dramatically less energy while guaranteeing a sufficiently greater packet success rate, especially with large scale networks.

The rest of the paper is organized as follows. The proposed BoP-MAC mechanism is presented in Section II. Section III presents performance evaluation of BoP-MAC and TMPQ-MAC protocols based on numerical experiments, and our conclusion is given in Section IV.

II. Proposed Backoff Priority-based Media Access Control Mechanism

A. Media Access Control Mechanism

Fig. 1 describes the main principles of SMAC [18] and TMPQ-MAC [12] protocols which have two schemes that can solve the delay reduction and priority handling at the MAC layer. SMAC works synchronously and uses RTS/CTS/DATA/ACK (Request-To-Send/Clear-To-Send) where the contention window is applied to RTSs. The RTS/CTS mechanism helps to avoid hidden terminals and also helps to reduce the level of conflicts due to the small size of RTS/CTS compared to DATA packets. In this mechanism, the contention window is determined by a random value in the range 0-CW. SMAC does not handle data prioritization, all data sent from sensor nodes are treated equally. TMPQ-MAC is improved from SMAC and MPQ by using the same operating mechanism as SMAC but adding four priority levels (p_4 , p_3 , p_2 , p_1) for data like MPQ and using this priority to prioritize data. Sending RTS in TMPQ-MAC follows p-persistent CSMA-CA principle, in addition to distinguish sending priority of data TMPQ-MAC treats RTS receiving with highest priority (p_4) as SMAC while with RTS of lower priority (p_3 , p_2 , p_1) will be sorted in priority order at the end of fixed window T_w (equivalent to CW of SMAC). This ensures that priority data is received in priority order and the highest priority data is sent at the earliest. However, using the p-persistent CSMA-CA mechanism in combination with Tw increases the average packet sending delay and also causes RTS loss by allowing many RTSs to be sent before only at most one RTS is received in a cycle.

B. Backoff Behavior

The backoff behavior is described in [19], with backoff time counter is decremented as long as the sender senses the idle state of the channel, stopped when the sender detects transmission on it (channel busy), and revived as it senses the channel and finds the idle state again for greater than a distributed interframe space (DIFS). The sender transfers a frame as the backoff time gets to zero. At every transmission period, the backoff time is assigned uniformly in the range

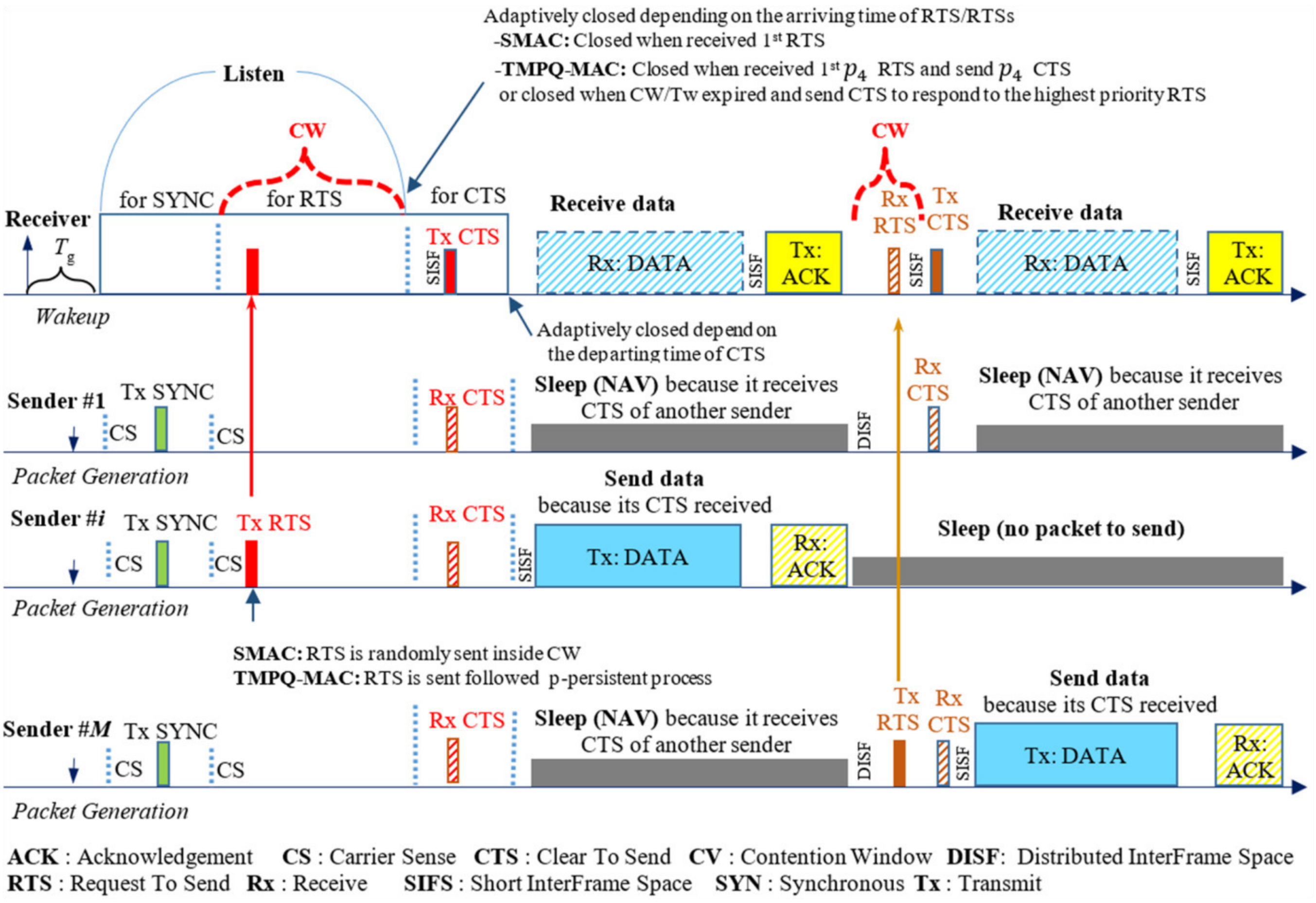

Adaptively closed depending on the arriving time of RTS/RTSs
-SMAC: Closed when received 1st RTS
-TMPQ-MAC: Closed when received 1st p_4 RTS and send p_4 CTS
or closed when CW/Tw expired and send CTS to respond to the highest priority RTS
Listen
CW
for SYNC
for RTS
for CTS
Receiver
T_g
Wakeup
Tx CTS
SISF
Receive data
Rx: DATA
SISF
Tx: ACK
CW
Rx RTS
Tx CTS
SISF
Receive data
Rx: DATA
SISF
Tx: ACK
Adaptively closed depend on the departing time of CTS
Sender #1
Tx SYNC
CS
CS
Rx CTS
Sleep (NAV) because it receives CTS of another sender
DISF
Rx CTS
Sleep (NAV) because it receives CTS of another sender
Packet Generation
Sender #i
Tx SYNC
Tx RTS
CS
CS
Rx CTS
SISF
Send data because its CTS received
Tx: DATA
Rx: ACK
Sleep (no packet to send)
Packet Generation
SMAC: RTS is randomly sent inside CW
TMPQ-MAC: RTS is sent followed p-persistent process
Sender #M
Tx SYNC
CS
CS
Rx CTS
Sleep (NAV) because it receives CTS of another sender
DISF
Tx RTS
Rx CTS
SISF
Send data because its CTS received
Tx: DATA
Rx: ACK
Packet Generation
ACK : Acknowledgement CS : Carrier Sense CTS : Clear To Send CV : Contention Window DISF: Distributed InterFrame Space
RTS : Request To Send Rx : Receive SIFS : Short InterFrame Space SYN : Synchronous Tx : Transmit

Fig. 1. Principles of S-MAC and TMPQ-MAC protocols

between 0 and (w-1). At the first attempt of transmission, w is fixed at the minimal value of backoff window. When the transmission is failed, the value of w is twofold until it reaches the pre-determined maximum value.

C. Proposed Approach

Our proposed BoP-MAC protocol employs a SMAC duty cycle and duration of active and sleep periods that are fixed, depending on the application requirements. It also inherits the data prioritization approach introduced in the work of [12] (TMPQ-MAC). We divide levels of priority into four types of data that are urgent, most important, important, and normal consecutively. The contention window is adaptively split into separate sections based on the data priority levels and the number of consecutive collisions.

Fig. 2 illustrates the contention window sizes used in our proposed BoP-MAC mechanism. An RTS is transmitted from a sender with its collision window adaptable to the data priority level and the busy condition of the channel. If a sender has data, it senses the channel to determine whether the medium is idle or not and randomly transfers its RTS frame within its priority window. In case of a busy medium found, the sender will double its priority contention window. Here, to prevent the collision of the same priority level RTSs from different senders, RTS sending will be started randomly within its contention window duration. Consequently, our MAC protocol is able to lessen the waiting time of receiving the not-selected-senders' CTSs, which can compare to T_w of TMPQ-MAC protocol or CW in SMAC, the earlier sending CTS (like Rx-Beacon in TMPQ-MAC) mechanism also enables other senders to ward off sending frames, and reduce the energy consumption by sleeping in the time of NAV.

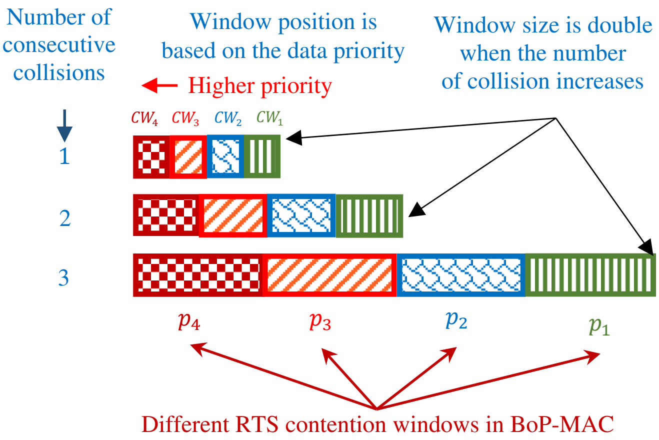

Number of consecutive collisions
Window position is based on the data priority
Window size is double when the number of collision increases
Higher priority
CW_4
CW_3
CW_2
CW_1
1
2
3
p_4
p_3
p_2
p_1
Different RTS contention windows in BoP-MAC

Fig. 2. Priority contention window control of BoP-MAC

III. Performance Evaluation and Discussions

In this section, we have simulated and evaluated the performance of IoT sensor networks that utilize our developed BoP-MAC protocol exploiting the adaptive contention window and backoff mechanism. Numerical simulations are conducted on Castalia 3.3 [20] with CC2420 transceivers [21]. The key experimental parameters and values applied are summarized in Table I. To assess the overall network performance, the three performance indicators taken into account is listed as follows:

- *Average packet delay:* is defined as the duration for senders from the data generated time to the time their data arrive the sink.
- *Average energy consumption:* is determined as the mean consumed energy per bit.
- *Average packet success rate:* is calculated as a ratio of the entire number of different packets (not count for duplicate packets) that are received to the sent packet total.

TABLE I. Main Experimental Parameters

Parameter	Value
Size of sensor network area	10m x 10m
Number of sender nodes	1 to 10
Link bandwidth	250kb/s
Radio	CC2420
Size of SYN	6 bytes
Sizes of RTS/Tx-Beacon	13/14 bytes
Sizes of CTS/Rx-Beacon	13 bytes
MAC overhead size	11 bytes
Listen interval	17ms
RTS/Tx Beacon retransmission number	10
Packet arrival rate	1 packet/s
Application header length	5 bytes
Sensor Startup Randomization	1ms
DATA packet size	28 bytes
ACK packet size	11 bytes
CCA Check Delay	0.128ms
Physical frame overhead	6 bytes
T_g	6.7ms
CW for SMAC and T_w for TMPQ-MAC	10ms
CW_{min} - CW_{max} for BoP-MAC	4-16

1) Average end-to-end delay of different packet priorities

Fig. 3 demonstrates the average end-to-end latency of various packet priorities of TMPQ-MAC and BoP-MAC. The end-to-end packet delay in the network with TMPQ-MAC is greater than in that of BoP-MAC for all four packet priority levels.

A T_w timer used at the receiver for TMPQ-MAC to collect TxBeacons from all senders. If p_4 TxBeacon is received by the receiver, the receiver sends back RxBeacon to p_4 sender, other senders knows and wait until the next frame (for NAV duration shown in the RxBeacon). If receiver does not receive p_4 TxBeacon but other lower priority ones, it will have to wait till the end of T_w timer, and then select the greatest and earliest priority TxBeacon to determine. Therefore, the average delay of p_4 packets is lowest ranging from 13.4 to 69.7ms while the average delay of p_3, p_2, and p_1 packet is higher, in the ranged from 30.7, 30.7 and 30.6 to 152.8, 153.5, and 138.0ms, accordingly. The results seem to show different trend in the delay of p_3, p_2, and p_1 packet because with the limited number of RTS retransmission, RTS of higher priority level will reach the receiver more with higher number of retransmission compared to the RTS of lower priority level.

BoP-MAC uses the scheme of accepting first RTS, the contention window is close and receiver immediately send CTS, that bring about a less packet delay compared to TMPQ-MAV. Furthermore, with the adaptive window size based on priority, the average delay of p_4 packets is lowest ranging from 12.3 to 16.9ms while the average delay of

p_3, p_2, and p_1 packet is ranged from 12.9, 13.8 and 14.8 to 20.8, 23.7 and 25.8ms, accordingly.

Actually, the graphs present a gradual growth in the average end-to-end packet delay of each priority level as the number of sending nodes increases because as the number of senders increases, the probability of contention is higher.

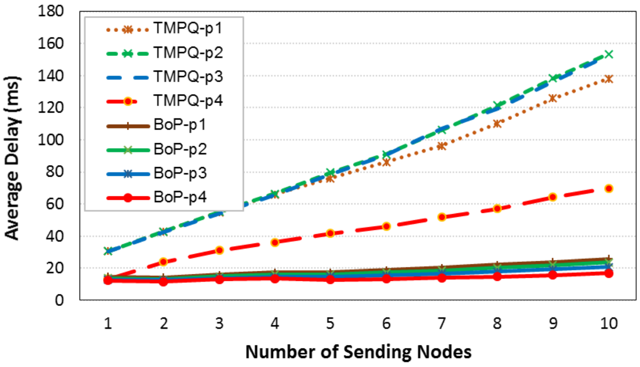


Fig. 3. Average multi-priority packet delay comparison between TMPQ-MAC and BoP-MAC

2) *Average end-to-end delay*

Fig. 4 describes the average end-to-end delay of all priority packets obtained by the proposed BoP-MAC protocol in comparison with that of the comparable TMPQ-MAC protocol. It is showed that the average delay attained in the proposed BoP-MAC-based network is greatly cut down compared to that of the TMPQ-MAC one. The reason for the cut down is that BoP-MAC soon adopted RTS scheme for all RTSs, not just p_4 one. The average delay of the entire packet using BoP-MAC when the number of nodes increases from 1 to 10 steadily increases from 13.5ms to 21.8ms while the corresponding delay with TMPQ-MAC increases rapidly from 26.3ms to 114.4ms.

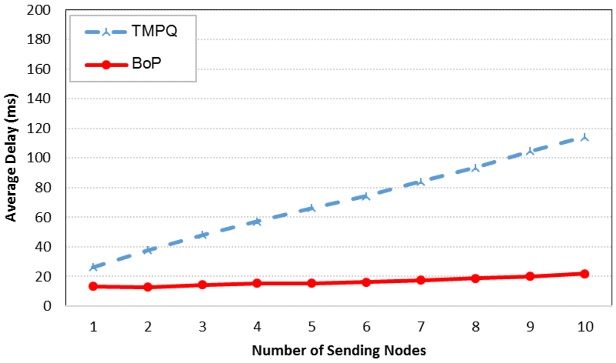


Fig. 4. Average packet delays of TMPQ-MAC and BoP-MAC

3) *Average energy consumption per successful delivered bit*

The simulated results in Fig. 5 illustrate that the average consumed energy (calculated in *mJ per bit*) of the proposed BoP-MAC-based network is much less than that of the network with TMPQ-MAC, and as the number of nodes increases the difference in energy consumption turn into more observable. Specifically, the average power consumption with BoP-MAC when the number of simultaneous sending nodes increases from 1 to 10 is 0.23*mJ/bit* to 0.25*mJ/bit* while with TMPQ-MAC the corresponding power consumption is 0.22 *mJ/bit* to 0.59*mJ/bit*.

The BoP-MAC makes the use of adaptive contention window, so the earliest RTS sender could transmit its packets while other nodes will sleep until the next period. Meanwhile, in TMPQ-MAC, using p-persistent mechanism, all senders have to seed and wait until they can send their TxBeacons. In that case, if p_4 TxBeacon reaches the receivers, the TxBeacon contention window is closed and p_4 RxBeacon is sent from the receiver to confirm the p_4 sender, if not, all senders must stay awake until T_w is expired. Hence, the average senders' wake-up time is much more than that of BoP-MAC, this explains the expanded consuming energy in TMPQ-MAC for wakeup state.

Moreover, as the sender number is enlarged, the competition level becomes greater and more energy is consumed, too. In this circumstance, BoP-MAC offers many advantages over TMPQ-MAC with an adaptive closing window when receiving the earliest incoming RTS, the node number becomes greater, the total congestion window time of each node and the power consumption become less than that of TMPQ-MAC.

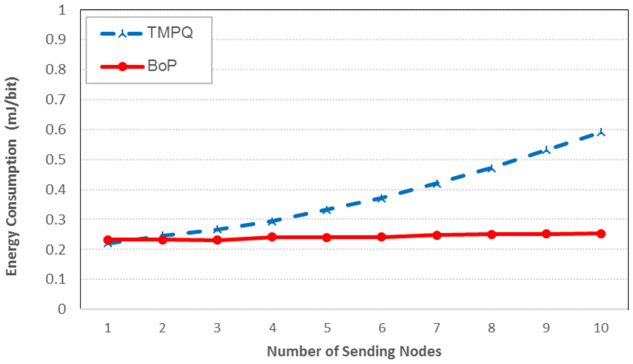
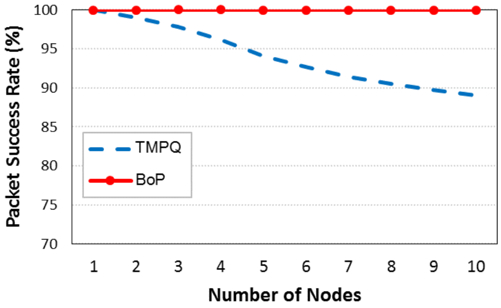
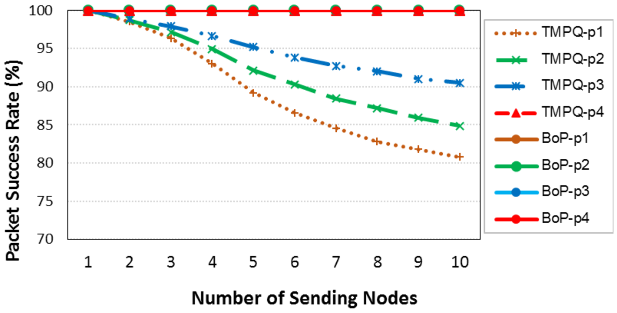


Fig. 5. Energy consumption comparison

4) *Average packet success rate*



a) Average packet success rate



b) Different priotity packet success rate

Fig. 6. Packet success rate analysis

Fig. 6 describes the average packet success rate (PSR) comparison between TMPQ-MAC and BoP-MAC protocols. Fig. 6a describes the total average for all packet priority levels. We can see that TMPQ-MAC has 88% PSR when the number of senders reaches 10 while BoP-MAC gets 100% PSR. Fig. 6b demonstrates the separate priority packet PSR of the two MAC protocols. For BoP-MAC, the PSR is 100% with all four priority level packets, but for TMPQ-MAC the PSRs are nearly 100% for p_4 packets, while for p_3, p_2, p_1 are lower with the corresponding value of 90.5, 84.8, 80.8 % PSR when the number of senders reaches 10.

This result also shows that TMPQ-MAC has a higher packet loss rate despite having the same retransmission number as BoP-MAC. And if the transmission rate has not reached 100%, the number of retransmissions needs to be increased to get better PSR, and so the packet transmission delay will also increase. If the number of retransmissions is restricted, the higher priority packets will be sent more than the lower priority packets, which will reduce the delay between non-highest priority in TMPQ-MAC because the end-to-end delay only counts for successfully transmitted packets. This further explains the results in Fig. 3, the simulation delay of TMPQ-MAC with the three lower priority categories is almost not much different.

IV. Conclusions

In this paper, we have investigated media access control mechanisms for modern IoT wireless sensor networks that are able to provision multi-priority data and effectively deal with critical challenges on the QoS requirement. To enhance the overall performance, in terms of end-to-end delay, power consumption and packet loss rate, of multi-event IoT sensor networks, we have successfully proposed a backoff priority-based media access control scheme that exploits the duty cycle of traditional MAC protocols like SMAC and fully take the advantages of active/sleep durations with the RTS/CTS handshaking method while being capable of serving multi-priority data effectively and adjusting the backoff contention window for the data transmission with the order of data priority. We have also simulated and evaluated the performance of the proposed BoP-MAC solution in comparison with TMPQ-MAC protocol, one of the notable conventional WSN MAC protocols under various network conditions. Numerical simulations demonstrate that, under the same network and traffic conditions, our proposed BoP-MAC solution offers significantly higher performance than the TMPQ-MAC. The developed BoP-MAC scheme is able to lower the average delay remarkably and consume dramatically less energy while ensuring a greater network packet success rate.

Acknowledgment

This research was funded by Qualcomm Technologies, Inc under Research Agreement Number POS-458996 (SOW Number POS-459341).

References

[1] J. Manyika, M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin, and D. Aharon, “Unlocking the Potential of the Internet of Things,” McKinsey & Company, 2015. Accessed on: September 8, 2022. [Online]. Available: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

[2] O. Ajayi, A. Bagula, I. Chukwubueze and H. Maluleke, "Priority Based Traffic Pre-emption System for Medical Emergency Vehicles in Smart Cities," *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1-7, doi: 10.1109/ISCC50000.2020.9219628.

[3] K. F. Firdaus, S. A. Wibowo and K. Anwar, "Multiple Access Technique for IoT Networks Serving Prioritized Emergency Applications," *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring),* 2019, pp. 1-5, doi: 10.1109/VTCSpring.2019.8746519.

[4] L. Chen, X. Dong, X. Kuang, B. Chen and D. Hong, "Towards Ubiquitous Power Distribution Communication: Multi-service Access and QoS Guarantees for IoT Applications in Smart Grid," *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2019, pp. 894-898, doi: 10.1109/ISGT-Asia.2019.8881751.

[5] F. Shabnam, T. -U. Islam, S. Saha and H. Ishraque, "IoT Based Smart Home Automation and Demand Based Optimum Energy Harvesting and Management Technique," *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 1800-1803, doi: 10.1109/TENSYMP50017.2020.9230940.

[6] A. Kumar, M. Zhao, K. Wong, Y. L. Guan and P. H. J. Chong, "A Comprehensive Study of IoT and WSN MAC Protocols: Research Issues, Challenges and Opportunities," *in IEEE Access*, vol. 6, pp. 76228-76262, 2018, doi: 10.1109/ACCESS.2018.2883391.

[7] N. T. T. Hang, N. C. Trinh and N. T. Ban, "Energy aware event driven routing protocol and dynamic delivering scheme for multievent wireless sensor network," *2018 2nd International Conference on Recent Advances in Signal Processing, Telecommunications & Computing (SigTelCom)*, 2018, pp. 224-229, doi: 10.1109/SIGTELCOM.2018.8325795.

[8] S. K. Swain and P. K. Nanda, "Adaptive Queue Management and Traffic Class Priority Based Fairness Rate Control in Wireless Sensor Networks," in IEEE Access, vol. 9, pp. 112607-112623, 2021, doi: 10.1109/ACCESS.2021.3102033.

[9] M. B. Attia, K. -K. Nguyen and M. Cheriet, "Dynamic QoE/QoS-Aware Queuing for Heterogeneous Traffic in Smart Home," in *IEEE Access*, vol. 7, pp. 58990-59001, 2019, doi: 10.1109/ACCESS.2019.2914658.

[10] T. C. Nguyen, H. -C. Le, S. Sarang, M. Drieberg and T. -H. T. Nguyen, "Priority and Traffic-Aware Contention-Based Medium Access Control Scheme for Multievent Wireless Sensor Networks," in *IEEE Access*, vol. 10, pp. 87361-87373, 2022, doi: 10.1109/ACCESS.2022.3199385.

[11] Kim S.C., Jeon J.H., Park H.J., "QoS Aware Energy-Efficient (QAEE) MAC Protocol for Energy Harvesting Wireless Sensor Networks," *Convergence and Hybrid Information Technology (ICHIT 2012), Springer Lecture Notes in Computer Science*, vol 7425, 2012. https://doi.org/10.1007/978-3-642-32645-5_6.

[12] A. N. Sakib, M. Drieberg, S. Sarang, A. A. Aziz, N. T. T. Hang, and G. M. Stojanović, “Energy-Aware QoS MAC Protocol Based on Prioritized-Data and Multi-Hop Routing for Wireless Sensor Networks,” *Sensors*, vol. 22, no. 7, p. 2598, 2022, doi: 10.3390/s22072598.

[13] M. Rasheed, I. U. Din, M. Adnan, A. Tariq, S. Malik and I. Syed, "ECM-MAC: An Efficient Collision Mitigation Strategy in Contention Based MAC Protocol," in *IEEE Access*, vol. 9, pp. 62880-62889, 2021, doi: 10.1109/ACCESS.2021.3074812.

[14] Q. Huamei, F. Linlin, Y. Zhengyi, Y. Weiwei, W. Jia, “An energy-efficient MAC protocol based on receiver initiation and multi-priority backoff for wireless sensor networks,” in *IET Communications*, vol. 15, no. 20, pp. 2503-2512, 2021, doi: 10.1049/cmu2.12283.

[15] I. Syed, S. Shin, B. Roh and M. Adnan, "Performance Improvement of QoS-Enabled WLANs Using Adaptive Contention Window Backoff Algorithm," in *IEEE Systems Journal*, vol. 12, no. 4, pp. 3260-3270, Dec. 2018, doi: 10.1109/JSYST.2017.2694859

[16] Trong-Minh Hoang, Van-Truong Nguyen, Nhu-Giap Nguyen and Tuan-Nguyen Lang, “Analysing the performance of unslotted sensor networks based on the IEEE 802.15.4 employed EIED algorithm,” *2017 International Conference on Information Networking (ICOIN)*, 2017, pp. 682-685, doi: 10.1109/ICOIN.2017.7899582.

[17] Trong-Minh Hoang, Van-Kien Bui, and Thanh-Tra Nguyen, “The performance evaluation of an IEEE 802.11 network containing misbehavior nodes under different backoff algorithms," *Security and*

Communication Networks, 2017, vol. 2017, Article ID 2459780, 8 pages, doi:10.1155/2017/2459780

[18] Wei Ye, J. Heidemann and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," in *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, June 2004, doi: 10.1109/TNET.2004.828953.

[19] G. Bianchi, "IEEE 802.11-saturation throughput analysis," in *IEEE Communications Letters*, vol. 2, no. 12, pp. 318-320, Dec. 1998, doi: 10.1109/4234.736171.

[20] T. Boulis, Castalia Version 3.3 Master. Access: September 8, 2022. [Online]. Available: https://github.com/boulis/Castalia/releases/tag/3.3

[21] Texas Instruments. “CC2420 single-chip 2.4 GHz RF transceiver,” Accessed on: September 8, 2022, [Online] Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 195–200 DOI: 10.15439/2022R33 ISSN 2300-5963 ACSIS, Vol. 33

Decoupling Sliding Mode Control of Underactuated Systems using a Takagi-Kang-Sugeno Fuzzy Brain Emotional Controller and Particle Swarm Optimization

Duc-Hung Pham
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Hai Duong, Vietnam.
phamduchunghp@gmail.com

Viet- Ngu Nguyen
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Hung Yen, Vietnam.
Ngunguyenviet77@gmail.com

Thi Minh- Tam Le
Faculty Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Hung Yen, Vietnam
leminhtamutehy@gmail.com

***Abstract*—A Tagaki-Kang-Sugeno fuzzy brain emotional controller (TFBEC) for decoupling control of underactuated nonlinear systems is developed in this paper. The decoupling sliding mode technique is used to achieve decoupling control performance. An amygdala cortex and a prefrontal cortex comprise the brain emotional model. The prefrontal cortex is an emotional neural network, while the amygdala cortex is a sensory neural network. The proposed TFBEC is adaptive, and the parameters can be adjusted to achieve efficient control performance. A TFBEC is used as the main controller to approximate an ideal controller and achieve the desired control performance, and a robust compensator is used to eliminate the remaining approximation error and achieve system stability. A particle swarm optimization is used to find the optimal learning rates of the proposed method. Finally, the TFBEC control system is demonstrated by controlling a bridge crane system with one degree of under actuation. Simulation results have confirmed the validity of the proposed approach.**

***Keywords*—Takagi-Kang-Sugeno fuzzy system, brain emotional controller, bridge-crane system.**

I. INTRODUCTION

For a specific type of underactuated nonlinear systems, the decoupling sliding mode control (DSMC) was developed. Separated into their own second-order systems, all subsystems need both a primary and secondary control function. Discrete subsystems' state variables can be thought of as sliding surfaces. For these sliding surfaces, we develop a primary goal condition and a secondary goal condition, with an intermediate variable taken from the sub-sliding surface condition to account for these subsystems [1]. The concept of sliding mode control (SMC) has recently been introduced as a means of controlling nonlinear systems whose dynamics are unknown.

LeDoux [3] initially observed in 1992 that emotions are critical to human perception and action. In 2001, Balkenius and Moren [4] created a model of emotional learning in the brain that was grounded in neurophysiology. To this end, they developed and tested in virtual reality a brain with a synthetic amygdala and frontal lobe. The creation of a model of emotional learning has received a lot of attention in recent years. The Brain Emotional Learning Controller (BELC), introduced by Lucas et al [5], is a notable example of a system built using this paradigm. The amygdala cortex and the sensory network in this BELC are in constant dialogue with one another, just as the orbitofrontal cortex and its counterpart do in the human brain. The BELC performs well in dynamic systems because of its rapid self-learning capability, low implementation complexity, and great robustness.

Decoupled sliding mode control (DSMC) alone is not enough to make highly nonlinear objects easier to control. Recent research that combines DSMC with neural networks, like the fuzzy neural network [6], has produced impressive results. Based on this direction of growth, this study suggests combining DSMC with a new controller called Takagi-Kang-Sugeno fuzzy brain emotional controller (TFBEC).

There are two kinds of fuzzy systems: 1) Takagi-Sugeno-Kang (TSK) fuzzy systems and 2) Mamdani fuzzy systems [7]. In TSK fuzzy systems, the "IF" parts of the TSK rules match the "IF" parts of other fuzzy inference rules. In general, the "THEN" part of TSK rules is a polynomial function of the input variables. This BELC has a new fuzzy neural network called TSK Fuzzy Brain Emotional Controller (TFBEC), which uses the TSK fuzzy inference algorithm. Both the TSK fuzzy neural network and the BELC have advantages that the TFBEC also has. The parameter update laws of the TFBEC are worked out, and a Lyapunov function is used to show that the control system is stable. A second controller is needed to act like the ideal controller. The auxiliary controller might be a good one [8].

The remainder of this paper can be summarized as follows: Section II discusses the problem formulation, Section III discusses the proposed TFBEC and the PSO method, Section IV discusses the simulation results, and Section V concludes the paper.

II. PROBLEM FORMULATION

Consider a nonlinear system with underactuated expressed in the following form [1]

$$\begin{cases} \dot{y}_1 = y_2 \\ \dot{y}_2 = g_1(y_1, y_2) + b_1(y_1, y_2)u + d_1 \\ \dot{y}_3 = y_4 \\ \dot{y}_4 = g_2(y_1, y_2) + b_2(y_1, y_2)u + d_2 \end{cases} \quad (1)$$

where $g_1(y_1, y_2)$, $g_2(y_1, y_2)$ and $b_1(y_1, y_2)$, $b_2(y_1, y_2)$ are bounded nominal nonlinear functions, $\mathbf{y} = [y_1\ y_2\ y_3\ y_4]^T \in \Re^4$ is the state vector, u is the control input; and d_1 and d_2 are the external disturbances. This system can be viewed as two subsystems, each with a second-order canonical two subsystems with second-order canonical form that includes the corresponding states (y_1, y_2) and (y_3, y_4). The decoupling control seeks to develop a single input u that simultaneously controls the states (y_1, y_2) and (y_3, y_4) to achieve the desired performance. The tracking error is defined as follows:

$$\begin{cases} e_1 = y_{d1} - y_1 \\ e_2 = y_{d2} - y_2 \\ e_3 = y_{d3} - y_3 \\ e_4 = y_{d4} - y_4 \end{cases} \quad (2)$$

where $\mathbf{y}_d = [y_{1d}\quad y_{2d}\quad y_{3d}\quad y_{4d}]^T \in \Re^4$ is the reference trajectory. Defining the coupling sliding surface for this system as [8]

$$s = \xi_1(e_1 - \omega) + e_2 \quad (3)$$

$$z = \xi_2 e_3 + e_4 \quad (4)$$

where ξ_1, ξ_2 selected coefficients correspond to those of a Hurwitz polynomial, and ω is derived from z and is defined as

$$\omega = \mathrm{sat}(z/\Phi_\omega)\omega_u, \quad 0 < \omega_u < 1 \quad (5)$$

where Φ_ω is the boundary layer of Z . Φ_ω transfers z to the correct range of y_1, and the definition of sat(.) function is:

$$\mathrm{sat}(z/\Phi_\omega) = \begin{cases} \mathrm{sgn}(z/\Phi_\omega), & \text{if } |z/\Phi_\omega| \geq 1 \\ z/\Phi_\omega, & \text{if } |z/\Phi_\omega| < 1 \end{cases} \quad (6)$$

where $\omega_u \leq 1$ therefore ω is a decaying oscillation signal

From (1), an ideal controller u_{ideal} can be represented as

$$u_{ideal} = b_1^{-1}\left(-\xi_1 y_2 - \xi_1\dot{\omega} - g_1 + \xi_1\dot{y}_{1d} + \dot{y}_{2d} - d_1\right) \quad (7)$$

u_{ideal} in (7) is unavailable due to an unidentified problem. Thus, a TFBEC is presented as a controller that approaches perfection.

III. Takagi-Kang-Sugeno Fuzzy Brain Emotional Controller

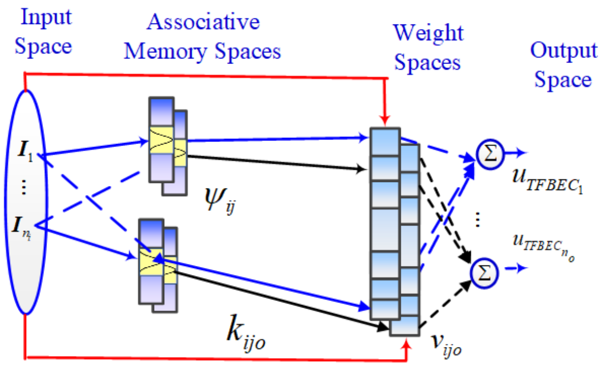


Fig. 1. The structure of TFBEC

The suggested TFBEC is depicted in Figure 1. The TFBEC employs the TSK fuzzy inference rules:

If I_1 is f_{1jo}, I_2 is f_{2jo} ,…, and I_{n_i} is $f_{n_i jo}$

then $u_o = I_1 k_{1jo} + I_2 k_{2jo} + \ldots + I_{n_i} k_{n_i jo}$ (8)

for $o = 1, 2, \ldots, n_o$; $j = 1, 2, \ldots, n_j$

(8)

where I_i is the input of the fuzzy inference system; k_{ijo} and u_o are respectively the TSK weight and the TFBEC output. Signal propagation and fundamental function in each TFBEC space are described as follows:

A. Input Space

For the input data, $I = \left[I_1, I_2, \ldots, I_{n_i}\right] \in \Re^{n_i}$, n_i is the input dimension.

B. Association Memory Space

In Sensory cortex space, the sigmoid function is adopted represented as:

$$\psi_{ij} = \exp\left(-\left(\frac{I_i - m_{ij}^B}{\sigma_{ij}^B}\right)^2\right) \quad (9)$$

where ψ_{ij} is the Gaussian function of prefrontal system input and amygdala system input for sensory cortex output, m_{ij}^B and σ_{ij}^B are respectively mean and variance.

C. Emotional Weight Space

This space uses the inference fuzzy rules as follows:

If I_1 is ψ_{1j}, I_2 is ψ_{2j} , …., and I_{n_i} is $\psi_{n_i j}$ then

$$v_{jo} = \sum_{i=1}^{n_i}(I_i k_{ijo}) \quad (10)$$

for $o = 1, \ldots, n_o$; $j = 1, \ldots, n_j$; $i = 1, \ldots, n_i$

D. Output Space

The output space is represented as follows

$$u_{TFBEC_o} = \sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\psi_{ij}v_{jo} = \sum_{i=1}^{n_i}\sum_{j=1}^{n_j}\psi_{ij}\sum_{i=1}^{n_i}(I_i k_{ijo}) \qquad (11)$$

for $o = 1,\ 2,\ldots,\ n_o$

A cost function is defined as $V_1 = \frac{1}{2}s^2$, then the derivate of it is $\dot{V}_1 = s\dot{s}$. Using the gradient descent method to find the proposed method's updating laws. The following equations describes the updated laws.

$$k_{ijo}(t+1) = k_{ijo}(t) + \Delta k_{ijo} \qquad (12)$$

$$m_{ij}^B(t+1) = m_{ij}^B(t) + \Delta m_{ij}^B \qquad (13)$$

$$\sigma_{ij}^B(t+1) = \sigma_{ij}^B(t) + \Delta\sigma_{ij}^B \qquad (14)$$

$$\Delta k_{ijo} = -\lambda_k.\frac{\partial\dot{V}_1}{\partial k_{ijo}} = -\lambda_k.\frac{\partial(s\dot{s})}{\partial k_{ijo}} =$$

$$-\lambda_k.\frac{\partial(s\dot{s})}{\partial u_{TFBEC_o}}.\frac{\partial u_{TFBEC_o}}{\partial k_{ijo}} = -\lambda_k.b_1.s.\psi_{ij}\sum_{i=1}^{n_i}I_i \qquad (15)$$

$$\Delta m_{ij}^B = -\lambda_m.\frac{\partial\dot{V}_1}{\partial m_{ij}^B} = -\lambda_m.\frac{\partial(s\dot{s})}{\partial m_{ij}^B} =$$

$$-\lambda_m.\frac{\partial(s\dot{s})}{\partial u_{TFBEC_o}}.\frac{\partial u_{TFBEC_o}}{\partial\psi_{ij}}.\frac{\partial\psi_{ij}}{\partial m_{ij}^B}$$

$$= -\lambda_m b_1 s\sum_{i=1}^{n_i}(I_i k_{ijo}).\ \frac{2(I_i - m_{ij}^B)}{(\sigma_{ij}^B)^2} \qquad (16)$$

$$\Delta\sigma_{ij}^B = -\lambda_\sigma.\frac{\partial\dot{V}_1}{\partial\sigma_{ij}^B} = -\lambda_\sigma.\frac{\partial(s\dot{s})}{\partial\sigma_{ij}^B} =$$

$$-\lambda_\sigma.\frac{\partial(s\dot{s})}{\partial u_{TFBEC_o}}.\frac{\partial u_{TFBEC_o}}{\partial\psi_{ij}}.\frac{\partial\psi_{ij}}{\partial\sigma_{ij}^B}$$

$$= -\lambda_\sigma.b_1.s\sum_{i=1}^{n_i}(I_i k_{ijo}).\frac{2(I_i - m_{ij}^B)^2}{(\sigma_{ij}^B)^3} \qquad (17)$$

where λ_k, λ_m, and λ_σ are learning rates with positive values. The learning rates can be optimized by the PSO algorithm. The approach error causes a tracking error in the control system because the TFBEC cannot perfectly replicate the ideal controller. As a result, a compensating controller is required to ensure the control system's robust stability. The control system is depicted in Figure 2.

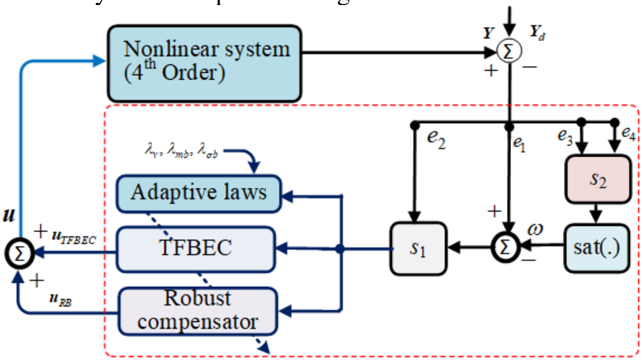


Fig. 2. Block diagram of TFBEC control system

When we take the derivative of (3), we get the following:

$$\dot{s} = \xi_1(\dot{e}_1 - \dot{\omega}) + \dot{e}_2 = \xi_1\left(\dot{y}_{1d} - \dot{y}_1 - \dot{\omega}\right) + \dot{y}_{2d} - \dot{y}_2 \quad (18)$$

Due to the inevitability of an approximation error between the TFBEC and the ideal controller, the latter can be expressed as the sum of the former and the latter, with the assumption that the approximation error is defined as

$$\Delta = u_{ideal} - u_{TFBEC} \qquad (19)$$

The control system is structured as follows to account for the approximation error:

$$u = u_{TFBEC} + u_{RB} \qquad (20)$$

$$u_{RB} = \hat{\mu}\text{sign}(s) \qquad (21)$$

where $\hat{\mu}$ is an estimated value of the variable μ being looked up. Then the adaptive law of $\hat{\mu}$ is as follows.

$$\hat{\mu} = \lambda_\mu|s| \qquad (22)$$

Picking a Lyapunov function using the formula

$$V = \frac{1}{2}s^2 + \frac{\tilde{\mu}\tilde{\mu}}{2\lambda_\mu} \qquad (23)$$

Then, the derivate of V is attained

$$\dot{V} = s(\sigma - u_{RB}) + \frac{\tilde{\mu}\dot{\tilde{\mu}}}{\lambda_\mu} = s\left(\sigma - \hat{\mu}\text{sign}(s)\right) + \frac{\tilde{\mu}\dot{\tilde{\mu}}}{\lambda_\mu}$$

$$= s\sigma - \hat{\mu}|s| + \frac{\tilde{\mu}\dot{\tilde{\mu}}}{\lambda_\mu} \qquad (24)$$

Because $\mu = \tilde{\mu} + \hat{\mu}$ is constant so: $\dot{\tilde{\mu}} = -\dot{\hat{\mu}} = -\lambda_\mu|s|$ therefore

$$\dot{V} = s\sigma - \hat{\mu}|s| - (\mu - \hat{\mu})|s| = -\left(\mu - |\sigma|\right)|s| \le 0 \qquad (25)$$

Since $\dot{V} \le 0$, $\dot{V} \le \dot{V}(0)$, provides that $\tilde{\mu}$ and s are bounded. Defining $\Lambda = \left(\mu - |\sigma|\right)s \le \left(\mu - |\sigma|\right)|s| \le -\dot{V}$. Integrating Λ with respect to time, obtains:

$$\int_0^t \Lambda(\tau)d\tau \ \le V(0) - V(t) \qquad (26)$$

Since $V(0)$ and $\dot{V}(t)$ are both constrained, $\dot{V}(t)$ is not increased, so $\lim_{t\to\infty}\int_0^t \Lambda(\tau)d\tau < \infty$. This points to the fact that $t \to \infty \Rightarrow s \to 0$. As a result, the TFBEC control method that was suggested has a high degree of guaranteed stability.

E. Particle Swarm Optimization (IPSO)

Particle swarm optimization (PSO) is an efficient optimization method recommended by Eberhart and Kennedy [12]. To obtain suitable learning rates λ_k, λ_m, and λ_σ for the update laws of TFBEC, the improved PSO algorithm is used [12]. The algorithm calculates the fitness function of each set, and then each set can be adjusted based on the local optimization position of the particles

L_{best_i} and the global optimization position of the swarm G_{best_i}. In this study, the fitness function is chosen as follows.

$$F=\frac{1}{|e|+0.1} \tag{27}$$

$p_i(t)$, $\theta_i(t)$ are respectively the current velocity and the current velocity of the particle. The update of $p_i(t)$ and $\theta_i(t)$ are given as follows.

$$p_i(t+1)=p_i(t)+\theta_i(t+1) \tag{28}$$

$$\begin{aligned}\theta_i(t+1)=\zeta.\theta_i(t)+\kappa_1.\mu_1.\left(P_{best_i}-p_i(t)\right)+\\ \kappa_2.\mu_2.\left(G_{best_i}-p_i(t)\right)\end{aligned} \tag{29}$$

where ζ is inertia weight, κ_1 and κ_2 are learning factors, μ_1 and μ_2 are two random variables in the range [0,1], $0\le\mu_1\le1$, $0\le\mu_2\le1$. The PSO algorithm can be represented as follows.

Step 1: Initialization of the swarm, the position of the particles is chosen randomly.

Step 2: Calculate the fitness function for each particle. $Fit\left(p_i\right)$

Step 3: Comparison of the fitness function with its best fitness function.

If $F\left(p_i\right)<F\left(G_{best_i}\right)$ then $F\left(G_{best_i}\right)=F\left(p_i\right)$ and $p_i=G_{best_i}$

Step 4: Comparison of the fitness function $F\left(p_i\right)$ of each particle with the best global particle $F\left(G_{best_i}\right)$

If $F(p_i)<F\left(G_{best_i}\right)$ then $F\left(G_{best_i}\right)=F(p_i)$ and $p_i=G_{best_i}$.

Step 5: Update the position and velocity with (28) and (29).

Step 6: Go back to step 2 and repeat until convergence.

IV. Simulation Results

For the purposes of this section, consider the control problem of a two-dimensional bridge crane system, as seen in Figure 2 and formulated as follows [9].

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=u \tag{30}$$

where $q=\begin{bmatrix}x & \theta\end{bmatrix}^T$ and matrices and vectors are expressly defined as follows.

$$M(q)=\begin{bmatrix} m_x+m & ml\times\cos\theta \\ ml\times\cos\theta & ml^2\end{bmatrix},$$

$$C(q,\dot{q})=\begin{bmatrix}0 & -ml\times\sin\theta\times\dot{\theta}\\ 0 & 0\end{bmatrix},$$

$G(q)=\begin{bmatrix}0 & mgl\times\sin\theta\end{bmatrix}^T$, $u=\begin{bmatrix}F-f_{rx} & d\end{bmatrix}^T$. Table I provides a place to define the various parameters of the system.

TABLE I
THE PAREMTERS OF BRIDGE CRANE SYSTEM

Parameter	Definiton
m_x	cart mass
m	load mass
l	cable length
x	cart displacement
θ	load swing angle
F	control input
f_{rx}	non-linear friction between the cart and the bridge
d	external disturbance
f_{r0x}, ε	static friction coefficients
k_{rx}	viscous friction

$$f_{rx}=f_{r0x}\tanh(\dot{x}/\varepsilon)-k_{rx}|\dot{x}|\dot{x} \tag{31}$$

k_{rx} can be determined through offline experimental testing and data collection. Consider the control problem in terms of putting the load at the appropriate location by simultaneously managing the horizontal motion of the carriage and minimizing the load's wobble. Specifically, we wish to push the position of the carriage x to the required place $p_d\in\Re$ while eliminating θ, which can be specified mathematically as.

$$\lim_{t\to\infty}x(t)=p_d,\qquad \lim_{t\to\infty}\theta(t)=0 \tag{32}$$

Define $y_1=\theta(t),\ y_2=\dot{\theta}(t),\ y_3=x(t),\ y_4=\dot{x}(t)$.

From (3)-(5), the following specifications, we select $\xi_1=5$, $\xi_2=0.5$, $\Phi_\omega=5$, $\omega_u=0.95$. The initial values are as follows: $m_x=7\ [kg]$, $m=1.025\ [kg]$, $l=4\ [m]$, $p_d=0.6\ [m]$, $x(0)=0$, $\theta(0)=0$. To test the effect of control effort in the presence of external disturbances, add wind speed as a disturbance during transport to validate at t=7 [s] $d=-45$ [N], at t=8 [s] $d=45$ [N], at t=9 [s] $d=-55$ [N], and at t=10 [s] $d=55$ [N]. The initial values for TFBEC are:

n_i=2, n_j=8, n_k=2, n_o=2,

$m_{11}^B=m_{21}^B=-1, m_{12}^B=m_{22}^B=-0.75, m_{13}^B=m_{23}^B=-0.5,$

$m_{14}^B=m_{24}^B=-0.25, m_{15}^B=m_{25}^B=0, m_{16}^B=m_{26}^B=0.25,$

$m_{17}^B=m_{27}^B=0.5, m_{18}^B=m_{28}^B=0.75$;

$\sigma_{11}^B=\sigma_{21}^B=-0.8, \sigma_{12}^B=\sigma_{22}^B=-0.6, \sigma_{13}^B=\sigma_{23}^B=-0.4,$

$\sigma_{14}^B=\sigma_{24}^B=-0.2, \sigma_{15}^B=\sigma_{25}^B=0, \sigma_{16}^B=\sigma_{26}^B=0.2,$

$\sigma_{17}^B=\sigma_{27}^B=0.4, \sigma_{18}^B=\sigma_{28}^B=0.6$; λ_k=0.05, $\lambda_m=0.05$, and λ_σ =0.01. The initial value of PSO: ζ=0.9 , $\kappa_1=\kappa_2=0.02$, $\mu_1=\mu_2=0.001$, and $\theta_i=0.002$.

The results of the simulation are presented in Figures 4-7, and Table II has a listing of the Root Mean Square Error (RMSE). The root mean square error (RMSE) of the sliding mode control (SMC) [10] and the FBELC [11] is smaller than the RMSE of the proposed TFBEC for θ (RMSE_ θ), which is 2.48 times and 1.01 times smaller, respectively. And the root mean square error of the suggested TFBEC for X, which is denoted by RMSE X, is 1.67 times smaller than the RMSE of

SMC and 1.17 times smaller than the RMSE of FBELC, respectively. However, because of the complexity of TFBEC, the amount of time needed to complete computations using our method is significantly longer. The process of changing the learning rates λ_k, λ_m, and λ_σ using the PSO algorithm is depicted in Figure 4. This demonstrates that the PSO algorithm is effective since after the first fluctuation in the learning rate values, those values eventually settle down to a constant positive value. Figure 5 demonstrates that the output of is less volatile when utilizing the proposed TFBEC in comparison to the output of conventional controllers, and that this leads to a more rapid transition to steady state. The results of X that are displayed in Fig. 6 are comparable to those displayed in Fig. 5. Figure 7 illustrates how the control efforts are altered when an external disturbance is present. The suggested TFBEC is able to adjust to noise more quickly than the SMC and the FBELC.

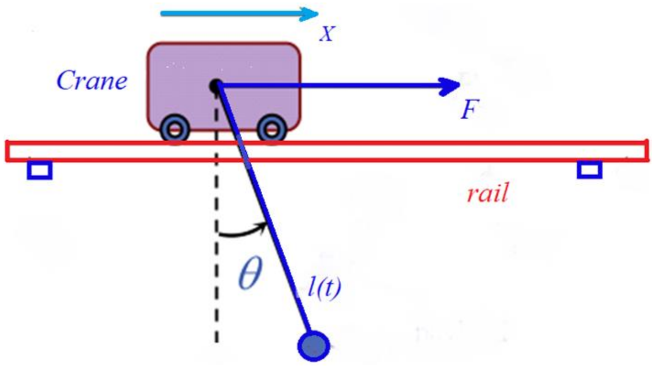


Fig. 3. A 2-dimensional underactuated bridge-crane

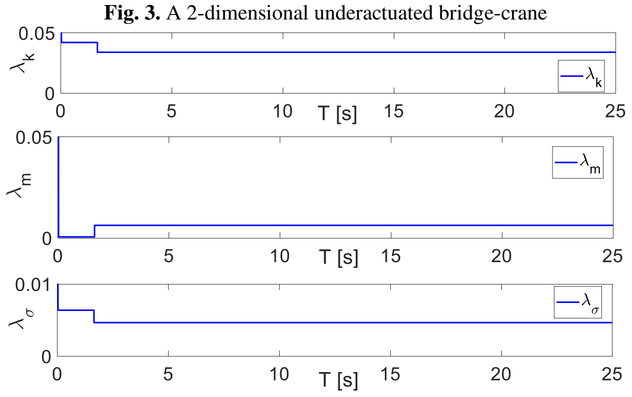


Fig. 4. The updating of learning rates using PSO algorithm.

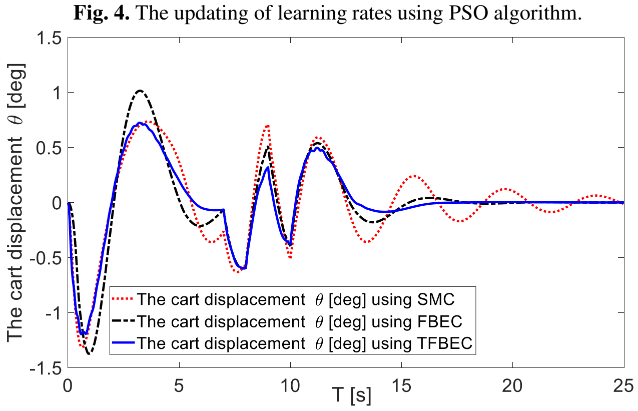


Fig. 5. The control results for the bridge crane for θ

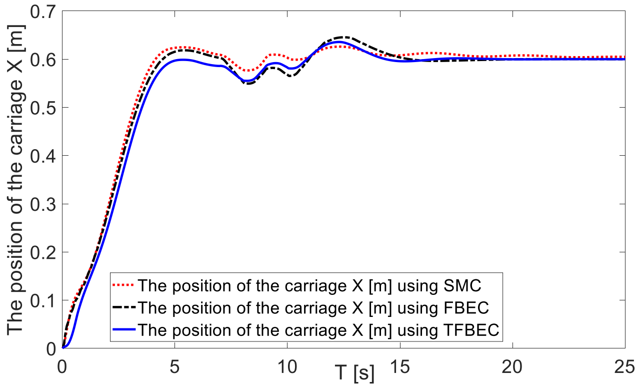


Fig. 6. The control results for the bridge cranes for X

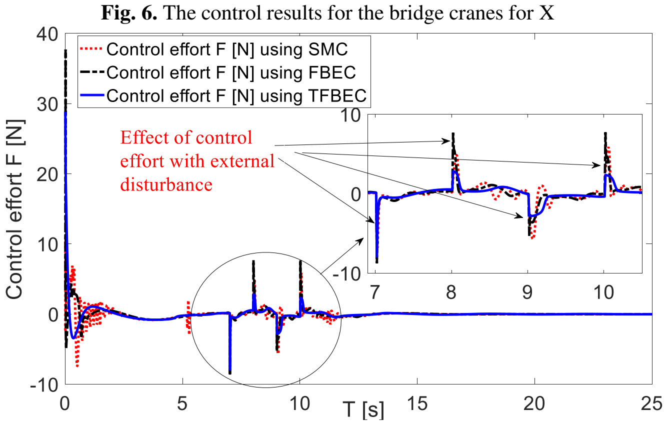


Fig. 7. The control effort F [N] for the bridge cranes

TABLE II
COMPARISION IN RMSE FOR BRIDGE CRANE SYSTEM

Method	Computation time [s]	RMSE_θ	RMSE_X
SMC [10]	0.085	0.380	0.01
FBELC [11]	0.175	0.155	0.007
The proposed TFBEC	0.234	0.153	0.006

V. Conclusion

It has been suggested that TFBEC can be used to decouple the control of nonlinear systems. In addition, the optimum values of the learning rates can be determined by the use of PSO. The results of a simulation run on a nonlinear bridge crane are shown, demonstrating that the effectiveness of the suggested control system was confirmed by the simulation. The developed controller is successful in properly tracking the target while making only minor errors and demonstrates rapid convergence.

References

[1] F. Yorgancioglu, and H. Komurcugil, “Decoupled sliding-mode controller based on time-varying sliding surfaces for fourth-order systems,” *Expert Systems with Applications*, vol. 37, no. 10,pp. 6764-6774, 2010.

[2] K. J. Å ström and B. Wittenmark, Adaptive Control, Reading, *Addision-Wesly,* Massachusetts, 1995.

[3] J. E. LeDoux, The Amygdala: Neurobiological Aspects of Emotion, *Wiley-Liss, New York*, pp. 339-351, 1992.

[4] C. Balkenius and J. Moren, "Emotional learning: A computational model of the amygdala," *Cybernetics and Systems*, vol. 32, no. 6, pp. 611-636, 2001.

[5] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami, "Introducing BELBIC: Brain emotional learning based intelligent controller,"

International Journal of Intelligent Automation and Soft Computing, vol. 10, no. 1, pp. 11-21, 2004.

[6] L.–C. Hung and H.-Y. Chung, “Decoupled sliding-mode with fuzzy-neural network controller for nonlinear systems,” *International Journal of Approximate Reasoning*,vol. 46, no. 1, pp 74-97, 2007.

[7] Lin, C. M., Pham, D. H., & Huynh, T. T. “Encryption and Decryption of Audio Signal and Image Secure Communications Using Chaotic System Synchronization Control by TSK Fuzzy Brain Emotional Learning Controllers,” *IEEE Transactions on Cybernetics*, 2021, . doi: 10.1109/TCYB.2021.3134245.

[8] T.-T. Huynh, C. –M. Lin, N.-Q. –K. Le, N. P. -Nguyen, and F. Chao. "Intelligent wavelet fuzzy brain emotional controller using dual function-link network for uncertain nonlinear control systems." *Applied Intelligence*, vol. 52, no. 3, pp. 2720-2744, 2022.

[9] N. Sun, Y.-C. Fang, and X.-Q. Wu, "An enhanced coupling nonlinear control method for bridge cranes," *IET Control Theory and Applications,* vol. 8, no.13, pp. 1215-1223, 2014.

[10] C. M. Lin, and Y. J. Mon, "Decoupling control by hierarchical fuzzy sliding-mode controller," *IEEE Transactions on Control Systems Technology,* vol. 13, no. 4, pp. 593-598, 2005.

[11] C.-M. Lin, D.-H. Pham, and T.-T. Huynh, “Synchronization of chaotic system using a brain-imitated neural network controller and its applications for secure communications,” *IEEE Access,* vol. 9, pp. 75923-75944, 2021.

[12] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43: IEEE.

DOI: 10.15439/2022R04
ISSN 2300-5963 ACSIS, Vol. 33

udCATS: A Comprehensive Unsupervised Deep Learning Framework for Detecting Collective Anomalies in Time Series

1st Truong Son Pham
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
0000-0001-6320-8501

2nd Viet Hung Nguyen
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
0000-0002-9818-4455

3rd Anh Thang Le
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
0000-0003-2019-4781

4th Van Duong Bui
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
0000-0002-0905-5214

***Abstract*—Anomaly detection has recently gained enormous attention from the research community. It is widely applied in many industrial areas, such as information security, financing, banking, and insurance. The data in these fields can mainly be represented as time series data, the corollary being that time series anomaly detection plays an essential role in these applications. Therefore, many authors have tried to solve the problem of collective anomaly detection in time series. They have proposed several approaches, from classical methods such as Isolation Forests to modern deep learning networks such as Autoencoders. However, a comprehensive framework for handling this problem is still lacking. In this work, firstly, we propose using an Attention-based Bidirectional LSTM Autoencoder (Att-BiLSTM-AE) as an anomaly detection model. Furthermore, in the essential part of this paper, we developed a comprehensive unsupervised deep learning framework, udCATS, to solve the problem of detecting collective anomalies in time series. Our experiments show that the Att-BiLSTM-AE outperforms other detection models, and using it within the udCATS framework increases the detection accuracy.**

***Index Terms*—collective anomaly, time series, unsupervised, deep learning**

I. Introduction

Anomaly detection plays an essential role in many industrial areas, for example, financing, banking, information security, and insurance. Many data in these domains can be represented as time series. Because of that, anomaly detection in time series data has recently gained massive attention from the research community.

A time series can be univariate or multivariate, discrete or continuous. In this work, we focus only on discrete univariate time series. Therefore, the term ”time series” used in the rest of this article refers to a discrete univariate time series. **Time series** by its definition, is a set of data collected at successive, discrete timestamps and can be written as $\{X_t, t \in Z\}$ [1]. The term anomaly of a time series can be considered an outlier. From the traditional point of view, an **outlier/anomaly** is an observation that varies ”extensively from the other one as to produce suspicions that it was generated by a different mechanism [2].”

An anomaly in time series can deliver important information. For example, it could be some unwanted data points that were produced or collected incorrectly. In this case, anomaly detection is essential for data cleaning, which is crucial for developing proper machine learning models. In addition, the anomaly can also represent the events of interest, such as machine breakdowns, cyber-attacks, and insurance frauds, which are the main applications of anomaly detection in time series.

The anomalies in time series can be divided into three main categories: point, collective, and contextual anomaly [3]. A time series data point is an anomaly when it behaves out of the ordinary compared to most other points. The term collective anomaly refers to consecutive data points with unusual behavior. It is crucial to mention that each point of an abnormal sub-sequence is not necessary an outlier. Contextually anomaly is used when some time series points are typical in a specific context but anomalous in another context [3].

We focus here on collective anomaly detection because detecting the collective outliers is much more challenging than detecting the unusual points. As mentioned above, a single data point in a sub-sequence may not be an outlier; however, they will build up an abnormal sub-sequence when considering them in consecutive order. That makes the research problem much more challenging. Besides that, the problem of point anomaly detection is already well-researched [4]. In contrast, the detection accuracy can still be improved in the problem of collective anomaly detection by proposing or applying contemporary deep learning networks. In our work, firstly, we propose using an attention-based bidirectional LSTM Autoencoder (Att-BiLSTM-AE) as an anomaly detection model. Furthermore, in the essential part of this paper, we developed a comprehensive unsupervised deep learning framework called udCATS to solve the problem of detecting collective anomalies in time series. Our experiments show that the Att-BiLSTM-AE outperforms other detection models while using it within the udCATS framework increases the detection accuracy.

The rest of this paper is organized as follows. First, section II concerns some selected unsupervised learning approaches to detect collective anomalies. Next, the udCATS framework, which includes four primary processes, is described in Section III. Finally, section IV details our experiments and discusses their results before we clarify in Section V how we would like to improve the framework continually.

II. Related Work

Many methods and approaches have been proposed to detect collective anomalies in time series. They can be grouped into two categories: supervised and unsupervised detection methods. In comparison, the approaches can be divided into three groups: statistical, classical machine learning, and deep learning models [3].

Supervised methods typically produce increased detection precision; however, they are pretty unuseful because they require labeled data sets, which are usually unrealistic. The labeling process is nowadays one of the most costly steps in a Machine Learning Pipeline. On the other hand, unsupervised methods are much more practical and valuable. However, receiving a high accuracy with unsupervised learning models is very demanding. Deep learning models have demonstrated their robustness and accuracy in an unsupervised manner compared to statistical and classical machine learning models [5], [6]. In this section, unsupervised approaches applied for collective anomaly detection problems and time series are discussed briefly [6]–[11].

One of the most straightforward ideas to detect the anomalies in an unsupervised manner is applying clustering algorithms such as K-Means Clustering [8] or Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [9]. The detailed descriptions of these clustering algorithms are provided by [12], [13] and [14].

C. Mete, F. Dadaşer-Çelik, and A. Dokuz [9] applied DBSCAN to detect anomalies in a dataset that contains the daily average temperature over 33 years. The author segmented the time series into monthly sequences, normalized them by their mean and variance, and then clustered them with DBCAN. The results show that DBSCAN can detect collective anomalies even if there is no significance between them and the usual data points. Keogh and Lin [8], nevertheless, have indicated that using clustering algorithms for collective anomalies detection is senseless. They showed that the cluster centers discovered for several runs of the K-means algorithm on the same dataset are not remarkably contrasting to the one of a random walk process. Some authors tried to analyze and overcome this problem. However, it remains unsolved [15].

L. Bontemps, V.L. Cao, J. McDermott and N.A. Le-Khac [7] proposed a LSTM-based collective anomaly detection model. Firstly, the time series is modeled with an LSTM RNN [16]. The predictive model is then adapted to propose a circular array containing prediction errors from several recent time steps. Finally, a predetermined threshold is applied to indicate a collective anomaly. To evaluate the model, the authors converted the KDD 1999 dataset [17] into a time series version. The results showed that without any false alarm, the model could detect 86% of the collective anomalies. If the threshold is set to capture all the anomalies, the number of false alarms is increased to 63.

Besides LSTM Network, some other deep learning models are also proposed for detecting collective anomalies in time series, such as Convolutional Neural Networks (CNN) [6], Gated recurrent unit (GRU) [10], and Autoencoder [11]. The results show that, in general, deep learning models perform very well for collective anomaly detection problems in time series data.

We can make some important conclusions based on the knowledge gained from a comprehensive literature review, especially from the selected publications discussed above:

- There is still no comprehensive framework for detecting time series collective anomalies. The task of detecting collective anomalies is not trivial as putting the time series into a detection model to get the results. It requires several steps, for example, splitting the time series into sub-sequences, reducing the data dimension, scaling the features, etc.
- Clustering-based approaches are not suitable for this kind of problem.
- Deep learning models produce highly accurate results when solving the problem of collective anomaly detection.

For these reasons, we propose a comprehensive framework, called udCATS, for detecting collective anomalies in time series in an unsupervised manner. The framework uses an Attention-based Bidirectional Long Short-Term Memory Autoencoder as the anomaly detection engine. All the components of the udCATS framework are essential for solving the problem.

III. udCATS Framework

This section explains the udCATS framework in detail. It first clarifies the architecture and then each component of the framework.

A. Framework Architecture

The framework contains four components: time series segmentation, representation, scaling, and anomaly detector engine. The time series is first segmented into sub-sequences, later transformed to reduce the high dimensionality. These processes are called segmentation and representation. The output of the representation process is then used as the input for the data scaling process. In the end, an Attention Bidirectional

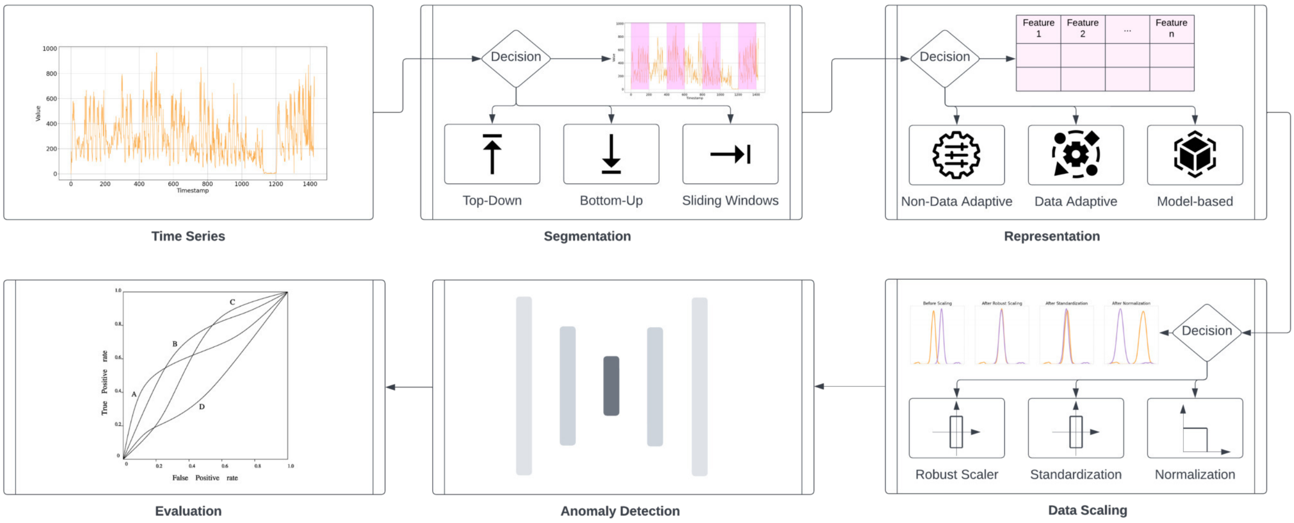


Fig. 1. Architecture of the udCATS Framework

Long Short-Term Memory Autoencoder is used to detect abnormal samples. For example, suppose a sample is classified as an anomaly. In that case, it can be used to identify the original sub-sequence to determine the collective abnormalities. Figure 1 illustrates the architecture of the udCATS framework.

Each of the components mentioned above is a selection process, which means different methods can be selected based on the nature of the input time series. For time series segmentation, top-down, button-up, or sliding windows can be selected, while non-data adaptive, data-adaptive, and model-based approaches are the most prominent time series representation approaches. Data-dictated representation can also be discovered in the literature. However, it is not widely used for this task. We experimentally recommend an Attention Bidirectional Long Short Term Memory Autoencoder as an anomaly detection engine. Although it is not mandatory, another deep learning network can also be used for this part. It depends, as explained, on the nature of the input data. Last, udCATS establishes standardization, normalization, and robust scaling for the data scaling process.

The remainder of this section expresses each element of the framework in detail.

B. Time Series Segmentation

Time series segmentation is a method of time-series analysis in which an input time series is divided into a sequence of discrete segments, called sub-sequence, to reveal the underlying properties of its source [18]. An optimal segmentation algorithm is defined as the one with minimal approximation error, calculated based on the difference between the segmented sub-sequences and the original time series. Figure 2 visualizes the segmentation process of the proposed udCATS framework. This is inspired by the work of M. Lovric, M. Milanovic, and M. Stamenkovi [18].

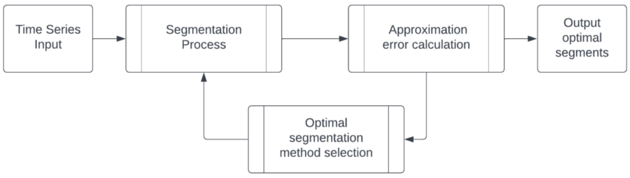

Time Series Input
Segmentation Process
Approximation error calculation
Output optimal segments
Optimal segmentation method selection

Fig. 2. The time series segmentation process

The following paragraphs describe the most well-known segmentation algorithms: sliding windows, top-down, and bottom-up [18].

Sliding Windows, also called ”brute-force” or ”one-pass” algorithm [18], it is one of the most widely involved time series segmentation algorithms. It starts with appointing the first data point as the anchor. Afterward, the window size is initially determined, and based on this size, the approximation error for the potential segment is calculated. Next, the window size is increased until the approximation error exceeds a predetermined threshold. Finally, a segment is created with the possible largest window size. This process is repeated until the sliding windows are across the entire time series. The new anchor is updated as the next data point right behind the created segment.

The **Top-Down** algorithm considers the original time series as one major segment. It starts with finding the breaking point, which divides the time series into two parts with the maximal difference between them. The approximation error is then calculated for both segments and compared with the predetermined threshold. These steps are repeated for all of the segments until the approximation error exceeds the threshold [18].

The **Bottom-Up** algorithm is the opposite of the top-down algorithm described above. It starts with segmenting the time series of length n into $n-1$ segments. Then, a segment is decided to merge with the one on the left or the right based on increasing the approximation error. Finally, it takes the one with a minor error increase. The merging process is repeated until the approximation error of a segment exceeds a predetermined threshold [18].

C. Time Series Representation

Unsupervised detection methods often do not directly use the original time series data points as the input. Instead, representations of the time series will be used. The representation is helpful for dimension reduction and similarity measurement and often helps produce better results [19].

There are four main approaches to time series representation: non-data adaptive, data-adaptive, model-based, and data-dictated representation [20]. The parameters can be fine-tuned with the first three approaches to find the best time series compression for the particular application. However, the time series dictates the compression itself with the last one. For this reason, only non-data adaptive, data-adaptive, model-based approaches are used for the selection process of the time series representation process.

In **non-data adaptive** algorithms, the represented parameters remain the same for all time series, independent of their nature. Some of the most widely used non-data adaptive algorithms are Discrete Fourier Transform (DFT), Piecewise Aggregate Approximation (PAA), DCT (Discrete Cosine Transform), or Wavelets [20].

In **data adaptive** representations, the parameters vary depending on the available data. In the literature, we can find some well-known methods for data-adaptive representation, such as Symbolic Aggregate Approximation, Piecewise Linear Approximation, or Singular Value Decomposition [20]

The **model-based** approaches assume that the observed time series was created based on the basic model. The aim is to find the parameters of such a model as a representation. Two time series are then considered similar if an identical set of parameters can model them. The model can be a Hidden Markov, statistical, or even deep learning one [20].

D. Data Scaling

For the scaling process, we propose selecting from three of the most famous and standard techniques: **normalization**, **standardization**, and **robust scaling**. Readers are referred to [21] for more detailed explanations of these scaling methods and how to select the right one based on the data distribution and the applications.

E. Attention-based Bidirectional Long Short Term Memory Autoencoder as the Anomaly Detection Engine

As mentioned above, several detection models can detect collective anomalies after the segmentation, representation, and scaling process. Some examples are the One-Class Support Vector Machine, Isolation Forest, or AutoEncoder. However, we recommend using an Attention-based Bidirectional Long Short Term Memory Autoencoder as the anomaly detection engine. The previous works [22]–[25] also inspire this recommendation. The authors have proved the efficiency and robustness of LSTM- and Bidirectional LSTM- Autoencoder for the anomaly detection problem. Figure 3 illustrates a simplified structure of an Attention-based Bidirectional Long Short Term Memory Autoencoder.

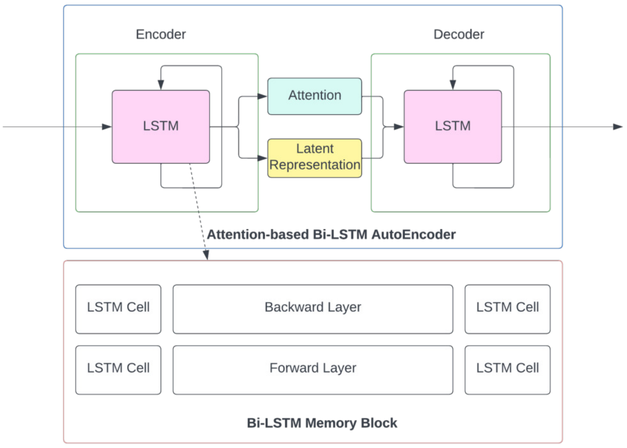


Fig. 3. Attention-based Bidirectional Long Short Term Memory Autoencoder

Because of the limitation of the pages, we will not describe the network in detail. Instead, readers, who are interested in this network, are referred to [25]–[27] for more information.

IV. Experiments and Results

This section describes the dataset, accuracy measurement, and the results of the experiments.

A. Dataset Description and Experiment Settings

The data used for the experiments in this article is the S5 dataset, provided by Yahoo [28]. This is a labeled benchmark dataset for anomaly detection. We compared the above-mentioned unsupervised methods based on their performance with this dataset. Therefore, it is essential to mention that the data labels are only used for the performance evaluation and not for the model training process.

The time series dataset represents the traffic of Yahoo services. The anomalies were labelled by experts. This dataset consists of 67 different time series. Each of them has 1400 data points, which were recorded hourly. About 1.9% of the data are anomalies. The dataset is divided into training and test sets where 70% of the data are used for training and 30% for testing. The training set does not contain any abnormal sub-sequence. Figure 4 visualizes a time series with collective anomalies colored red.

B. Accuracy Measurement

Because we have the labeled anomalies in the test set, AUC can be used to evaluate the framework's performance.

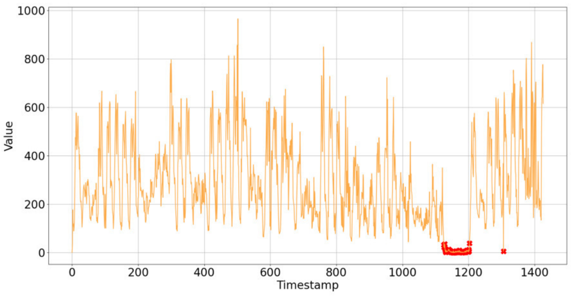


Fig. 4. A Time series with collective anomalies

AUC is the abbreviation of ”Area under the ROC (Receiver Operating Characteristic) curve.” That is, AUC represents the entire two-dimensional area under the ROC curve. ROC curve is a diagram showing the performance of a detection model at all values of thresholds. This curve illustrates two parameters: true positive rate (TPR) and false positive rate (FPR). The true positive rate is also known as the recall.

C. Results

In this part of the section, the results of the experiments are discussed. After the segmentation process, which is mandatory, the optimal length of a sub-sequence is experimentally set to 4. The most suitable segmentation method for this dataset is the sliding windows algorithm. Because the window size is tiny, the non-data adaptive method was applied for the representation process. The transformed vectors are at the end scaled with a robust scaler. The experimental results show that all four main processes of the framework are essential for high detection accuracy. Missing one of them will lead to lower performance.

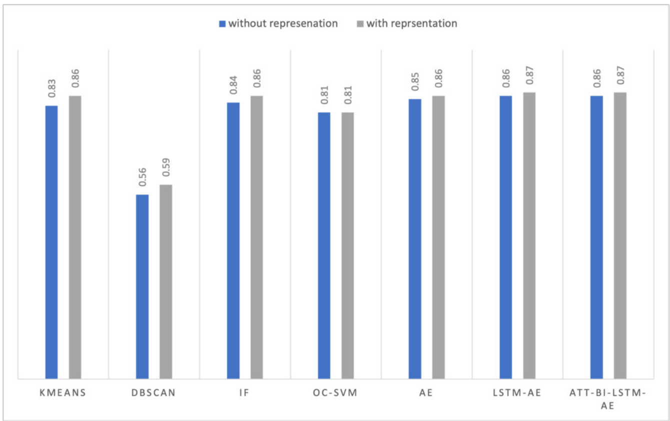


Fig. 5. Importance of the representation process

Figure 5 illustrates the importance of the representation process. The figure shows that the accuracy of six models (out of seven) is improved while applying the representation process, while the accuracy of the last one remains the same. Another critical remark is, together with LSTM AutoEncoder, Attention Bidirectional Long Short-Term Memory Autoencoder outperformed other detection models in both cases, with or without the time series representation process.

Figure 6 visualizes the performance ace of the udCATS framework with different detection models.

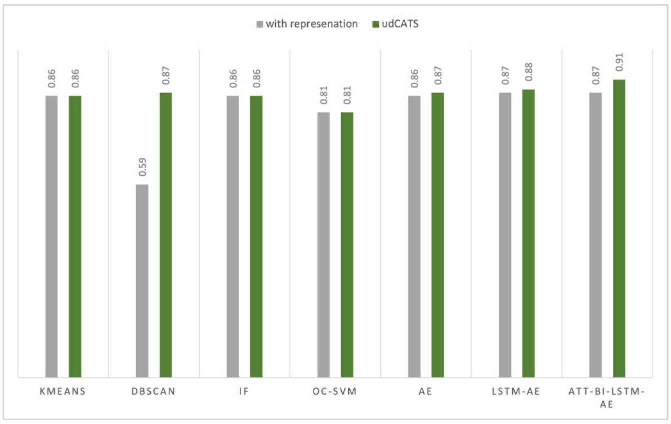


Fig. 6. Performance of udCATS Framework

From the graphic, it is crucial to observe that the scaling process of the comprehensive udCATS framework improved the accuracy of five detection models. The remaining two models performed at the same level. Besides, the udCATS framework with Attention-based Bidirectional Long Short Term Memory Autoencoder as the anomaly detection engine received the highest accuracy, represented by the AUC values. To obtain the best results, the confidence interval of the detection model is predetermined with a value of 0.95.

Table I shows the averaging AUCs of the models in different settings, while figure 7 illustrates the box plot of the udCATS framework’s AUCs over the whole dataset. Besides the mean of the AUCs, which is 0.91, the box plot also shows their median. The median is very high, around 0.97. The box plot is short, which means the udCATS framework performs with a high level of agreement over the whole data set of 67 time series.

TABLE I
EXPERIMENTAL RESULTS

Detection Model	**Without TS Representation**	**With TS Representation**	**udCATS**
K-Means	0.83	0.86	0.86
DBSCAN	0.56	0.59	0.87
IF	0.84	0.86	0.86
OC-SVM	0.81	0.81	0.81
AutoEncoder	0.85	0.86	0.87
LSTM AE	0.87	0.86	0.88
Att-Bi-LSTM-AE	0.87	0.86	**0.91**

V. Conclusion and Outlook

In this work, we provided two main contributions. Firstly, we experimentally demonstrated that an Attention-based Bidi-

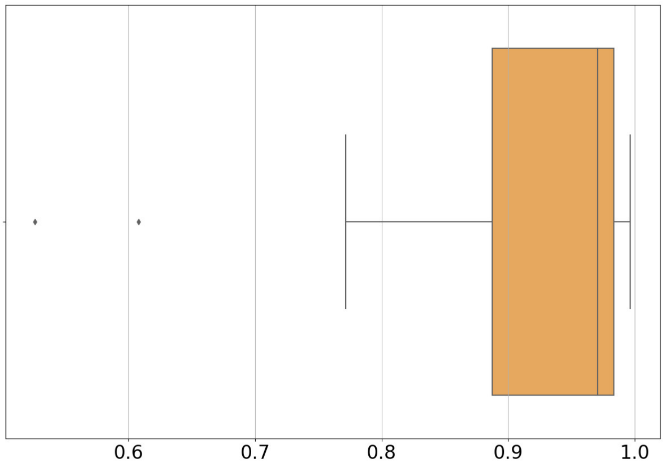


Fig. 7. Performance of udCATS Framework with Att-Bi-LSTM-AE

rectional LTSM Autoencoder could handle the collective anomaly detection of a time series. Secondly, and most importantly, we proposed a comprehensive framework, called udCATS, for solving the problem, which contains four main selecting processes: time series segmentation, representation, data scaling, and anomaly detection. To the best of our knowledge, this is the first comprehensive framework to handle this problem. The experimental results show that the Attention-based Bidirectional LTSM Autoencoder model performed better than the other detection models. Using it within the udCATS framework significantly improved the detection accuracy.

The following steps will assess the framework with more benchmark data sets. First, this would guide to an improvement of the framework architecture. Afterward, we will extend the selection processes with other methods and try to find a method to implement these processes to work fully automatically. Last but not least, we could combine the loss function of the four individual processes into one total loss function. The idea is to develop an end-to-end training process that improves accuracy.

References

[1] Judith D Singer and John B Willett. It's about time: Using discrete-time survival analysis to study duration and the timing of events. *Journal of educational statistics*, 18(2):155–195, 1993.

[2] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[3] Mohammad Braei and Sebastian Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*, 2020.

[4] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019.

[5] John Cristian Borges Gamboa. Deep learning for time-series analysis. *arXiv preprint arXiv:1701.01887*, 2017.

[6] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. Deepant: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access*, 7:1991–2005, 2018.

[7] Loïc Bontemps, Van Loi Cao, James McDermott, and Nhien-An Le-Khac. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International conference on future data and security engineering*, pages 141–152. Springer, 2016.

[8] Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2):154–177, 2005.

[9] Mete Çelik, Filiz Dadaşer-Çelik, and Ahmet Şakir Dokuz. Anomaly detection in temperature data using dbscan algorithm. In *2011 international symposium on innovations in intelligent systems and applications*, pages 91–95. IEEE, 2011.

[10] Wentai Wu, Ligang He, and Weiwei Lin. Local trend inconsistency: a prediction-driven approach to unsupervised anomaly detection in multi-seasonal time series. *arXiv preprint arXiv:1908.01146*, 2019.

[11] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014.

[12] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[14] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[15] Seyedjamal Zolhavarieh, Saeed Aghabozorgi, and Ying Wah Teh. A review of subsequence time series clustering. *The Scientific World Journal*, 2014, 2014.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Stephen D. Bay, Dennis F. Kibler, Michael J. Pazzani, and Padhraic Smyth. UCI machine learning repository, 1999.

[18] Miodrag Lovrić, Marina Milanović, and Milan Stamenković. Algoritmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, 1(1):31–53, 2014.

[19] Xiaoyue Wang, Abdullah Mueen, Hui Ding, Goce Trajcevski, Peter Scheuermann, and Eamonn Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2):275–309, 2013.

[20] Chotirat Ratanamahatana, Eamonn Keogh, Anthony J Bagnall, and Stefano Lonardi. A novel bit level time series representation with implication of similarity search and clustering. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 771–777. Springer, 2005.

[21] Pallavi Pandey and Avinash Navlani. Feature scaling: Minmax, standard and robust scaler, Nov 2020.

[22] Sanket Mishra, Varad Kshirsagar, Rohit Dwivedula, and Chittaranjan Hota. Attention-based bi-lstm for anomaly detection on time-series data. In *International Conference on Artificial Neural Networks*, pages 129–140. Springer, 2021.

[23] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45, 2020.

[24] HD Nguyen, Kim Phuc Tran, Sébastien Thomassey, and Moez Hamad. Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282, 2021.

[25] Ashima Chawla, Paul Jacob, Brian Lee, and Sheila Fallon. Bidirectional lstm autoencoder for sequence based anomaly detection in cyber security. *International Journal of Simulation–Systems, Science & Technology*, 2019.

[26] Ariyo Oluwasanmi, Muhammad Umar Aftab, Edward Baagyere, Zhiguang Qin, Muhammad Ahmad, and Manuel Mazzara. Attention autoencoder for generative latent representational learning in anomaly detection. *Sensors*, 22(1):123, 2021.

[27] Jing Wang, Guigen Nie, Shengjun Gao, Shuguang Wu, Haiyang Li, and Xiaobing Ren. Landslide deformation prediction based on a gnss time series analysis and recurrent neural network model. *Remote Sensing*, 13(6):1055, 2021.

[28] N Laptev and S Amizadeh. Yahoo anomaly detection dataset s5. *URL http://webscope. sandbox. yahoo. com/catalog. php*, 2015.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 207–211
DOI: 10.15439/2022R28
ISSN 2300-5963 ACSIS, Vol. 33

An investigation of the complexities of a malignant tumor's fractional-order mathematical model

Rakhi Singh
Ph.D Scholar,
Department of Mathematics
UIT, RGPV, Bhopal, India
rakhirekha1990@gmail.com

Vijay Kumar Gupta
Prof., Department of Mathematics
UIT, RGPV, Bhopal, India
vkgupta@rgtu.net

Jyoti Mishra
Prof., Department of Mathematics
GGITS, Jabalpur, India
jyotimishra@ggits.org

***Abstract*—In this paper, model of a malignant tumor & associated problems are examined using fractional-order method. We consider a case where the malignant tumor cells' net death rate is solely time-dependent. Fractional homotopy decomposition method (HDM) has been applied to determine model's series solution. The answer to the HDM is given using the Maclaurin expansion. This method's use of the Mathematica software package allows for fast and simple computation of series solutions, which is one of its benefits.**

***Index Terms*—Homotopy decomposition method, Series solution, Caputo fractional derivative, Fractional malignant tumor model.**

I. Introduction

Fractional calculus and its applications are currently experience rapid development, with more and more compelling real-world applications. Fractional calculus is an attractive area of applied analysis that is utilized to simulate biological problems in science. Fractional calculus has been very beneficial in the modeling of many diseases. To understand its assessment, presence, stability, and control, research into mathematical models of infectious diseases is crucial. Fractional differential equations have been used in recent years to provide mathematical representations of biological processes [1–6].This work aims to apply the homotopy decomposition technique HDM [7], a new method, to solve the Time-Fractional Malignant Tumor Growth Model [5]. Malignant tumor is another term for a cancerous tumor. When a tumor is called "malignant," it means it is cancerous and has a high chance of spreading beyond its original location. Although they can infect other organs, the cancer cells that spread to other body regions are identical to the original ones. For instance, if lung cancer spreads to the liver, the cancer cells there are still lung cancer cells [2-4].

Korpinar Z et al. [4] explain how RPS method is used to analyze the most recent series solutions of several fractional cancer tumor models. Saadeh R et al. [3] described a new approach LRPS method to find numerical solution of some model of cancer tumor. Iyiola OS et al. [5] released their paper in which they said that 3 distinct scenarios of net death rate are taken into consideration, even in situations when the net death rate of cancerous cells depends on cluster of cells. With help of q-HAM, find answer of the time-fractional partial differential equation. Gandhi H et al. [2] says that the situation where net death rate & tumor growth are considered & therapy is time-dependent. To find model's solution, the fractional RDT method was used.

Our study is moving forward with a time-fractional malignant tumor model under a beginning condition to account for the fact that the model is provided via a time-fractional differential equation and that the "net death rate of tumor cells" is solely dependent on time.

$$\frac{\partial^{\alpha}\wp(s,w)}{\partial w^{\alpha}} = \frac{\partial^{2}\wp(s,w)}{\partial s^{2}} - w^{2}\wp(s,w)$$

where $w \geq 0,\ 0 \leq s \leq 1,\ 0 < \alpha \leq 1$, with initial condition as

$$\wp(s,0) = e^{rs}.$$

where $\wp(s,w)$ is tumor cell density at position s at time w & r is dependent on therapy death rate at time w.
In this work, Caputo fractional derivative is used.

II. Preliminaries

Definition 2.1. Riemann-Liouville (RL) fractional integral operator for order $\alpha \geq 0$, of a function $\phi \in L^{1}(a,b)$ is given as in [12,13],

$$I^{\alpha}\varphi(w) = \frac{1}{\Gamma(\alpha)}\int_{0}^{w}(w-\tau)^{\alpha-1}\varphi(\tau)\,d\tau,\ w>0, \quad (2.1)$$

where Γ indicates the Gamma function and $I^{\alpha}\varphi(w) = \varphi(w)$.
Definition 2.2. In Caputo's definition, a fractional derivative is

$$D^{\alpha}\varphi(w) = I^{i-\alpha}\varphi(w) = \frac{1}{\Gamma(i-\alpha)}\int_{0}^{w}(w-\tau)^{i-\alpha-1}\varphi^{(i)}(\tau)\,d\tau, \quad (2.2)$$

where $i - 1 < \alpha \leq i, i \in \mathrm{N}, w > 0$, assuming the right side is present.

Lemma 2.1. Let $w \in (a,$ b]. Then

$$[I^{\alpha}(w-a)^{\beta}](w) = \frac{\Gamma(\beta+1)}{\Gamma(\beta+\alpha+1)}(w-a)^{\beta+\alpha}, \quad (2.3)$$

where $\alpha \geq 0, \beta > 0$.

III. Basic concept of the HDM

In order to show how this technique's basic structure works as in [16,18], focus on a common nonlinear non-homogeneous fractional PDE using the following structural beginning conditions:

$$\frac{\partial^\alpha \varpi(s,w)}{\partial w^\alpha} = L(\varpi(s,w)) + N(\varpi(s,w)) + \wp(s,w), \qquad (3.1)$$

$where \alpha > 0.$

subject to initial conditions

$$D_0^{\alpha-l}\varpi(u,0) = \wp_l(s), \quad l = (0,.....,n-1),$$

$$D_0^{\alpha-n}\varpi(s,0) = 0, \quad n = [\alpha],$$

$$D_0^{l}\varpi(s,0) = \hbar_l(s), \quad l = (0,.....,n-1),$$

$$D_0^{n}\varpi(s,0) = 0, \quad n = [\alpha],$$

where, $\frac{\partial^\alpha}{\partial w^\alpha}$ indicate Caputo or Riemann-Liouville fractional derivative, $\wp$ is commonly used function, *N* is common nonlinear fractional differential operator, & *L* represent linear fractional differential operator. HDM prime pace is used to switch fractional PDE to fractional PIE via using inverse operator I_v^α is utilized on both sides of "(3.1)". Riemann-Liouville fractional derivative case,

$$\varpi(s,w) = \sum_{j=1}^{n-1}\frac{\wp_j(s)}{\Gamma(\alpha-j+1)}w^{\alpha-j} + \frac{1}{\Gamma(\alpha)}\int_0^w (w-\alpha)^{\alpha-1} \qquad (3.2)$$
$$[L(\varpi(s,\tau)) + N(\varpi(s,\tau)) + \wp(s,\tau)]d\tau;$$

Caputo fractional derivative case,

$$\varpi(s,w) = \sum_{j=1}^{n-1}\frac{\hbar_j(s)}{\Gamma(\alpha-j+1)}w^{j} + \frac{1}{\Gamma(\alpha)}\int_0^w (w-\alpha)^{\alpha-1} \qquad (3.3)$$
$$[L(\varpi(s,\tau)) + N(\varpi(s,\tau)) + \wp(s,\tau)]d\tau.$$

or by placing generally

$$\sum_{j=1}^{n-1}\frac{\wp_j(s)}{\Gamma(\alpha-j+1)}w^{\alpha-j} = G(s,w)$$ or

$$G(s,w) = \sum_{j=1}^{n-1}\frac{\hbar_j(s)}{\Gamma(\alpha-j+1)}w^{j},$$

we find

$$\varpi(s,w) = G(s,w) + \frac{1}{\Gamma(\alpha)}\int_0^w (w-\alpha)^{\alpha-1}$$
$$[L(\varpi(s,\tau)) + N(\varpi(s,\tau)) + \wp(s,\tau)]d\tau \qquad (3.4)$$

The answer may be expressed as a power series in *p*, according to a Key statement of HPM technique:

$$\varpi(s,w) = \sum_{n=0}^{\infty} p^n \varpi_n(s,w), \qquad (3.5a)$$

$$\varpi(s,w) = \lim_{n\to\infty}\varpi(s,w,p) \qquad (3.5b)$$

Decomposing a nonlinear term yields

$$N\varpi(s,w) = \sum_{n=0}^{\infty} p^n \mathcal{H}_n(\varpi) \qquad (3.6)$$

where parameter for embedding is $p \in (0,1]$. $\mathcal{H}_n(\varpi)$ is He's polynomial & is created through

$$\mathcal{H}_n(\varpi_0,....,\varpi_{n)}) = \frac{1}{n!}\frac{\partial^n}{\partial p^n}\left[N\left(\sum_{n=0}^{\infty} p^j \varpi_j(s,w)\right)\right],$$

$$n = 0,1,2..... \qquad (3.7)$$

HDM is achieved via an elegant mixture of homotopy method & Abel integral.

$$\sum_{n=0}^{\infty} p^n \varpi_n(s,w) - G(s,w) = \frac{p}{\Gamma(\alpha)}\int_0^w (w-\tau)^{\alpha-1}$$
$$\left[\wp(s,w) + L\left(\sum_{n=0}^{\infty} p^n \varpi_n(s,w)\right) + N\left(\sum_{n=0}^{\infty} p^n \varpi_n(s,w)\right)\right]d\tau. \qquad (3.8)$$

By comparing the components of similar degrees of p, solutions in various orders with the initial term can be obtained.

$$\varpi_0(s,w) = G(s,w). \qquad (3.9)$$

Theorem ([7]) Assume that F: U→V is a contraction of nonlinear mapping and that U and V are Banach spaces. If the sequence produced by the HDM technique is considered

$$\varpi_n(s,w) = F(\varpi_{n-1}(s,w)) = \sum_{i=0}^{n-1}\varpi_i(s,w),$$

n=1,2,3....., If so, the following is true:

(1) $\|\varpi_n(s,w) - \varpi(s,w)\| \le \varphi^n \|G(s,w) - \varpi(s,w)\|;$

(2) $\varpi_n(s,w)$ is forever in neighborhood of $\varpi(s,w)$ meaning

$\varpi_n(s,w) \in B(\varpi(s,w),r) = \{\varpi^*(s,w) / \|\varpi^*(s,w) - \varpi(s,w)\|\};$

(3) $\lim_{n\to\infty}\varpi_n(s,w) = \varpi(s,w).$

Proof: (1) By induction on n, $\|\varpi_1 - \varpi\| = \|H(\varpi_0) - \varpi\|$, & based on Banach fixed point theorem, *F* has a fixed point ϖ indicating $F(\varpi) = \varpi$; for that reason,

$$\|\varpi_1 - \varpi\| = \|H(\varpi_0) - \varpi\| = \|H(\varpi_0) - H(\varpi)\|$$
$$\le \varphi\|\varpi_0 - \varpi\| = \varphi\|G(s,w) - \varpi\|$$

because *F* is contraction mapping.

Suppose that $\|\varpi_{n-1} - \varpi\| \le \varphi^{n-1}\|G(s,0) - \varpi(s,w)\|$ is an induction hypothesis, then

$$\|\varpi_n - \varpi\| = \|H(\varpi_{n-1}) - H(\varpi)\| \le \varphi\|\varpi_{n-1} - \varpi\| \le \varphi\varphi^{n-1}\|G(s,w) - \varpi\|.$$

(2) First demonstrate that $G(s,w) \in B(\varpi(s,w), r)$, and this is find by induction on m. So, for $m = 1$,

$\|G(s,w) - \varpi(s,w)\| = \|\varpi(s,0) - \varpi(s,w)\| \le r$ with $\varpi(s,0)$ initial condition.

Suppose that $\|G(s,w) - \varpi(s,w)\| \le r$ for m-2 is induction hypothesis, then

$$\|G(s,w) - \varpi(s,w)\| = \left\| G_{m-2}(s,w) - \frac{\wp_m(s)}{\Gamma(\alpha - m - 1)} w^{\alpha - m} \right\|$$

$$\le \|G_{m-1}(s,w) - \varpi(s,w)\| + \left\| \frac{\wp_m(s)}{\Gamma(\alpha, m, 1)} w^{m-1} \right\| = r.$$

Now, utilizing (1), we have for all $n \ge 1$

$$\|\varpi_n - \varpi\| \le \varphi^n \|G(s,w) - \varpi\| \le \varphi^n r \le r.$$

(3) By (2) & the fact $\lim_{n\to\infty} \varphi^n = 0$ produces that $\lim_{n\to\infty}\|\varpi_n - \varpi\| = 0$; therefore,

$$\lim_{n\to\infty} \varpi_n = \varpi.$$

Ⅳ. TIME-FRACTIONAL MALIGNANT TUMOR GROWTH MATHEMATICAL MODEL

Malignant cells' net death rate is solely time-dependent as in [5] & model comes by time-fractional differential equation

$$\frac{\partial^\alpha \wp(s,w)}{\partial w^\alpha} = \frac{\partial^2 \wp(s,w)}{\partial s^2} - w^2 f(s,w) \tag{4.1}$$

where $w \ge 0$, $0 \le s \le 1$, $0 < \alpha \le 1$, with initial condition as

$$\wp(s,0) = e^{rs} \tag{4.2}$$

where $\wp(s,w)$ is tumor cell density at position s at time w & r is dependent on therapy death rate at time w.

Ⅴ. ANALYTICAL SOLUTION

The following problem in "(4.1)" can be found using the HDM technique.

$$\sum_{n=0}^{\infty} p^n \wp_n(s,w) - \wp(s,0) = \frac{p}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^2}{\partial s^2}\left(\sum_{n=0}^{\infty} p^n \wp_n(s,\tau) \right) - w^2 \sum_{n=0}^{\infty} p^n \wp_n(s,\tau) \right) d\tau \tag{5.1}$$

At this time, compare the coefficient of the similar order of p, integral equations receive:

$$p^0 : \wp_0(s,w) = \wp(s,0) = e^{rs}, \ \wp_0(s,0) = \wp(s,0) \tag{5.2}$$

$$p^1 : \wp_1(s,w) = \frac{1}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^2 \wp_0}{\partial s^2} - w^2 \wp_0 \right) d\tau,$$

$$\wp_1(u,0) = 0, \tag{5.3}$$

$$p^2 : \wp_2(u,v) = \frac{1}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^2 \wp_1}{\partial s^2} - w^2 \wp_1 \right) d\tau,$$

$$\wp_2(u,0) = 0, \tag{5.4}$$

$$p^3 : \wp_3(s,w) = \frac{1}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^3 \wp_2}{\partial s^3} - w^2 \wp_2 \right) d\tau,$$

$$\wp_3(u,0) = 0, \tag{5.5}$$

$$p^4 : \wp_4(s,w) = \frac{1}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^4 \wp_3}{\partial s^4} - w^2 \wp_3 \right) d\tau,$$

$$\wp_4(s,0) = 0, \tag{5.6}$$

$$\vdots$$

$$p^n : \wp_n(u,v) = \frac{1}{\Gamma(\alpha)} \int_0^w (w-\tau)^{\alpha-1} \left(\frac{\partial^2 \wp_{n-1}}{\partial s^2} - w^2 \wp_{n-1} \right) d\tau,$$

$$\wp_n(s,0) = 0. \tag{5.7}$$

The following solution are obtained:

$$\wp_0(s,w) = e^{rs}, \tag{5.8}$$

$$\wp_1(s,w) = \frac{e^{rs}(r^2 - w^2)}{\Gamma(1+\alpha)} w^{\alpha}, \tag{5.9}$$

$$\wp_2(s,w) = \frac{e^{rs}(r^2 - w^2)^2}{\Gamma(1+2\alpha)} w^{2\alpha}, \tag{5.10}$$

$$\wp_3(s,w) = \frac{e^{rs}(r^2 - w^2)^3}{\Gamma(1+3\alpha)} w^{3\alpha}, \tag{5.11}$$

$$\wp_4(s,w) = \frac{e^{rs}(r^2 - w^2)^4}{\Gamma(1+4\alpha)} w^{4\alpha}, \tag{5.12}$$

$$\vdots$$

$$\wp_n(s,w) = \frac{e^{rs}(r^2 - w^2)^n}{\Gamma(1+n\alpha)} w^{n\alpha}. \tag{5.13}$$

Using the software mathematica, then find all the terms of the above integral equations. Several components of the series solutions are taken into consideration in this case, & approximate result is provided-

$$\wp(s,w) = \wp_0(s,w) + \wp_1(s,w) + \wp_2(s,w) + \wp_3(s,w) + \wp_4(s,w) + \ldots.. \tag{5.14}$$

$$\wp(s,w) = e^{rs} + \frac{e^{rs}(r^2 - w^2)}{\Gamma(1+\alpha)} w^{\alpha} + \frac{e^{rs}(r^2 - w^2)^2}{\Gamma(1+2\alpha)} w^{2\alpha} + \frac{e^{rs}(r^2 - w^2)^3}{\Gamma(1+3\alpha)} w^{3\alpha} + \frac{e^{rs}(r^2 - w^2)^4}{\Gamma(1+4\alpha)} w^{4\alpha} + \ldots\ldots\ldots \tag{5.15}$$

$$\wp(s,w)=e^{rs}\begin{pmatrix}1+\dfrac{(r^2-w^2)}{\Gamma(1+\alpha)}w^{\alpha}+\dfrac{(r^2-w^2)^2}{\Gamma(1+2\alpha)}w^{2\alpha}+\\ \dfrac{(r^2-w^2)^3}{\Gamma(1+3\alpha)}w^{3\alpha}+\dfrac{(r^2-w^2)^4}{\Gamma(1+4\alpha)}w^{4\alpha}+.........\end{pmatrix}$$

$$=e^{rs}\sum_{n=0}^{\infty}\frac{(r^2-w^2)^n}{\Gamma(1+n\alpha)}w^{n\alpha}. \qquad (5.16)$$

Using "(5.16)" in

$$\wp(s,w)=\sum_{n=0}^{\infty}\wp_n(s,w) \qquad (5.17)$$

In particular as $\alpha\longrightarrow 1$

$$\wp(s,w)=e^{su}\begin{pmatrix}1+\dfrac{(r^2-w^2)}{1!}w+\dfrac{(r^2-w^2)^2}{2!}w^{2}+\\ \dfrac{(r^2-w^2)^3}{3!}w^{3}+\dfrac{(r^2-w^2)^4}{4!}w^{4}+........\end{pmatrix}$$

$$=e^{rs}e^{(r^2-w^2)w}, \qquad (5.18)$$

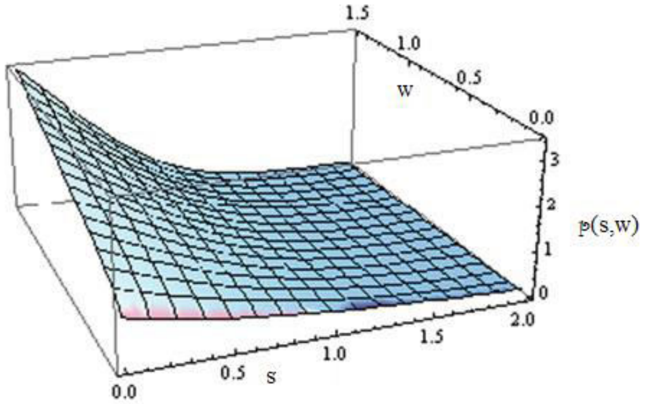
$$\wp(s,w)=e^{-w^3+wr^2+rs} \qquad (5.19)$$

Researchers found an accurate solution to the problem by using the homotopy decomposition method, whereas Iyiolaa and Zaman et al. [5] used the homotopy analysis method and found an approximate analytical solution to the problem. The HDM is a useful method for dealing with non-linear PDEs, as we have seen.

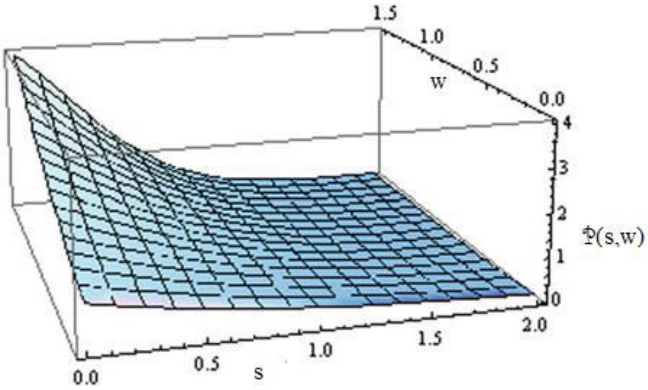
Ⅵ. TIME-FRACTIONAL MALIGNANT TUMOR GROWTH MODEL GRAPHICAL SOLUTION

Finally, using the analytical solution "(5.16)", some graphical results were produced for various value of the fractional parameter α , that lies between 0 and 1 with parameters $r=-1,0\le u\le 2,0\le v\le 1.5$.

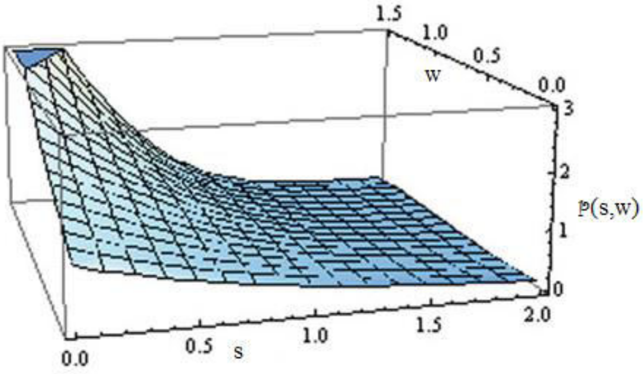
Outcome demonstrates that under the initial conditions and the chosen parameters, the death rate *r* caused the tumor cells quantity to drop over time & eventually approach zero. When we modified value for α , it affected the convergence resolution using Figs 1, 2, 3, 4, 5, 6 changed when we changed the value $\alpha=0.5,0.8,0.9,0.95,0.98,1$.



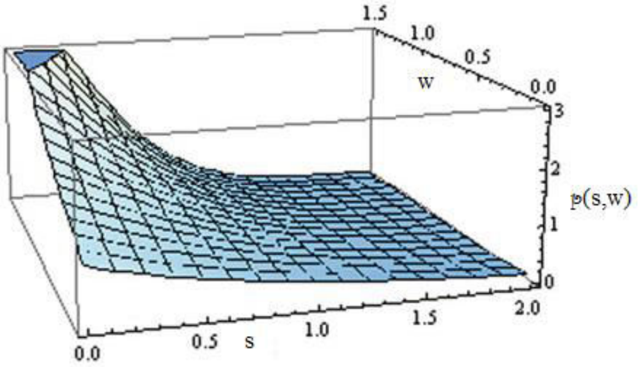
$Fig.1\ r=-1,\alpha=0.5,0\le s\le 2,0\le w\le 1.5$



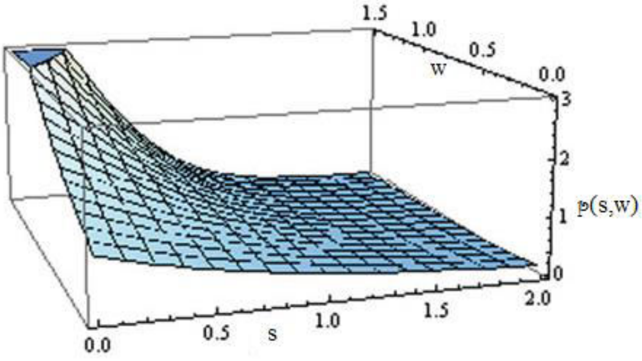
$Fig.2\ r=-1,\alpha=0.8,0\le s\le 2,0\le w\le 1.5$



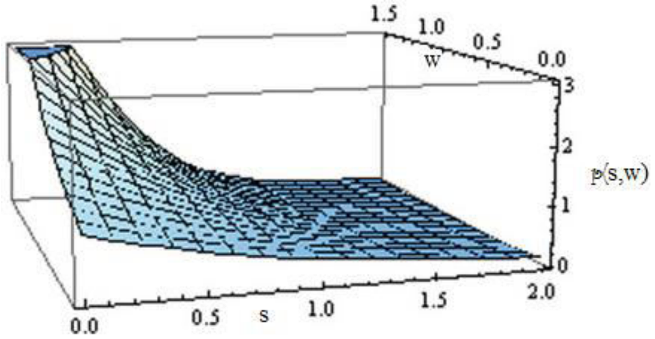
$Fig.3\quad r=-1,\alpha=0.9,0\le s\le 2,0\le w\le 1.5$



$Fig.4\qquad r=-1,\alpha=0.95,0\le s\le 2,0\le w\le 1.5$



$Fig.5\quad r=-1,\alpha=0.98,0\le s\le 2,0\le w\le 1.5$



$Fig.6 \quad r = -1, \alpha = 1, 0 \leq s \leq 2, 0 \leq w \leq 1.5$

Ⅶ. CONCLUSION

We observed that HDM is a highly useful mathematical tool that may be used to determine an exact or very close to precise solution to a model of time-fractional malignant tumor growth. In malignant tumor growth model, when net death rate of the tumor cells is solely time-dependent, we effectively employed it to produce the precise solution. We also produced other convergent solutions using a fractional parameter α . We discussed the fractional derivative concept's brief history and some of its features.

References

[1] Atangana A, Alabaraoye E. Solving a system of fractional partial differential equations arising in the model of HIV infection of CD4+ cells and attractor one-dimensional Keller-Segel equations. Advances in Difference Equations. 2013 Dec;2013(1):1-4.

[2] Gandhi H, Tomar A, Singh D. A predicted mathematical cancer tumor growth model of brain and its analytical solution by reduced differential transform method. In Proceedings of International Conference on Trends in Computational and Cognitive Engineering 2021 (pp. 203-213). Springer, Singapore.

[3] Saadeh R, Qazza A, Amawi K. A New Approach Using Integral Transfarm to Solve Cancer Models. Fractal and Fractional. 2022 Aug 31;6(9):490

[4] Korpinar Z, Inc M, Hınçal E, Baleanu D. Residual power series algorithm for fractional cancer tumor models. Alexandria Engineering Journal. 2020 Jun 1;59(3):1405-12.

[5] Iyiola OS, Zaman FD. A fractional diffusion equation model for cancer tumor. AIP Advances. 2014 Oct 13;4(10):107121.

[6] Mishra J. A study on the spread of COVID 19 outbreak by using mathematical modeling. Results in Physics. 2020 Dec 1;19:103605.

[7] Atangana A, Secer A. The time-fractional coupled-Korteweg-de-Vries equations. InAbstract and Applied Analysis 2013 Jan 1 (Vol. 2013). Hindawi.

[8] Burgess PK, Kulesa PM, Murray JD, Alvord Jr EC. The interaction of growth rates and diffusion coefficients in a three-dimensional mathematical model of gliomas. Journal of Neuropathology & Experimental Neurology. 1997 Jun 1;56(6):704-13.

[9] Moyo S, Leach PG. Symmetry methods applied to a mathematical model of a tumour of the brain. InProceedings of Institute of Mathematics of NAS of Ukraine 2004 (Vol. 50, No. Part 1, pp. 204-210).

[10] Ali SM, Bokhari AH, Yousuf M, Zaman FD. A spherically symmetric model for the tumor growth. Journal of Applied Mathematics. 2014 Jan 1;2014.

[11] Bokhari AH, Kara AH, Zaman FD. On the solutions and conservation laws of the model for tumor growth in the brain. Journal of Mathematical analysis and Applications. 2009 Feb 1;350(1):256-61.

[12] El-Ajou A, Arqub OA, Zhour ZA, Momani S. New results on fractional power series: theories and applications. Entropy. 2013 Dec 2;15(12):5305-23.

[13] El-Ajou A, Arqub OA, Momani S. Approximate analytical solution of the nonlinear fractional KdV–Burgers equation: a new iterative algorithm. Journal of Computational Physics. 2015 Jul 15;293:81-95.

[14] Khater MM, Attia RA, Lu D. Numerical solutions of nonlinear fractional Wu–Zhang system for water surface versus three approximate schemes. Journal of Ocean Engineering and Science. 2019 Jun 1;4(2):144-8.

[15] Atangana A, Baleanu D. Nonlinear fractional Jaulent-Miodek and Whitham-Broer-Kaup equations within Sumudu transform. InAbstract and applied analysis 2013 Jan 1 (Vol. 2013). Hindawi.

[16] Chen X, Zheng L, Zhang X. Convergence of the homotopy decomposition method for solving nonlinear equations. Advances in Dynamical Systems and Applications. 2007;2(1):59-64.

[17] Mishra J, Atangana A. Numerical analysis of a chaotic model with fractional differential operators: from Caputo to Atangana–Baleanu. InMethods of Mathematical Modelling 2019 Sep 17 (pp. 167-188). CRC Press.

[18] Atangana A, Bulutv H. EXTENSION OF HOMOTOPY DECOMPOSITION METHOD (HDM) TO COUPLED NONLINEAR VAN DER POL TYPE'S EQUATION. Acta Marisiensis. Seria Technologica. 2013 Jul 1;10(2):54.

[19] Mishra J. Fractional hyper-chaotic model with no equilibrium. Chaos, Solitons & Fractals. 2018 Nov 1;116:43-53.

[20] Saadeh R, Qazza A, Amawi K. A New Approach Using Integral Transform to Solve Cancer Models. Fractal and Fractional. 2022 Aug 31;6(9):490.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 213–218 DOI: 10.15439/2022R22 ISSN 2300-5963 ACSIS, Vol. 33

Energy-Efficient Distributed Cluster-Tree Based Routing Protocol for Applications IoT-Based WSN

Nguyen Duy Tan
Faculty of Information Technology, Hung Yen
University of Technology and Education,
Hungyen, Vietnam
tanndhyvn@email.com

Hong-Nhat Hoang
School of Computer Science and Engineering
Pusan National University Busan,
Republic of Korea
nhat.hqh@gmail.com

***Abstract*—With the growth of wireless sensor network (WSN) technologies, the applications of IoT-based WSNs allow the interconnection of smart objects or sensors through the Internet. However, energy constraint is a major obstacle in WSN, which directly affects the lifespan of the network. Hence, many researchers have focused on how to program routing protocols to maximize energy conservation in WSNs. The clustering mechanism is demonstrated that separating the network into clusters may significantly decrease energy consumption. In this paper, we propose distributed tree-based clustering routing protocol for IoT applications (EE-DTC). In order to enhance efficient energy, EE-DTC chooses cluster head nodes based on the remaining energy, the location, and the density of nodes. In addition, to lengthen the network lifespan, we create multi-hop routes with short communication links intra-clusters by building the minimum spanning tree (MST) using the Kruskal algorithm. Our experiment results exhibit that the performance of EE-DTC overcomes the TBC and LEACH-VA protocols in terms of increasing network lifespan, reducing energy consumption, and improving efficient energy.**

***Index Terms*—Wireless sensor networks, routing protocol, energy-efficient, IoT, tree-based clustering.**

I. Introduction

In 1999, Kevin Ashton claims that the idea of the "Internet of Things" (IoT) was launched by enabling connectivity among various things at any time and location via the Internet [1]. A wireless sensor network (WSN) comprises a set of smart sensor devices and a gateway or base station (BS) device that can communicate with each other using radio channels. WSNs are profitable in IoT applications for assembling and processing data before transferring it to the gateway node or end-user [2]. IoT-based WSN will build a network system for connecting, calculating, and sharing information in order to create applications in human life, for example, early earthquake warning systems, battlefield monitoring, weather forecast, intelligent traffic system, smart cities, smart agriculture, environment monitoring, remote patient monitoring, healthcare and so on [3, 4, 5, 6]. Figure 1 exhibits that an application of IoT-based WSN for the early detection system of forest fire is deployed in the jungle to keep track of the environment, in which smart sensors (IoT devices) are used to measure the environmental temperature or humidity [7]. The sensed data from these sensors will be processed and sent to the computers of staff in the office or at home through mobile devices or the Internet. From this information, users can give decisions in real-time to solve the problem that occurred.

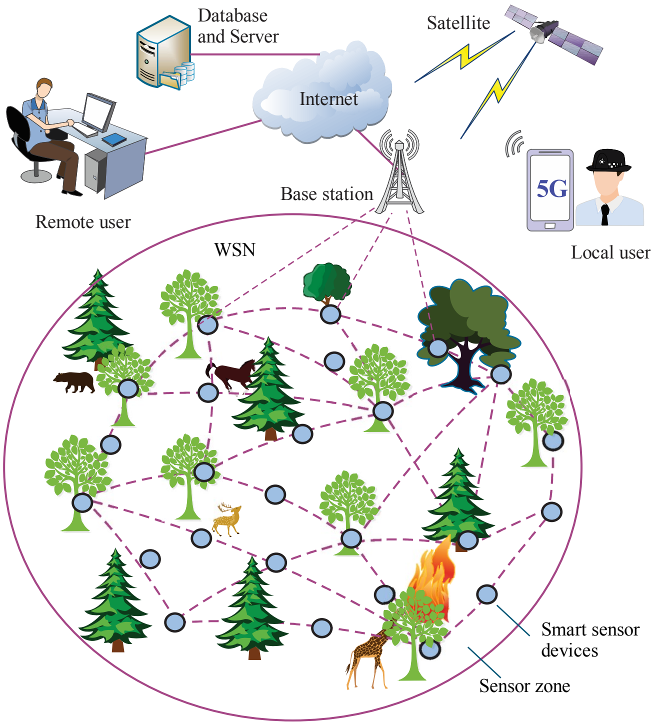


Figure 1. An application of IoT-based WSN for early detection of forest fire

However, smart sensing nodes are tightened in terms of energy battery and it is too difficult to recharge or substitute a new battery because the nodes are often deployed in the harsh zone that humans cannot reach [8]. Therefore, the greatest challenge in applications IoT-based WSN is how to reduce the energy dissipation of smart sensor devices, as a result, prolong the lifespan of the network, which should be considered during the designing of routing protocols. To attain this goal, numerous researchers have proposed efficient energy routing schemes for WSN, in which the clustering mechanism is one of the most useful methods [9, 24, 25]. According to the clustering method, the network system is partitioned into various clusters; each cluster involves one cluster head node (CHN) and some cluster members (CMs) [10]. CMs only sense the external environment and transmit their acquired data to their CHN, whereas a CHN is responsible for receiving and aggregating data and then relaying the assembled data packets to the BS [19].

The top popular routing protocol inspired by clustering is the low-energy adaptive clustering hierarchy (LEACH) [11] which applies the distributed scheme approach for constructing clusters and CHNs election. According to the simulation results, the LEACH protocol drastically reduces the overall energy consumed in the network but it still has some limitations that need to be improved. For example, selecting CHNs at random locations can lead to both a far distance from the CHNs to the BS and an unequal cluster distribution because the CHNs may be too close together. Recently, there have been many improvements to the LEACH protocol to enhance the network lifetime for applications of IoT-based WSNs. One of them is I-LEACH [12], in which CHNs are selected based on the threshold value of the remaining energy and energy cost for the receiving, aggregating, and transmitting of the nominee CHN in the current round. However, the links among CMs and their CHN are still far due to one-hop communication and as the result, the sensor nodes still consume a lot of energy in data communication.

Consequently, we propose a distributed tree-based clustering routing protocol for IoT applications namely EE-DTC, in which nodes are self-configuring and form a network base on LEACH. EE-DTC builds multi-hop routes intra-cluster to reduce the energy depletion of sensing nodes by using Kruskal. Furthermore, the considerations of the remaining energy and the distance from nominee CHNs to the BS as the main criteria in CHNs selection also help to improve energy efficiency. Our simulation outcomes depict that EE-DTC can be achieved better energy efficiency than TBC and LEACH-VA about 15% and 21% in comparison to TBC and LEACH-VA, respectively. The rest of this study is organized as follows: The related work presented in Section 2, Sections III, and IV is described the network model and detail of EE-DTC. In Section V, we analyze and discuss the simulation outcomes. Finally, the conclusion of the study is presented in Section VI.

II. Related Works

Many algorithms have suggested preserving the precious energy of sensor nodes and this section will review some of these algorithms.

Heinzelman et al. [11], proposed a routing algorithm inspired by a clustering scheme for WSNs called LEACH. LEACH employs a algorithm distribution to group sensor nodes into clusters by self-organizing. The working of LEACH is segmented into several rounds, each round has two phases: the formation cluster phase and the data transferring phase. In the clustering phase, some CHNs are randomly selected from all alive nodes. A node creates randomly a number between 0 and 1 for determining whether to become CHN or not. If this number is smaller than the probability value *T(n)* indicated by Equation (1), the node will elect itself as the CHN of the current round [11].

$$T(n)=\begin{cases}\dfrac{k}{1-k\left(r \bmod \dfrac{1}{k}\right)}, & if\, n\in G\\ 0, \quad otherwise & (n\notin G)\end{cases} \qquad (1)$$

where k is the percentage of CHNs in the network ($k \approx 5\%$ [13]), r indicates the currently considering round and G is a list of sensor nodes that are not elected as CHN in the last ($1/k$) round. After selecting CHNs, nodes participate in the corresponding cluster depending on the strong signal of the message transmitted to them by the CHNs.

In the data transferring phase, sensor nodes congregate data and send it to their CHN in each cluster. CHN will be responsible for aggregating data and forwarding it to the BS device. The LEACH protocol significantly reduces energy consumption by organizing the network into clusters and fusing data before delivering them to the BS device. However, network nodes still consume lots of energy for inside cluster communication since the distances between nodes and CHN are still far away. To solve this problem, Kim et al. [14] presented a novel clustering based on tree approach for efficient energy in WSNs based on LEACH called (TBC). According to this proposal, the nodes in each cluster will construct a tree with CHN as a root based on their distance information. The simulation results illustrate that TBC can reduce the energy dissipation required intra-communication and significantly strengthen the lifespan of the network in comparing to the three existing protocols which are LEACH, PEGASIS [15], and TREEPSI [16]. However, in both LEACH and TBC protocol, the remaining energy of the nominee CHNs is not considered in electing itself to become leader cluster node, so some CHNs may early run out of energy, therefore, the collecting data of CMNs cannot send to the BS device. To defeat this drawback, Xu et al. [17] have introduced an improvement of the LEACH algorithm, namely E-LEACH, in which the remaining energy of the nominee CHNs is considered in order to balance the energy consumption load. In addition, E-LEACH employs the MST algorithm to form routes for multi-hop communication among CHNs and BS devices. As a result, E-LEACH prolongs the network lifespan and enhances energy. The threshold value *T(n)* of E-LEACH for the nominee CHNs to become CHN is expressed as follows [17]:

$$T(n)=\begin{cases}\dfrac{k}{1-k\left(r \bmod \dfrac{1}{k}\right)}\times\dfrac{E_{curent}}{E_{init}}, & if\, n\in G\\ 0, \quad otherwise & (n\notin G)\end{cases} \qquad (2)$$

where, E_{curent} and $E_{initial}$ are the residual energy and initial of the nominee CHN, respectively.

Liang et al. [18] proposed an improved LEACH called LEACH-VA, where the whole network monitoring zone is separated into unequal polygons based on Voronoi Diagram. Firstly, the optimal quantity of CHNs is selected according to the total energy consumed in each round to diminish the cost of distributing excessiveness clusters per round. Then, the nodes in the same polygon are grouped in a cluster with the CHN at the center of the polygon. Finally, CHNs are joined by employing an ant colony algorithm to acquire the multi-hop routing optimization from CHNs to the BS device. According to the experimental results, LEACH-VA significantly decreases the energy utilization in data transmission and prolongs the lifespan of the network.

Osamy et al. [19] proposed a clustering routing algorithm based on entropy information theory for heterogeneous WSNs, namely EBCS. EBCS organizes the network into three layers: Cluster heads selection layer, cluster formation, and data transmission layer. Especially, EBCS combines multiple parameters in the selection nodes working as CHN based on entropy schemes such as the distance from the node to the gateway device, residual energy, and the distribution of the density of the node in the monitoring field. The experiment results exhibit that EBCS outperforms LEACH in terms of the throughput and lifetime of the network. Hau et al. [21] introduced the improvement of the AODV routing protocol by using intelligent agents for efficient energy. Tan et al. [22] proposed a sector tree routing protocol based on a clustering technique for efficient energy, in which the nodes in a cluster communicate with each other based on the MST connecting. Gamal et al. [10] have proposed hybrid particle swarm optimization and a K-means clustering hierarchy algorithm to construct clusters to enhance the lifespan of WSNs. According to this proposal, the Fuzzy logic theory is used to choose CHN with two criteria: residual energy level and distance from CHNs to BS. Moreover, the authors have integrated the K-means clustering and particle swarm optimization algorithm to construct clusters. Khoshrangbaf et al. [26] proposed an optimal ant colony algorithm to advance the coverage region of WSNs and maintain the connectivity between the nodes.

III. Proposed Method

A. Network Model

In our network model, we suppose that a network system involves a gateway or base station device and lots of sensor nodes that can support IoT applications [20].

- The gateway device is fixed and not limited to energy and computation capacity, which can aggregate and forward the data packets.
- N sensor nodes are random implementations in the network zone of two-dimensional.
- All sensor nodes are stationary state after deployment and know the location of each other based on GPS
- All sensing nodes use the battery energy and cannot recharge or replace
- The radio channels are symmetry

B. Energy Consuming Model

The radio energy consuming model employed in our proposal to transmit the *b-bit* data packet ETX [11, 19, 23]:

$$E_{TX}(b,d)=\begin{cases} b\times E_{elec}+b\times E_{friis}\times d^2 & ,if\ d<d_0 \\ b\times E_{elec}+b\times E_{tworay}\times d^4 & ,if\ d\geq d_0 \end{cases} \quad (3)$$

where E_{elec} is the energy consumption part for the electric circuit, E_{towray} and E_{friis} are the energy consuming for the transmitter amplifier by two ray ground mode and free space mode with the distance d, corresponding, and d_0 is the threshold distance in which sensor nodes use the free space model. The geographical distance between nodes a and b is calculated as Equation (5) below:

$$d(a,b)=\sqrt{(x_a-x_b)^2+(y_a-y_b)^2} \quad (4)$$

The energy consumed to receive a packet is calculated below:

$$E_{RX}=bE_{elec} \quad (5)$$

IV. Distributed Tree-Based Clustering Routing

The operation of EE-DTC is partitioned into numerous rounds; each round comprises B three stages: Cluster formation, distributed tree construction, and steady stage which are described as several steps like the following:

Step 1: Start a new round, check, if the sensor node is dead, go to **Step 11**, otherwise, nodes calculate their own threshold value *T(i)* as Equation (6) and randomly generate a number *R* in the scope from 0 to 1. If *R* is indicated smaller than the threshold value *T(i)*, the node is selected as CHN, otherwise, the node is CMN.

Step 2: If the node is CHN then broadcast the *ADV_CH* message and go to **Step 5**.

Step 3: If the node is CMN, receives the message *ADV_CH* and calculates the fitness function as Equation (7).

Step 4: CMNs send *JOIN_Adv* messages that contain the identifier, residual energy, and its position to CHN corresponding to the CHN whose fitness function value is the highest, and goes to **Step 9**.

Step 5: CHNs receive *JOIN_Adv* messages from CMNs and connect nodes in their cluster by constructing the MST with the CHN as a root

Step 6: CHNs transmit the information about the tree and the slot time for communication (TDMA) to CMNs in the cluster, go to **Step 8**.

Step 7: CMNs receive formation about trees and TDMA

Step 8: Every node periodically collects environmental information and transmits sensed data packets to their parent or CHN along the tree

Step 9: CHNs receive information from CMN nodes, aggregate it with their own data, and forward it to the BS device.

Step 10: If the round time is expired, go to **Step 1**, otherwise go to **Step 9.**

Step 11: The end.

A. Cluster Formation

In this study, we use the criteria parameters such as residual energy, initial energy, the distribution density of nodes, and the distance among nodes and BS device to make the threshold value for electing itself to become CHN as described in Equation (3) below:

$$T(i)=\begin{cases} \dfrac{k}{1-k\left(r \bmod \dfrac{1}{k}\right)}\times\left(\dfrac{c_1*N_i*E_{curent}}{d_{\max}}+\dfrac{c_2*E_{init}}{d(i,BS)}\right), & if\ n\in G \\ 0, \quad otherwise & (n\notin G) \end{cases} \quad (6)$$

where N_i is the number of neighbor nodes of node i-th, $d_{max,}$ and *d(i, BS)* is the network diameter and distance from the nominee CHN *i-th* to BS, respectively. The constant values c_1 and c_2 are the weights for the parameters and $c_1+c_2=1$.

This threshold *T(i)* ensures that every node elected to become CHN has a high energy level and is as near the BS as possible. After the node selects itself as a CHN, it broadcasts an advertising (*ADV_CH*) message to other nodes. When a CMN *i-th* received an *ADV_CH* message from CHN, it will compute the fitness function value as Equation (7) and transmit the join-request (*JOIN_REQ*) message, which contains the identification of node and position information to the corresponding CHN whose the *ff()* function value is maximum.

$$ff\left(i,CH_j\right)=\frac{E\left(CH_j\right)_{residual}}{d\left(BS,CH_j\right)+d\left(i,CH_j\right)} \qquad (7)$$

where $E(CH_j)_{residual}$ is the remaining energy of CHN *j-th*.

B. Distribute Tree Construction

Unlike LEACH, the proposed protocol employs a multi-hop communication scheme intra-cluster by constructing a tree for each cluster. To do this, we assume that each cluster in WSN is considered as a graph directionless *G(V, E, D)*, in which *V* contains a list of smart sensor nodes, *E* denotes a list of links for communicating of the sensor nodes and *D* describes the set of distances value on *E*, respectively. For routing discovery intra-cluster, each CHN will construct a MST in their cluster with the CHN as root inspired by Kruskal, which is depicted as Algorithm 1.

Algorithm 1: Distributed Tree Formation

Input: *V: a list of sensor nodes in a cluster*
E: a list of links in a cluster
D: a list of the distance value
Output: *Tree with CHN as a root*

```
count = 0;
ei =1;
TREE = TREE +{CHN};
for i = 1 to sizeof(E) do
    E[i].state = NOT_SELECT;
end for
Sort the set of edges in E increases the distance value
while (ei < sizeof (E) - 1) do
    Choose ei in E with E[ei].state = NOT_SELECT;
    u = get edge of (E[ei].u);
    v = get edge of (E[ei].v);
    if (v and u are two various trees) then
        Join(E[ei].v, E[ei].u) into tree;
        E[ei].state = SELECTED;
        count ++;
    if (count = sizeof(V)) then break;
    end if
    ei++;
end while
for i =1 to sizeof (E) do
    if (E[i].state = SELECTED) then
        Combine {E[i].v, E[i].u} into TREE;
    end if
end for
Create time slots for all nodes on the tree based on
    TDMA mechanism
Broadcast the time slots and TREE to nodes in the
    cluster
return {TREE};
```

C. Data Transmission

After forming routes for communication, sensor nodes start observing and transmitting sensed data to its parent node toward CHN on the tree. Firstly, the leaf nodes will communicate with their parent node on the tree. The parent nodes accept the data and combine it with their own acquired data, and send it towards the upper-level of the tree. Whenever the CHN node receives the data from overall living CMNs, it will also aggregate and forward it to the BS devices. After a period of time, the new round will be restarted by reelecting CHN as well as reconstructing MSTs within clusters for a new round.

V. Performance Evaluation and Discussion

The performance of EE-DTC is evaluated and compared to TBC [14] and LEACH-VA [18] based on ns2 [23] in this section.

A. Scenario and Simulation Parameters

The scenario is exhibited in Figure 2 and the parameters utilized in the simulation scenario are presented in Table 1 [11, 20, 22, 23, 24].

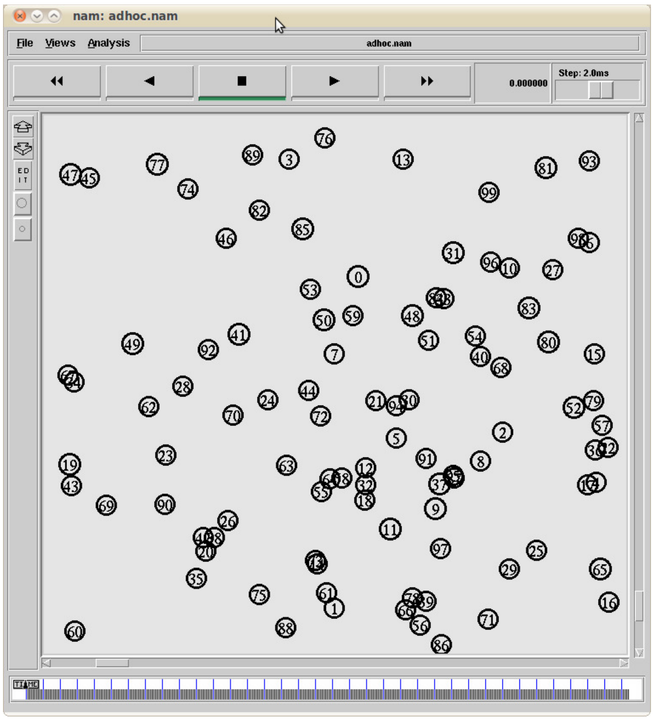


Figure 2. A simulation scenario configured in ns2 with random distributed 100 sensor nodes

TABLE I. The simulation Parameters

No. Item	Parameters explanation	Value
1	Size of simulation field	100m ×100m
2	Number of sensor nodes (*N*)	100 nodes
3	Energy consumption: two-ray ground model (E_{towray})	10 pJ/bit/m^2
4	Energy consumption: free space model (E_{friis})	0.0013pJ/bit/m^4
5	Energy consumption: Electric circuit (E_{elec})	50 nJ/bit
6	Energy consumption: Data fusion (E_{DF})	5 nJ/bit/packet
7	The initial energy of node (E_{init})	2J
8	Packet size	1024 bytes
9	Simulation time	3600s
10	BS location	49,175

B. *Simulation results*

Figure 3 shows the simulation results of the total number of living nodes with increasing the lifetime of the network according to rounds. In Figure 3, we can see that the time of the first node death of both TBC and LEACH-VA protocols is approximately 600 rounds, while the first node death of EE-DTC protocol is 1000 rounds. So, our proposed protocol can improve the lifespan of the network by nearly 25% because the EE-DCT diminishes the communication distance among nodes intra-cluster by building distributed tree. In addition, the EE-DCT considers the residual energy level and the cost of data transmission as criteria parameters inside the threshold probability and fitness function for the selection CHNs and forming the cluster.

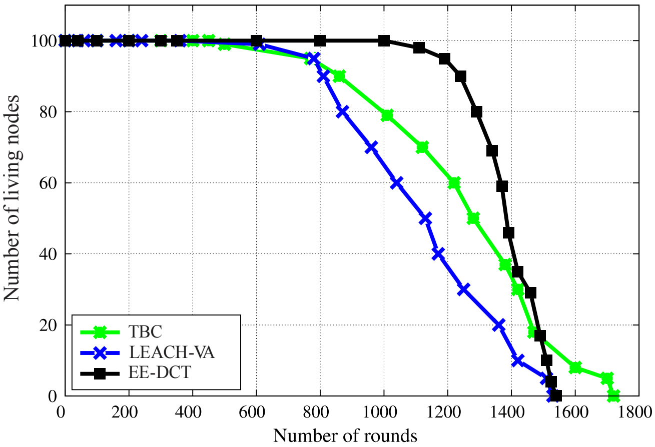


Figure 3. Number of living nodes per round

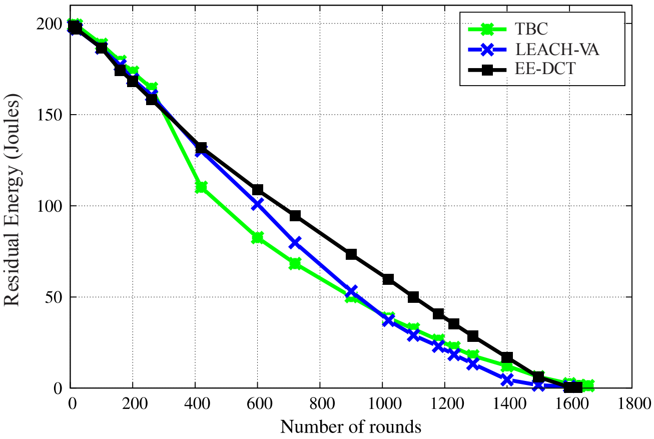


Figure 4. The overall residual energy of nodes per round

Figure 4 shows that the overall residual energy of sensor nodes reduces while the amount of rounds increases with TBC, LEACH-VA, and EE-DTC. It is clear that the residual energy level of sensor nodes running our proposed protocol is significantly higher than the other two protocols. This result is that EE-DTC organizes better clustering, is more suitable for the number of clusters and chooses the shorter data transmission routes in each cluster. In LEACH-VA, the acquired data transmitted from CMNs to CHN may be over-links long-distance which is the cause of more energy depletion during the data transmission stage.

In Figure 5, the efficient routing protocol is expressed depending on the throughput. The more throughputs accomplish, the better the routing protocol is. Figure 5 clearly shows that the throughput of the EE-DCT is higher than that of the two existing protocols. Specifically, the overall number of data bytes transmitted to the BS in LEACH-VA is limited to 120 × 10^6 bytes, in TBC is increased to 140 × 10^6 bytes, whereas EE-DTC rose to 160 × 10^6 bytes as determined in Figure 5.

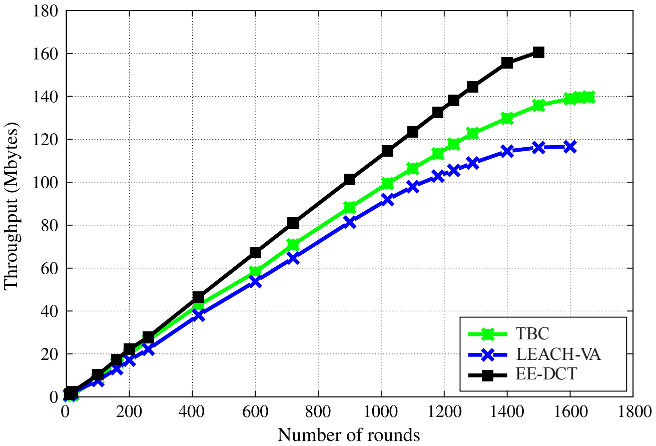


Figure 5. The throughput of the network per round

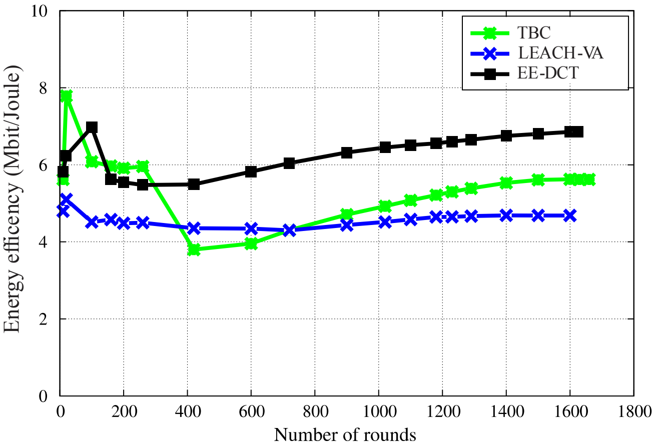


Figure 6. The energy efficiency of the network per round

In addition, Figure 6 illustrates the efficient energy of TBC, LEACH-VA, and EE-DTC with increasing the number of rounds. As can be observed in Figure 6, the TBC protocol has the highest energy efficiency in the first 100 rounds with approximately 8×10^6 bit per Joule and the shortest in the range of 400 to 500 at about 3.9×10^6 bit per Joule. Besides, the energy efficiency of the LEACH-VA protocol is again relatively stable at approximately 4.3×10^6 bit per Joule. Meanwhile, EE-DTC achieves efficient energy of approximately 6 ×10^6 bits per Joule. This is because EE-DTC uses short multi-hop communication links intra-cluster and better-balanced energy consumption than TBC and LEACH-VA, thus EE-DTC achieves better energy efficiency than TBC and LEACH-VA about 15% and 20% in comparison with TBC and LEACH-VA, respectively.

VI. Conclusions

In this study, we have presented distributed clustering routing based on tree scheme for applications of IoT-based WSN. Our proposal protocol not only lengthens the lifespan of the network but also balances energy dissipation among the CHN, and conserves the rare battery energy of the nodes in the cluster. To provide multi-hop routing intra-cluster, we built the distributed MST in each cluster with short links among nodes by using Kruskal. Furthermore, the selection of CHNs considers the remaining energy of the nominee CHN and the distance from them to the BS which also helps to improve energy efficiency. The proposal is experimented by using ns2 and compared to two existing protocols. The results show that EE-DTC achieves performance better than the TBC, and LEACH-VA in terms of the network lifespan, energy consumption, throughput, and efficient energy. Our future work of us is on how to apply lightweight data compression algorithms such as Klein and Gain in IoT applications to reduce the energy consumption of IoT devices.

Acknowledgment

This work was supported by Hung Yen University of Technology and Education under grant code UTEHY.L.2022.01.

References

[1] J. A. Afonso, R. A. Sousa, J. C. Ferreira, V. Monteiro, D. Pedrosa and J. L. Afonso, "IoT system for anytime/anywhere monitoring and control of vehicles' parameters," in IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). Bari, Italy, September 2017, pp. 193–198.

[2] K. Bajaj, B. Sharma, and R. Singh, Integration of WSN with IoT Applications: A Vision, Architecture, and Future Challenges. Cham: Springer International Publishing, 2020, pp. 79–102.

[3] A. Alphonsa, and G. Ravi., "Earthquake Early Warning System by IOT using Wireless Sensor Networks," in International Conference on Wireless Communications, Signal Processing and Networking (WISPNET). India, March 2016, pp. 1201–1205.

[4] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, and J. C.-W. Lin, "Applications of Wireless Sensor Networks and Internet of Things Frameworks in the Industry Revolution 4.0: A Systematic Literature Review," Sensors, vol. 22, no. 6, pp. 1–36, 2022.

[5] K. Haseeb, I. U. Din, A. Almogren, and N. Islam, "An Energy Efficient and Secure IoT-Based WSN Framework: An Application to Smart Agriculture," Sensors, vol. 20, no. 7, pp. 1–14, 2020.

[6] J. J. P. C. Rodrigues, D. B. R. Segundo, H. A. Junqueira, M. H. Sabino, R. M. Prince, J. Al-Muhtadi, and V. H. C. Albuquerque, "Enabling Technologies for the Internet of Health Things," IEEE Access, vol. 6, pp. 13 129–13 141, 2018.

[7] S. Verma, S. Kaur, D. B. Rawat, C. Xi, L. T. Alex, and N. Z. Jhanjhi, "Intelligent Framework Using IoT-Based WSNs for Wildfire Detection," IEEE Access, vol. 9, pp. 48 185–48 195, 2021.

[8] Y. Gao, G. Kang, J. Cheng, and N. Zhang, "A new Energy Efficient Clustering Algorithm based on Routing Spanning Tree for Wireless Sensor Network," IEICE Transactions on Communications, vol. E100.B, pp. 1–12, 2017.

[9] A. Rady, E. L.-S. M. El-Rabaie, M. Shokair, and N. Abdel-Salam, "Comprehensive survey of routing protocols for Mobile Wireless Sensor Networks," International Journal of Communication Systems, vol. e4942, pp. 1–30, 2021.

[10] M. Gamal, N. E. Mekky, H. H. Soliman, and N. A. Hikal, "Enhancing the Lifetime of Wireless Sensor Networks Using Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization," IEEE Access, vol. 10, pp. 36 935–36 948, 2022.

[11] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Transactions on wireless communication, vol. 1, no. 4, pp. 660–670, 2002.

[12] T. M. Behera, U. C. Samal, and S. K. Mohapatra, "Energy-efficient modified LEACH protocol for IoT application," IET Wireless Sensor Systems, vol. 8, no. 6, pp.223–228, 2018.

[13] S. E. Pour and R. Javidan, "A new energy aware cluster head selection for LEACH in wireless sensor networks," IET Wireless Sensor Systems, vol. 11, no. 1, pp. 45–53, 2021.

[14] K. T. Kim, C. H. Lyu, S. S. Moon, and H. Y. Youn, "TreeBased Clustering (TBC) for Energy Efficient Wireless Sensor Networks," in IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA). Perth, Australia, April 2010, pp. 680–685.

[15] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information System," in IEEE Aerospace Conference Proceedings, March 2002, pp. 1125–1130.

[16] S. S. Satapathy and N. Sarma, "TREEPSI: TRee based Energy Efficient Protocol for Sensor Information," in IFIP International Conference on Wireless and Optical Communications Networks. India, April 2006, p. Siddhartha Sankar Satapathy and Nityananda Sarma.

[17] J. Xu, N. Jin, X. Lou, T. Peng, Q. Zhou, and Y. Chen, "Improvement of LEACH protocol for WSN," in The 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). Chongqing, China, May 2012, pp. 2174–2177.

[18] H. Liang, S. Yang, L. Li, and J. Gao, "Research on routing optimization of WSNs based on improved LEACH protocol," EURASIP Journal on Wireless Communications and Networking, vol. 194, pp. 1–12, 2019.

[19] W. Osamy, A. Salim, and A. M. Khedr, "An information entropy based-clustering algorithm for heterogeneous wireless sensor networks," Wireless Networks, vol. 26, p. 1869–1886, 2020.

[20] Tan, N, D.,& Viet, N, D. "SSTBC: Sleep Scheduled and Tree-Based Clustering Routing Protocol for Energy-Efficient in Wireless Sensor Networks", IEEE-RIVF International Conference on Computing and Communication Technologies, Cantho, Vietnam, 2015, pp. 180-185.

[21] Van-Hau, N, Nam, V. H., Linh, D, M., and Quy, V. K., (2021), An improved agent-based AODV routing protocol for MANET, EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 8(27), 2021.

[22] Tan, N, D., Quy, K, V., Hung, P, N., & Vinh, L, V. (2021). Sector Tree-Based Clustering For Energy Efficient Routing Protocol in Heterogeneous Wireless Sensor Network, International Journal of Computer Networks & Communications (IJCNC), 13(5), 57-74.

[23] T. N. Nguyen, C. V. Ho, and T. T. T. Le, "A Topology Control Algorithm in Wireless Sensor Networks for IoT based Applications," in The International Symposium on Electrical and Electronics Engineering (ISEE). Ho Chi Minh City, Vietnam, 2019, pp. 141–145.

[24] VINT Project. (1997, aug) The network simulator - NS2. [Online]. Available: http://www.isi.edu/nsnam/ns.

[25] T. José, A. R. L. Ribeiro and E. D. Moreno, "Performance Analysis and Application of Mobile Blockchain in Mobile Edge Computing Architecture", the 17th Conference on Computer Science and Intelligence Systems - Annals of Computer Science and Information Systems, vol. 32, pp. 191-197, 2022.

[26] M. Khoshrangbaf, V. K. Akram and M. Challenger, "Ant Colony based Coverage Optimization in Wireless Sensor Networks", the 17th Conference on Computer Science and Intelligence Systems - Annals of Computer Science and Information Systems, vol. 32, pp. 155-159, 2022.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 219–225
DOI: 10.15439/2022R16
ISSN 2300-5963 ACSIS, Vol. 33

CRSDF: Improved Network Lifespan through Chain-routing Scheme and Data Fusion in Wireless Sensor Network

Nguyen Duy Tan
Faculty of Information Technology,
Hung Yen University of Technology and Education,
Hungyen, Vietnam
tanndhyvn@email.com

Hong-Nhat Hoang
School of Computer Science and Engineering
Pusan National University
Busan, Republic of Korea
nhat.hqh@gmail.com

***Abstract*—How to use efficient energy in wireless sensor networks (WSN) is one of the major challenges due to limited energy batteries and computation capacity. Therefore, in this paper, we propose combining a chain-base routing scheme and data fusion sensor information (CRSDF for short). CRSDF contains two major works: Firstly, the chain-based routing method is applied to connect sensor nodes into a chain in which each node transmits only with the nearest neighbor using the remaining energy and distance of nodes as standard parameters to determine which node will be selected the chain leader, secondly, we fuse and compress one or more data packets to generate a result packet with small size base on the Slepian-Wolf and Dempster-Shafer theory. The simulation results exhibit that the energy efficiency of our proposed protocol can be improved by 40%, 20%, and 15% compared to low-energy adaptive clustering hierarchy (LEACH), power-efficient gathering in sensor information system (PEGASIS), and an improved energy-efficient PEGASIS-Based protocol, respectively.**

***Index Terms*—Energy-efficient, routing protocol, chain-based clustering, wireless sensor networks, data fusion.**

I. Introduction

WSN consists of many sensor nodes that are deployed in diverse fields with different applications such as healthcare, environment monitoring, smart cities, intelligent transportation system, and so on [1]. Sensor nodes are small in size, limited bandwidth, processor abilities, memory, and resources, particularly, the battery energy is very difficult to recharge and substitute. Therefore, it is very valuable for designing routing protocols to efficiently employ energy resources and lengthen the lifespan of the network. Chain-based clustering routing protocol and data compression are satisfactory methods to decrease energy consumed and extend the lifespan of sensor networks [2, 3]. If the chain-based clustering routing method enormously reduces the distance communication among nodes in the network, then data aggregation will remove the data redundancy gathered from surrounds in various sensors to obtain more accurate and concise information. Furthermore, data pressure also is a suitable solution for maintaining battery energy by abating the number of bits to be transferred, however, the normal compression method as Lempel-Ziv; Huffman is difficult acceptance for sensor networks because they need a strong processor and large storage capability [4, 5] for activities. Distributed Source Coding (DSC) technique [6, 7], which is proposed by Slepian-Wolf, perform compresses the lossless data of two correlated sources employing side information. The data correlated can be the observed stream or the data in the past of its neighbors. DSC is one of the most relevant solutions for WSNs to conserve energy due to the storage and processing constraint of sensor nodes. Furthermore, in WSN, sensor nodes usually are deployed densely in a sensor region to monitor the environment, so, this correlated condition can be easily satisfied [8, 9]. The hierarchical clustering routing technique is known as a good solution to limit energy consumption, and prolong the lifespan of the network. For instance LEACH [10], in LEACH the nodes are grouped into a few clusters. Each cluster votes for a leader node called the cluster head node (CHN) to take the responsibility for transmitting data to the sink device, other nodes (called cluster members) will only send measured data to its CHN. However, the communication single-hop intra-cluster consumes a lot of energy because the coverage area of clusters is wide. To overcome this limitation, the chain-based routing technique is proposed to maximize the network lifespan such as PEGASIS [2]. According to this technique, sensor nodes are connected and communicated with the closest neighbor in a long chain. It has been verified that the performance of the PEGASIS protocol is better the LEACH protocol in terms of network lifespan and energy efficiency. However, the selecting chain leader node (CLN) is the same as CHN without considering the remaining energy, and the distance from candidate nodes to the sink is also a limitation. Therefore, there have been many researchers focused on improving the chain routing scheme base on PEGASIS, typically, an improved energy-efficient PEGASIS-based protocol is introduced by Sen et al. [11], an energy-efficient chain-based routing protocol for orchard WSN, [12], or enhanced energy-efficient routing for WSN using extended power-efficient gathering in sensor information systems [13], and so on. However, none of the above improvements consider the collaboration between chain routing and data fusion to enhance energy efficiency.

In this paper, we propound a data fusion and chain-based clustering routing scheme (called CRSDF), which can achieve energy and bandwidth efficiency by eliminating redundant data by fusing and avoiding "long links" in the chain. In CRSDF, the Greedy algorithm is employed to build the chain like IEEPB [11], but CRSDF chooses CHN in rounds by deliberating the remaining energy of candidate nodes and the distance between theirs and the sink to determine which node will be voted CHN. CHNs will be responsible for forwarding data to the sink. In addition, the Slepian-Wolf and Dempster-Shafer (DS) evidence theory is also used to aggregate data going along the chain, and then the DSC scheme is used to reduce data packet size in correlated sources. The simulation results exhibit that the network lifetime of CRSDF can be lengthened by 40%, 30%, and 10% compared to LEACH, PEGASIS, and IEEPB respectively.

The remainder of this paper is organized as follows. Section II presents the related works and Section III presents the system model, Section IV analyzes the data fusion architecture, Section V describes the details of CRSDF. In Section VI, we evaluate and analyze the experiment results. Finally, we conclude the study in Section VII.

II. Related works

There are many researchers focused on improving energy-efficient chain-based routing protocols. For example, in [11], Sen et al. proposed an improved energy-efficient PEGASIS-based (IEEPB for short), in which the criteria for selecting CLN is considered both the remaining energy and distance among nodes and the sink. Moreover, in order to reduce the "long link" between nodes in the building chain, the IEEPB selects a node that is not only compared to the end node of the chain but also compared to the other nodes in the chain to find the closest node for connecting into the chain. The simulation results show that IEEPB overcomes PEGASIS in terms of network lifespan and energy efficiency. However, the chain still contains long links since the algorithm for building the chain is still not optimal.

Recently, Zi et al. [3] proposed a novel chain-based routing protocol (BranChain) to avoid long links in a chain by connecting nodes in an independent branched chain (IBC) in the chain constructing phase. Each IBC contains several nodes that are connected by finding the optimal paths between two small chains. Although this method can improve network lifespan, the complexity algorithm is increased. Sadhana et al. [13] proposed an extended PEGASIS protocol called E-PEGASIS to enhance energy-efficient by combining the remaining energy and distance between the node and the sink in the chain leader (CL) selection phase. In addition, Wu et al. [12] propose an improved chain-based clustering hierarchical routing based on the LEACH algorithm called ICCHR, where the remaining energy and distance among nodes and the sink is attached to the probability function T(n) as a criterion for electing CHN, and then, CHNs are connected together into chain beginning the furthest CH from the sink based on the greedy algorithm. However, most of the current routing protocols do not consider data aggregation/fusion to reduce the number of packets that need to be transmitted to the sink, thereby significantly reducing energy consumption. Huy and Viet [14] proposed sliding windows for multi-sensor data fusion (DF-SWin) where the concepts of data mining, sliding window, and Rough set are integrated to get several specimens from sensor nodes in the cluster based on the energy level, distance from the node to CHN, and the residual packet properties to select value parameters for data fusion in CH node. In [9], Ullah et al. proposed multi-sensor data fusion based on modified belief entropy in Dempster–Shafer theory for heterogeneous sensors (DFUDS). DFUDS applied the rules of measuring uncertainty in DS theory for incorporating the data from sensors to achieve more accurate and concise quantification of data. Besides data aggregation, data pressure is also a technique to diminish the size of data packets transmitted to the sink, thus also saving energy. Hau et al. [20] introduced the improvement of the AODV routing protocol by using intelligent agents for efficient energy. Tan et al. [21] proposed a sector tree based on a clustering routing protocol for efficient energy, in which the nodes in a cluster communicate with each other based on the minimum spanning tree connecting. Sadler et al. [15] proposed the sensor Lempel Ziv Welch (S-LZW) algorithm based on the LZW algorithm for sensor systems by balancing three parameters: the dictionary size, and the data size for compression and processes with a full dictionary. In [4], Malleswari et al. introduced the implementation of modified Huffman coding for WSN, where the Huffman algorithm is employed for the compression and decompression of data in order to minimize the size of the data packet and save energy.

III. System Model

A. Network Model

In our network model, we hypothesize that a network system consists of a gateway or sink device and a lot of sensor nodes that can support applications [3, 10, 11].

- The gateway device is fixed and not limited to energy and computation capacity, which can aggregate and forward the data packets.
- N sensor nodes are randomly deployed in the network zone of two-dimensional.
- All sensor nodes are static state after deployment and know the location of each other based on the global positioning system (GPS)
- All sensing nodes use the battery energy and cannot recharge or replace
- The radio channels are symmetry

B. Energy Consuming Model

In the radio energy dissipation model used in our proposal to transmit *q-bit* data, the energy consumed may be computed as Equation (1) [10, 12]:

$$E_{TX}(q,d)=\begin{cases} q\times E_{elec}+b\times E_{friis}\times d^2 & ,if\ d<d_0 \\ q\times E_{elec}+q\times E_{tworay}\times d^4 & ,if\ d\geq d_0 \end{cases} \tag{1}$$

where E_{elec} is the energy consumption part for the electric circuit, E_{towray} and E_{friis} are the part of energy consumption by the amplifier in two ray ground mode and free space mode with the distance *d*, corresponding, and d_0 is the threshold distance between two nodes. The geographical distance between nodes *a* and *b* is calculated as Equation (2) below:

$$d_{ab}=\sqrt{(x_a-x_b)^2+(y_a-y_b)^2} \tag{2}$$

The energy dissipated in receiving *b-bit* data is calculated by the number of bits and energy dissipated by the electric circuit as Equation (3) below:

$$E_{RX}=qE_{elec} \tag{3}$$

IV. Data Fusion and Framework

A. Distributed Source Coding

DSC based on the Slepian-Wolf theorem is one of the most efficient techniques, to compress correlated data sources [4, 16]. In DSC, the correlated signals from a few sensor nodes are compressed with a totality rate greater than or equal to the joint entropy thus they decrease data packet size. For instance, we have two distributed derivations *X* and *Y* which create *u* bits of binary data as demonstrated in Fig.

1. We assume that the data of sources X and Y are highly correlated and can be different by no more than 1 bit. In other words, the Hamming distance between X and Y is $d_H(X, Y) \leq 1$ or $P_r(X_i = 0) = P_r(Y_i = 0) = P_r(X_i = 1) = P_r(Y_i = 1) = 0.5$, $i = 1, ..., u$. So, $H(X) = H(Y) = u$ bits, where $P_r(X)$ and $H(X)$ denote the probability distribution density function of random source X and the entropy function, respectively. In this case, if Y is sent to the decoder as side information that contains u bits, then we can send X with $H(X|Y) = b$ bits per specimen, where b is the index number of the subset, without any loss data at the joint decoder.

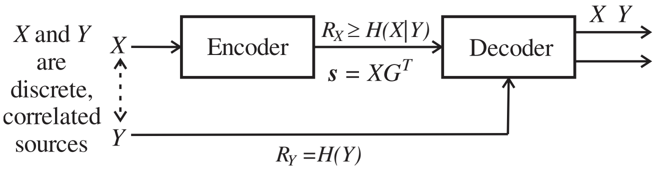


Fig. 1. Distinct encoding and common decoding of two correlated data sources Y and X

We suppose that Slepian-Wolf coding of X and Y origins are equiprobable 2^u specimens (u = 4 bits binary per specimen).

Let $k = u - b$, where k is a integer and $2^k - 1 \geq d_H(X, Y)$, then $H(X) = H(Y) = 4$ bits per specimen, $H(X|Y) = b = 3$ bits, and $H(X, Y)$=4+3=7 bits per pair of specimen for common decoding, where $H(X|Y)$ and $H(X,Y)$ denote the conditional entropy and the joint entropy, respectively. Consequently, we can group 2^4 specimens as $2^3 = 8$ subsets, in each subset the Hamming distance between any two elements is greater than or equal to 4, and assign 8 different binary index numbers by Z_{xxx}, respectively, as follow:{Z_{000} = (0000, 1111): 000, Z_{001} = (0111, 1000): 001, Z_{010} = (0100, 1011): 010, Z_{011} = (0011, 1100): 011, Z_{100} = (0010, 1101): 100, Z_{101} = (0101, 1010): 101, Z_{110} = (0110, 1001): 110, Z_{111} = (0001, 1110): 111}. We study how to send data of source Y, whose data is used as side information, as {0100} and the index of X as {110}. When the decoder received the index as {110}, the probable value of X is set {0110, 1001} in Z_{110}. The decoder will obtain the precise value of X as {0110} since the Hamming distance between X and Y is less than or equal to one ($d_H(X, Y)$ =1).

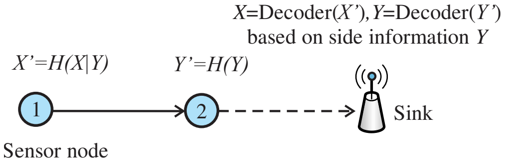


Fig. 2. Execution of distributed source coding in WSN

In fact, the side information Y is perfectly available at the decoder (e.g. sink device) since, in a sensor network, where the sensor node is deployed densely in a sensor field, these correlated data conditions can be satisfied easily. However, the collaboration of encoders is not easy to perform in the sensors node, as exhibited in Fig. 1 and 2, the Y is only ready for the decoder but not the encoder, therefore, how can we press two origins X and Y into a total (u + b) bits as the mentioned example with lossless decoding? To solve this puzzle, we can employ a Linear Block Code (LBC) model in Galois Field two ($GF(2)$) representing in the form of a (u, b) LBC, which is given by $\boldsymbol{G = [I_m: P]}$ and $\boldsymbol{H = [P^T: I_{u-b}]_{(u-b) \times u}}$ with the property that $GH^T = 0$, where G is a b by u binary generator matrix and $\boldsymbol{P^T}$ is the exchange of the b by ($u - m$) matrix $\boldsymbol{P}$. $\boldsymbol{H}$ is a parity check matrix used at the decoder using syndrome decoding, while G is used to encode the message. With the above example, we have an H and G matrix as shown follows:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$$

Therefore, the encoder can simply compute the syndrome of the source $\boldsymbol{s = XG^T}$, (associated with all $X \in Z_s$), which is the index of X using $H(X|Y) = b$ bits of syndrome instead of u bits original data, achieving the Slepian-Wolf coding in a pressure ratio of u:b [16]. The decoder will associate the side information Y and all subsets to find out the original of X.

B. Dempster-Shafer theory

In WSN applications, sensor nodes measure the environment and periodically send correlated high data packets to the sink that is connected to the user's PC or through the internet. In order to preserve the battery energy of nodes, we can apply DSC techniques as above. However, it will be much extravagant energy in sending the unnecessary collected data because when nodes are very close, the observed values are the same. Fig. 3(a) illustrates the number of packets transmitted in a chain without data aggregation, in which there are seven nodes in a chain transmitting the overall seven packets along the chain to the sink. The total demolish energy may be computed as Equation (4):

$$E(a) = 14E_{Rx}(q) + 21E_{Tx}(q, d(i, j)) + 7E_S(q) \qquad (4)$$

$$E(b) = 6E_{Rx}(q) + 7E_{Tx}(q, d(i, j)) + 4E_{DF}(q) + 7E_S(q) \qquad (5)$$

where $E(b)$ and $E(a)$ are energy consumption totality with data fusion and without data fusion, respectively. Let the distance between every two nodes is d and the packet size is q = 1 bit, then we have:

$$E(a) = 35E_{elec} + 21E_{friss}d^2 + 7E_G + 7E_S \qquad (6)$$

$$E(b) = 13E_{elec} + 7E_{friss}d^2 + 4E_{DF} + 7E_G + 7E_S \qquad (7)$$

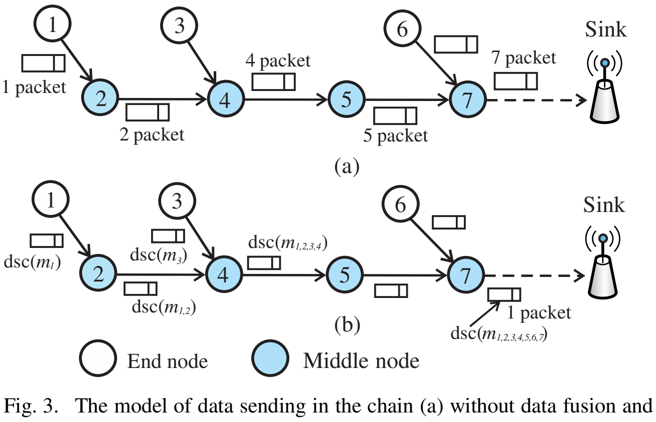


Fig. 3. The model of data sending in the chain (a) without data fusion and (b) with data fusion

Accordingly, it is clear that the energy exhaustion without data fusion is higher than about three times in comparison to data fusion. However, how to aggregate two or more correlated data packets from many sensor nodes (multi-sensor for short) into an individual packet as shown in Fig. 3(b)? In our proposal, we apply the DS evidence theory to reduce redundant information and improve the reliability of inference based on collected data from the external environment [17, 18, 22]. According to DS reasoning, all mutually exclusive events have the same style are enumerated in "a finite set Θ". Assuming that Θ = { θ_1, θ_2, ..., θ_v}, where θ_i ($i=1,v$) is the observation mutually exclusive or hypothetical possibilities that are obtained from the source sensor. The overall possible subsets of Θ are 2^v subsets.

$$2^v=\{\phi, \theta_1, \theta_2, ..., \theta_v, \{\theta_1, \theta_2\}, ..., \{\theta_1, \theta_2,..., \theta_i\},..., \Theta\} \quad (8)$$

Let m_i be "probability mass function" of the sensor *i-th* (i= identification), which the detected value in subset "*A*", and denoted by a “confidence interval” as [*Belief(A), Plausibilityl(A)*] [9].

$$\begin{cases} \sum_{A\subseteq\Theta} m(A)=1 \\ m(\phi)=0 \end{cases} \quad (9)$$

where ϕ and *m(A)* is the null subset of Θ and probability mass function that shows how the accurate evidence for supporting hypothesis A is possible, respectively.

The lowest zone of the confidence interval is called *Belief(A)*, which is the total of the probability mass *m(A)* and supports for the reasons to believe in the values "θ_i" in subset *A*, including itself.

$$Belief_i(A) = \sum_{P_A\subseteq A} m_i(P_A), \ \forall A \subseteq 2^v \quad (10)$$

The upper zone of the confidence interval is the sum of all the probability mass of the subsets that intersect with subset *A*.

$$Plausibility_i(A) = 1 - \sum_{P_A\cap A=\phi} m_i(P_A), \ \forall A \subseteq 2^v \quad (11)$$

DS has also given a rule of combining for fusing (⊕) two sensor sources called *$m_{12}(A)$* with the subset $A \neq \phi$ and $m_{12}(\phi)=0$ [18].

$$m_{12}(A) = m_1 \oplus m_2(A) = \frac{\sum_{B\cap C=A\neq\phi} m_1(B)m_2(C)}{1-\sum_{B\cap C=\phi} m_1(B)m_2(C)} \quad (12)$$

For example, as shown in Fig. 3(a), consider two sensors 1 and 2 observe together the outside environment temperature, obtain values Θ={"θ_1=30^{0}C", "θ_2=27^{0}C", "θ_3=unknow"}, where the θ_3 cannot determine the measurement value.

$$2^3=\{\phi,\theta_1,\theta_2, \theta_3, \{\theta_1, \theta_2\},\{\theta_1, \theta_3\},\{\theta_2, \theta_3\},\{\theta_1, \theta_2, \theta_3\}\}$$

TABLE I. THE RESULTS OF DATA FUSION OF THE DS’S RULE OF COMBINATION

			Sensor 1			**Fused data**
	Mass function		$m_1(\theta_1)$	$m_1(\theta_2)$	$m_1(\theta_3)$	
			0.2	0.7	0.1	1.0
Sensor 2	$m_2(\theta_1)$	0.6	{θ_1}=0.12	{ϕ}=0.42	{θ_1}=0.06	0.385
	$m_2(\theta_2)$	0.3	{ϕ}=0.06	{θ_2}=**0.21**	{θ_2}=**0.03**	**0.596**
	$m_2(\theta_3)$	0.1	{θ_1}=0.02	{θ_2}=**0.07**	{θ_3}=0.01	0.019
Fused data		1.0	0.385	**0.596**	0.019	1.0

The probability mass function achievements as follows: with sensor 1, $m_1(\theta_1)$=0.2, $m_1(\theta_2)$=0.7, $m_1(\theta_3)$=0.1; with sensor 2, $m_2(\theta_1)$=0.6, $m_2(\theta_2)$=0.3, $m_2(\theta_3)$=0.1, the other value are zero.

Table 1 illustrates the results of the calculation after applying the combination rule of DS, in which:

$$m_{12}(\theta_1) = \frac{0.12+0.02+0.06}{1-(0.06+0.42)} = \frac{0.2}{0.52} = 0.384$$

$$m_{12}(\theta_2) = \frac{0.21+0.07+0.03}{1-(0.06+0.42)} = \frac{0.31}{0.52} = 0.596$$

$$m_{12}(\theta_3) = \frac{0.01}{1-(0.06+0.42)} = \frac{0.01}{0.52} = 0.019$$

Accordingly, to fuse the data of *N* sensor nodes, each node generates a sequence of *v* values, we can apply Equation (13) as below:

$$m_{12,...,v}(A) = m_1 \oplus m_2,... \oplus m_v(A) = \frac{\sum_{\cap_i C_i = A\neq\phi}\left(\prod_{0\le i\le v} m_i(C)\right)}{1-\sum_{\cap_i C_i=\phi}\left(\prod_{0\le i\le v} m_i(C)\right)} \quad (13)$$

V. PROPOSED PROTOCOL

Here, we depict the detail of CRSDF protocol, which is inspired by the PEGASIS protocol [2, 3, 19]. The operation of the CRSDF protocol is divided into rounds, each round composed of four phases: (1) chain leader selection, (2) chain formation, (3) data fusion, and (4) data transmission phase.

A. Phase 1: Chain Leader Selection

In this phase, all nodes alive will send a hello message containing ID, location information, and residual energy to the sink. The sink will select the CLN having the residual energy higher than E_{ave} and maximum cost function as Equation (14) and (15):

$$E_{ave} = \frac{1}{n}\sum_{i=1}^{n} E_{res}(i) \quad (14)$$

Where *n* and *$E_{res}(i)$* are the overall living nodes and the residual energy of contestant node i-*th* at the present time, respectively.

$$\cos t(i) = Max\left(\frac{c_1}{c_2} \times \frac{E_{res}(i)}{d_{i,Sink}}\right) \quad (15)$$

Where $d_{i,Sink}$ is the distance from the node i-*th* to the sink device, which is computed as Equation (2). Besides, the coefficient of cost factors values c_1 and c_2 are constant, and ($c_1 + c_2=1$).

Algorithm 1 Chain Leader Selection Phase

```
Input: N sensor nodes
Output: chain leader node (CLN)
1: for i = 1 to N do
2:     Send a HELLO message containing the remaining energy and the location to the Sink
3: end for
4: The Sink calculates average energy as in Equation (15)
5: Chooses the chain-leader node, of which the cost function value as in Equation (16) is the highest
6: Broadcast CLN-ADV to all nodes alive in the network
7: Nodes receive the CLN-ADV message and go to Algorithm 2
```

```
return: CLN
```

B. Phase 2: Chain Formation

Fig. 4 shows an example of the network topology acquired by running different schemes with 100 sensor nodes in one round. Here, our proposed chain is shorter distance communication than the PEGASIS pattern of the network. After selecting CLN, the sink distributes this information to all living nodes to construct a chain as in Algorithm 2 below based on the Greedy algorithm to reduce the "long link" in the undirected weight graph problem.

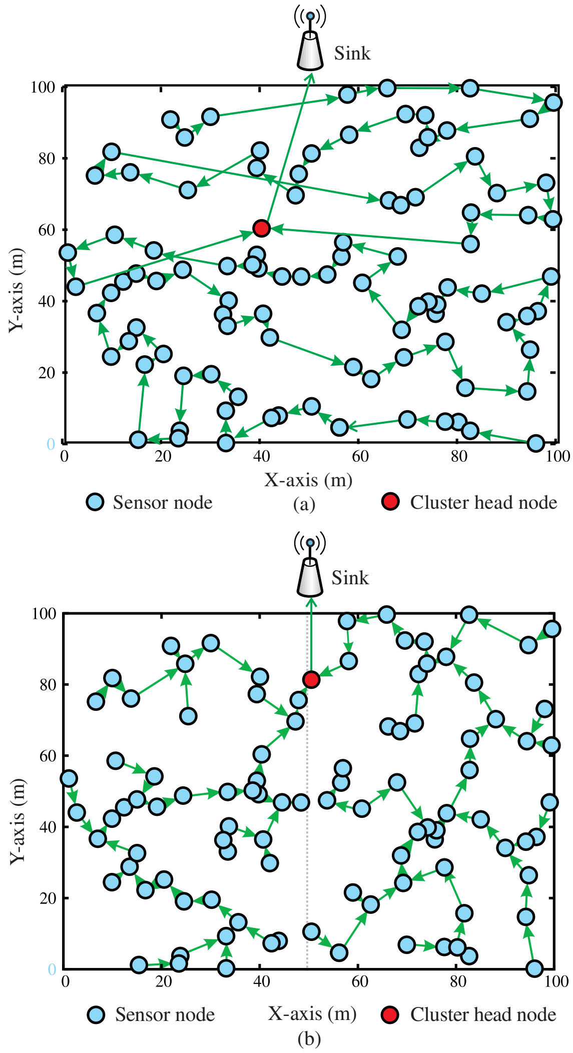


Fig. 4. The network topology with: (a) PEGASIS and (b) CRSDF protocols

Algorithm 2 Chain Formation Phase

```
Input: N sensor nodes, CLN
Output: CHAIN

CHAIN = {CLN} // the chain begin by leader node
cloestNode = {the closest node to the CLN in N};
CHAINP = {cloestNode } // the chain begin by the closest node to the sink
$N = N\setminus\{cloestNode,$ CLN$\}$;
while ($N \neq \emptyset$) do
    Find node $i$ in $N$ and node $j$ in CHAIN, whose $d_{i,j}$ is minimum;
    Find node $k$ in CHAINP, whose $d_{i,k}$ is minimum;
    if ($d_{i,j} < d_{i,k}$) then
        Connect node $i$ to node $j$ in CHAIN;
        Append node $i$ into CHAIN;
    else
        Connect node $i$ to node $k$ in CHAINP;
        Append node $i$ into CHAINP;
    endif
    Discard node $i$ in $N$
end while
Connect CHAINP to CHAIN by connecting the node cloestNode with CLN
Create time slots according to the TDMA schedule for all member node
Broadcast CHAIN and TDMA information to the network
return CHAIN;
```

C. Phase 3: Data Fusion

After constructing the chain, CLN generate a TDMA schedule and broadcast them to all member nodes in its chain. The sensor nodes start measuring, aggregating, and compressing data as in Algorithm 3 follow:

Algorithm 3 Data Fusion Phase

```
Input: The list of data packets Lp
Output: a single data packet with a small size

if (node $i$ in {CHAIN} is a middle node or CLN) then
if (the data packet needs to decompress) then
for each packet $p$ in {list of data packets}
for every 7 bits in compressed data packet p in the decoder do
    $Y$ = read the high 4 bits as side information
    $s$ = read the lower 3 bits
    Search value $X$ in subset $Z_s$, which has Hamming distance with $Y$ is less than or equal to 1;
    Append $Y$ and $X$ into data packet $pp$ decompression;
end for
    Delete the packet p in {list of data packets}
    Update the packet $pp$ into {list of data packets}
end for
  end if
//Data fusion based on D-S
    for each packet $p$ in {the list of data packets} do
        Calculate the mass function $m_i(A)$ based on Equation (9)
    end for
    Fuse data for all packets as Equation (13)
  end if
// This code for data lossless compress as DSC
    for every 8 bits in the data packet do
        $Y$ = read the high 4 bits as side information
        $X$ = read the lower 4 bits as original data
        Calculate $s = X*G^T$
        Append $Y$ and $s$ into a new data packet compression p
    end for
return: {a single packet p}
```

D. Phase 4: Data Transmission

After completing the first two phases above, the data packets are processed and transmitted in CRSDF starting at the furthest node in the chain. Nodes will transfer the observed data to the next node in the time slots designated by the TDMA mechanism along the chain. With the chain constructed in CRSDF have more than two end nodes as shown in Fig. 4(b). The middle nodes or CLN will receive the data, decompress these data packets and aggregate them with their own data, re-compress, and then it forwards this single packet to the next node along the chain or the sink. After a period of time, the next round will be restarted by reselecting CLN and reconstructing the chain for a new round.

VI. Experimental And Performance Evaluation

A. Simulation Parameters

To evaluate the performance of CRSDF, we have simulated CRSDF, LEACH, and PEGASIS in the network simulator ns-2.34 [21, 22, 23] and C/C++ using the scenario with some parameters that are indicated in Table 2, [10, 13].

TABLE II. THE SIMULATION PARAMETERS

No. Item	Parameters Description	Value
1	Size of simulation field	100m ×100m
2	Number of sensor nodes (N)	100 nodes
3	Energy consumption: two ray ground model (E_{towray})	10 pJ/bit/m^2
4	Energy consumption: free space model (E_{friis})	0.0013pJ/bit/m^4
5	Energy consumption: Electric circuit (E_{elec})	50 nJ/bit
6	Energy consumption: Data fusion (E_{DF})	5 nJ/bit/packet
7	The initial energy of node (E_{init})	2J
8	Packet size	1024 bytes
9	Simulation time	3600s
10	Sink location	49,175

B. Simulation results

Fig. 5 displays the simulation result of the total number of dead nodes in the network according to the number of rounds of four schemes: LEACH [10], PEGASIS [2], IEEPB [11], and our proposed CRSDF. We can see that the CRSDF has a longer network lifespan than LEACH and PEGASIS and IEEPB. The network using the LEACH protocol stops working at about 100 rounds, the PEGASIS prolongs its life to about 1400 rounds, and IEEPB can preserve the lifecycle to about 1650 rounds, whereas the CRSDF protocol can lengthen its lifespan to about 1750 without compression and 1900 rounds with compression. This improvement is due to the reduction of the distance transmission from the CHs node to the sink device and the avoidance of long links in the chain, the more rounds, the longer the network lifetime.

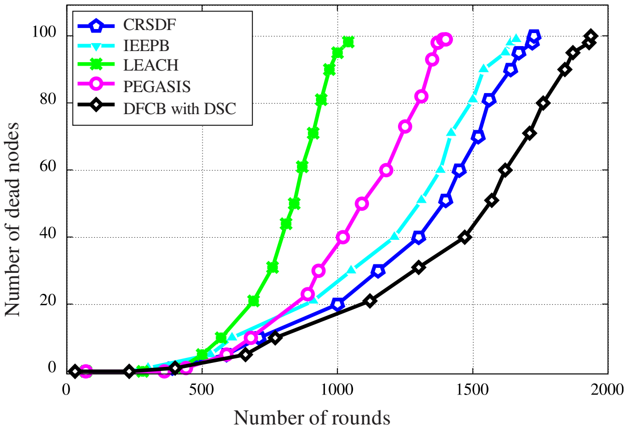


Fig. 5. Number of nodes alive during the simulation time in (rounds)

In Fig. 6, we describe the energy consumption of four protocols according to the simulation time (rounds). Based on the results displayed in Fig. 6, we can observe that the energy consumption nodes in the network running the proposed protocol is less than about 50%, 30%, and 20% compared to LEACH, PEGASIS, and IEEPB with data compression, respectively.

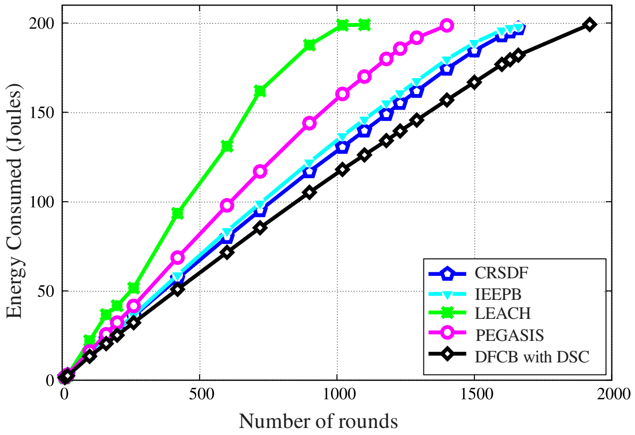


Fig. 6. Energy consumption of three protocols

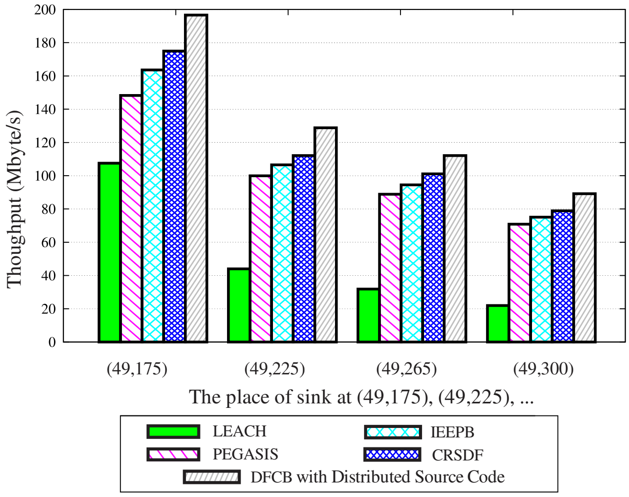


Fig. 7. The network throughput with the different places of the sink

As shown in Fig. 7, the network throughput is described as the amount of data packets transported success from the overall sensor nodes to the sink device in unit time. We can observe that network throughput will decrease if the distance from network nodes to the sink device is because CHN nodes consume a lot of energy for transmitting data over links with far distances, however, CRSDF still has in better performance terms of throughput as compared to the other three protocols mentioned.

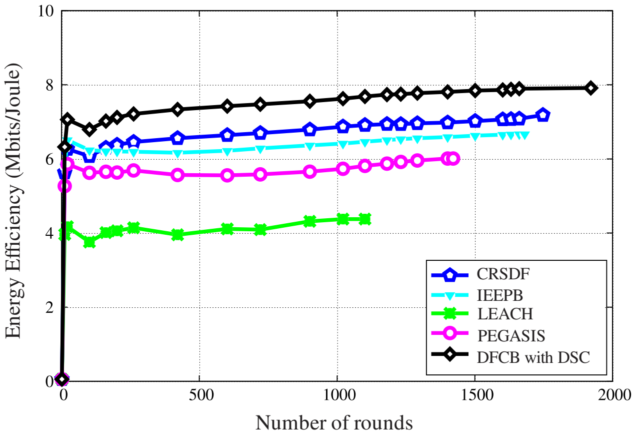


Fig. 8. The number of messages received at the sink

Fig. 8 presents the energy efficiency in Megabits per Joule according to the total number of rounds in running various routing protocols. We can observe clearly that our proposed protocol achieves higher energy-efficient than the other three protocols: LEACH, PEGASIS, and IEEPB about 30%, 20%, and 15%, respectively. The reason for this enhancement is the CRSDF minimizes the energy consumption in the communication process among nodes by decreasing the data packet size before forwarding it to the next node within the chain and the sink.

VII. Conclusion

In this paper, we presented collaboration among a chain-based routing scheme and data fusion to improve energy efficiency, namely CRSDF, which can extend the network lifespan by constructing a chain connecting the nearest neighbor nodes and fusing, compressing sensed data in order to reduce the number of bits in the data packet before sending to the sink device. Simulation results display that the energy efficiency of CRSDF is better than LEACH, PEGASIS, and IEEPB about 30%, 20%, and 15%, respectively. In the future, we study energy efficiency hierarchical routing protocol for applications IoT based on wireless sensor networks.

Acknowledgment

This work was supported by Hung Yen University of Technology and Education under grant code UTEHY.L.2022.01.

References

[1] Yang, X., Pei, X., Chen, G., Li, T., Wang, M., & Wang, C. (2019). A Strongly Unforgeable Certificateless Signature Scheme and Its Application in IoT Environments. *Sensors*, 19(12), 1 – 27.

[2] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information system," in *IEEE Aerospace Conference Proceedings*, March 2002, pp. 1125 – 1130.

[3] L. Zi, W. Chen, X. Liu, and X. Chen, "A novel chain-based routing protocol, branchain, in wireless sensor networks," *International Journal of Embedded Systems*, vol. 11, no. 3, pp. 259–268, 2019.

[4] M. Malleswari, B. A. Krishna, N. Madhuri, and M. K. Chowdary, "Implementation of modified huffman coding in wireless sensor networks," *International Journal of Computer Applications*, vol. 177, no. 1, pp. 14–17, 2017.

[5] K. Sayood, *Lossless Compression Handbook*. California, USA: Academic Press, 2003.

[6] D. Slepian and J. K. Wolf, "Noiselesscoding of correlated information source," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[7] P. Kumar and V. Gupta, "Data compression using distributed source coding in wireless sensor network," *International Journal of Electronics Communication and Computer Technology*, vol. 2, no. 1, 2011.

[8] K.-C. Lan and M.-Z. Wei, "A compressibility-based clustering algorithm for hierarchical compressive data gathering," *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2550–2562, 2017.

[9] I. Ullah, J. Youn, and Y.-H. Han, "Multisensor data fusion based on modified belief entropy in dempster–shafer theory for smart environment," *IEEE Access*, vol. 9, pp. 37 813–37 822, 2021.

[10] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on wireless communication*, vol. 1, no. 4, pp. 660–670, 2002.

[11] F. Sen, Q. Bing, and T. Liangrui, "An Improved Energy-Efficient PEGASIS-Based Protocol in Wireless Sensor Networks," in *Eighth International Conference on Fuzzy Systems and Knowledge Discovery*. Shanghai, China, July 2011, pp. 2230–2233.

[12] H. Wu, H. Zhu, L. Zhang, and Y. Song, "Energy efficient chain based routing protocol for orchard wireless sensor network," *Journal of Electrical Engineering & Technology*, vol. 14, p. 2137–2146, 2019.

[13] S. S, S. E, and D. D, "Enhanced energy efficient routing for wireless sensor network using extended power efficient gathering in sensor information systems (e-pegasis) protocol," *Procedia Computer Science*, vol. 194, p. 89–101, 2021, the 18th International Learning & Technology Conference.

[14] H. Duong-Viet and V. Nguyen-Dinh, "Df-swin- sliding windows for multi-sensor data fusion in wireless sensor networks," in *The 9th International Conference on Knowledge and Systems Engineering (KSE)*. Hue, Vietnam, October 2017, pp. 54–59.

[15] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. Boulder, Colorado,USA, November 2006, pp. 265–278.

[16] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, no. 5, pp. 80 – 94, 2004.

[17] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[18] E. D. C. Bezerra, A. S. Teles, L. R. Coutinho, and F. J. da Silva e Silva, "Dempster–shafer theory for modeling and treating uncertainty in iot applications based on complex event processing," *Sensors*, vol. 21, no. 5, pp. 1–26, 2021.

[19] Tan, N, D.,& Viet, N, D. "SCBC: Sector-Chain Based Clustering Routing Protocol for Energy Efficiency in Heterogeneous Wireless Sensor Network", Advanced Technologies for Communications (ATC), Ho Chi Minh, Vietnam, 2015, pp. 314-319.

[20] Van-Hau, N, Nam, V. H., Linh, D, M., and Quy, V. K., (2021), An improved agent-based AODV routing protocol for MANET, EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, 8(27), 2021.

[21] Tan, N, D., Quy, K, V., Hung, P, N., & Vinh, L, V. (2021). Sector Tree-Based Clustering For Energy Efficient Routing Protocol in Heterogeneous Wireless Sensor Network, International Journal of Computer Networks & Communications (IJCNC), 13(5), 57-74.

[22] W. Heinzelman, "MIT uAMPS LEACH ns Extensions," https://blog.katastros.com/a?ID=01700-10dfaae3-e3c2-4f2f-8235-608c3010d995 (accessed: Feb 29, 2022), 2004.

[23] VINT Project, "The network simulator - NS2," http://www.isi.edu/nsnam/ns (accessed: Feb 2, 20212, 1997.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 227–234 DOI: 10.15439/2022R08 ISSN 2300-5963 ACSIS, Vol. 33

Root Rot Lentil and Healthy Lentil Detection Using Image Processing

1st Noshin Tasnia
Applied statistics and Data Science
Jahangirnagar University
Dhaka, Bangladesh
nousin.tasnia@gmail.com

2nd Moon Halder
American International
University–Bangladesh
Dhaka, Bangladesh
moon.halder892@gmail.com

3rd Mst. Jiasmin Ara
Biology Department
Cantonment English School and College
Chattogram, Bangladesh
mst.jiasmin@gmail.com

4th Md. Rejaul Karim
Farm Machinery and Postharvest Process Engineering Division
Bangladesh Agricultural Research Institute
Gazipur, Bangladesh
rejaulkarim@bari.gov.bd

5th Shakik Mahmud
Computer Science and Engineering
United International University
Dhaka, Bangladesh
smahmud172174@bscse.uiu.ac.bd

***Abstract*—The hardest thing to do in agriculture is to figure out which leaves are healthy and which ones are damaged. Bangladesh makes 80% of its money from farming. Most farmers cannot read or write. They didn't know how much fertilizer to put on a lentil with root rot or a healthy lentil. They sometimes spray medicine on the plants, which is terrible for them. As a result, agriculture has become much less productive. In this paper, a picture-segmenting algorithm is given that can automatically find and classify plant leaf diseases. Also included are surveys of different ways to classify diseases that can be used to find plant leaf diseases. The Convolution Neural Network model is used to segment images, an essential part of finding plant leaf diseases. Every country's growth is based on its agricultural production. To keep agricultural production at a certain level and keep growing sustainably, scientists need to study how to find and treat diseases. Standard methods in the literature for classifying leaf images involve extracting attributes and training classifier models, which makes them less accurate. The technique suggested gets rid of any unnecessary data from the image collection. Using the mixture model for region growth, we first find the area of interest based on the colors of the leaves in the image. After extracting the features, a deep convolution neural network model is used to classify the leaf photos. A convolutional neural network model can be used with the deep learning model to find different patterns in color photos. Examining the execution strategy of the proposed model using an unauthorized dataset. According to the results of the simulating replica, the suggested model outperforms the well-known current methods in the field, with mean classification accuracy and area under the characteristics curve of 95.35% and 94.7%, respectively.**

***Index Terms*—Image processing, Root Rot lentil, Healthy lentil, CNN, Tensor flow, Plant disease detection, classification.**

I. Introduction

In today's society, the agricultural field mass is used for more than only feeding. Agriculture productivity is critical to Bangladesh's economy. As a result, in agriculture, detecting sickness in plants is vital. An automatic disease detection system helps detect plant diseases in their early stages. For example, root rot leaf disease is a dangerous disease found in pine trees in Bangladesh. The fake tree grows slowly and dies after six years. Its impact can be seen in Rajshahi, Rangpur, and Dinajpur, all of which are in the north. The early discovery could have been beneficial in such cases.The only currently used approach for identifying and detecting plant ailments is a simple naked-eye examination by experts. Working with big farms necessitates a sizable team of specialists and ongoing plant monitoring, which are expensive. However, in some nations, farmers lack the resources or skills to consult professionals. Expert consultation is, therefore, both costly and time-consuming. The recommended method is helpful in such situations for keeping an eye on vast fields of crops. Simply glancing at the signs on the plant's leaves simplifies and reduces the cost of disease diagnosis. In addition, this facilitates machine vision by enabling image-based process automation control, monitoring, and robot guidance [1] [2] [3].

It requires more time and is less accurate to identify plant diseases by eye and can only be done in certain places. However, an automated detection method takes less time and work and is more accurate. In addition to brown and yellow marks, early and late scorch, including fungus, virus, and microbial illnesses, widely planted diseases included brown and yellow spots. Image processing is used to quantify the sick area's affected area and identify the afflicted area's color difference [1] [4] [5]. By leveraging additional datasets, such as Plant Village, and CNN models trained on this database, many object identification and picture segmentation issues may be significantly improved [6]. An AI procedure called image processing separates or organizes a picture into various components. There are several different picture segmentation techniques that may be used, from straightforward thresholding to sophisticated color image segmentation. Typically, these components relate to simple things for people to distinguish and see as distinct objects. Since computers can't intelligently determine items, many different techniques for segmenting photos exist. The image processing is based on various features in the image. This could be color information, boundaries, or photo segments [7] [8]. For classified images of root rot lentils and healthy lentils, we use a convolution neural network (CNN). These papers proposed a dashboard-based system for remote data collection to support environmental parameter monitoring and provide warnings about lake water levels and hydroelectricity generation [9] [10].

II. Literature Review

Several academics have discussed several methods for detecting and identifying lentil illnesses. Revathi et al. proposed using image processing edge detection techniques to classify Cotton Leaf Spot Diseases. Initially, picture segmentation based on edge detection was performed. Hiary and Bani-Ahamed used a three-pronged approach to detect illness. The infected zone was initially detected using k-means clustering. The characteristics are then extracted using color co-occurrence methods, and ultimately, CNN is used to classify them. Arivazhagan et al. presented a four-step process for illness classification. The green pixels are masked and deleted using a threshold value in the segmentation process once a color transformation structure for the input RGB image is constructed. The texture feature is then computed using the color co-occurrence approach. The retrieved features are then sent to the classifier. The Otsu approach is used in the picture segmentation ”Leaf Spot Disease Grading Method Using Image Processing. ”ShenWei Zhang and Wu Yachun contributed. The RGB image values were transformed into the HSI color model, and the H plane value was taken for future investigation. The detection technique presented by Dhaygude and Kumbhar consists primarily of four phases. The RGB photo is converted into the HSV layer. The masking operation is carried out by mainly utilizing the values of green-colored pixels. The following functional segments are retrieved from the resulting SGDM matrices using segmentation algorithms. Camargo and J.S. Smith offer an image-processing system that analyzes colored photos to identify visual indications of plant illnesses.The study is based on an analysis of fuzzy logic, ANN, SVM, PNN, and SELF ORG MAPS. It is challenging to comprehend the algorithm’s structure and choose the best parameters for a neural network when training data is not linearly separable [4]. The objective of this research is to design a color co-occurrence method and a vision-based detection system that masks green pixels. NNs may be used to increase the recognition rate of the classification process [1]. Utilizing neural networks and K-means clustering to automatically identify leaf diseases. Artificial neural networks, fuzzy logic, and other soft computing techniques can be used to classify crop illnesses [11]. Form, color, and ideal characteristics may be employed as input criteria for sickness detection in the color co-occurrence technique utilizing SVM classifier, which allows for the expansion of training samples [2]. Improved recognition rates may be achieved by using the Gabor filter for feature extraction and the ANN classifier for classification [3]. Different plant diseases may be categorized using texture segmentation utilizing the co-occurrence matrix approach and the K-means clustering methodology, as well as the Bayes classifier, the K-means clustering technique, and the major component classifier [5]. Employing advanced color extraction features and a huge database, the color co-occurrence texture analysis technique was developed using spatial gray-level dependency matrices [12]. The picture is smoothed using the median filter, and the Otsu technique may be used to calculate the threshold. Crop loss may be estimated using the illness spot area. The condition may be classified by looking at the size of the sick spot [13]. In order to increase the accuracy of the final classification process, a study of several methods for spotting leaf disease, the creation of hybrid algorithms, and the use of neural networks have been proposed [14]. In some circumstances, the implementation still lacks the accuracy of outcomes. More tweaking is required.

III. Materials and Methodology

A. Data Preprocessing and Augmentation

Figure 1 illustrates how color picture cameras were used to record diseased blade images with a resolution of 256 and an arbitrary length from a library of leaf illnesses (Bangladesh Agriculture Research Institute) to compile three different types of disordered leaf images. Only three types of blade data were used for this investigation since some leaf illnesses might be challenging to diagnose. To confirm the identification and demonstrate the generalizability of various CNN network topologies for various leaf diseases, Apple and cherry leaves both have diseased regions, and the severity of these diseases at different disease severity levels is equal. It better satisfies the criteria for our investigation. Compared to other leaf illnesses, these sick leaves are more able to express the distinctive ability of disease areas for different CNN structures and to reach the knowledge of different CNN structures in leaf classification.

Fig. 1. Training set image of healthy lentil sample.

First, 224 × 224 resolution was achieved by resizing all images of leaf diseases to the same length and breadth. Before being fed into different networks, photos are scaled to 224 × 224 to fit the various pretraining CNN topologies. The horizontal and vertical orientations of the photos of these disease types were then reversed since certain leaf disease types contain fewer photographs than others and the gathering of leaf disease images is random.

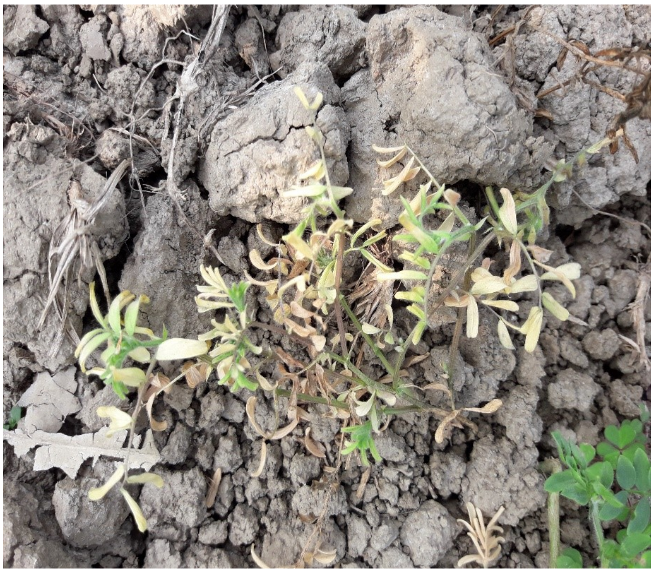


Fig. 2. Training set of root rot lentil sample

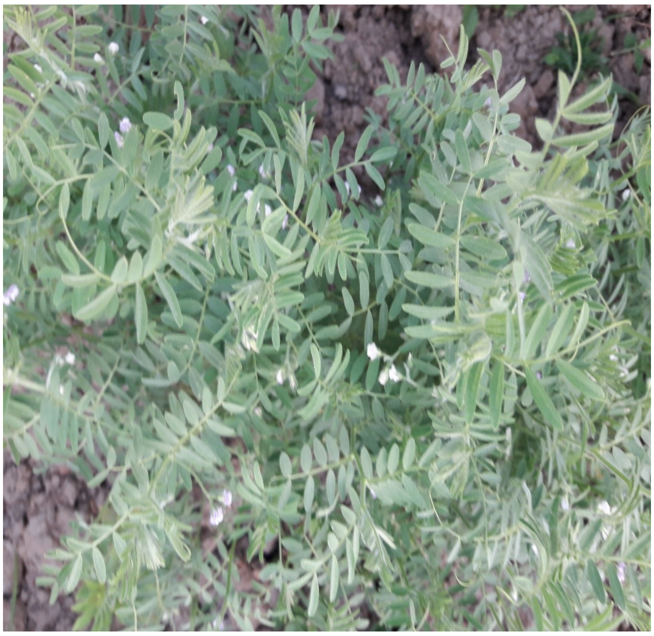


Fig. 3. Testing set of sample image which will predict CNN model what leaf is it.

B. Convolutional Neural Network (CNN)- Based Method

Our deep learning-based network is composed of the Squeeze and Excitation (SE) module, the Inception framework, and the VGG16 convolutional layers. The first five convolutional layers were developed using the VGG16 model for self-learning low-to-high training picture features. Deeper convolutional layers extract more abstract high-level qualities by reducing the feature map resolution. The subsequent max-pooling layer cleans up the feature maps produced by the preceding convolutional layer. The feature fusion that is produced by the inception structure also increases the number of features that can be acquired from feature maps and extracts the unique features using multidimensional analysis. The greatest average pooling layer is used instead of the fully connected layer, which reduces the training parameters, speeds up convergence, and improves classification accuracy. The fully connected layer is replaced with the largest average pooling layer using the inbuilt SE module, which recalculates the special features in the channel dimension. In Fig 2 and Table 1, the modified model's network architecture is displayed together with the appropriate parameters.

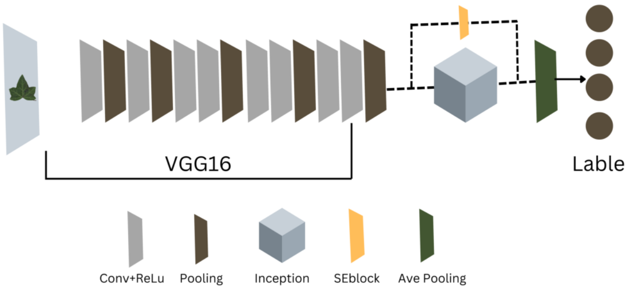


Fig. 4. The structure of the proposed convolutional neural network (CNN).

Which layers of the initial network must be frozen during the pre-training phase and which layers can continue to learn at a certain learning rate are determined by the VGG16 pre-training model. Since low-level characteristics are better able to adapt to a variety of conditions, the top few levels are often frozen. To train the model on our dataset, we used a stochastic gradient descent optimization technique. With momentum and weight attenuation set to 0.0005 and 0.9, respectively, the initial learning rate was set at 0.001. In our tests, the dropout layer was employed to avoid over-fitting training and improve the model's performance.

C. Statistical Method

The suitability of deep convolutional neural networks for the classification as described in the prior problem is assessed. We focus on two well-known designs that were created for the ImageNet dataset as part of the ”Large Scale Visual Recognition Challenge” (ILSVRC): AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015).

The LeNet-5 architecture from the 1990s's design pattern is followed by the AlexNet architecture (shown in Figure 1). (LeCun et al., 1989). Stacks of convolutional layers are often followed by one or more fully linked layers in the LeNet-5 architecture. A normalizing layer and a pooling layer may come after the convolution layers, if preferred. ReLu non-linear activation units are often used to link all network layers. Five convolutional layers, three fully linked layers, and a SoftMax layer make up AlexNet. Normalization and a pooling layer come after the first two convolution layers (conv1, 2), while only one pooling layer comes after the final convolution layer (conv5). With 38 outputs (same amount

TABLE I
CONVOLUTIONAL NEURAL NETWORK (CNN)-BASED MODEL SIMILARITY

FRAMEWORK

Type	Size	Output Size
Conv1 (Convolutional layer 1)	9 × 9/4	96 × 55 × 55
Max-pool1	3 × 3/2	96 × 27 × 27
Conv2	3 × 3/1	128 × 112 × 112
Max-pool2	3 × 3/1	128 × 56 × 56
Conv3	3 × 3/1	256 × 56 × 56
Max-pool3	3 × 3/1	256 × 28 × 28
Conv4	3 × 3/1	512 × 28 × 28
Max-pool4	3 × 3/1	512 × 14 × 14
Conv5	3 × 3/1	512 × 14 × 14
Max-pool5	3 × 3/1	512 × 7 × 7
Max-pool6	3 × 3/1	512 × 3 × 3
Inception	0	736 × 3 × 3
Max-pool7	3 × 3/2	736 × 1 × 1
Fully connection	0	256 × 1 × 1
Linear	0	10 × 1 × 1
Softmax	0	10

the total number of classes in our dataset), the last fully-connected layer (fc8) in our modified version of AlexNet feeds the SoftMax layer. The input from (fc8) is then exponentially normalized by the SoftMax layer, producing a distribution of values throughout the 38 classes that sum up to 1. These values show the network's perception of the classes that the supplied input picture belongs to. Each of AlexNet's first seven layers is connected to a ReLu nonlinearity activation unit. A dropout layer with a dropout ratio of 0.5 is connected to the first two fully connected levels (fc6, 7), in contrast. Though the network has somewhat fewer parameters (5 million parameters) than AlexNet, the GoogleNet design is more deeper and larger, with 22 layers (60 million parameters). The GoogleNet design makes extensive use of the "network in-network" architecture (Lin et al., 2013) in the form of inception modules. The inception module uses parallel 1 x 1, 3 x 3, and 5 x 5 convolutions, as well as a max-pooling layer, to simultaneously record a number of properties. The amount of labor required must be taken into account in terms of implementation practicality, which is why 1 x 1 convolutions are added for dimensionality reduction before the previously mentioned 3 x 3, 5 x 5, and also after the max-pooling layer. The outputs of all of these parallel layers are then simply concatenated in a filter concatenation layer. This is one of nine inception modules of the GoogLeNet architecture, which was employed in our research. This architecture's complete description may be found in (Szegedy et al., 2015).

In order to compare the performance of the two architectures on the Unauthorized dataset (BARI), we train the model from scratch in one case and then use transfer learning to modify already learned models (trained on the ImageNet dataset) in the other. The weights of the layer fc8 of AlexNet and the loss 1, 2, and 3 classifier layers of GooglLeNet are reset in the case of transfer learning. Then, unlike what is often done in transfer learning, we do not restrict the learning of any of the layers while training the model. To put it another way, the main difference between these two learning strategies (transfer vs. learning from scratch) is in the initial state of a few layers' weights, allowing the transfer learning strategy to take advantage of the substantial amount of visual knowledge already learned by the pre-trained AlexNet and GoogleNet models extracted from ImageNet (Russakovsky et al., 2015). Using the FDTD method to simulate wave propagation in a 2D environment, a band-pass filter (BPF) to filter out noise from the measured particle velocity of the shear wave, and then the Algebraic Helmholtz Inversion (AHI) algorithm to directly estimate CSM are the new approaches we suggest in this paper. The results of numerical simulations have demonstrated the usefulness of the recommended approach [15].

To summarize, In total of 45 experimental configurations, which depend on the following parameters:

For deep learning architecture:

- The AlexNet,
- The GoogLeNet.

Training mechanism:

- The Transfer Learning,
- The Training from Scratch.

Dataset type:

- Color,
- Grayscale,
- Leaf Segmented.

Training-testing set distribution:

- Train: 80%, Test: 20%, (In general case)
- Train: 60%, Test: 40%, (Random Case)
- Train: 50%, Test: 50%, (Random Case)
- Train: 40%, Test: 60%, (Random Case)
- Train: 20%, Test: 80%. (Rare case)

This article has made extensive use of the Architecture: Training Methodology: Datasets for Train-Test-Set-Distribution are related to certain experiments. We will use the term Google Net: 80-40 on the Transfer Learning: Grayscale to refer to the experiment employing the Google Net architecture, which was trained using transfer learning on the gray scaled BARI dataset with a train-test set distribution of 80:40. A total of 30 epochs, or the number of training iterations during which the neural network has fully traversed the whole training set, are executed for each of these 45 experiments. The decision to use 30 epochs was made based on the empirical finding that learning always converged within 30 epochs in all of these experiments (as demonstrated by the aggregated plots (Figure 2) for all of the trials).To enable a fair comparison between the outcomes of all the experimental configurations, we also attempted to standardize the hyper-parameters throughout all the experiments, using the following hyper-parameters for each experiment:

- The solver type: Stochastic Gradient Descent,
- Base learning rate is: 0.005,
- The learning rate policy: Step (decreases by a factor of 10 every 30/3 epochs)
- Momentum is : 0.9,

- The weight decay is: 0.0005,
- For gamma: 0.1,
- For batch size: 24 (in case of GoogLeNet), 100 (in case of AlexNet).

All of the aforementioned tests were conducted using our own branch of Caffe (Jia et al., 2014), a fast, open-source deep learning framework. Using a typical instance of Caffe, it is possible to reproduce the basic results, such as the overall accuracy [16].

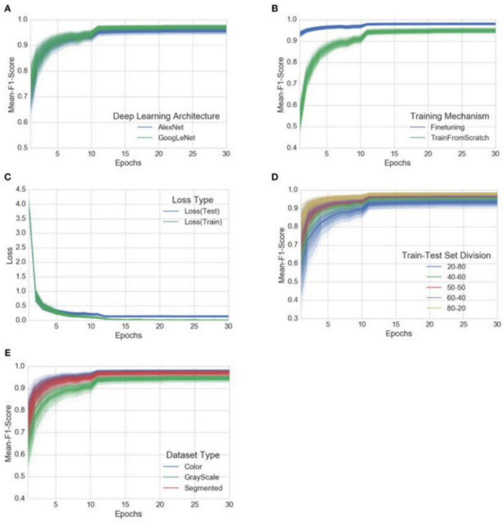


Fig. 5. The evolution of the mean F1 score and loss over the course of 30 training epochs, grouped by experimental configuration parameters.

RMSProp is chosen for our executing program for sparse reward problems.

IV. Required Technologies and Software

A. Deep Learning

Machine learning (ML), a subset of artificial intelligence, includes deep learning (DL). Artificial Intelligence is the expansive field of intelligent computer output. Machine learning focuses on developing self-learning computer algorithms that can access and utilize data. DL is a machine learning technique that uses many nonlinear transformations for data abstractions at the highest level and model designs.

B. Convolutional Neural Networks

One of the Deep Learning algorithms, a Convolutional Neural Network, can take an input image, evaluate distinct characteristics or objects in the image, and distinguish between them. Based on the properties of the design and invariance translation of their shared weights, these are also known as shift invariant or space invariant artificial neural networks (SIANN).

C. Object Detection

Object detection is a broad topic of deep learning. Object identification utilizing a convolution neural network, which is employed in many real-time areas. The deep learning neural network is used in profound learning to detect the object. Identifying items in deep learning basically involves recognizing objects in photos.

D. Tensor Flow

Tensor flow is an open source software library for high-accuracy numerical computing. Developed initially by Google Brain team researchers and engineers in the Google AI organization, it provides significant support for machine learning and in-depth learning, and the scalable digital computing base is employed in a variety of different scientific domains.

E. Keras

Keras is a Python-based high-level neural network API that can be used with TensorFlow, Theano, and CNTK. It was created with the goal of permitting. For simple experiments. Keras considers rapid and easy prototyping (through ease of use, measured quality, extensibility). Both recurrent and convolutional systems are strengthened. Runs on both the GPU and the CPU.

V. Proposed System

Digital cameras or similar technologies are used to capture photographs of various types of leaves, which are then utilized to determine the afflicted region in the leaves. Then, various image-processing techniques are used to them in order to analyse those photos and extract various and valuable information for further analysis.

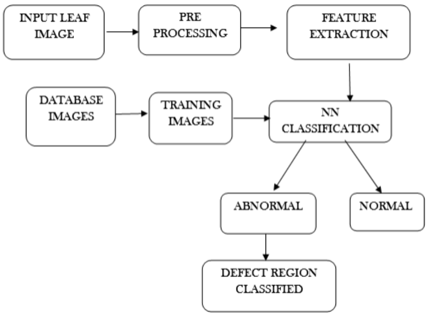


Fig. 6. Block diagram for leaf disease detection, and classification.

The following algorithm illustrates the step-by-step technique for the suggested picture recognition and segmentation processes:

- The first stage is picture acquisition, which includes capturing an image using a digital camera.
- Pre-processing the incoming image enhances image quality and removes unintended distortion. To achieve the appropriate image region, the leaf image is cropped. Then, the image is smoothed using the smoothing filter. To increase contrast, image augmentation is also employed.

- Many studies have utilized various image processing approaches to identify the leaf illness, including the following steps: picture acquisition, image preprocessing, image segmentation, feature extraction, and classification. Abirami Devaraj and colleagues [17].
- K Means clustering is used for segmentation, while CNN is utilized for classification. PCA is used in this study to decrease the feature set.
- Collect important segments for classifying leaf diseases. Using a genetic algorithm, segment the components. This paper is designed particularly for identifying leaf diseases. The work is separated into two key stages here. The ring project-based segmentation model is created in the first step to investigate the characteristics of leaf pictures.
- This paper is designed particularly for identifying leaf diseases. The work is separated into two key stages here. The ring project-based segmentation model is created in the first step to investigate the characteristics of leaf pictures.

VI. Dataset Details and Results

All the experiments are performed in Jupyter python, took 45 images for experiment. 19 images were root rot lentil and the others were healthy. Divided them into three sectors which are Training, testing and validation data set. Sample is lentil leaf images, after segmentation images will be classified which is root rot and which is healthy leaf. Firstly, read the images in our code, after then divided the training and validation dataset into two classes, then set the model CNN and library Keras layers, Maxpooling 2D, preprocessing image generator, optimizer RMSprop, activation “relu”,”sigmoid” function. To get better accuracy metrics set the loss model binary cross entropy. To model fit epochs 30. Then observation of the training model our model could predict which image is root rot and which is healthy from the validation dataset. We could use ARIMA and LSTM model for time series prediction [18].

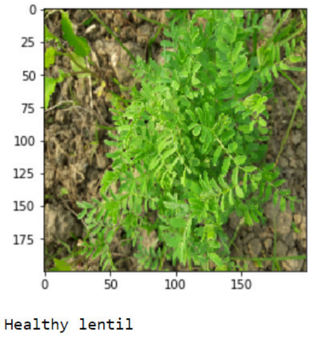


Fig. 7. Prediction output of healthy lentil leaf.

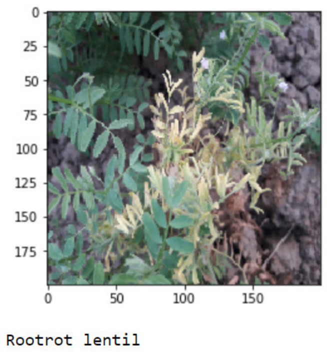


Fig. 8. Prediction of output Root Rot lentil leaf.

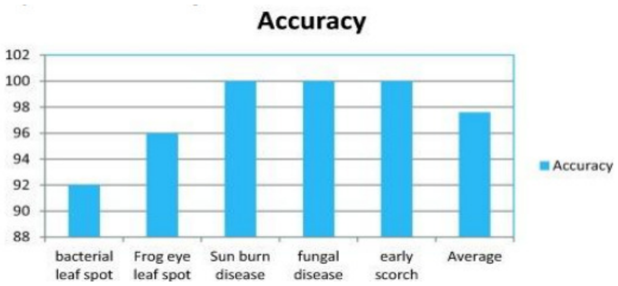


Fig. 9. Classification results per class for the proposed method.

Using the best model to accurately predict the right class label (i.e., leaf and disease information) from among two probable class labels, we obtained an overall accuracy of 31.40 percent in dataset 1 and 31.69 percent in dataset 2. A random classifier will have an average accuracy of just 2.63 percent, it is important to note. In 52.89 percent of cases in dataset 1 and in 65.61 percent of cases in dataset 2, respectively, the correct class was among the top five guesses.

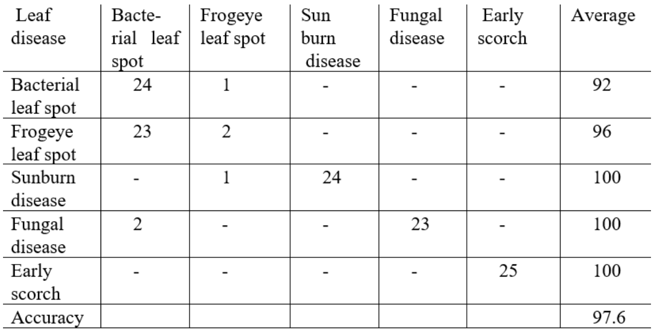
GoogLeNet: Segmented: TransferLearning:80-20 was the best model for dataset 1 while;

GoogLeNet: Color:TransferLearning:80-20 was the best model for dataset 2.

Figures 7 and 8 show an example image from these datasets, as well as its depiction of activations in the first levels of an AlexNet architecture.

TABLE II
Plant Leaf Disease Dataset Demography

Class	Training images(before data augmentation)	Training images(after data augmentation)	Number of testing image
Leaf disease	20	18	2
Bacterial leaf spot	25	22	0
Frog eye leaf spot	33	7	1
Sun burn disease	28	14	5
Fungal disease	32	18	4
Early scorch	33	19	3



Leaf disease	Bacterial leaf spot	Frogeye leaf spot	Sun burn disease	Fungal disease	Early scorch	Average
Bacterial leaf spot	24	1	-	-	-	92
Frogeye leaf spot	23	2	-	-	-	96
Sunburn disease	-	1	24	-	-	100
Fungal disease	2	-	-	23	-	100
Early scorch	-	-	-	-	25	100
Accuracy						97.6

Fig. 10. Proposed method showing the grouping results by means of class.

TABLE III
PLANT LEAF DISEASE DATASET DEMOGRAPHY

Model	Validation Accuracy	Test Accuracy
CNN model and drop out	99.21%	99.32%
CNN model and L2 Regularization	98.62%	98.73%

Data source is collected from Bangladesh Agriculture Research Institute(BARI), the image was collected two categories one is root rot lentil leaf other is healthy lentil leaf. The data set is divided 75% for training dataset, 10% for validation and 15% for testing.

This paper outlines the key image processing techniques used for identifying leaf diseases, which include k-means clustering and CNN. This method can greatly aid in the precise diagnosis of leaf disease. Picture capture, image preprocessing, segmentation, feature extraction, and classification are the five procedures for identifying leaf diseases. By calculating the quantity of illness present in the leaf, we may use an appropriate amount of insecticides to successfully control the pests, increasing crop output. We may extend this strategy by employing other segmentation and classification algorithms.

VII. CONCLUSION

Object detection using a convolution neural network is widely employed in the emerging generation. These have a variety of applications in medicine and agriculture. The number of convolution layers can be increased or decreased to enhance accuracy. CNN validation is a type of production environment in which we may test and improve accuracy. Following the accuracy measurements in the test model and the final stage, the product implementation stage will begin. The suggested method was created with the benefits to farmers and the agricultural sector in mind. The proposed technology can identify plant illnesses and deliver disease-fighting treatments. A proper understanding of the disease and the remedy may be utilized to improve the plant's health. The suggested method is Python-based and has an accuracy of roughly 98 percent for each ailment diagnosed. Using Google's GPU for processing can improve accuracy and speed. The technology may be put on Drones to allow for aerial surveillance of agricultural areas. The project's major goal is to lessen the burden on farmers by identifying the diseases that have attacked their crops in the early stages before applying the wrong pesticide to them, causing further crop damage. Using our algorithm, we may integrate new plant disease combinations to broaden the system's scope and apply it to a variety of crops. Furthermore, the suggested technique provides a less expensive solution to reducing crop loss and educating farmers. Because Bangladesh is a country that is heavily reliant on agriculture, i.e., our agricultural exports make up a significant portion of the country's GDP, we seek to preserve the crops by presenting this model in light of the current agricultural environment in Bangladesh.

REFERENCES

[1] Dhaygude, S. B. and Kumbhar, N. P., Agricultural plant leaf disease detection using image processing. Int J Adv Res Electr Electron Instrum 3: 2(1), 2013

[2] S. Arivazhagan, R. Newlin Shebiah, S. Ananthi, S. Vishnu Varthini, Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. Agric Eng Int: CIGR Journal. 15(1): 211-217, 2013

[3] Kulkarni Anand H, Ashwin Patil RK. Applying image processing technique to detect plant diseases. Int J Mod EngRes. 2(5):3661–4, 2012

[4] Savita N. Ghaiwat, Parul Arora, Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review, IJRAET, Vol 2, Issue 3, 2014

[5] Sabah Bashir, Navdeep Sharma, Remote Area Plant Disease Detection Using Image Processing, IOSR Journal of Electronics and Communication Engineering, Volume 2, Issue 6, PP 31-34, 2012

[6] Shin HC, Roth HR, Gao M, Lu L, Xu Z, Nogues I, Yao J, Mollura D, Summers RM, Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics, and Transfer Learning. IEEE Trans Med Imaging. 35(5):1285-98, 2016

[7] B. J. Thompson, E. Dougherty, Mathematical Morphology in Image Processing (1st ed.), 1993

[8] B. Bhanu and Jing Peng, Adaptive integrated image segmentation and object recognition, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),427-441,2000

[9] QT. Hoang, N.C. Minh, DT. Tran, Designing a Remote Monitoring System for Lakes Based on Internet of Things. In: DT. Tran, G. Jeon, T.D.L. Nguyen, J. Lu, TD. Xuan, Intelligent Systems and Networks. ICISN; vol 243, 2021

[10] A. S. M. A. Akib, S. Mahmud and M. F. Mridha, Future Micro Hydro Power: Generation of Hydroelectricity and IoT based Monitoring System, Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021, pp. 298-302

[11] M. R. Badnakhe and P. R. Deshmukh, An Application of K-Means Clustering and Artificial Intelligence in Pattern Recognition for Crop Diseases", International Conference on Advancements in Information Technology, vol.20 (2011)

[12] Smita Naikwadi, Niket Amoda, Advances In Image Processing For Detection Of Plant Diseases, International Journal Of Application Or Innovation In Engineering and Management, 2(11), 2013

[13] B. Sanjay, Leaf disease severity measurement using image processing,in International Journal of Engineering and Technology, 3(5), 297-301, 2011

[14] P. Chaudhary, A. K. Chaudhari, A.N. Cheeran, Sharda Godara, Color Transeform Based Approach for Disease spot Detection on Plant Leaf,IJCST, 3(6), 2012

[15] T. Quang-Huy, V. T. Duong, L. Q. Hai and T. Duc-Tan, Image Reconstruction Utilizing Algebraic Helmoltz Inversion and Passband Filtering Applied to Viscoelasticity, International Conference on Multimedia Analysis and Pattern Recognition (MAPR), 1-5, 2020

[16] N. C. Minh, T. H. Dao, D. N. Tran, N. Q. Huy, N. T. Thu and D. T. Tran, Evaluation of Smartphone and Smartwatch Accelerometer Data in Activity Classification, 8th NAFOSTED Conference on Information and Computer Science (NICS), 33-38, 2021

[17] A. Devaraj, K. Rathan, S. Jaahnavi and K. Indira, Identification of Plant Disease using Image Processing Technique, International Conference on Communication and Signal Processing (ICCSP), 0749-0753, 2019

[18] N. Tasnia, S. Mahmud and M. F. Mridha, COVID-19 Future Forecasting Tool: Infected Patients Recovery and Hospitalization Trends Using Deep Learning Models, International Conference on Science and Contemporary Technologies (ICSCT), 1-6, 2021

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 235–241 DOI: 10.15439/2022R06 ISSN 2300-5963 ACSIS, Vol. 33

An XAI-Based Deep Learning Framework for Coronary Artery Disease Diagnosis using SPECT MPI polar map images

1st Ton Do Thanh
Faculty of Information Technology
Hung Yen University of Technology and Education
Hungyen, Vietnam
thanhtonvk@gmail.com

2nd Chi Thanh Nguyen*
Institute of Information Technology, AMST
Hanoi, Vietnam
thanhnc80@gmai.com
*Corresponding author

3rd Nhu Hai Phung
Institute of Information Technology, AMST
Hanoi, Vietnam
hainda59@gmail.com

4th Van-Hau Nguyen
Hung Yen University of Technology and Education
Hungyen, Vietnam
nvhau66@gmail.com

5th Trung Kien Tran
Institute of Information Technology, AMST
Hanoi, Vietnam
t2kien@gmail.com

6th Thanh Trung Nguyen
Department of Medical Equipment 108 Military Central Hospital, Hanoi
Hanoi, Vietnam
thanhtrungys@yahoo.com

***Abstract*—Our study aimed to develop an explanatory method for predicting Coronary Artery Disease (CAD) classification using spect images. As we all know, deep neural networks usually consist of many layers connected to each other through interlocking network nodes. Even if we check the classes and describe their relationships, it is difficult to understand entirely how active neural networks make predictions. Therefore, deep learning is still considered a "Black box". Existing XAI (eXplainable Artificial Intelligence) approach can provide insights into the inside of a Deep Learning model allowing for transparency and interpretation. Our previous research helps doctors diagnose the CAD of patients by developing deep learning models using a multi-stage transfer learning framework. The model achieved 0.955 accuracy, 0.932 AUC, 0.944 sensitivity, and 0.889 specificity, showing effective performance. Our dataset includes 218 SPECT images from 218 imported patients collected at 108 Hospital in Hanoi, Vietnam. In this paper, We propose an explainable Deep Learning framework using three popular XAI approaches: LIME, GradCam, and RISE. These XAI approaches are effective tools for interpreting the prediction of deep learning models. We evaluate the effectiveness of the interpretation by visualizing the explained regions and using improved deletion and insertion with a threshold limit suitable for Binary Classification. The experiment results show that our model effectively diagnoses CAD and provides medical interpretation. Furthermore, the proposed method for evaluating the deletion and insertion metrics is considered more efficient for binary classification than the traditional metrics.**

***Index Terms*—XAI, Transfer Learning, Deep Learning, LIME, GradCAM, RISE**

I. Introduction

Coronary artery disease (CAD) is the most common form of cardiovascular disease in the elderly. It is one of the leading causes of death in the world. In Vietnam, the rate of patients dying from coronary heart failure is relatively high, accounting for 11% - 36%, and is showing signs of increasing yearly.

Nowadays, SPECT-MPI is a remarkably effective method concerning the diagnosis of CAD [9]. This approach provides 3D information about the distribution of radioactive compounds in the heart, reducing the number of unnecessary angiograms and allowing for appropriate treatment planning. MPI is a noninvasive imaging modality where injected radiopharmaceutical uptake is measured using SPECT to diagnose CAD [12]. Research in CAD diagnostic support using SPECT MPI has been carried out a lot. Some studies such as [1] and [2] use Machine Learning (ML) algorithms such as Adaptive Boosting, Gradient Boosting, Random Forests, and Xgboost. Although these methods achieve >90% accuracy, it requires a large amount of data and has to go through many processes before the doctor can conclude whether he has CAD. The emergence of Deep Learning (DL) has recently become more widely used in CAD diagnosis. This trend gradually improves the training time and cost significantly compared to ML. Therefore, several DL-based CAD diagnostic models have been developed and achieved promising results. Papandrianos et al. [2] improved the model using DL. Specifically, the accuracy reaches 94.58%.

Deep learning has made significant progress in image analysis and prediction. Image Analysis was often introduced using a human design system in the past. In CAD diagnosis using SPECT MPI image, the doctor will have to analyze the image and predict whether the patient has the disease or not based on the area of the lesion. In deep learning, this is learned by neural networks to optimize the input. However, neural networks usually consist of many layers that are connected through interlocking network nodes. Even if we examine the classes and describe their relationships, it is difficult to understand how all active neural networks make predictions. Before that, deep learning was still considered a "Black box". It is this that makes it impossible for users to trust the model without knowing how its predictive logic works, whether it is similar to expert diagnostic logic or not. It also cannot explain to the patient why he has CAD. It is for these reasons that eXplainable Artificial Intelligent (XAI) was born to solve those problems.

By the 1990s, researchers had begun to study whether extracting the rules generated by neural networks was possible. [9] Researchers have created neural network-based decision-making for experts to figure out how to develop explanations to allow the technology to become more reliable.

Recently there have been studies to do "Black boxes" transparent, They include Decision Trees, Bayesian Networks, Linear Models, etc. The Association of Computing Machinery Conference on Fairness, Accountability, and Transparency (ACM FAccT) was established in 2018 to study trust and explainability in engineering and AI systems.

It can be said that the concept of XAI has been around for a long time, but due to technical limitations, it has not been developed. Recently, when advanced CNN network models and technologies appeared, XAI was developed and promoted its effectiveness because the more accurate the DL model, the more complex it is and the easier it is. There are many popular XAI methods such as LIME [10], SHAP [11], Influence Functions [12], Integrated Gradient [13], Grad-Cam [15], RISE [14], etc. In this study, we propose an explainable DL Framework using three methods: Grad-Cam, LIME, and RISE. The main contributions of this paper are the following:

- Multiple interpretation methods are used to interpret the model in many respects.
- We evaluate the effectiveness of XAI methods in detail for each label (CAD and NonCAD) on aspects: visualization, an improved deletion algorithm, and an improved insertion algorithm.

The organization of this paper is structured as follows: Section 2 reviews the work related to CAD interpretations: GradCam, RISE, and LIME. Section 3 describes the methodology of the usage model and the XAI methods. Section 4 presents the results of the interpretation. Section 5 compares the effectiveness of the XAI methods, the work involved, and the paper conclusions of the paper.

II. RELATED WORK

Nikolaos I.Papandrianos et al. [16] used transfer learning with the VGG16 network for CAD classification problems. The study used a dataset that included cases from 625 patients as representative of stress and rest, including 127 infarct, 241 ischemic, and 257 normal cases that were disproved. Result: The model achieves 93.3% accuracy and 94.58% AUC. The explanation for the model using GradCAM-based color visualization.

Liu et al. [17], approximately DL to help improve the accuracy of CAD diagnosis. There were 37243 tested patients used in this study and SPECT MPI images were extracted from the records. Also, clinical data including BMI, sex, height, etc. They used handover learning with the Resnet-34 network architecture. The DL model was evaluated by 5-fold cross-comparison. The obtained AUC result is 0.872 ± 0.002. The result is quite positive with such a large data set.

Apostolopoulos et al. [18] built a DL Hybrid Random Forest model and used polar map images and clinical data to classify CAD. Compare predictions and results from nuclear experts. This model was evaluated by cross-validation 10 times and achieved an accuracy 0.7915, specificity 0.7925, and sensitivity 0.7736. The result is not really good.

Our previous research [3] used 218 SPECT images from 218 patients in the Department of Nuclear Medicine of 108 Hospital, Hanoi, Vietnam. With a number of CAD: 112 (51.37%) and a number of non CAD: 106(48.63%). The dataset has been processed and consulted by experts and doctors. We proposed a multi-stage transfer learning framework and evaluated the proposed framework by 15 pre-trained deep CNNs, well-trained on ImageNet (more than 1 million images for 1000 classes). Since the dataset is small, we used Global Average Pooling (GAP) instead of Flatten function at the end of the feature extraction to prevent overfitting. Furthermore, the GAP is more native to the convolution structure by enforcing correspondences between feature maps and categories so the feature maps can be easily interpreted as category confidence maps. We used all the most popular pre-trained CNNs such as VGG, Xception, EfficientNet, Inception, DenseNet, ResNet, and the deeper version (VGG19, ResNet152V2,...). The results demonstrated that all the pre-trained CNNs models performed very well, with Acc > 86.4%. Moreover, we found that ResNet152V2-based model showed the best performances for all metrics: Acc 95.5%, AUC 93.2%, Sen 94.4%, Pre 96.4%, and F1-score 95.2%. To interpret the model, we used CAM (Classification Activation Model).

Overall, in the above research, several studies have used XAI to apply in the interpretation of CAD. However, the explanation just has intuitive and is not deep, as well as the application of different explanatory methods to compare the effectiveness of the research, making the explanatory models unreliable. Therefore, further studies and experiments are needed in this area. With the traditional metrics insertion and deletion methods[20], the weights fluctuate from 0 to 1. This is only effective for the multi-label model because when deleting or inserting features, the probability will be divided among other labels. As for binary labels, when deleting or inserting, the probability of the label is not evenly divided, but there will be a preference for a label. Therefore, we propose the metrics deletion and insertion method using the threshold limit according to the area of the explanatory region.

III. MATERIAL AND METHODS

A. CAD Dataset

This paper is a follow-up to our previous work [3]. So we use the same data set to study for this paper, the data includes SPECT MPI polar maps from 218 patients collected in the Department of Nuclear Medicine of 108 Hospital, Hanoi, Vietnam. This dataset was obtained after a processing procedure and consultation with many technicians and doctors.

B. Proposed methods

As described in Fig.1, the methodological flow includes the following four parts: (1) Loading the dataset and preprocessing; (2) Training model with Transfer Learning; (3) Explaining the model with LIME, Grad-Cam, and RISE; (4) Visualize, Metrics Insertion and Deletion. The following is a detailed explanation for each part:

1) Loading dataset and preprocessing

The dataset has been preprocessed and divided into a training set consisting of 174 train images and 44 test images.

2) Traning model with Transfer Learning

This paper uses the Resnet152V2-based model, which showed the best performance in the previous research [3]. The results obtained on the test are Acc 95.5%, AUC 93.2%, Sen 94.4%, Pre 96.4%, and F1-score 95.2%. We use XAI methods to interpret this model and evaluate it against metrics.

3) Explain Model with LIME, Grad-Cam and RISE

a) LIME

LIME [10] is a post hoc method by making small increments instead of explaining the whole thing, we bring the facts to the local and show which part of the data has the most influence on our model predictions.

Algorithm 1 Sparse Linear Explanations using LIME

```
Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x Length of explanation K
        Z ← {}
        For i э {1,2,3,...N} do
                Z'I ←sample_around(x')
                Z← Z U (z'_I,f(z_i), π(z_i))
        end for
        w← K-Lasso(Z,K)
        return w
```

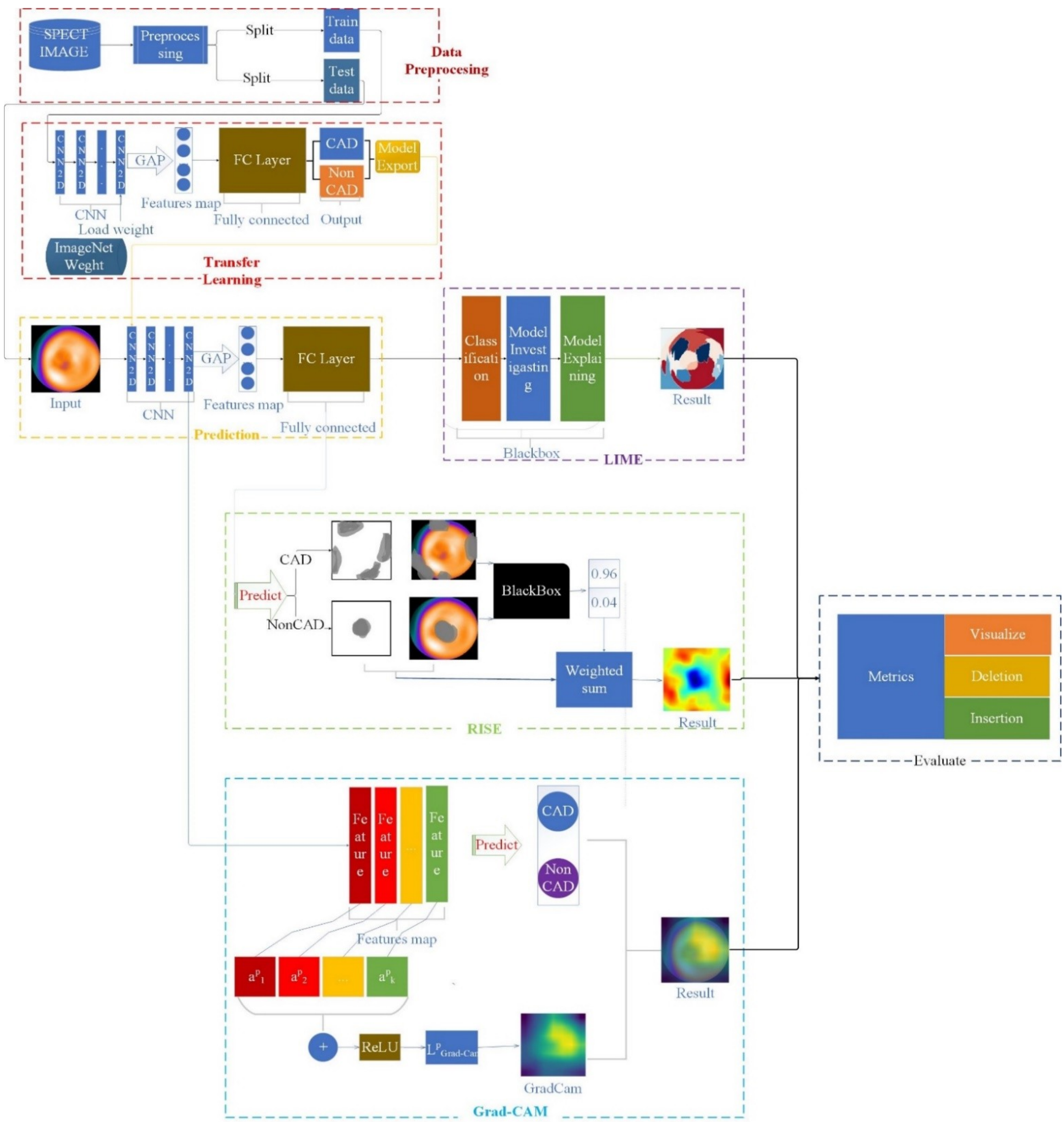


Fig. 1. Detailed explainability approach using LIME, Grad-Cam, and RISE for CAD Classification

b) GradCAM

GradCam [15] is a generalized instance of the CAM used in [10], which can be used with any network architecture. The idea of Grad-CAM is similar to CAM, and we use spatial information contained in convolution layers to understand which regions are essential for a particular class. Grad-CAM can be used on any feature maps from any layer in the network, but to demonstrate the efficiency and explanatory power of the model, the GradCAM paper uses the last convolution layer because the last convolution layer usually contains the most information:

$$L^{p}_{\text{Grad-CAM}} = ReLU(\Sigma_k a^{p}_{k} A^{k}) \qquad (1)$$

where A^k is the features map.

c) RISE

RISE [14] explains by measuring the importance of an image region as obscuring or 'contaminating' that region and observing how this affects black-box decisions. This is done by setting the pixel's intensity to zero, which opens the region or adds noise. RISE models by multiplying an image by [0,1] value masked. From there, the area affected by the decision of the model will be determined.

$$S_{I,f}(\lambda) \overset{\text{MC}}{\approx} \frac{1}{\mathbb{E}[M]\cdot N}\sum_{i=1}^{N} f(I \odot M_i)\cdot M_i(\lambda). \qquad (2)$$

4) Visualization, Metrics Insertion and Deletion

In this section, we present the interpretations of LIME, GradCAM, RISE and evaluate the effectiveness of the interpretation.

We propose two auto-evaluation metrics to evaluate the interpretation effect: deletion and insertion, researched by [20]. We change the threshold of the algorithm. Instead of a fixed value from 0 to 1, we use early stopping according to the weighted ratio of the interpreted positive area.

a) Deletion

In deletion, the metric is the removal of the model's decision-making agents. Specifically, this metric will measure the prediction's probability reduction as superpixels have been removed, which will be removed based on heatmap importance. So, the stronger reduction in probability means a better explanation.

Algorithm 2 Traditional Deletion to compute deletion score

```
Procedure Deletion
        Input: black box $f$, image $I$, importance map $S$, number of
        pixels $N$ removed per step
        Output: deletion score $d$
        $n \leftarrow 0$
        $h_n \leftarrow f(I)$
        while $I$ has non-zero pixels do
                According to $S$ ,set next $N$ pixels
                in $I$ to 0
                $n \leftarrow n+1$
                $h_n \leftarrow f(I)$
        $d \leftarrow$ AreaUnderCurve($h_i$ vs $i/n$, $\forall$I = 0,…$n$)
        return $d$
```

Algorithm 3 Proposed Deletion to compute deletion score

```
Procedure Deletion
        Input: black box $f$, image $I$, importance map $S$, number of
        pixels $N$ removed per step
        Output: deletion score $d$
        $n \leftarrow 0$
        $threshold \leftarrow Area(S)/Area(I)$
        $h_n \leftarrow f(I)$
        while $I$ has $threshold$ pixels do
                According to $S$ ,set next $N$ pixels in $I$ to
                $threshold$
                $n \leftarrow n+1$
                $h_n \leftarrow f(I)$
        $d \leftarrow$ AreaUnderCurve($h_i$ vs $i/n$, $\forall$I = 0,…$n$)
        return $d$
```

b) Insertion

In insertion, the process is made the opposite of deletion, which is the addition of superpixels. In addition, instead of the initialization being erased, they will be superimposed on a kernel layer to blur. Then delete the kernel blur to reveal the critical area of the interpretation. Insertion measures the increase of the prediction probability when superpixels are included, with the larger the area of the probability, the better explanation.

Algorithm 4 Traditional Insertion to compute deletion score

```
Procedure Insertion
        Input: black box $f$, image $I$, importance map $S$, number of
        pixels $N$ removed per step
        Output: insertion score $d$
        $n \leftarrow 0$
        $I' \leftarrow Blur(I)$
        $h_n \leftarrow f(I)$
        while $I \neq I'$ do
                According to $S$ ,set next $N$ pixels in $I'$ to
                corresponding pixels in $I$
                $n \leftarrow n+1$
                $h_n \leftarrow f(I)$
        $d \leftarrow$ AreaUnderCurve($h_i$ vs $i/n$, $\forall$I = 0,…$n$)
        return $d$
```

Algorithm 5 Proposed Insertion to compute deletion score

```
Procedure Insertion
        Input: black box $f$, image $I$, importance map $S$, number of
        pixels $N$ removed per step
        Output: insertion score $d$
        $n \leftarrow 0$
        $I' \leftarrow Blur(I)$
        $h_n \leftarrow f(I)$
        $threshold \leftarrow Area(S)/Area(I)$
        threshold
        while $I \neq I'*threshold$ do
                According to $S$ ,set next $N$ pixels in
                $I'*threshold$ to corresponding pixels in $I$
                $n \leftarrow n+1$
                $h_n \leftarrow f(I)$
        $d \leftarrow$ AreaUnderCurve($h_i$ vs $i/n$, $\forall$I = 0,…$n$)
        return $d$
```

IV. RESULT AND DISSCUSIONS

A. Results

1) Explanations

a) LIME

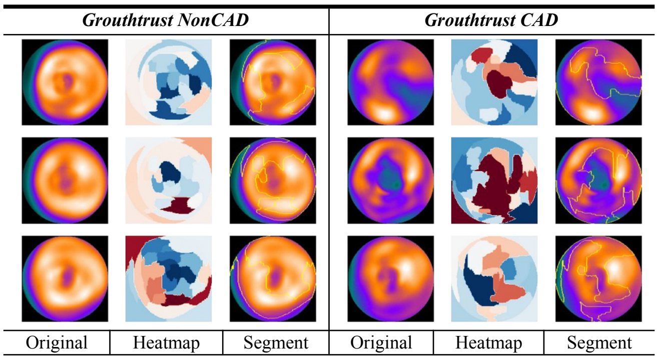


Fig. 2. Visualization LIME technique

Fig. 2 presents some visualization for LIME, "Original" original SPECT MPI images; "Heatmap" describe the areas of explanation and influence on the model prediction. The green area is positive, the dump area is negative, and "Segment" visualizes the results generated by LIME.

For NonCAD interpretation, looking at the heatmap and segment, we can see that the blue area (for heatmap) or circled (for the segment) is the explanation area for NonCAD prediction, which is relatively accurate when describing the bright color SPECT image area and for CAD describing into the dark SPECT image area (the damaged area). Thus, LIME is very effective for our model.

b) GradCAM

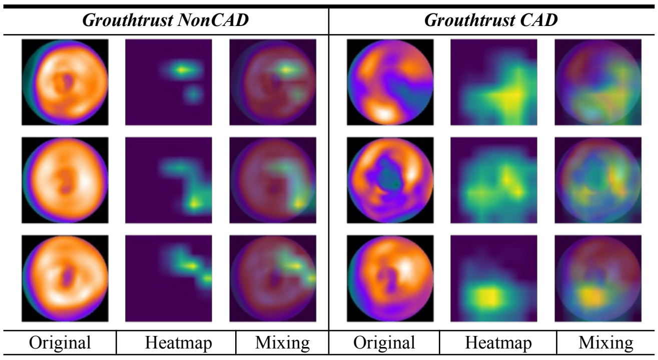


Fig. 3. Visualization GradCam technique. “Original” original SPECT MPI images; “Heatmap” explanation of GradCam, “Mixing” visualized result generated by Grad-CAM

c) RISE

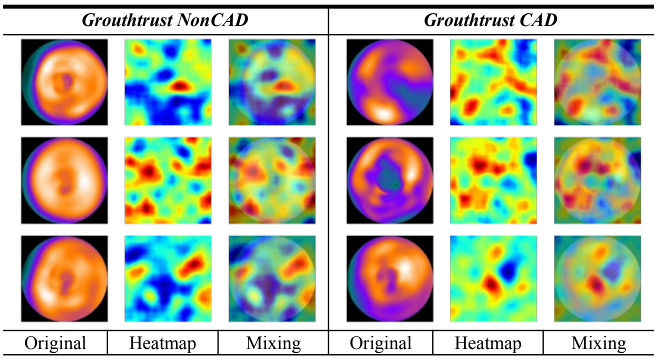


Fig. 4. Visualization RISE technique

Fig. 4 presents the explanations for RISE. "Original" original SPECT MPI images; "Heatmap" explanation of GradCam, "Mixing" visualized result generated by RISE. The RISE Explanation method is very effective for the model.

The interpretation is highlighted by hot color. NonCAD is very standard when looking at the bright areas of the spectrum, and for CAD, the interpreted area is relatively large and is the SPECT image's dark (damaged area).

2) Metric

Here we evaluate the deletion and insertion for two methods: traditional [20] and our improved method. The vertical y-axis describes the prediction probability of the label being indexed. The horizontal x-axis describes the area of the area to be deleted and inserted, with traditional from 0 to 1; and our method from 0 to threshold (all scaled to 0-1). Furthermore, we separate the Grouthtrust into two columns for evaluation: CAD and NonCAD. The presentation includes Original Image, Deletion/Insertion game (This is the metric we use to evaluate the effectiveness of interpretation), and Mixing (visualized results generated by the interpretation method).

a) LIME

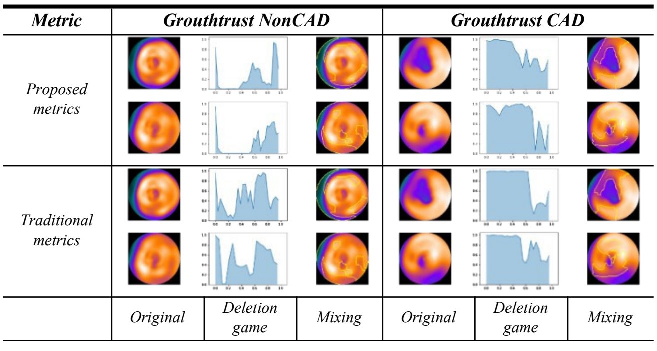


Fig. 5. Visualization LIME Deletion Technique; "Original" original SPECT MPI images;”Deletion game” visualization for prediction probability using deletion algorithm; “Segment” visualized results generated by LIME

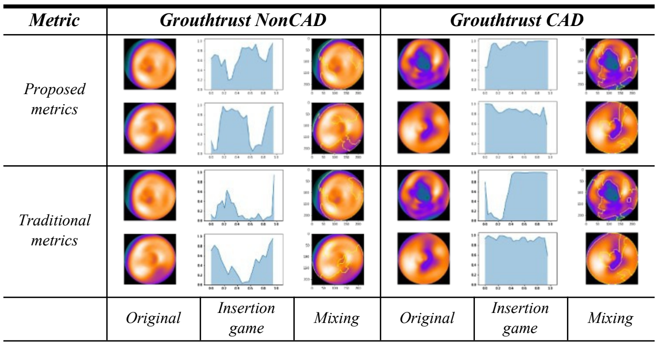


Fig. 6. Visualization LIME Insertion Technique; "Original" original SPECT MPI images; "Insertion game" visualization for prediction probability using insertion algorithm; "Segment" visualized results generated by LIME

b) GradCAM

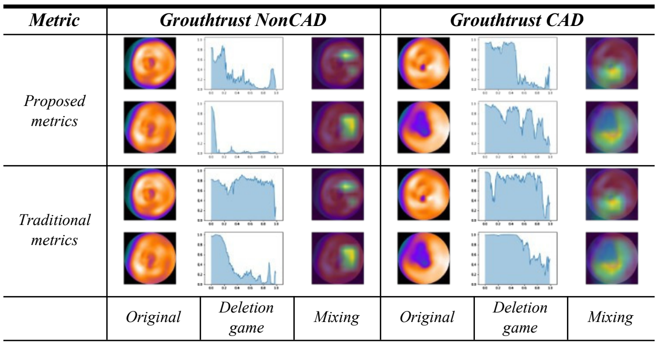


Fig. 7. Visualization GradCAM Deletion Technique; "Original" original SPECT MPI images; "Deletion game" visualization for prediction probability using insertion algorithm; "Mixing" visualized results generated by GradCAM

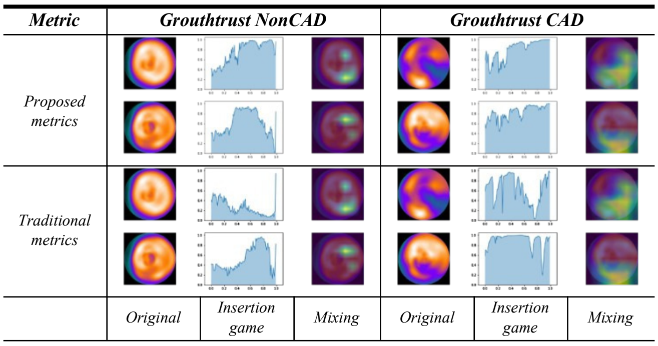


Fig. 8. Visualization GradCAM Insertion Technique; "Original" original SPECT MPI images; "Insertion game" visualization for prediction probability using insertion algorithm; "Mixing" visualized results generated by GradCAM

c) RISE

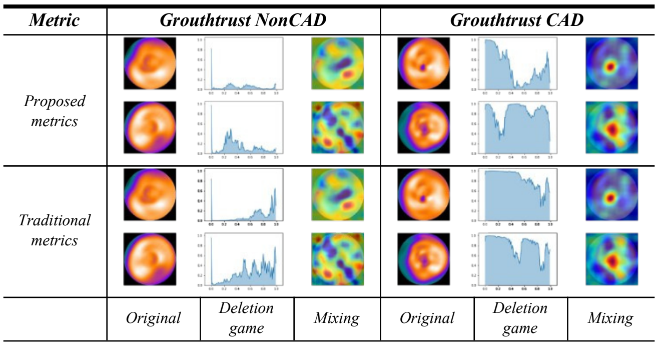


Fig. 9. Visualization RISE Deletion Technique; “Original” original SPECT MPI images; "Deletion game" visualization for prediction probability using insertion algorithm; "Mixing" visualized results generated by RISE

By applying the proposed algorithm, we have seen its effectiveness. In the deletion, instead of deleting all the features of the image, we only erase up to the threshold, and in the final result, the probability of the label scoring is 0.2-0.3 instead of 0.4 for NonCAD labels and 0.6 for CAD labels. In addition, the area of the Deletion game of the proposed algorithm is smaller than that of the traditional algorithm.

According to the traditional method, the image before being inserted has all the features deleted, and according to the proposed method, the image is deleted only in the area to be explained. The area of the Insertion game of the proposed algorithm is bigger than that of the traditional algorithm. The details are evaluated in Table I.

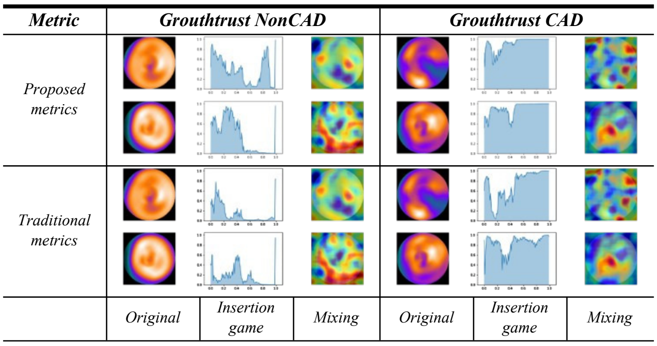


Fig. 10. Visualization RISE Insertion Technique; "Original" original SPECT MPI images; "Insertion game" visualization for prediction probability using insertion algorithm; "Mixing" visualized results generated by RISE.

TABLE I. EVALUATES THE AVERAGE OF INSERTIONS AND DELETIONS OF THREE XAIS, COMPARING THE PERFORMANCE OF TRADITIONAL AND OUR IMPROVED METRICS ON THE TEST DATASET, INCLUDING 44 IMAGES.

XAI	Deletion		Insertion	
	Traditional	Improved	Traditional	Improved
LIME	0.55	**0.51**	0.53	**0.58**
GradCAM	0.52	**0.36**	0.61	**0.72**
RISE	0.41	**0.38**	0.57	**0.63**

B. Discussions

In this study, we have developed three explanatory methods for the Resnet152V2-base model from our previous research [3]. As can be seen, CNNs do not provide transparency and interpretability in their decisions, which is an essential obstacle to their full integration into medical image analysis. Therefore, doctors cannot rely on the predictions provided. We have implemented LIME, GradCAM, and RISE techniques to solve it, generating heatmaps for interpretation. And the methods we used to give reasonably well-explainable results are presented in SPECT MPI images, dark areas are CAD areas, and light and light areas are NonCAD areas.

In addition, we also implement two more techniques to improve Delete and Insert by changing the threshold to interpret the three methods LIME, GradCAM, and RISE. It has overcome the disadvantages of traditional deletion and insertion methods for binary classification interpretation.

In summary, all three interpretation methods for export can interpret images for CAD using SPECT MPI images. Overall, the present study constitutes an innovation in understanding image classification models and evaluating deletion and insertion metrics with promising results.

V. CONCLUSION

The proposed paper presents efforts known to develop an interpretable path to CAD diagnosis using SPECT MPI imaging and sophisticated modern interpretive and DL techniques. In addition to deploying a highly accurate model from research [3], it is necessary to address the ability to interpret images through visualization. For this study, the effectiveness of the LIME, GradCAM, and RISE interpretation tools was investigated and yielded promising results for the automatic and accurate diagnosis in nuclear cardiology. Doctors can use visualization techniques LIME, GradCAM, and RISE to compare and make effective and confident decisions, taking advantage of the visual explanations provided. Therefore, LIME, GradCAM, and RISE methods have been proven to be effective tools in providing explanations for CNN-based decisions in SPECT MPI images.

In conclusion, this study contributes to the effective diagnosis of coronary artery disease. Thus it will promote confidence in using an interpretable artificial intelligence model for diagnosis in nuclear medicine.

REFERENCES

[1] Erito Marques de Souza Filho, Fernando de Amorim Fernandes, Christiane Wiefels, Lucas Nunes Dalbonio de Carvalho et al., “Machine Learning Algorithms to Distinguish Myocardial Perfusion SPECT Polar Maps,” 2021 Nov.

[2] Nikolaos I Papandrianos, Anna Feleki, Elpiniki I Papageorgiou, Chiara Martini, “Deep Learning-Based Automated Diagnosis for Coronary Artery Disease Using SPECT-MPI Images,” 2022 July.

[3] P. N. Hai, N. C. Thanh, N. T. Trung, T. T. Kien, “Transfer Learning for Disease Diagnosis from Myocardial Perfusion SPECT Imaging,” Computers, Materials and Continua, Vol.3. pp. 5925-5941, 2022 July.

[4] Fagan, L. M.; Shortliffe, E. H.; Buchanan, B. G. (1980), “Computerbased medical decision making: from MYCIN to VM. Automedica,” Heuristic Programming Project, Departments of Medicine and Computer Science Stanford University, Stanford, California.

[5] Alizadeh, Fatemeh (2021). "I Don't Know, Is AI Also Used in Airbags?: An Empirical Study of Folk Concepts and People's Expectations of Current and Future Artificial Intelligence". Icom. 20 (1): 3–17.

[6] Brown, John S.; Burton, R. R.; De Kleer, Johan, “Pedagogical, natural language, and knowledge engineering techniques,” SOPHIE I, II, and II. Intelligent Tutoring Systems. Academic Press, Vol. 4, pp. 98-111, 2016.

[7] Bareiss, Ray; Porter, Bruce; Weir, Craig; Holte, Robert, Protos, “An Exemplar-Based Learning Apprentice. Machine Learning,” Morgan Kaufmann Publishers Inc, Vol. 3, pp. 112–139, 2019.

[8] Bareiss, Ray, “Exemplar-Based Knowledge Acquisition: A Unified Approach to Concept Representation, Classification, and Learning. Perspectives in Artificial Intelligence,” 2001.

[9] Tickle, A. B.; Andrews, R.; Golea, M.; Diederich, J. ,”The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural network,” IEEE Transactions on Neural Networks, Vol.5. , No. 10-12, pp 1057-1068, 2018 Nov.

[10] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, “Why Should I Trust You?” Explaining the Predictions of Any Classifier,” Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Vol.2., No.1-3, pp 97-101, 2016 June.

[11] Scott M. Lundberg, SuIn Lee: “A Unified Approach to Interpreting Model Predictions,” School of Computer Science University of Washington Seattle, WA 98105, 2017.

[12] Pang Wei Koh, Percy Liang, “Understanding Black-box Predictions via Influence Functions,” 2017.

[13] Mukund Sundararajan, Ankur Taly, Qiqi Yan, “Axiomatic Attribution for Deep Networks,” 2017.

[14] Vitali Petsiuk, Abir Das, Kate Saenko, “RISE: Randomized Input Sampling for Explanation of Black-box Models,” 2018.

[15] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradientbased Localization,” 2019.

[16] N. I. Papandrianos, A. Feleki, S. Moustakidis, E. I. Papageorgiou Ioannis, D. Apostolopoulos, D. J. Apostolopoulos, "An Explainable Classification Method of SPECT Myocardial Perfusion Images in Nuclear Cardiology Using Deep Learning and Grad-CAM," 2021.

[17] Liu, H.; Wu, J.; Miller, "Diagnostic Accuracy of Stress-Only Myocardial Perfusion SPECT Improved by Deep Learning," Eur. J. Nucl. Med. Mol. Imaging, Vol.48., pp. 2793–2800, 2021.

[18] Otaki, Y.; Singh, A.; Kavanagh, P.; Miller, R.J.H.; Parekh, T.; Tamarappoo, B.K.; Sharir, T.; Einstein, A.J.; Fish, M.B.; Ruddy, T.D, "Clinical Deployment of Explainable Artificial Intelligence of SPECT for Diagnosis of Coronary Artery Disease," JACC Cardiovasc. Imaging 2021, Vol.4. , No.3-5. , pp. 99-110, 2019

[19] Ruth C Fong and Andrea Vedaldi, "Interpretable Explanations of Black Boxes by Meaningful Perturbation," In IEEE International Conference on Computer Vision 2017 Oct, Vol.3., pp.115-121, 2017.

[20] Laura Ruis, Mitchell Stern, Julia Proskurnia, William Cha: "Insertion-Deletion Transformer," 2020 Jan.

DOI: 10.15439/2022R09 ISSN 2300-5963 ACSIS, Vol. 33

GreedySlide: An Efficient Sliding Window for Improving Edge-Object Detectors

1st To Hai Thien
University of Transport Technology
Hanoi, Vietnam
thienth@utt.edu.vn
0000-0002-2099-1863

2nd Tung-Lam Duong
ASI Company
Hanoi, Vietnam
adamduong26111996@gmail.com
0000-0002-9459-4705

3rd Chi-Luan Le
University of Transport Technology
Hanoi, Vietnam
luanlc@utt.edu.vn

***Abstract*—The recent development in deep learning and edge hardware architecture has provided artificial applications with a robust foundation to move into real-life applications and allow a model to inference right on edge. If a well-trained edge object detection (OD) model is acquired, multiple scenarios such as autonomous driving, autonomous hospital management, or a self-shopping cart can be achieved. However, to make a model well-inference on edge, a model needs to be quantized to scale down the size and speed up at inference. This quantization scheme creates a degradation in the model where each layer is restricted to at most lower representations, forcing an output layer only to have fewer options to circle an object. Furthermore, it also limits model generalization where the behavior of the dataset gets cut off each activation layer. We proposed a novel method GreedySlide by sliding window that divides a capture into windows to make an object fits better on the quantization bound to address this problem. Even though the technique sounds simple, it helps increase the number of options for bounding an object and clips the variance that can have by scanning the whole image. Our work has improved an original edge model on its corresponding benchmark by experimenting and increasing the model generalization on other related datasets without retraining the model.**

I. Introduction

Recent developments in artificial intelligence have given researchers a powerful tool for data exploration and analysis in many fields, especially in Computer Vision. Many Computer Vision tasks have been overcome by artificial intelligence models and have reached state-of-the-art results such as classification, detection, and recognition,... To take advantage of this premise, researchers are now making ways to bring these SOTA to edge devices to extend AI to another level. Bringing a model into an edge device can blow away people's concerns about leaking personal data when every piece of information and process stays inside a device, and only their encrypted features can be transferred outside for server computing and analysis. This helps AI operate closer to the human world and turn into personal assistance in many tasks

However, most SOTA models contain a huge trained number of parameters in full precision on servers, containing a huge number of layers and parameters. This incident makes them heavy in computation, and it is not reasonable to run the whole model on any edge devices. While high computation causes the edge device to lose more power, it also prevents incompatible edge hardware from running in real-time inference. Thus, they have to work simultaneously on building edge hardware for a friendlier deep learning configuration and reducing the number of bit inferences on a model that an edge has to handle. Newly edge AI chips such as Coral, Ambient,... typically rely on in-memory or near-memory data flow designs that place the logic and the memory data closer together for faster inference [3]. However, even with a strong hardware configuration, some model still takes quite a long time to generate a good result on the server-side. Therefore, the demand of **Model Compression** i.e finding a good model structure and reducing its size for low computational and relatively small power consumption, is leading in the latest research works. By deploying model compression, a two-fold benefit of minimizing the total number of energy-intensive memory accesses [17] and increasing the inference time due to effectively higher memory bandwidth, reducing the overall latency [12].

Regardless of the promising benefits, model compression has limited the capacity for generalization. In a well-inferencing model that can perform well after a model's compression, weights, biases, and activation values have to be retrained to match the new configuration. This step can be time-consuming and prohibit researchers from exploiting SOTA results that take days and months to complete. However, this is not the biggest problem that model compression can cause to a model. In Kim's work [6], he proposed a position-based scaled gradient as a training optimizer that scales the gradient depending on the position of a weight vector for friendly model compression. While for previous work of [9], [7] and [5], they focus on mimicking activation by mean and variance to represent the distribution of activation in the training dataset. By forcing the model to choose the parameters lower bit that fit with the distribution of weights and biases in a network in a higher precision scheme, it has taken away the uncertainty of a model and made it too robust to the training data's behavior, ie. poor model generalization. The lower the bit range, the less degree of freedom to tune with the parameters, which restricts the search space [8]. In addition, the validation for a deep learning model compression from previous work addresses only the same package's testing data, which can not verify a compressed model's true generalization.

Moreover, when considering bringing model compression to an object detection scheme, the model is more numerically sensitive than the Image Classification process. For [6] and [2] work of Image Classification, the final result for a model is a certain value which is clamping by softmax distribution between 0 and 1 and pushing its maximum a posteriori on a correct class. While a maximum posterior distribution value does not need to be determined exactly, the bounding boxes represented for object detection have to be fitted with the image's pixel location, especially for lower scale objects in the image.

By naively inspecting an input image for an object detection task, the representation of output, when scaled into a lower range bit, could be insufficient. For an input image that has a size (weight x height) larger than the current bit range, a set of quantization range can naively draw as a 255x255 grid for 8-bit inference. Vertices of a bounding box can fall diversely to fit with the ground truth label as they are assigned to each intersection of the grid, which is the corresponding lower-bit representation. After compression, for example, if a vertice value has not passed entirely to a new intersection on the grid, it will be forced heavily back to the previous intersection, which causes shifting in the entire bounding box. A small shifting pixel may not affect big objects, but it can cause problems for small and medium size objects in the scene (Figure 1), which causes a drop in model performance after a quantization process.

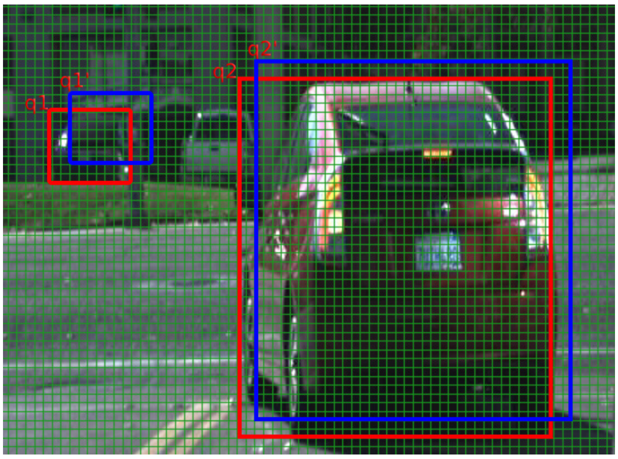


Fig. 1: By naively dividing an image in 255x255 grid quantization range (green lines), vertices q1 and q1' are different by two quantized values. Same for q2 and q2'. The detection between 2 close quantized values can affect differently on the final result based on the different scales of objects in an image

Our paper proposes a new method called GreedySlide, which addresses these problems dynamically within the scope of maintaining a good generalization for the model and easy integration on the edge interface. In GreedySlide, we take the trained model's outputs as the purest components to exploit how well a model explores the hidden data pattern and its restriction in generalization. We will then perform a greedy sliding window policy to scan over the image to obtain sub-location bounding boxes in the image and compare that with the current outputs to select the global bounding boxes for the whole image. Details of the policy will be explained in III. By scanning over the image instead of using a single image for detection, we emphasize better size for lower-scale objects and create a suitable range that matches the model's training size during compression. Therefore, it can bypass the overfitting compression when lowering the bit inference.

II. Background - Related Work

Existing methods for object detection using CNNs can be classified into two-stages and one-stage approaches. In two-stages methods such as FasterRCNN [14], R-FCN [1], Retinanet [10] classification and localization are implemented using two separate steps involving classification and region proposal. In contrast to this, the ones-stage approaches (such as Yolo [13], SSD-MobilenetV2 [11] classify and localize objects in only one step. Generally, one-stage detection models are faster by combining two stages as one, while the accuracy of two-stages models is higher. However, if scale down the accepted accuracy to a smaller intersection of the ground truth and the predicted object (intersection of union (IoU) = 0.5), one-stage models can achieve nearly the same accuracy of the two-stages methods. To make use of a one-stage detection structure for fast and real-time inference on an edge device, researchers [9] [15] [4] have tried to replace the feature extraction part of the one-stage scheme with a smaller model and a roughly same efficiency on the full precision setting.

Small objects detection: However, despite fast inferencing, object detections from the one-stage model usually get problems in detecting small-scale objects as the convolution features of these objects generally disappear in the last layers. Due to this problem, the normal solution is enlarging the input image so that the small object's pixels will increase in the training pictures or applying adding features map of upper layers. While upsampling the image size costs more time of inferencing and hyperparameters volume, the feature-map of upper layers makes the one-stage model behave as the second-stage model [16]. As the original model for the single stage has already struggled to handle small objects, a quantized version of it can not assure better performance and small object detection at the edge remains challenging. In our work, we fused both these ideas together. Instead of upsampling the whole image, we emphasize only the area may contain small objects through sliding windows and by providing a subsequent of the original image, we have naively provided the raw feature map of the model.

III. Proposed GreedySlide Algorithm

As briefly discussed our method focus on the post-training phase rather than bringing it into the training pipeline. By applying at the post-training stage, GreedySlide allows to take advantage of any model inference performance ie. different bit scale and improve model generalization. To perform our GreedySlide Algorithm, we divide our work into three phases: Sliding Windows Detection, Bounding Boxes Suppression and

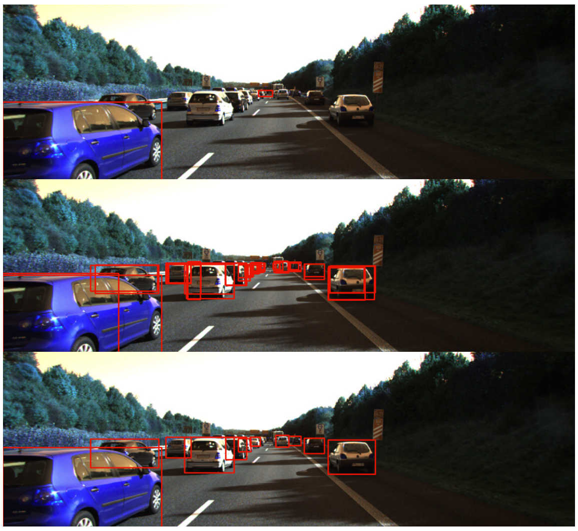


Fig. 2: Top picture is the result of bounding boxes by SSD-MobilenetV2 quantized edge model "occurring shifting detection"; Middle picture is the "partition" bounding boxes after Sliding Windows Detection; Bottom picture is the final bounding boxes after Bounding Boxes Suppression and Greedily Bounding Boxes Selection

Greedily Bounding Boxes Selection. The illustration of this whole pipeline is shown in Figure 2.

A. *Sliding Windows Detection*

Instead of taking only the whole image as an input for the model, the first phase of the GreedySlide algorithm divided the image into overlapped windows W size (l, l) combined with the original image I for inferencing. There are two main points in this basic approach. First, by overlapping windows on each other by γ threshold, we aim to avoid big objects being sliced into separated parts. Each window through the same ConvNet model only tells what object is detected in them, without knowledge of the whole image. Therefore, we combined the windows with the whole image as a true input for the model. For the second point, the optimal sliding width of the window l should be close to the training size of the model as the resizing do not distort much on the original image, and the image from each window can fit better to the quantized layer's behavior of low-bit range model which can overcome the problem of shifting bounding box for quantization scheme.

$$SLW(W, I, l, \gamma) = W', I' \tag{1}$$

This can be considered batch inference for multi-segment of the same image, which gives the model more elucidated input of the original image and supports multiple resolutions. Each subsequent range of the image will emphasize low-scale objects better as the appearance of those are dynamically bigger in a window. In addition, the window image can represent again the raw feature location of the image, which is useful for the model that has been trained without looking in this data and especially for the quantized model where the feature learning can be saturated.

B. *Bounding Boxes Suppression*

The reason we called this phase suppression is that there will be a lot of partial bounding boxes coming from big objects that can be scattered in the results. This phase will act as a constrained filter to remove those by greedily remove from the output. In detail, depending on the frequent location of the objects as well as the baseline performance of the model, GreedySlide algorithm takes into account another two threshold numbers to help suppress the bounding boxes. α is designed as the confidence to select a bounding box for each window, and β is the confidence to believe in the baseline model. In general, most object detection models are confident in getting high-scale objects rather than low-scale objects. By selecting correct β we can take out the most confident bounding boxes (global bounding boxes) from the whole image and use it to suppress scattered partial bounding boxes (local bounding boxes) from the windows.

Normally, two bounding boxes are the same if their IOU is high. However, because partial bounding boxes are relatively smaller than the groundtruth bounding boxes, using IOU is insufficient. Thus, we use self-intersection factor (f) to evaluate how strong the local bounding boxes attach to the global bounding boxes. A self-intersection factor f is calculated by the amount of the intersection of a bounding box over its own size, as described in (2).

$$f = \frac{intersection}{bounding\ box\ area} > \epsilon \tag{2}$$

For each local bounding box, it will be compared to global bounding boxes by a matrix relationship between bounding boxes. Each row of the matrix is a self-intersection factor f. Multiple local bounding boxes think to belong to one if f is bigger than ϵ amount. However, the selected ϵ is tricky to obtain a good result. A small ϵ can leave duplicating bounding boxes, while a big ϵ can leave small bounding boxes that scatters around an object due to its low certainty. A good ϵ forces the bounding boxes that are heavily related to a bigger box to be one and avoid wrongly taking close contacted bounding boxes of another objects.

C. *Greedily Bounding Boxes Selection*

After filtering the partial bounding boxes of the global bounding boxes, there can be the partial bounding boxes of small and medium-scale of objects in among the windows. However, these bounding boxes have no global bounding boxes to suppress them. Therefore, in this phase, the procedure is slightly different from the previous stage. GreedySlide will perform to compare each local bounding box together by the self-intersection-factor f. Then, bounding boxes that are strongly related to each other will be merged as a group G represents for an object. From the group of bounding boxes G, GreedySlide selects a bounding box that represents all of other boxes by select the top vertex *x1,y1* and the bottom

TABLE I: COCO BenchMark of two models (SSD-MobilenetV2, Retina) with its Greedy version in different precision. The benchmark COCO protocol allows to addess the IoU from 0.5 to 0.95 and estimate the performance in terms of recall and precision. The SSD-Mobilenetv2 is conducted on Google Coral Board, while the heavy Retina is measured on Jetson-Nano

	300x300-91 classes-Int8		640x640-80 classes-Fp16		640x640-80 classes-Fp32	
Models Metrics	Mobinet SSD	Greedy Mobile SSD	Retina	Greedy Retina	Retina	Greedy Retina
AP(0.5) - (S,M,L)	0.21	**0.24**	0.35	**0.41**	0.39	**0.41**
AP(0.5) - S	0.02	**0.06**	0.11	**0.22**	0.14	**0.21**
AP(0.5) - M	0.13	**0.25**	0.39	**0.45**	0.44	**0.46**
AP(0.5) - L	**0.45**	0.42	**0.55**	0.54	**0.55**	0.54
AR(0.5) - (S,M,L)	0.21	**0.26**	0.37	**0.46**	0.41	**0.45**
AR(0.5) - S	0.02	**0.06**	0.11	**0.23**	0.15	**0.23**
AR(0.5) - M	0.14	**0.27**	0.41	**0.51**	0.47	**0.51**
AR(0.5) - L	**0.49**	0.47	0.60	**0.62**	0.60	**0.62**
AP(0.5:0.95) - (S,M,L)	0.14	**0.16**	0.23	**0.26**	**0.28**	**0.28**
AP(0.5:0.95) - S	0.01	**0.03**	0.07	**0.13**	0.09	**0.13**
AP(0.5:0.95) - M	0.07	**0.16**	0.25	**0.29**	0.30	**0.31**
AP(0.5:0.95) - L	**0.31**	0.29	**0.39**	0.38	**0.41**	0.39
AR(0.5:0.95) - S	0.01	**0.03**	0.07	**0.14**	0.10	**0.14**
AR(0.5:0.95) - M	0.08	**0.18**	0.29	**0.35**	0.34	**0.36**
AR(0.5:0.95) - L	**0.36**	0.35	**0.46**	**0.46**	**0.48**	0.47
Overall Score	4/15	11/15	3/15	13/15	3/15	12/15

vertex *x2,y2* by (3) with a group mean confidence c (4). The reason that this method is called GreedySlide is because of this final execution as it takes the biggest box consisting of every box in the group. By taking the biggest bounding boxes instead of the intersection, we enlarge the area of capturing an object rather than taking a bounding box that can be a partition of an object. Furthermore, to neglect tiny boxes that are bare to tell due to the zooming feature of this method, every box needs to be large than δ scale concerning the window.

$$\mathit{x1} = \min_{X} G,\ \mathit{y1} = \min_{Y} G,\ \mathit{x2} = \max_{X} G,\ \mathit{y2} = \max_{Y} G \tag{3}$$

$$c = \frac{1}{N} \sum_{G} c_g \tag{4}$$

IV. Experimental results

A. Benchmark Settings

Our goal is to provide an algorithm that can help to improve object detection model performance for quantized edge models and maintain a good generalization for those. Therefore, we addressed two experiments to verify the method's performance: external dataset validation and same dataset validation. We choose the Google Coral Board (which only supports int8 for inference), and Jetson Nano (for FP16 and FP32) as two most popular edge devices nowadays for our experiments. Noted that, in the different dataset scenarios, we emphasize the importance of our method at int8 settings.

Model and Dataset Selection: To test the detection performance, we address the GreedySlide detector on Google-provided models trained on the COCO dataset: SSD-MobilenetV2 (6M params) and Retina (32M params) model for object detections. We choose this model because it is one of the most famous architectures for edge object detection with fast and light capabilities. Moreover, it also emphasizes the flexibility of our approaches. Meanwhile, we use these three external datasets: KITTI, CrownAI and Autti, for a throughout benchmark when bringing the general detectors from COCO dataset to the narrow scope task like (vehicle detection/pedestrian detection). For KITTI dataset requirements, we follow their evaluation on three levels of difficulties: easy (big and clear), medium (average-size/slightly occlusion) and hard (small and highly occluded) with various of IoU rate. To apply that setting into CrownAI and Autti, we define the easy scope for bounding boxes that are bigger than 100 pixels and the hard scope for every scale of pixels.

Hyperparameters Selection: when evaluating on KITTI dataset, we have figured out: the ideal size k for sliding window is roughly 1/3 of the width of the image; γ confidence for each window should be 0.4 and γ' for the whole image is 0.7 to reach highest accuracy for detection. In addition, the reason we choose 0.7 is to balance between 0.6 and 0.8 result for generalization. Furthermore, this setting also allows our method to catch up to 70% speed of orignal models. Therefore, we use this configuration throughout our experiments. The benchmark of detection result is conducted and measured mAP in Pascal VOC Format.

B. Same Dataset Detection Performance

In this setting, we perform throughout evaluation on the pre-trained COCO dataset model published in an open repository of Google Coral and Jetson Nano. We omit the testing SSD-Mobinet v2 settings in Jetson Nano due to similar behavior results on Coralboard. We also discard the benchmark of the Retina model on Google Coral because of the slow inference of TPU on this device.

Overall the result at table I shows increased performance for the edge model regardless of int8 quantization, FP16 reduction, or full precision on 32 bits on both Jetson and Coralboard. However, it seems to lose a bit of performance for

TABLE II: COCO on traffic classes. N: non-GreedySlide. G: Greedyslide

Classes Metrics	Person		Car		Truck		Bus		Motorcycle		Bicycle	
	N	G	N	G	N	G	N	G	N	G	N	G
SSD-Mobilenet V2												
AP(0.5) - (S,M,L)	0.37	**0.43**	0.16	**0.29**	0.23	**0.27**	0.50	**0.57**	0.34	**0.4**	0.19	**0.30**
AP(0.5:0.95) - M	0.18	**0.28**	0.11	**0.28**	0.03	**0.13**	0.03	**0.21**	0.05	**0.14**	0.08	**0.19**
AP(0.5:0.95) - L	**0.53**	0.51	**0.49**	0.45	**0.40**	0.37	0.65	0.62	**0.47**	0.44	0.47	**0.49**
AR(0.5:0.95) - M	0.21	**0.31**	0.15	**0.34**	0.04	**0.17**	0.03	**0.23**	0.07	**0.18**	0.09	**0.22**
AR(0.5:0.95) - L	**0.60**	0.59	**0.57**	0.53	**0.46**	0.43	**0.69**	0.67	**0.53**	0.50	0.54	**0.56**
Overall Score	2/5	3/5	2/5	3/5	2/5	3/5	1/5	3/5	2/5	3/5	0/5	5/5
Retina												
AP(0.5) - (S,M,L)	**0.57**	**0.57**	0.43	**0.46**	0.31	**0.37**	0.64	**0.68**	0.52	**0.53**	0.38	**0.40**
AP(0.5:0.95) - S	0.15	**0.18**	0.17	**0.20**	0.05	**0.09**	0.14	**0.24**	0.11	**0.15**	0.06	**0.11**
AP(0.5:0.95) - M	**0.47**	0.46	**0.46**	0.45	0.16	**0.25**	0.37	**0.40**	**0.25**	0.24	**0.29**	0.28
AP(0.5:0.95) - L	**0.67**	0.61	**0.54**	0.48	**0.42**	0.37	**0.72**	0.68	**0.52**	0.51	**0.52**	0.51
AR(0.5:0.95) - S	0.17	**0.20**	0.18	**0.22**	0.06	**0.12**	0.15	**0.28**	0.11	**0.18**	0.06	**0.12**
AR(0.5:0.95) - M	**0.52**	0.50	**0.53**	**0.53**	0.21	**0.33**	0.39	**0.44**	**0.32**	0.29	**0.33**	**0.33**
AR(0.5:0.95) - L	**0.73**	0.69	**0.62**	0.58	**0.51**	0.50	**0.74**	0.73	**0.57**	0.55	**0.58**	**0.58**
Overall Score	5/7	3/7	4/7	4/7	2/7	4/7	2/7	5/7	4/7	3/7	4/7	5/7

TABLE III: mAP results for Greedy-SSD detector in comparison with SSD-MobilenetV2 and modified on KITTI, CrownAI and Autti Dataset

Models	easy (0.5)	medium (0.5)	hard (0.5)	easy (0.7)	medium (0.7)	hard (0.7)
KITTI						
Greedy-SSD	**0.5425**	**0.6421***	**0.5924**	**0.5162**	**0.5244**	**0.4204**
SSD-MobilenetV2	0.1632	0.1519	0.1517	0.1566	0.1176	0.1191
CrownAI						
Greedy-SSD	**0.6791***	**-**	**0.5672**	**0.5244**	**-**	**0.4019**
SSD-MobilenetV2	0.1877	-	0.1549	0.1172	-	0.0943
Autti						
Greedy-SSD	**0.6123***	**-**	**0.5772**	**0.5342**	**-**	**0.4446**
SSD-MobilenetV2	0.1784	-	0.1612	0.1123	-	0.1003

large objects but emphasizes the efficiency of small objects and medium-size ones. This can be understood as we fragment the picture to address, which makes the small objects more robust but accidentally divides the big objects into pieces. However, it is up to the shape and size of the objects in the picture. When inspecting some specific classes in Table II, we can notice there exists class can maintain good detection in large size like *bicycle* or *bus*, while others can degrade roughly. However, the results show promising in accuracy performance when gathering the good points among AP(average precision) and AR (average recall) that Greedyslide boosts the origin model (roughly triple the performance $12/15$ for overall classes and $5/7$ for specific classes inspection). In addition, it also shows the flexibility of our method on different model types (SSD and Retina) and edge configurations. It also hints our method can assist the model at full precision scale.

C. External Dataset Detection Performance

In this setting, we only address the SSD-Mobilenet as introduced in the Benchmark Settings. Table (III) summarizes the evaluation results from the three datasets. As shown, Greedy-SSD has overperformed to two baselines method for one-stage models. The accuracy mAP increases roughly five times in comparison with SSD-MobilenetV2, which are really sufficient detectors in computers or servers. For closer inspection, SSD-MobilenetV2, through quantization, has dramatically reduced its accuracy by six times according to SOTA full precision results on KITTI open dataset benchmark (from 0.61). Furthermore, Greedy-SSD performs better regardless of object scales in the image when the gap between easy and hard task is not severe. One of the most important factors for this method is that it can help the SSD-MobilenetV2 increase its accuracy to acceptable results without designing a new deep learning model structure for quantization like other approaches.

V. Conclusion & Future Work

In this paper, we have introduced our novel GreedySlide method on optimizing detection performance on the different type of datasets, especially for edge devices' scopes. It proves that the gap of detection performance with and without GreedySlide is considerable and well-improved in external dataset. Using GreedySlide can reduce the complexity and effort to optimize a model by quantization and ensure a better generalization for the model. In the future, we will try to integrate this behavior into the training pipeline as a multi-scale feature model for a better and more compact solution.

VI. Acknowledgement

This research is funded by University of Transport Technology (YTT) under grant number ĐTTĐ2021-06

References

[1] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[2] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4852–4861, 2019.

[3] Samuel Greengard. Ai on edge. *Commun. ACM*, 63(9):18–20, August 2020.

[4] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[5] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018.

[6] Jangho Kim, KiYoon Yoo, and Nojun Kwak. Position-based scaled gradient for model quantization and pruning. 2020.

[7] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[8] Dongsoo Lee, Se Jung Kwon, Byeongwook Kim, Yongkweon Jeon, Baeseong Park, and Jeongin Yun. Flexor: Trainable fractional quantization. *arXiv preprint arXiv:2009.04126*, 2020.

[9] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2810–2819, 2019.

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[11] W Liu, D Anguelov, D Erhan, C Szegedy, S Reed, CY Fu, and AC Berg. Ssd: Single shot multibox detector. arxiv 2016. *arXiv preprint arXiv:1512.02325*, 2020.

[12] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

[13] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arxiv 2018. *arXiv preprint arXiv:1804.02767*, pages 1–6, 2018.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[16] S. Zhai, D. Shang, S. Wang, and S. Dong. Df-ssd: An improved ssd object detection algorithm based on densenet and feature fusion. *IEEE Access*, 8:24344–24357, 2020.

[17] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

DOI: 10.15439/2022R13 ISSN 2300-5963 ACSIS, Vol. 33

Different Classifier Approaches Used For Fingerprint Classification

Meena Tiwari
Research Scholar department of CS, SRK
University Bhopal MP, India
phmeenatiwari@gmail.com

Dr. Ashish Mishra
Professor department of CS, Gyan Ganga
Institute of Technology and Science
Jabalpur, India
ashishmishra@ggits.org

***Abstract*—Fingerprints play an important role in public safety and criminal investigations such as: B. Legal Investigations, Law Enforcement, Cultural Access, and Social Security. It can also help to give people a comfortable and secure life. Various gender segregation strategies have been proposed. In this article, the fingerprint algorithm uses a variety of Naive Bayes classifiers, SVM, Logistics Regression and Random Forest which they use to obtain the best results of gender segregation, a new fingerprint method can be created by Naive Bayes classifier, SVM, Logistics Regression and The Random Forest used and compiled proposed from different divisions obtained the best possible division of results by Random Forest, with 98% accuracy compared to Naive Bayes, SVM and Logistics Regression, based on Random. The forest is the most sensitive to gender segregation.**

***Index Terms*—Naïve Bayes, SVM, Logistic Regression, Random Forest, Gender classification, Fingerprint database, Association Rule Mining.**

I. Introduction

Separation of fingerprints refers to the separation of each fingerprint in a section in a consistent and reliable way so that anonymous fingerprints can be searched and only need to be compared with the fingers in the details of the section. Comparison of fingerprints is usually based on small finger functions, such as apex and branches, while fingerprint classification is often based on larger functions, such as the shape of the rib cage.

To identify a person, you must compare your finger with all fingerprints in the message. A common strategy is to reduce the number of comparisons in finger recognition, thereby extending the recognition process to distinguish fingerprints through the various categories described previously [1]. The actual control of human thought is based on the development of models. Advanced computers support pattern recognition. Separation is an example of pattern recognition, which attempts to assign each input value to a different category. Your main goal is to find the best role model support under certain circumstances and to differentiate one category from another. Divorce performance is very important in making good decisions. However, the function of classification depends largely on the identity of the data to be categorized. Major Headings Major headings should be typeset in boldface with the first letter of important words capitalized.

A. Gender Classification

There are a number of ways to determine the most important gender, such as reducing the speed of detection of criminals in cybercrime, thereby reducing the time spent on application. The abuse of biological agents has created many ways to identify sex, such as face, finger, voice samples, etc. This allows sexual determination based on fingerprints. Fingerprint recognition completed. Two operations, one is used to find the number of fingers in the male and female numbers, the size of the ribs, the size of the ribs, and the base size and width of the ribs. And fingers. Model. Used for gender segregation. Second, measurement compares only the incoming finger with other information, which is very important in accelerating finger recognition. Recovery technology and automatic hunting. [25] With the availability of cheap and high-quality scanners and their good compatibility with a variety of biometric technologies, many citizens and commercial applications are beginning to get bored or get involved with fingerprints.

II. Classifiers Used

In this paper four classifier are used for fingerprint classification which are as follows:

A. Naive Bayes Classifier

Naive Bayesian Separator is an algorithm that uses the Bayes theorem to separate objects. The ignorant Bayesian separator assumes strong or irrational independence between data point signals. Popular use of Bayesian naive classifiers for spam filtering, text analysis, and medical diagnostics. This separator is widely used in machine learning because it is easy to use.

Unjustified Bayesian segregation can be easy opportunities supported by Bayesian theory and with a stable sense of independence. For example, suppose domain variables are separate categories and each variable contains a limited number of values.

B. SVM Classifier

Support Vector Machine is a learning program that uses speculative space for specific tasks in large spaces. SVM is trained using a learning algorithm designed for processing that considers learning bias based on mathematical learning concepts. In the late 1970's, Vapnik and colleagues introduced this learning strategy and combined it with various mathematical, mechanical, and neural networks. Risk reduction (SRM) policy is contained in SVM. It turns out that this is an improvement in the traditional goal of minimizing the high-risk (ERM) used by conservative neural networks. Compared to ERM, SRM reduces the expected amount of risk by reducing errors in training data. Therefore, SVM is unique and comes with excellent summaries and capabilities to achieve mathematical learning objectives. In fact, SVM is a binary separator.

C. Random Forest

Random Forest is a widely used machine learning engine and is a trademark of Leo Braiman and Adele Cutler. Combine results from multiple decision trees to get one result. Easy use and flexibility lead to acceptance because you can overcome differences and frustrations.

The Random Forest (or the Forest Forest) are trademarks of various species. It contains a lot of stems, the first step to removing the stem from the tree. A random forest is a group of slightly different trees.

D. Logistice Regression

Asset recovery is an independent control system algorithm used to predict the probability of target variation. The type of target or variation you rely on is dichotomous, that is, there are only two possible categories.

Simply put, the dependent variable is binary and the data is encoded as 1 (indicating success / yes) or 0 (indicating failure / no).

Statistically, the systematic model predicts P (Y = 1) as the X function. It is one of the simplest ML methods used for diagnostic problems such as spam detection, diabetes prognosis, cancer detection, etc.

III. Related Work

Nithin MD et al. [3] 200 A study of the age group 18 to 30 years (100 men and 100 women) was used. Use the newly designed structure to determine left-to-left compression and perform a statistical evaluation. Tests show that women in both study areas, either individually or in combination, tend to have more than one fingerprint.

Pattanawit Soanboon et al. [4] used the size of the fingertips and also explained that males have stronger fingers than females, meaning that males have smaller glands than females in a given region and therefore less soaps. The high prevalence of female fingerprints is due to the fact that the epidermal columns of women tend to be smoother than those of men. Men tend to have stronger mountains than women, with a difference of around 10%.

Ashish Mishra et al. [5] in this article, The difference in overcrowding between men and women in some areas may be due to the fact that the proportion of men is greater than that of women, so the amount of overcrowding is the same. Placed in a larger area of males, so males are smaller in size.

Suchita Tarare et al. [6] also explain the whole process of the scheme above. The DWT modification provides multiple fingerprint structures from a data set (image reading) to create a database of functions used as a viewing table to distinguish unknown fingerprints from other fingerprints (fingerprints) used for testing. The divider Knn designates one of the two sets for testing fingerprints.

Alessandra Lumini et al. [7] describes the many systems and structures associated with the integration of biometric systems, both informal and multidisciplinary, and classifies them according to certain tax statistics. In addition, we address the issue of testing biometric systems and discuss performance indicators and processes.

Swapnil R. Shinde et al. [8] described a complete comparison of conventional domain strategies, with a particular focus on DWT and its combinations. It also uses a canvas on the edge of the Canny and a DWT-based hair filter.

Neeti Kapoor et al. [9] to find a significant difference in the strength of tread between boys and girls in most Indians, a study was conducted on 200 subjects (100 boys and 100 girls) aged 18 years to find a sexual relationship - Year 30.

Meena Tiwari et al. [10] in this work, the study used four divisions: the Bases organization, the multi-stakeholder organization, the closest neighbors, and additional vector equipment. In addition, classification was tested in four prominent studies. These are cases that are questioned with 70% correction, 30% test, 60% preparation, 20% test, finally 60% preparation, 40% test, 10 cases. From the results it can be concluded very well that all the emergence of a common division completes representation of more than 90%. However, SVM is still the best divider proposed to be counted. Fingerprints are strong evidence of legitimacy in court. With regard to the amazing power as a finger recognition system, this post tries to explore relationships between women. Reading and standing are the result of our statistics and part of the basic meaning, to see the right values for sex. This design requires a proven test structure for unique products to reduce numerical time and improve performance.

IV. Results and discussion

The purpose of this proposed project is to utilize new technologies based on mindless Bayesian classifiers, vector support machines, random forests and retrospect in order to properly train and test fingerprint scanners. The proposed system is based on the results of comparing the four categories selected by the categories: Support Vector Machines and Naive Bayes, Random Forest and Logistic Regression which are divided into left and right category categories. Specifically, it has two independent modules: a training data selection module and a fingerprint separation module. Fig. 1 shows a block diagram for the implementation of the finger splitting system.

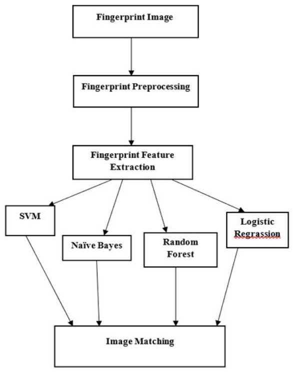


Fig.1. Block diagram of the system implemented to classify fingerprints.

TABLE.1. MASTER CLASSIFIER FINGERPRINT DATABASE

W			X			Y		
termination x	termination y	Angle	bifurcation x	bifurcation y	Angle 1	Angle 2	Angle 3	Class
115	15	109	133	32	137	39	120	1
17	121	20	54	80	58	130	60	1
204	21	114	84	76	106	76	94	1
22	108	24	80	81	82	139	87	1
111	33	130	146	90	228	92	167	1
40	104	41	93	143	97	162	98	1
110	42	166	168	103	190	109	115	1
47	107	51	110	88	112	52	114	1
100	53	145	173	120	196	120	108	1
57	189	57	123	56	125	200	125	1
70	61	235	80	130	123	132	162	1
63	181	64	132	156	134	128	137	1
76	65	187	202	138	53	140	219	1
65	59	66	141	237	144	232	148	1
193	67	71	174	157	166	163	115	1
70	179	74	166	84	168	76	180	1
186	77	211	52	181	82	183	151	1
81	218	81	184	47	187	110	196	1
178	83	78	57	198	48	199	68	1
88	126	88	199	60	204	57	217	1
187	89	177	159	221	85	233	74	1
93	183	96	238	0	0	0	0	1
78	100	176	0	0	0	0	0	1
103	182	105	0	0	0	0	0	1
103	107	128	0	0	0	0	0	1
107	122	114	0	0	0	0	0	1
176	127	177	0	0	0	0	0	1
134	205	149	0	0	0	0	0	1
89	161	104	0	0	0	0	0	1
163	147	170	0	0	0	0	0	1
154	171	57	0	0	0	0	0	1
172	88	173	0	0	0	0	0	1
174	174	183	0	0	0	0	0	1
177	134	178	0	0	0	0	0	1
141	179	156	0	0	0	0	0	1
179	98	180	0	0	0	0	0	1
109	185	169	0	0	0	0	0	1
190	52	192	0	0	0	0	0	1
124	192	65	0	0	0	0	0	1
193	38	197	0	0	0	0	0	1
179	198	161	0	0	0	0	0	1
200	148	201	0	0	0	0	0	1
78	206	164	0	0	0	0	0	1

V. FINGERPRINT NOTES

The editorial board is edited by Naive Bayes and delivered to a unique library. half of the information index is used for preparation, and the remaining half is used for testing. The specifications found are shown in Table 1. Repeat the same rehash system (W, X, Y), separated by fingers left (1) and right (2). These three types are a combination of X coupling, Y coupling, point, X branch, Y branch, point 1, point 2, and point 3.

TABLE.2.THE RESULTS OBTAINED FROM THE NAIVE BAYES CLASSIFICATION APPLIED TO THE CLASS (W, X, Y) ON CASIA DB FINGERPRINTS

	W			X			Y		
	termination x	termination y	Angle	bifurcation x	bifurcation y	Angle 1	Angle 2	Angle 3	Class
Accuracy (%)	71								

TABLE.3.THE RESULTS OBTAINED FROM THE SVM CLASSIFICATION ARE APPLIED TO 3 CLASSES (W, X, Y) IN CASIA DB . FINGERPRINT DATABASE

	W			X			Y		
	termination x	termination y	Angle	bifurcation x	bifurcation y	Angle 1	Angle 2	Angle 3	Class
Accuracy (%)	68								

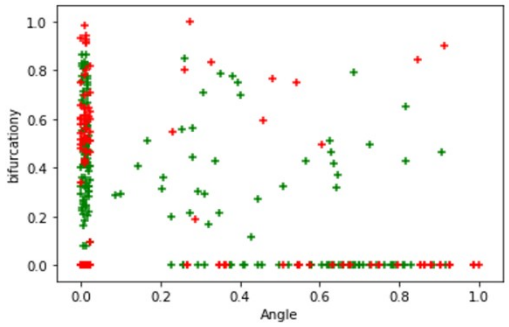


Fig.2. Plot the classification boundaries and visualize the svm classifier

TABLE.4. RESULTS OBTAINED FROM RANDOM FOREST, APPLIED IN 3 CLASSES (W, X, Y) IN CASIA DB . FINGERPRINT DATABASE

	W			X			Y		
	termination x	termination y	Angle	bifurcation x	bifurcation y	Angle 1	Angle 2	Angle 3	Class
Accuracy (%)	98								

TABLE.5. THE RESULTS OBTAINED FROM LOGISTIC REGRESSION, APPLIED IN 3 CLASSES (W, X, Y) IN CASIA DB . FINGERPRINT DATABASE

	W			X			Y		
	termination x	termination y	Angle	bifurcation x	bifurcation y	Angle 1	Angle 2	Angle 3	Class
Accuracy (%)	88								

A. Comparison Accuracy Naive Bayes , SVM method , Random Forest and Logistic Regression

This comparison tests the ability to distinguish by simple separators commonly used in image classification. Four classifiers were selected for the test: Naive Bayes, SVM, Random Forest, Logistic Regression and related categories to compare classification results. Compared to the Bayesian classifier class, SVM classifier and logistic regression, the random forest division has yielded excellent results. Random forest planning sets well 98% model attributes and reduced model sets. Depending on these results, it can be improved by further processing the data and synchronizing the separation. Accurate comparisons of the mindless Bayes approach, random forest and logistic SVM regression are shown in Table 2 and Table 3.

B. Confusion Matrix

Confusion matrices are used to measure the performance of a two-step task on a given data set. False negative) incorrect sample classification.

TABLE.6. CONFISION MATRIX

Actual Class	**Predicted Class**	
	Positive	Negative
Positive	True Positive (TP)	False Negative
Negative	False Positive (FP)	True Negative (TN)

Total number of copies = items ordered correctly + items ordered correctly

Classification example = TP + TN

Example of misclassification = FP + FN

Table 5 uses non-statistical records. 185 cases. The fingerprint category is randomly selected from the fingerprint record. When using the algorithm for the design of the Bayesian naive in a set of data, a confusion matrix is generated in a two-digit finger class, i.e. Left thumb and right thumb.

TABLE .7. NAIVE BAYES CLASSIFIER CONFUSION MATRIX

Actual Class	**Predicted Class**	
185	Positive	Negative
Positive	71	7
Negative	9	41

The table above is suitable for calculating the matrix of confusion of these actual and hypothetical scales, where 185 is perfectly correct in grade 71 and is not entirely accurate. True lies in seventh grade. Not at all. The absolute best result of the category was 9 and nothing. The positive result of lying in class was 41.

TABLE.8. SVM CONFUSION MATRIX

Actual Class	**Predicted Class**	
185	Positive	Negative
Positive	68	37
Negative	32	48

Table - 8 sets of data used do not match. 185 cases. The fingerprint category is randomly selected from the fingerprint record. When the SVM partition algorithm is applied to a data set, the fingerprint confusion matrix is made up of two values; Left and Right Thumbs The table above is used to calculate the confusion matrix of this real and predicted phase, i.e. the true positive result of this phase is 68 and is negligible. From false good to section 37. Impossible. The best result for the class was 32 points, which was not the case. The positive class result of the class was 48.

TABLE.9.RANDOM FOREST CONFUSION MATRIX

Actual Class	**Predicted Class**	
185	Positive	Negative
Positive	98	7
Negative	9	51

The table uses countless records. 185 cases. The fingerprint category is randomly selected from the fingerprint record. When using a random forest algorithm in a data set, the confusion matrix is created for a two-digit fingerprint category, i.e.. Left and Right Thumbs The table above is used to calculate the confusion matrix of this real and predicted phase, i.e. the actual positive result of this phase is 98, at all. True lies in seventh grade. Not at all. The absolute best result of the category was 9 and nothing. The positive result of lying in class was 51.

TABLE.10. LOGISTIC REGRESSION CONFUSION MATRIX

Actual Class	**Actual Class**	**Actual Class**
185	Positive	Negative
Positive	88	7
Negative	8	65

Table - 10 of the used data sets do not match. 185 cases. The fingerprint category is randomly selected from the fingerprint record. When using a logistic regression algorithm on a data set, a two-digit confusion matrix is developed for the fingerprint category, i. Left and Right Thumbs The table above is used to calculate the confusion matrix of this real and predicted phase, i.e. the true positive result of this phase is 88, but not at all. True lies in seventh grade. Not at all. The overall good result for the class was 8 points, which was certainly not true. The positive result of lying in class was 65.

C. *Accuracy*

Described as the ratio between well-separated samples and the total number of samples. Figure 2 shows the comparative accuracy of the Naive Bayes, SVM, Random Forest and Logistic Regression.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

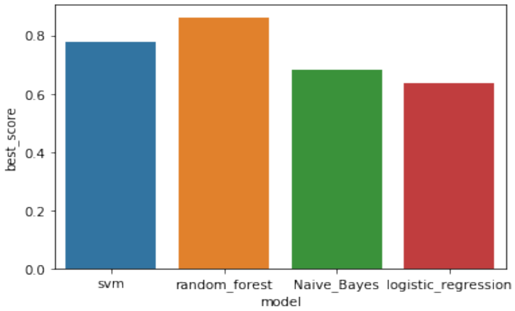


Fig.2. Comparative accuracy with naive Bayes, SVM method, random forest and logistic regression

VI. Conclusions

Gender segregation can be done through the Naive Bayes Classifier, SVM, Random Forest and job adjustment. By limiting the following search categories to the sixth left and right information, the search time is effectively reduced. Once the left and right face thumbs are separated, the biological features of each can be used for further separation. In this category, four categories have been selected: Naive Bayes, SVM, Random Forest and Logistic Regression. Compared to the mindless Bayesian, SVM and logistic regression classifier, the best results were obtained with random forest planning.

References

[1] Alessandra A. Paulino et al. (2015) *Latent unique mark Matching Using Descriptor-Based Correlation* IEEE TRANSACTIONS ON INFORMATION FORENSICS & SECURITY, VOL. 8, NO. 1, page 31-45.

[2] Ashish Mishra et al. (2017) *A Novel Technique for Fingerprint Classification based on Naive Bayes Classifier and Support Vector Machine* 0975 – 8887, Volume 169 – No.7.

[3] Hasan, Haitham, and S. Abdul-Kareem (2013) *Fingerprint image enhancement and recognition algorithms: a survey* Neural Computing and Applications 23, no. 6 : 1605-1610.

[4] Meena Tiwari, et al (2021) *Development of Association Rule Mining Model for Gender Classification* 1st International Conference on Computational Research and Data Analytics (ICCRDA 2020)

[5] Neeti Kapoor et al. (2016) *Sex differences in thumbprint ridge density in a central Indian population* ScienceDirect, Elsevier, Egyptian Journal of Forensic Sciences 5, 23–29.

[6] Nithin MD et al. (2011) *Gender differentiation by finger ridge count among South Indian population* J Forensic Leg Med,18:79–81.

[7] Pattanawit Soanboon a et al. (2016) *Determination of sex distinction from unique mark edge thickness in northeastern Thai young people* ScienceDirect, Elsevier Egyptian Journal of Forensic Sciences 6, 185–193.

[8] S. Lee Hong et al. (2012) *Neural correlates of unpredictability in behavioral patterns of wild-type and R6/2 mice* communicative & integrative Biology 5:3, 1-3.

[9] Suchita Tarare et al. (2015) *Fingerprint Based Gender Classification Using DWT Transform* IEEE International Conference on Computing Communication Control and Automation, 978-1-4799-6892-3/15.

[10] Swapnil R. Shinde et al (2015) *Gender Classification with KNN by Extraction of Haar Wavelet Features from Canny Shape Fingerprints* IEEE International Conference on Information Processing Vishwakarma Institute of Technology. 978-1-4673-7758-4/15.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 255–260 DOI: 10.15439/2022R01 ISSN 2300-5963 ACSIS, Vol. 33

Simple and Efficient Convolutional Neural Network for Trash Classification

1st Binh-Giang Tran
Department of IT
Vinh University of Technology and Education
Nghe An, Vietnam
binhgiangktv@gmail.com

2nd Duy-Linh Nguyen
School of Engineering, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
ndlinh301@mail.ulsan.ac.kr

***Abstract*—Strong economic and city developments have given a great amount of trash. Trash is made continuously from families, public and commercial areas, construction places, hospitals, etc. The enlarging trash amount is a much more serious problem than climate change, and the cost of trash treatment will be a big burden to countries in the world. One of the effective trash treatment measures is to separate trash right from its source, especially domestic trash. The countries have applied many trash classification systems, but the requirements for infrastructure, implementation, and operation are quite complicated. In order to help people easily sort household trash at home, this paper proposes a simple convolutional neural network for trash classification. The network is trained and evaluated on the TrashNet dataset with an accuracy of 90.71%. In addition, this work also tests in real-time on low-computation devices such as CPU-based personal computer and Jetson Nano devices.**

***Index Terms*—Convolutional neural network, trash classification, trash recognition.**

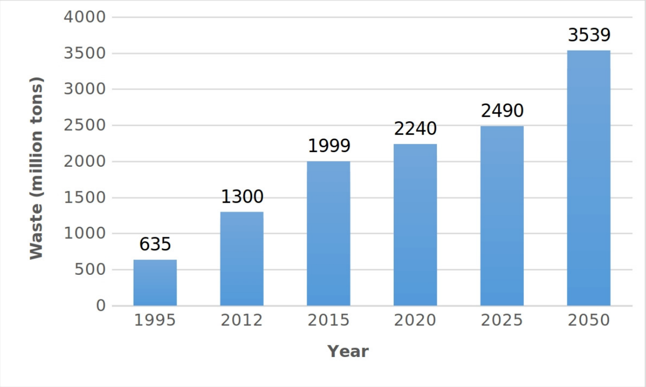


Fig. 1. The chart of the annual amount of trash on the globe.

I. Introduction

Trash not only pollutes the environment but also directly affects people's health. Funding and ways to treat trash are also a dilemma for many countries around the world. More specifically, in 2012, humans released 1,300 tons of trash into the environment, double the amount of trash in 1995. In 2015, the total amount of trash worldwide was 1,999 tons [1]. The quantity of trash in 2020 is 1.1 times higher than in the previous five years, about 2,220 tons [2]. And with the current rapid trash generation rate, it is expected that by 2050, the weight of trash that humans generate every day could be 3,539 million tons [3]. Fig. 1 shows the chart of the annual trash on the globe. Trash is an inevitable consequence of human life and production. On average, a person generates 0.74 kg of trash every day. This amount of daily household trash includes leftovers, paper, old clothes, damaged electronics, old furniture, garbage from the yard, etc. The benefits of trash classification at the place of residence help to save resources, and bring a great economic source. From that experience, it contributes to reducing the total quantity of trash that needs to be destroyed, reducing the load on the environment, and saving costs of collection, transportation, and treatment. This also contributes to raising public awareness about the protection and rational use of natural resources and the environment. There are many ways to sort trash at home. The first way is based on chemical characteristic, we can separate organic trash and inorganic trash, then continue to classify organic trash for easy handling. The second way is to use the product standards listed on the packaging to separate them into recyclable trash and non-recyclable trash. The third way is optical sorting or eddy current-based sorting classification techniques [4]. However, to reduce the complexity for people, household trash can be classified into the following six groups: glass, paper, cardboard, plastic, metal, and trash. There are many trash sorting systems in the world. In 2010, Glave et al. announced an automated vacuum trash collection (AVAC) system, which transports trash at high speed through underground pneumatic tubes to a collection station, where they are compacted and sealed in containers. This system helps to classify and recycle garbage automatically but requires a large area of land, pipe system, big industrial fan system, sensors, and automated software [5]. In 2019, the identify system called Multi Re-use Facility (MRF) was used to recognize and classify trash on conveyor belt. Each stage is identified and the trash is gradually eliminated. The order of identification is hazardous trash such as batteries, aerosols; soft plastic materials; paper, bottles, cans and finally ferrous metals. This system also requires high installation costs, making it suitable for small trash treatment plants [6]. To ensure high accuracy, low installation cost, and suitability for each family's size, the paper proposes a simple convolutional neural network (CNN) to recognize trash in daily life according to the six groups mentioned above. This network exploits basic features such as convolutional layers, pooling layers, optimization method, and especially the global average pooling technique to replace fully connected layers to reduce computational cost. Because of the high accuracy and the low delay during real-time testing, the network is applied to classify the trash and help the person to put the correct trash in the designated bin.

The main contributions of this paper are shown as:
1 - Proposes a simple and efficient CNN with two new connectors to classify trash.
2 - Develops a trash classification system on low-computing devices such as personal computers and embedded devices.
The rest of the paper is organized: Section II presents some traditional techniques and CNN in trash identification and analyzes their advantages and disadvantages. Section III describes the proposed network in detail. Section IV analyzes the experiments and evaluates the results. Finally, the conclusion and future development direction are presented in section V.

II. Related Work

In this section, the paper introduces several traditional and CNN-based techniques which has been applied to identify trash and analyzes the advantages and disadvantages of those techniques.

A. Traditional methods

The commonly used identification and classification algorithms include the K-Nearest Neighbor algorithm (KNN), Bayesian Classifier, and Support Vector Machine (SVM). The authors in [7] using the KNN algorithm in identifying and classifying trash with an accuracy of 93.8%. [8] presents the application of a Recurrent Neural Network (RNN) to monitor the amount of trash in the bin. The paper [9] developed a deep learning classifier to classify domestic solid trash into different categories. The classification results show that the trash classification accuracy is from 82% to 96%. In 2018, Yijian Liu et al. designed a trash classification system in which the software is based on the SURF-BoW algorithm and multi-class SVM classifier. This system focuses on sorting five types of trash: batteries, bottles, cans, paperboard, and paper boxes. Experimental results show that pin identification is the best [10]. However, the limitation of this algorithm is that it cannot recognize trash and broken bottles, which is one of the common types of household trash in every household.

B. CNN-based methods

Currently, convolutional neural networks are one of the mainstream approaches to image recognition. To classify and identify trash, the following CNN networks can be used: Alexnet, VGGNet, GoogleNet, ResNets, DenseNet [11]. Faced with India's trash boom, Thanawala D. et al. approached a variety of techniques that use CNN to classify trash into three categories: recyclable, non-recyclable, and organic. Along with the proposed neural network, five standard CNN architectures including VGG-16, DenseNet, InceptionNet, MobileNet, and ResNet are also tested on the given dataset. The highest test accuracy is 92.65% and the lowest is 81.25% [12]. Gary has built an automatic trash classification technique based on CNN at the edge, allowing smart and quick decisions to be made without access to data from the cloud. With this technique, trash is classified into six categories: paper, cardboard, glass, metal, plastic, and others [13]. This method is suitable when applied to large sorting centers to automatically separate trash into different categories, making recycling easier. Hua Zheng and Yu Gu used CNN to identify domestic solid trash. First, three CNN networks including GoogLeNet, ResNet50, and MobileNetV2 are used as component classifiers for separate prediction.Next, UPMWS is used to determine the weight coefficients of the composite models. This technique separates trash into four groups: wet trash, recyclable trash, hazardous trash, and dry trash. This algorithm is difficult to install and achieves high efficiency with ensemble learning [14]. The authors in [15] proposed a CNN network to classify trash into three categories: recyclable, non-recyclable and organic trash, with an accuracy of 81.22%. The paper also mentions other structures such as VGG16, InceptionNet, DenseNet and MobileNet. Out of all these transfer learning models, MobileNet showed the highest accuracy of 92.65%. K.Ahmad et al. used the CNN network to classify trash into four groups: cardboard, paper, metal, and plastic. Through testing for 100 times, with images of size 50×50, the accuracy reaches 76% [16]. The work in [17] apply multilayer hybrid convolution neural network technique, consider TrashNet dataset to identify trash through images; classified into glass, metal, paper, plastic. This network has a similar structure to VGGNet but is simpler, with fewer parameters, faster processing time, and up to 92% accuracy. The disadvantage of this method is that it can't classify recyclable and non-recyclable trash.

III. Proposed Methodology

The proposed trash classification network is described in Fig. 2. The network consists of feature extraction and classification modules.

A. Feature extraction module

The feature extraction module plays an important role in the proposed CNN network. It performs feature extraction at different levels and is input to the processing in the next module. This module combines the advantages of convolution layer (Conv), depthwise separable convolution layers (DWConv), average pooling layer, batch normalization method (BN), and two new connectors to extract the best feature maps and optimize the network parameter. Its architecture consists of a stem and four connectors. The stem contains five main blocks of which three have exactly the same structure. The first block is designed based on two 7×7 Conv layers followed by a BN and a ReLU activation function. This block uses Conv layers with large kernel sizes to capture the basic information of the object in the input image. Therefore, the input image of size $224 \times 244 \times 3$ after passing through this block will reduce the

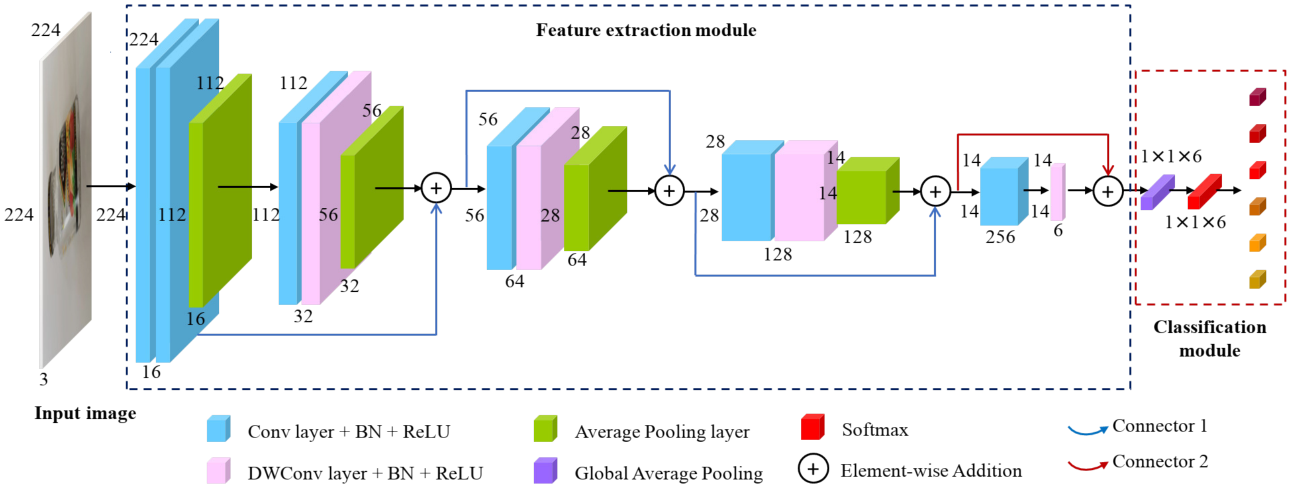


Fig. 2. The proposed trash classification network. This network consists of feature extraction and classification modules.

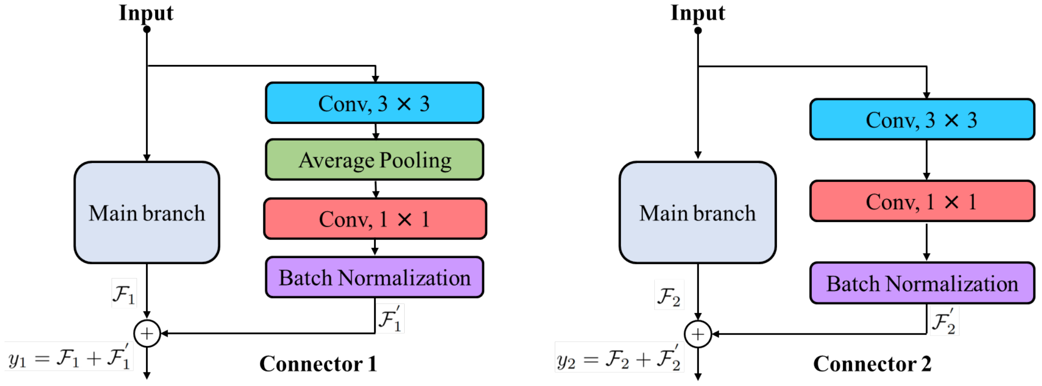


Fig. 3. The architecture of the connector 1 and connector 2.

dimension to $112 \times 122 \times 16$. At the intermediate level are three blocks with the same structure. Each block contains a Conv layer, a DWConv layer, an average pooling layer followed by a BN, and a ReLU activation function. Convolution layers use kernels with sizes reduced from 5×5 to 3×3. These blocks act as an intermediate feature extraction with more detailed information about the object. Through each block, the feature map has further reduced the dimension twice to form the output and link with the connectors. As a result, it generates three feature maps with dimensions of $56 \times 56 \times 32$, $28 \times 28 \times 64$, and $14 \times 14 \times 28$ respectively. The final block is made up of a Conv layer and a DWConv layer. This block acts as a bridge between the feature extraction module and the classification module. The feature map through this block will maintain the dimension 14x14 with the number of channels increasing to 256 and then decreasing to 6 (corresponding to the number of class in the dataset). The output from this block will also be connected to the previous feature map at the level through connector 2 using the element-wise addition operation.

As mentioned above, to combine feature maps at different levels this paper has proposed two types of connectors. Each connector consists of two branches, the main branch (stem) and the sub-branch. In the sub branch on connector 1 use a 3×3 Conv layer, an average pooling layer, and a 1×1 Conv layer followed by a BN. The two branches are then joined together via the element-wise addition operation. In connector 2, the sub-branch has the same architecture as the first connector, but it doesn't use average pooling in between the two Conv layers. With these connectors, the amount of information in the current feature maps is enriched with the information extracted from the previous level. The architecture of these connectors is shown in Fig. 3.

B. *Classification module*

Typically, popular classification networks use fully connected layers in the classification module. However, this technique increases the network parameter significantly because it contains a lot of connections. To solve that, this work also proposes to replace all fully connected layers with just one global average pooling layer while still ensuring classification accuracy. Specifically, this module uses a global average pooling layer to extract a 1x1x6 feature map. It then applies a softmax function on this feature map to calculate the probability of each class (six classes).

C. Loss function

The difference between the predicted value and the target value is calculated using the loss function during training. In this paper, the cross-entropy loss has been used and is defined as follows:

$$\mathcal{L}_{class} = -\sum_{i=0}^{5} \mathcal{P}_i^* .log(\mathcal{P}_i), \tag{1}$$

in which, i is the index of a each class (i from 0 to 5). $\mathcal{P}_i^*$ presents the target indicator (0 or 1) and $\mathcal{P}_i$ denotes the predicted probability from the proposed network. Function *log* is a natural logarithm function.

IV. EXPERIMENT

A. Dataset preparation

The proposed network is trained and evaluated on the TrashNet dataset [18]. This dataset contains 2,527 images divided into six subclasses: glass (501 images), paper (594 images), cardboard (403 images), plastic (482 images), metal (410 images), and trash (137 images)). The images were taken by placing the subject on a white posterboard using sunlight or room light. The devices used to take photos include the Apple iPhone 7 Plus, Apple iPhone 5S, and Apple iPhone SE. The original image size is 512 × 384. For a fair comparison with other research on this dataset, the experiment splits the dataset into 70% for training and 30% for evaluation.

B. Experimental setup

The proposed trash classification network is built using the Python programming language based on the Keras framework. Training and evaluation were performed on a GeForce GTX 1080Ti GPU. In addition, this experiment also uses an Intel Core i7-4770 CPU 3.40 GH CPU (PC) and an Nvidia Maxwell GPU (Jetson Nano device) to evaluate the speed of the network in real-time. Training goes through 200 epochs with a batch size of 16 and an initial learning rate of 10^3 (then reduced by a factor of 0.55 after 20 epochs if the accuracy is not improved). The Adam optimization method is used to update the weights during training. In addition, the network also applies some data augmentation methods to avoid overfittings such as random shift, random zoom, and random brightness.

C. Experimental results

In order to evaluate the performance, this experiment conducts training and evaluation of the proposed network. Proceed to refine the common classifier networks by removing all fully connected layers and replacing with a global average pooling layer, and train and evaluate under the same conditions of the dataset. In addition, to ensure objectively the evaluation results of the proposed network are also compared with other researches using the same dataset. The results in Table 1 show that the proposed network achieves a quite high accuracy of 90.71% with only 653,888 network parameters. This result is superior to both the refined networks and the networks in other studies. When compared with the best-refined competitor (VGG19), the proposed network has 11.26% greater accuracy but its network parameters are 31.44 times less. For other best networks in other research, the proposed network is higher at 1.71% of accuracy. Several samples of the qualitative results of the garbage classification are shown in Fig. 4.

TABLE I
THE COMPARISON RESULT BETWEEN PROPOSED TRASH CLASSIFICATION NETWORK WITH REFINED CLASSIFICATION NETWORKS AND OTHER RESEARCH IN SAME DATASET. THE RED COLOR DENOTES THE BEST COMPETITOR. SYMBOL (∗) DENOTES THE REFINED NETWORKS

Network	Parameters	Accuracy (%)
SqueezeNet∗	257,846	64.82
Proposed	**653,888**	**90.71**
MobileNetV2∗	3,575,878	58.10
MobileNetV1∗	4,284,614	75.69
NASNetMobile∗	5,358,234	75.49
DenseNet∗	8,093,254	64.63
Xception∗	15,246,150	78.46
VGG16∗	15,246,150	78.66
VGG19∗	20,555,846	**79.45**
InceptionV3∗	23,907,110	71.34
ResNet50∗	25,692,038	64.70
LeNet∗	78,430,076	64.63
Method in [18]	-	63.00
CNN in [19]	-	75.00
InceptionV4 [20]	-	**89.00**
DenseNet169 [20]	-	84.00
MobileNet [20]	-	84.00

The confusion matrix in Fig. 5 represents the prediction ratio of the proposed network per class in the dataset. Accordingly, the classes have equally balanced prediction rates from 91% to 94%, except for the two classes, plastic and metal, which have low prediction rates of 84% and 89%, respectively. This is easy to see because the structure and shape of several objects in the glass and trash classes cause confusion for the network during the learning process. Therefore, in the training process, data augmentation is essential to help the network learn a variety of shapes, structures, and contexts.

For evaluating the speed, the network was also tested with a simple real-time system based on a camera connected to a PC and Jetson Nano devices. With VGA camera resolution, the proposed network achieved 24.59 frames per second (FPS) and 15.89 FPS on PC and Jetson Nano devices, respectively. This result shows that the network can be implemented on low-computing devices for real-time trash classification with very low latency. However, the experimental process also presents that the accuracy of the system depends on several factors such as the distance from the camera to the object to be classified (trash), camera resolution, and light of the environment.

D. Ablation study

Each component in the proposed network architecture has a specific role that affects the performance of the entire

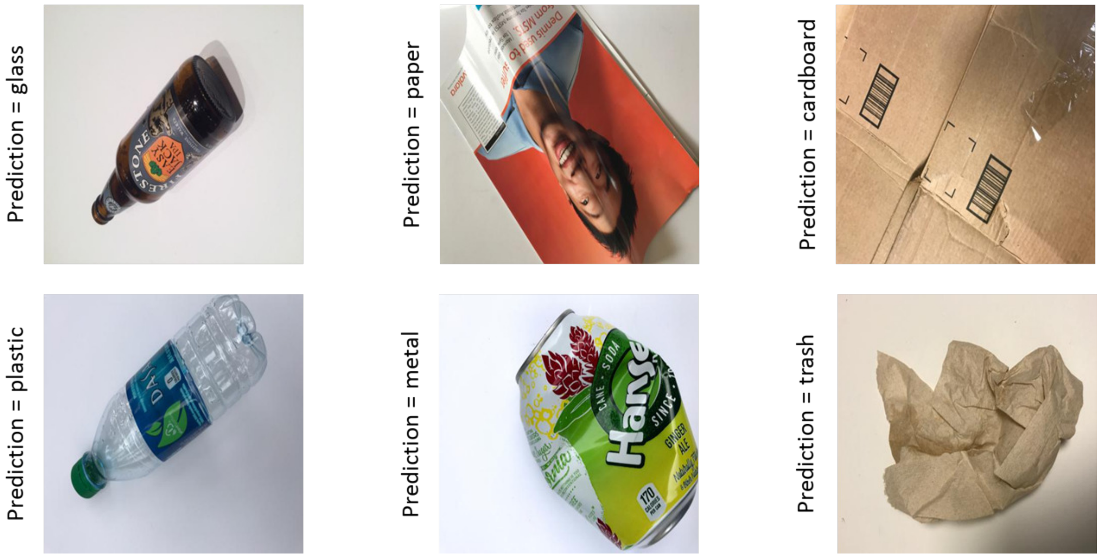


Fig. 4. Several qualitative results of trash classification on TrashNet dataset.

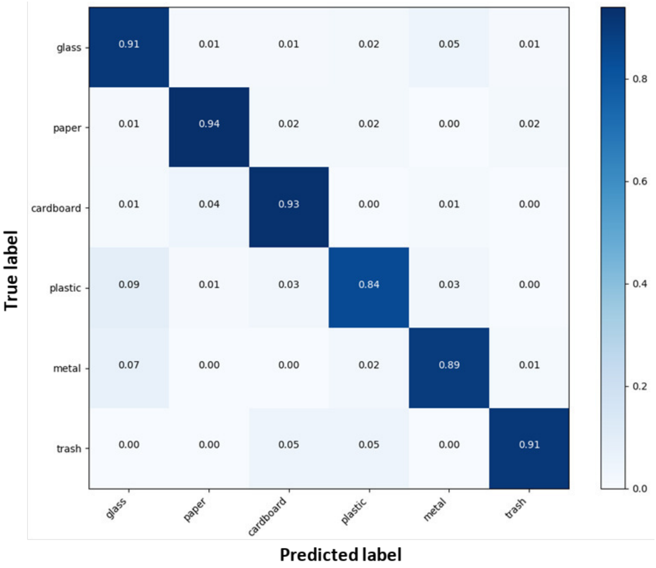


Fig. 5. The confusion matrix of trash classification network on TrashNet dataset.

network. This work conducted several ablation studies to evaluate such effects. First, the experiment conducted training and evaluation on the network without any connector (Stem). Then, gradually increase the connectors from one to four according to the depth of the network. The results in Table II show that when increasing the number of connectors from one to three (connector 1), the network parameter increases, but the accuracy tends to decrease or increase slightly. Until using the fourth connector (connector 2), the best accuracy is achieved.

TABLE II
ABLATION STUDY ON DIFFERENT NUMBER OF CONNECTOR. THE RED COLOR PRESENTS THE BEST NETWORK ARCHITECTURE.

Connectors	Parameters	Accuracy (%)
0 (just stem)	445,608	87.88
1 (connector 1)	448,552	87.62
2 (connector 1)	460,072	88.01
3 (connector 1)	505,640	87.75
4(Proposed)	**653,888**	**90.71**

Next, the experiment investigates the influence of each type of convolution layer on the entire network parameter. The experiment uses all Conv layers instead of DWConv layers and then training and evaluation. The results in Table III demonstrate that the use of DWConv layers has resulted in significant savings in network parameters (195,872 parameters). While the combination of Conv layers and DWConv layers still maintains higher accuracy of the network.

TABLE III
ABLATION STUDY ON DIFFERENT CONVOLUTION TYPE. THE RED COLOR PRESENTS THE BEST NETWORK ARCHITECTURE.

Convolution type	Parameters	Accuracy (%)
All Conv layer	849,760	87.88
Conv+DWConv layers	**653,888**	**90.71**

V. CONCLUSION AND FUTURE WORK

This paper has proposed a lightweight CNN for trash classification consisting of two main modules: feature extraction and classification. The network is built based on the features of Conv, DWConv, average pooling layers, and proposed connectors to extract feature maps. Then apply the global average pooling layer and softmax function to classify the objects. The network is trained and evaluated on the TrashNet dataset with high accuracy and negligible latency when tested in a real-time system. In the future, this trash classification network will be developed into an application that can be integrated into mobile devices.

REFERENCES

[1] "The world's growing municipal solid waste: trends and impacts." Accessed: Jun. 24, 2022. [Online]. Available: https://iopscience.iop.org/article/10.1088/1748-9326/ab8659.

[2] "Solid waste management." Accessed: Jun. 24, 2022. [Online]. Available: https://www.worldbank.org/en/topic/urbandevelopment/brief/solid-waste-management.

[3] "Humans on course to triple daily waste by 2100." Accessed: Jun. 24, 2022. [Online]. Available: https://www.nbcnews.com/sciencemain/humans-course-triple-daily-waste-2100-8c11497511.

[4] "Waste sorting." Accessed: Jun. 24, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Waste_sorting.

[5] G. Mangialardi, G. Trullo, F. Valerio, and A. Corallo, "Sustainability of a pneumatic refuse system in the metropolitan area: A case study in southern apulia region," *Procedia - Social and Behavioral Sciences*, vol. 223, pp. 799–804, 2016. 2nd International Symposium "NEW METROPOLITAN PERSPECTIVES" - Strategic planning, spatial planning, economic programs and decision support tools, through the implementation of Horizon/Europe2020. ISTH2020, Reggio Calabria (Italy), 18-20 May 2016.

[6] "Recycling facilities in the us." Accessed: Jun. 24, 2022. [Online]. Available: https://ibisworld.com.

[7] J. Yang, Z. Zeng, K. Wang, H. Zou, and L. Xie, "Garbagenet: A unified learning framework for robust garbage classification," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 4, pp. 372–380, 2021.

[8] A. Camero, J. Toutouh, J. Ferrer, and E. Alba, "Waste generation prediction in smart cities through deep neuroevolution," in *Smart Cities* (S. Nesmachnow and L. Hernández Callejo, eds.), (Cham), pp. 192–204, Springer International Publishing, 2019.

[9] T. Sheng, M. Shahidul Islam, N. Misran, M. H. Baharuddin, H. Arshad, M. R. Islam, M. Chowdhury, H. Rmili, and M. Islam, "An internet of things based smart waste management system using lora and tensorflow deep learning model," *IEEE Access*, vol. PP, pp. 1–1, 08 2020.

[10] Y. Liu, K.-C. Fung, W. Ding, H. Guo, T. Qu, and C. Xiao, "Novel smart wgaste sorting system based on image processing algorithms: Surf-bow and multi-class svm," *Computer and Information Science*, vol. 11, p. 35, 06 2018.

[11] "Modern convolutional neural networks." Accessed: Jul. 10, 2022. [Online]. Available: https://d2l.ai/chapter_convolutional-modern/index.html.

[12] D. Thanawala, A. Sarin, and P. Verma, "An approach to waste segregation and management using convolutional neural networks," in *Advances in Computing and Data Sciences* (M. Singh, P. K. Gupta, V. Tyagi, J. Flusser, T. Ören, and G. Valentino, eds.), (Singapore), pp. 139–150, Springer Singapore, 2020.

[13] G. White, C. Cabrera, A. Palade, F. Li, and S. Clarke, "Wastenet: Waste classification at the edge for smart bins," 2020.

[14] H. Zheng and Y. Gu, "Encnn-upmws: Waste classification by a cnn ensemble using the upm weighting strategy," *Electronics*, vol. 10, no. 4, 2021.

[15] D. Ziouzios, D. Tsiktsiris, N. Baras, and M. Dasygenis, "A distributed architecture for smart recycling using machine learning," *Future Internet*, vol. 12, no. 9, 2020.

[16] K. Ahmad, K. Khan, and A. Al-Fuqaha, "Intelligent fusion of deep features for improved waste classification," *IEEE Access*, vol. 8, pp. 96495–96504, 2020.

[17] C. Shi, C. Tan, T. Wang, and L. Wang, "A waste classification method based on a multilayer hybrid convolution neural network," *Applied Sciences*, vol. 11, no. 18, 2021.

[18] M. Yang and G. Thung, "Classification of trash for recyclability status," *CS229 project report*, vol. 2016, no. 1, p. 3, 2016.

[19] "trashnet." Accessed: Jul. 24, 2022. [Online]. Available: https://github.com/garythung/trashnet.

[20] R. A. Aral, Ş. R. Keskin, M. Kaya, and M. Hacıömeroğlu, "Classification of trashnet dataset based on deep learning models," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2058–2062, 2018.

DOI: 10.15439/2022R07
ISSN 2300-5963 ACSIS, Vol. 33

A Deep Learning Approach with Stack of Sub-classifiers for Multi-label Classification of Obstructive Disease from Myocardial Perfusion SPECT

Ninh Ngan Trieu
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
trieuninhngan2012@gmail.com

Nhu Hai Phung
Institute of Information Technology
AMST
Hanoi, Vietnam
hainda59@gmail.com

Chi Thanh Nguyen*
Institute of Information Technology
AMST
Hanoi, Vietnam
thanhnc80@gmail.com
*Corresponding author

Thanh Trung Nguyen
Department of Medical Equipment
108 Military Central Hospital
Hanoi, Vietnam
thanhtrungys@yahoo.com

***Abstract*—Artificial intelligence applications, especially deep learning in medical imaging, have gained much attention in recent years. With the computer's aid, Coronary artery disease (CAD) - one of the most dangerous cardiovascular diseases - is diagnosed effectively without human interference and efforts. A lot of research involving predicting CAD from Myocardial Perfusion SPECT has been conducted and given impressive results. However, all existing methods detect whether there is a disease or not. They do not provide information about which obstructive areas are (mainly in the left anterior descending artery (LAD), left circumflex artery (LCx), and right coronary artery (RCA) territories) that result in CAD. To further diagnose CAD, we develop new classifiers to solve a multi-label classification problem with the highest accuracy and area under the receiver operating characteristics curve (AUC) when compared to different methods. Our proposed method is based on transfer learning to extract features from Myocardial Perfusion SPECT Polar Maps and a novel stack of sub-classifiers to detect particularly obstructive areas. We evaluated our methods with eight hundred and one obstructive images from a database of patients referred to a hospital from 2017 to 2019.**

***Index Terms*—CAD, Myocardial Perfusion SPECT, multi-label classification, transfer learning.**

I. Introduction

According to the World Health Organization (WHO), cardiovascular disease (CVD) is currently the leading cause of death globally, accounting for 32 percent of all deaths [1]. At the National Heart Conference 2017, a startling number was reported. Each year, Vietnam has about 200,000 people die from CVD, twice as many deaths from cancer. More importantly, the number of people suffering from CVD at a young age is increasing. Among CVD, coronary artery disease (CAD) and cerebral stroke are the leading causes of death or disability.

Coronary artery disease (CAD) [2] is the most common type of heart disease. It is sometimes called coronary heart disease or ischemic heart disease. CAD develops when the coronary arteries become too narrow, or cholesterol blockages (plaques) develop in the walls. Plaque consists of cholesterol, fatty substances, waste products, calcium and the clot-making substance fibrin. As plaque continues to collect on artery walls, arteries narrow and stiffen. Normally, there are three main obstructive regions of myocardium corresponding to three branches: left anterior descending artery (LAD), left circumflex artery (LCx), and right coronary artery (RCA) territories. This disease damages arteries and impedes supplying oxygen and blood to the heart. Eventually, the blood flow is reduced, causing chest pain (angina), shortness of breath, or other coronary artery disease signs and symptoms. A complete blockage can dangerously cause a heart attack. Particularly CAD caused by acute myocardial infarction, acute coronary syndrome can cause immediate death or lead to heart failure and death later. Moreover, diagnosing CAD usually requires many processes and experienced doctors. Human mistakes are sometimes unavoidable, and those flaws are dangerous, especially in clinical decision-making. Therefore, early and accurate detection of CAD becomes even more urgent nowadays.

In the light of technological developments, the abundance of modern machines are invented, which help diagnose diseases in general and CAD in particular. CAD can be detected by a combination of taking medical history with tests and imaging methods. Currently, there are several methods of diagnosing

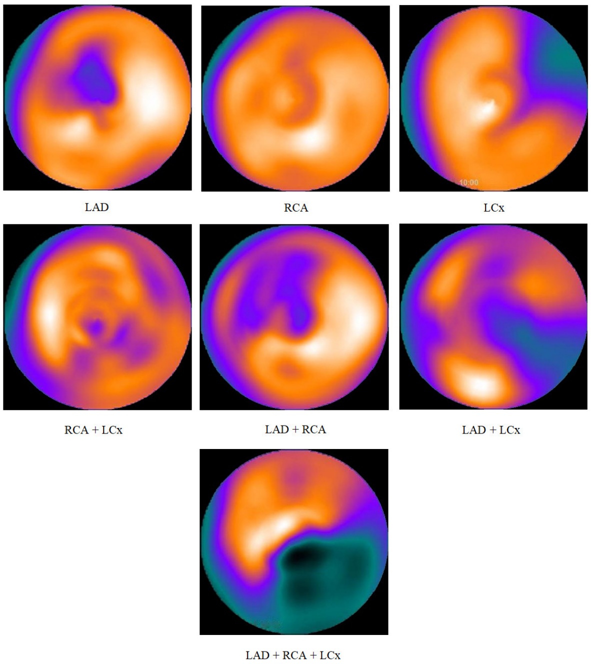


Fig. 1. Examples of SPECT images in our datasets.

CAD, such as electrocardiograph tests which record the electrical activity of the heart, a blood test that analyses factors in the blood that affect arteries, or computed tomography angiogram, which uses CT and contrast dye to view 3D pictures of moving heart and detect blockages in the coronary arteries. Among these techniques, Conventional single-photon emission computed tomography (SPECT) myocardial perfusion imaging (MPI) is one of the most widely used methods. SPECT MPI evaluates the present, extent, and degree of infarction by using gamma rays, providing 3-dimensional images. SPECT is based on the flow-dependent and/or metabolism-dependent selective uptake of a radioactive tracer by functional myocardial tissue. SPECT MPI has gained great success over the past decades as the modality of choice for accurately diagnosing patients with suspected coronary artery.

Many researches have been conducted [3]–[5], [9], [10], [12] based on SPECT MPI to help diagnose CAD accurately and reduce burden on doctors. In [12], four ensemble machine learning algorithms (Adaptive Boosting, Gradient Boosting, Random Forests, and Extreme Gradient Boosting) have been investigated in a dataset including one hundred and seven polar maps. Especially, the features extraction process generating inputs for four algorithms is intuitive and efficient. Each image was sliced into five horizontal and five vertical segments. After that, ten features were created by summing up pixel intensities from each segment. The results are impressive: all models had accuracy $>$ 90 percent and AUC approximately 0.8.

References [4] and [5] analyze 1638 (67% male) and 1160 (64% male) patients without known CAD, respectively. The authors apply deep learning models for polar maps and additional input images. Besides, sex information is also included to produce feature maps. Deep learning models are compared with current quantitative method (TPD - total perfusion deficit). The results demonstrate that deep learning models outperform TPD in terms of area under the receiver operating characteristic curve and sensitivity per patient and vessel.

TABLE I
STRESS POLAR MAPS CHARACTERISTICS.

Number	LAD	RCA	LCx	Image
Train	416	433	210	601
Test	102	109	53	200
Total	518	542	263	801

In [10], the authors utilize a predefined CNN-based model, termed RGB-CNN, which was proposed for other clinical problems, to solve binary-classification detecting CAD. RGB-CNN gives promising results (accuracy = 93.47% ± 2.81%, AUC score =0.936). The proposed methods are then compared with various state-of-the-art CNN backbones for the particular dataset.

Although many existing methods give impressive results in solving binary-classification to classify normal and abnormal SPECT images, none of them work on multi-label classification problems. That means, the computers now are able to predict very well whether or not a patient has disease, but having no clue about specific areas are being damaged.

The contribution of our research is two-fold:

1) We analyze the multi-label classification problem for SPECT images - which has not been studied before, in order to help further diagnose and give the patients and doctors information of areas causing CAD.
2) We propose a novel neural network-based structure solving multi-label classification problems in general.

The paper is organized as follows. In Section I introduces the process of generating datasets. Section III describes the proposed stack for classifying obstructive areas based on a fully connected neural network. Section IV presents the experimental evaluations and analysis. Finally, Section V gives the concluding remarks.

II. MATERIALS

This section generally presents the process of acquiring data. Our SPECT images datasets are collected at the Department of Nuclear Medicine, 108 Military Central Hospital from 2017 to 2019. The datasets were captured at stress by three kinds of specialized SPECT scanners (Infina, Optima, Ventri). It includes 801 polar maps, which are diagnosed with obstructive disease. Before the datasets were collected, all the patient's personal information was removed. This research was conducted with permission from the Department of Nuclear Medicine, 108 Military Central Hospital.

All images are read by three specialists with at least ten years of experience, trained in nuclear medicine in developed countries such as America, Japan, and Australia. Generating

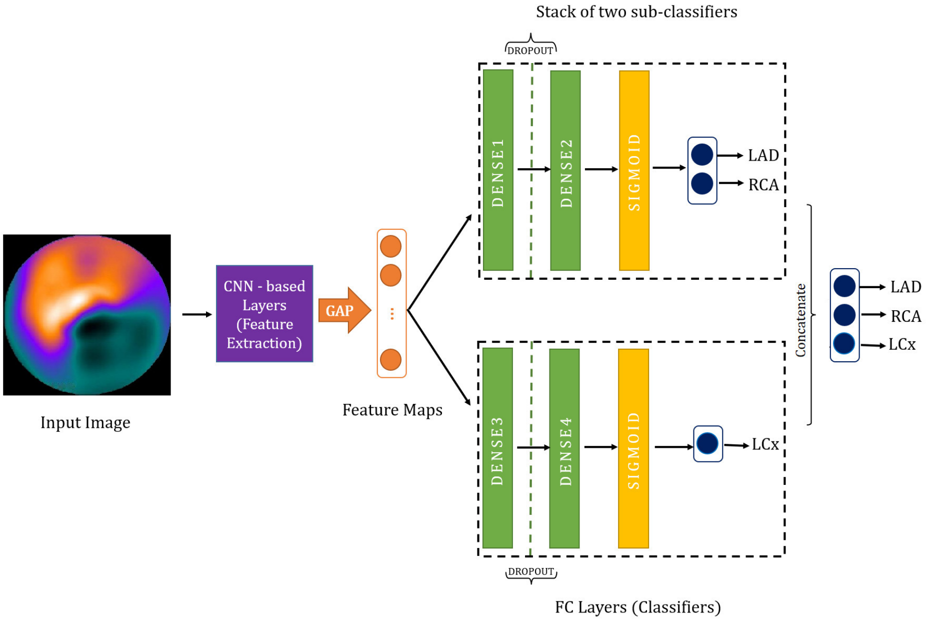


Fig. 2. An example of proposed stack of sub-classifiers solving multi-label classification problem.

SPECT images requires many processes and technical parts, from the pharmacist preparing the radiopharmaceuticals, the injection and imaging technicians, the image quality engineers, and doctors responsible for reading results. Every step requires high accuracy and correct sequence to ensure no errors in the whole implementation process.

A total of 1250 patients without known CAD, injected Tc^{99m_MIBI} (tracer). The patient's body weight determines rest-stress doses by a factor of 0.31mCi/kg. For patients unable to undergo stress physically, they get Diyridamole at a dose of 0,56mg/kg for 4 minutes after heart rate has reached 85 percent [8] [7], to increase the blood flow to the heart muscle as if patients were exercising.

After being injected, traces mix with blood and are taken up by the heart muscle as the blood flows through heart arteries. This radioactive material stored in the myocardium emits gamma rays with an energy level of 140 keV - which is captured by a special camera to show how well the heart muscle is perfused. We use Xeleris - a specialized program of Ge Healthcare for image reconstruction, processing 2D SPECT images, and integrating polar maps. After that, only stress polar maps are kept for further analysis.

Three specialists read and classified images as five levels from normal to surely abnormal. From these five categories, they are grouped more generally into two classes and each group is binary-labeled as normal or abnormal. In case of abnormal polar maps, obstructive areas are indicated, including the left anterior descending artery (LAD), the left circumflex artery (LCx), and the right coronary artery (RCA). From 1250 cases, merely 801 images with CAD and labeled obstructive areas are used to solve the multi-label classification problem. After pre-processing, RGB clinical images are exported in .png format with matrix size 352x352. Figure 1 above illustrates examples of all CAD cases having in our datasets. The datasets are separated into train and test set with the ratio 8:2 respectively. Because of the incidence of patients in Vietnam is uneven (usually LAD and RCA), our data are imbalanced. We can easily notice that the last label (LCx) merely has a half instances compared to others. Table I showcases our datasets in more details.

III. Proposed Stack of Classifiers

In this section, we describe in detail our proposed stack of sub-classifiers - a promising solution solving a multi-label classification problem for SPECT polar images.

For classification, after extracting useful features, classifiers solve the rest of problem. In most deep learning models, fully connected (FC) layers are the potential candidates to take responsibility for classifying objects. In [4], [5], [10], FC layers are also implemented to discern non-obstructive and obstructive SPECT images. However, for the multi-label classification problem, whether or not applying the same architecture to classify is good. Coming up with the idea of finding a multi-label classifier, which is less cumbersome but effective, we proposed a stack of sub-classifiers suitable for our datasets. The proposed idea can be applied to other multi-label classification problems.

Assume our multi-label classification has n labels, and we already found a good feature extractor that be able to extract useful image features. The intuitive idea is transforming multi-label classification to multi-binary classifications, by finding a suitable classifier for each label. Instead of using merely one classifier applied for all labels or a stack of n sub-classifiers for each label, we choose m ones ($0 < m < n$)

TABLE II
HYPERPARAMETERS OF NINE MODELS AFTER IMPLEMENTING HYPERBAND ALGORITHM.

Based model	Branch	Learning rate	Dense node		
			LAD	RCA	LCx
VGG16	3	0.01	576	640	192
	2	0.001	448		192
	1	0.01	64		
ResNet152V2	3	0.0001	640	128	576
	2	0.001	192		448
	1	0.001	128		
InceptionV3	3	0.0001	448	512	128
	2	0.0001	384		448
	1	0.001	640		

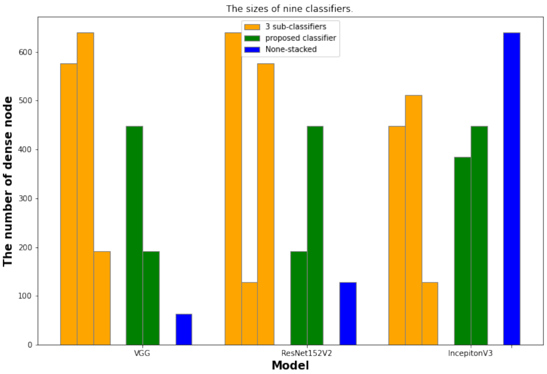


Fig. 3. A visual comparison of the number of dense nodes having in each model.

TABLE III
EXPERIMENTAL RESULTS OF NINE DEEP LEARNING ARCHITECTURES FOR ALL LABELS.

Based model	Number of sub-classifier	Accuracy	AUC
VGG16	3	0.772	0.736
	2 (proposed method)	**0.787**	**0.755**
	1	0.722	0.656
ResNet152V2	3	0.749	0.710
	2 (proposed method)	**0.785**	**0.774**
	1	0.747	0.701
InceptionV3	3	0.741	0.662
	2 (proposed method)	**0.760**	**0.736**
	1	0.737	0.656

and allocate n labels into those sub-classifiers. The features extracted from the previous part are mutual-used as inputs of all sub-classifiers. Based on how balanced our data is and how well the sub-classifier can detect each label, we can find suitable m to construct our stack. For example, we can group easy-to-detect labels into one classifier and others with their own classifiers. In case of our datasets, we have n = 3 (LAD, RCA, LCx), and m = 2, the first two more easy-to-detect labels (LAD and RCA) are classified by the same branch. Figure 2 above illustrates our proposed stack of two sub-classifiers.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Implementation details

In this study, to evaluate the proposed method, we conduct totally nine experiments with three different approaches on three feature extractors utilized from VGG16 [11], ResNet152V2 [6], and InceptionV3 [13] models. We apply transfer learning and use pre-trained feature extractors with weights from Imagenet. Useful features are extracted with these based extractors. Parameters from that part of model are not updated through training process. We mainly focus on finding out suitable classifiers and assessing how well they perform on our datasets. For each based model, three kinds of classifier were analyzed for comparison: a non-stacked classifier, a stack of two, and three sub-classifiers. Totally, nine models are analyzed and evaluated (3 feature extractors x 3 classifiers). To find the optimal hyperparameters of each model, we use Keras Tuner library. Particularly Hyperband algorithm was chosen for all hypermodels.

In addition to the model architecture, we define hyperparameter search space for learning rate and the number of units in FC layers. Our search space has three learning rate (0.01, 0.001, 0.0001) and various dense node for each classifier (from 64 to 640, step 64). As a result, in the case of stacked classifiers, each branch is equivalent to one classifier, having its own hyperparameter. For example, search space in a stack of three branches is much larger than two sub-classifiers and a none-stacked model.

We optimize all networks rigidly with Adam algorithm, Binary cross-entropy for loss function, metrics using are Binary accuracy and area under the receive operating characteristics curve (AUC). For searching hyperparameters, we set objective = validation binary accuracy, max epochs = 100, and early stopping with patience = 5. After each FC layers, we use dropout = 0.2. Moreover, according to table I above, we can easily see that datasets are unbalanced. The number of images labeled LCx (263) is only half of the diagnosed LAD (518) and RCA (542). Therefore, we use different class weights for each label—specifically, LAD, RCA, and LCx with 1, 1, and 2, respectively. After using Hyperband algorithm, the hyperparameters of nine models are chosen and described in Table II and Figure 3.

Nine deep learning models were implemented in Python 3 using open-source libraries, mainly are Tensorflow, Keras and Scikit-learn. All experiments are conducted on an HP computer equipped with an Intel Pentium (R) Core(TM) i5-8250U (1.60 GHz) and 8 GB of RAM, Windows 10 OS.

B. Results

After having hyperparameters, we train nine models with different epochs to optimize parameters with each architecture. We evaluate the model's performances by using the following metrics: accuracy, F1 score, recall, precision, and AUC. We evaluate those metrics for all labels and for each label separately. Table III and IV above showcase the performances of nine models in detail for all labels and each label, respectively.

TABLE IV
EXPERIMENTAL RESULTS OF NINE DEEP LEARNING ARCHITECTURES FOR EACH LABEL.

Based model	Area	Number of sub-classifier	Precision	Recall	F1-score
VGG16	LAD	3	**0.81**	0.84	**0.82**
		2(proposed method)	0.79	**0.85**	**0.82**
		1	0.79	0.76	0.77
	RCA	3	0.76	0.91	0.82
		2(proposed method)	**0.81**	0.89	**0.85**
		1	0.72	**0.97**	0.82
	LCx	3	0.71	**0.72**	**0.71**
		2(proposed method)	**0.74**	0.68	**0.71**
		1	0.68	0.37	0.48
ResNet152V2	LAD	3	0.81	0.77	**0.79**
		2(proposed method)	**0.82**	0.76	**0.79**
		1	0.81	**0.78**	**0.79**
	RCA	3	0.76	0.92	0.83
		2(proposed method)	**0.83**	0.86	**0.84**
		1	0.71	**0.96**	0.82
	LCx	3	0.67	0.60	0.63
		2(proposed method)	**0.73**	**0.75**	**0.74**
		1	0.67	0.68	0.68
InceptionV3	LAD	3	0.80	0.72	0.76
		2(proposed method)	**0.88**	**0.81**	**0.85**
		1	0.77	0.77	0.77
	RCA	3	0.82	**0.87**	**0.85**
		2(proposed method)	**0.85**	0.76	0.80
		1	0.84	0.84	0.84
	LCx	3	**0.80**	0.25	0.38
		2(proposed method)	0.64	**0.45**	**0.53**
		1	0.73	0.23	0.35

According to the results given in Table III above, our proposed method is outstanding in all three based models. In term of accuracy and AUC for all labels, our proposed approach completely outperforms the non-stacked and stack of three sub-classifiers models. In Table IV, performances of nine architectures are presented in particularly each label. In term of LCx - most difficult-to-detect label, our proposed method generally gives better results compared with other methods.

During experiments, we realize that in the case of three sub-classifiers - each label has its classifier, the model is fairly cumbersome. Moreover, the results even drop while having more parameters. Specifically, after just a few epochs, the value of loss function of model during the training process does not improve when the number of epochs increases. The model encountered a vanishing gradient problem.

In terms of non-stacked architecture, the model is not much different from traditional binary classification. Instead of having two units, the last FC layer now comprise n units, where n is the number of labels in multi-label classification. For example, in our datasets, n = 3. This kind of model can work well in binary classification, the idea can be found in [4], [5], [10]. However, as our results have proved, most non-stacked models have the poorest performance because of trouble to predict LCx. In our experiments, the models can detect with much higher accuracy for the first two labels (LAD, RCA) compared to the last label (LCx). The reason is that our dataset is imbalanced and detecting LCx is harder than the other two labels because the number of images having this label (263) is only half of LAD (518) and RCA (542) (Table I). Non-stacked architecture may still detect well on balanced datasets. However, imbalanced labels in multi-label problems are sometimes unavoidable, especially when the number of labels increases significantly.

The remaining problems of the two models above are addressed in terms of the two-sub-classifier model. By finding suitable branches for our stack and allocating labels in the appropriate sub-classifiers, we can deal with cumbersome architecture and give better result on unbalanced datasets. The results in Table III demonstrate the efficiency of our proposed method. This stack of two sub-classifiers effectively solves the multi-label classification problem with promising results.

V. Conclusion

This paper applied transfer learning methods, utilizing pre-trained features extractors from pre-knowledge of Imagenet, and proposed a practical stack of classifiers solving multi-label classification problems. We conduct extensive experiments on different models and classifiers on clinical polar SPECT datasets. Experimental results demonstrate that our proposed stack works well and gives the most outstanding results when combined with various features extractors while keeping a moderate number of parameters.

More importantly, our method can give better results in imbalanced datasets for each label - one of the most challenging obstacles that multi-label classification commonly faces. Our proposed method can help solve other multi-label classification problems with promising efficiency and accuracy. In the future, we will work more to address multi-label classification for imbalanced SPECT images datasets and go further with others multi-label classification problems. Hopefully, our study is able to apply in solving other multi-label classification problems for diverse kinds of objects and datasets.

References

[1] Cardiovascular diseases. World Health Organization, https://www.who.int/health-topics/cardiovascular-diseases, accessed on 2022-06-14

[2] Coronary artery disease. Mayo Foundation for Medical Education and Research (May 2022), https://www.mayoclinic.org/diseases-conditions/coronary-artery-disease/symptoms-causes/syc-20350613, accessed on 2022-06-14

[3] Apostolopoulos, I., Papathanasiou, N., Spyridonidis, T., Apostolopoulos, D.: Automatic characterization of myocardial perfusion imaging polar maps employing deep learning and data augmentation. Hellenic journal of nuclear medicine **23** (07 2020). https://doi.org/10.1967/s002449912101

[4] Betancur, J., Commandeur, F., Motlagh, M., Sharir, T., Einstein, A.J., Bokhari, S., Fish, M.B., Ruddy, T.D., Kaufmann, P., Sinusas, A.J., et al.: Deep learning for prediction of obstructive disease from fast myocardial perfusion spect: a multicenter study. JACC: Cardiovascular Imaging **11**(11), 1654–1663 (2018)

[5] Betancur, J., Hu, L.H., Commandeur, F., Sharir, T., Einstein, A.J., Fish, M.B., Ruddy, T.D., Kaufmann, P.A., Sinusas, A.J., Miller, E.J., et al.: Deep learning analysis of upright-supine high-efficiency spect myocardial perfusion imaging for prediction of obstructive coronary artery disease: a multicenter study. Journal of Nuclear Medicine **60**(5), 664–670 (2019)

[6] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. pp. 770–778 (06 2016). https://doi.org/10.1109/CVPR.2016.90

[7] Hesse, B., Tägil, K., Cuocolo, A., Anagnostopoulos, C., Bardiès, M., Bax, J., Bengel, F., Busemann Sokole, E., Davies, G., Dondi, M., et al.: Eanm/esc procedural guidelines for myocardial perfusion imaging in nuclear cardiology. European journal of nuclear medicine and molecular imaging **32**(7), 855–897 (2005)

[8] Holly, T., Abbott, B., Al-Mallah, M., Calnon, D., Cohen, M., DiFilippo, F., Ficaro, E., Freeman, M., Hendel, R., Jain, D., Leonard, S., Nichols, K., Polk, D., Soman, P.: Single photon-emission computed tomography (10 2010). https://doi.org/10.1007/s12350-010-9246-y

[9] Kaplan Berkaya, S., Ak, I., Gunal, S.: Classification models for spect myocardial perfusion imaging. Computers in Biology and Medicine **123**, 103893 (07 2020). https://doi.org/10.1016/j.compbiomed.2020.103893

[10] Papandrianos, N., Papageorgiou, E.: Automatic diagnosis of coronary artery disease in spect myocardial perfusion imaging employing deep learning. Applied Sciences **11**(14), 6362 (2021)

[11] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv 1409.1556 (09 2014)

[12] de Souza Filho, E.M., Fernandes, F.d.A., Wiefels, C., de Carvalho, L.N.D., dos Santos, T.F., dos Santos, A.A.S.M.D., Mesquita, E.T., Seixas, F.L., Chow, B.J.W., Mesquita, C.T., Gismondi, R.A.: Machine learning algorithms to distinguish myocardial perfusion spect polar maps. Frontiers in Cardiovascular Medicine **8**, 1437 (2021). https://doi.org/10.3389/fcvm.2021.741667, https://www.frontiersin.org/article/10.3389/fcvm.2021.741667

[13] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. pp. 2818–2826 (06 2016). https://doi.org/10.1109/CVPR.2016.308

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 267–271 DOI: 10.15439/2022R23 ISSN 2300-5963 ACSIS, Vol. 33

Performance Evaluation of Short-Packet Communications of Single-Hop System with Presence of Co-Channel Interference

1st Vu Le Anh Uyen
Faculty of Electrical and Electronics Engineering
Ho Chi Minh City
University of Transport
Ho Chi Minh City, Vietnam
vu.le@ut.edu.vn

2nd Tien-Tung Nguyen
Faculty of Electronics Technology
Industrial University
of Ho Chi Minh City
Ho Chi Minh City, Vietnam
nguyentientung@iuh.edu.vn

3rd Duy Tran Trung
Department of Electrical Engineering
Posts and Telecommunications
Institute of Technology
Ho Chi Minh City, Vietnam
trantrungduy@ptithcm.edu.vn

4th Lam-Thanh Tu
Communication and Signal Processing
Research Group
Faculty of Electrical and
Electronics Engineering
Ton Duc Thang University
Ho Chi Minh City, Vietnam
tulamthanh@tdtu.edu.vn

5th Tan Nguyen Nhat
Communication and Signal Processing
Research Group
Faculty of Electrical and
Electronics Engineering
Ton Duc Thang University
Ho Chi Minh City, Vietnam
nguyennhattan@tdtu.edu.vn

6th(*) Sang Nguyen Quang
Science and Technology Application
for Sustainable Development Research Group
Ho Chi Minh City
University of Transport
Ho Chi Minh City, Vietnam
sang.nguyen@ut.edu.vn
(*) Corresponding author

***Abstract*—An investigation of short-packet communications (SPCs) of a single hop system where there is a presence of co-channel interference at destination. The average block error rate (BLER) of the destination is derived in accurate as well as asymptotic closed-form expressions. The affects of some parameters like: the number of interference sources, the transmit power of the interference source on the system performance and the number of channel uses at the destination are considered. The results of the theory are evaluated by Monte Carlo simulation.**

***Index Terms*—average block error rate, short-packet communication, single-hop, co-channel interference**

I. Introduction

Recently, ultra-low latency and ultra-high reliability (URLL) is required in several operations serving as industrial automation, tele-surgery, tactile internet [1]. In order to meet the requirement, short-packet communications (SPCs) [2] is performed for the finite blocklength code transmission. Relied on the Shannon theory, in conventional communication systems, errors is dependent on both two parameters: the received signal-to-noise ratio (SNR), the coding rate. In the system deploying the short packet communication, in case of the coding rate is smaller the Shannon capacity, errors always exist; in addition, the packet blocklength affects the error rate. This leads to the classical Shannon theory not to be used to evaluate SPC systems, but to use another metric, which is BLER. Hence, SPC has attracted researchers and was investigated in different systems.

For single-hop system, an evaluation of the BLER of a non-orthogonal multiple access system with two users was presented in [3]. For dual-hop system, short packet communication were studied in [4], [5] in terms of BLER. The throughput and probability of coverage of SPC was evaluated for Multiple-input-multiple-output (MIMO) systems [6]. A performance for SPC in an unmanned-aerial-vehicle (UAV) system, where an UAV is a relay carrying out URLL connections between a user and a base station, was analyzed in [7]. The authors of [8] derived the BLERs of two destinations in an UAV system with presence of SPC. A solution to joint optimal power as well as blocklength was proposed to obtain max–min throughput for cooperative non-orthogonal multiple access [9]. The authors of [10] provided a deep-learning framework for predicting the BLERs for Internet of Things system. In [11], SPC was investigated in a cognitive radio network helped by an intelligent reflecting surface. However, these above works did not consider co-channel interference in the system. The authors in [12] study the affect of co-channel interference to the performance of the cooperative networks.

In this paper, the key contributions are listed as follows:

- We consider a single-hop system, where one source adopts short-packet communication to transmit the signal to its destination. In addition, we take into account co-channel interference in this system.
- Under setting, the BLERs of the destination are derived in closed-form and asymptotic expressions.
- The affects of some parameters, i.e., the number of interference sources, the transmission power of the interference source and the number of channel uses on the

performance also are evaluated.

- The simulation results are executed to validate the exact of our theory analysis. This simulation method is used effectively and widely in many studies [13]–[19]

A. Organization

In Section II, a description of the system model is presented. The performance of the system with regards to the BLER is analyzed in Section III. In Section IV, simulation and analysis results are discussed. Finally, a conclusion is provide in Section V.

II. System model

We considered a wireless system composed by one access point (i.e., S) and one Internet of Things (IoT) user (i.e., U) with presence of co-channel interference, as showed in Fig. 1

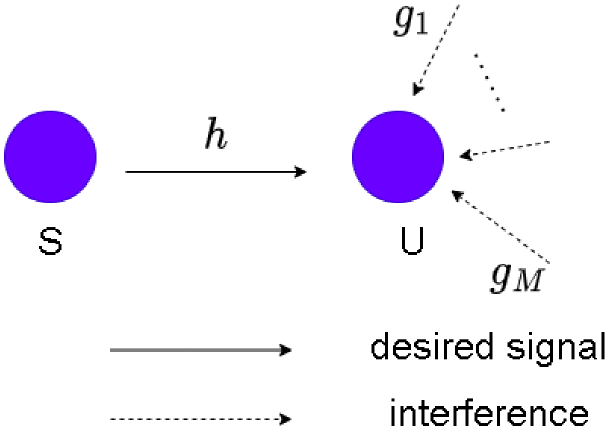


Fig. 1. System model of a single-hop communication system with the co-channel interference.

In this paper, the channel state information are assumed to know at all nodes. The signal at U is received from S as

$$y_{\mathrm{U}} = \sqrt{P_{\mathrm{S}}} x_{\mathrm{U}} h + \sum_{m=1}^{M} \sqrt{P_{\mathrm{I_m}}} g_{\mathrm{I_m U}} x_{\mathrm{I_m}} + n_{\mathrm{U}}, \tag{1}$$

where P_{S}, $P_{\mathrm{I_m}} = P_{\mathrm{I}}$ denote transmission power of the S, interference source, respectively. h, g_m present the channel coefficients of S-U link, and interference source m-U link, respectively. $n_{\mathrm{U}} \sim \mathcal{CN}(0, \sigma^2)$ is the additive white Gaussian noise (AWGN) at U. Therefore, the signal-to-interference-plus-noise ratio (SINR) at U is obtained as

$$\Lambda_{\mathrm{U}} = \rho_{\mathrm{S}} |h|^2 / [\rho_{\mathrm{I}} \kappa + 1], \tag{2}$$

where $\kappa = \sum_{m=1}^{M} |g_{\mathrm{I_m U}}|^2$.

III. Performance Analysis

Assuming that S transmits to U with the amount of information bit $\mathcal{N}$ and blocklength l. Hence, the average BLER to decode signal of U can be calculated by [2]

$$\bar{\epsilon}_{\mathrm{U}}(\Lambda_{\mathrm{U}}, \mathcal{N}, l) \approx \int_0^\infty Q\left(\frac{C(\Lambda_{\mathrm{U}}) - r_{\mathrm{U}}}{\sqrt{V(\Lambda_{\mathrm{U}})/l}}\right) f_{\Lambda_{\mathrm{U}}}(x) dx, \tag{3}$$

where $r_{\mathrm{U}} \triangleq \mathcal{N}/l$. In order to reduce the complexity of (3), $Q(x)$ is approximated as follows

$$\Xi(\Lambda_{\mathrm{U}}) \approx \begin{cases} 1, & \Lambda_{\mathrm{U}} \le v_{\mathrm{U}}, \\ 0.5 - \chi_{\mathrm{U}} \left(\Lambda_{\mathrm{U}} - \tau_{\mathrm{U}}\right), & v_{\mathrm{U}} < \Lambda_{\mathrm{U}} < u_{\mathrm{U}}, \\ 0, & \Lambda_{\mathrm{U}} \ge u_{\mathrm{U}}. \end{cases} \tag{4}$$

where $\chi_{\mathrm{U}} = \left[2\pi(2^{2r_{\mathrm{U}}} - 1)/l\right]^{-1/2}$, $\tau_{\mathrm{U}} = 2^{r_{\mathrm{U}}} - 1$, $v_{\mathrm{U}} = \tau_{\mathrm{U}} - 1/(2\chi_{\mathrm{U}})$, and $u_{\mathrm{U}} = \tau_{\mathrm{U}} + 1/(2\chi_{\mathrm{U}})$. By replacing (4) into (3), $\bar{\epsilon}_{\mathrm{U}}$ can be obtained as

$$\bar{\epsilon}_{\mathrm{U}} \approx \int_0^\infty \Xi(\Lambda_{\mathrm{U}}) f_{\Lambda_{\mathrm{U}}}(x) dx \approx \chi_{\mathrm{U}} \int_{v_{\mathrm{U}}}^{u_{\mathrm{U}}} F_{\Lambda_{\mathrm{U}}}(x) dx. \tag{5}$$

The average BLER, e_{U} of the destination is derived as following

$$e_{\mathrm{U}} = \chi(u_{\mathrm{U}} - v_{\mathrm{U}} - \Psi H), \tag{6}$$

where $\Psi = \frac{\Omega_h \rho_{\mathrm{S}}}{\bar{\Omega}_g \rho_I}$,
$H = \exp(-v_{\mathrm{U}}/(\Omega_h \rho_{\mathrm{S}})) \sum_{k=1}^{M-1} \frac{(k-1)!(-1/\Omega_h \rho_{\mathrm{S}})^{M-k-1}}{(M-1)!(v_{\mathrm{U}}+\Psi)^k} - \frac{(-1/\Omega_h \rho_{\mathrm{S}})^{M-1} \exp \Psi \ (\Omega \rho_{\mathrm{S}})}{(M-1)!} Ei(-(v_{\mathrm{U}} + \Psi) \ (\Omega_h \rho_{\mathrm{S}})) - \exp(-u_{\mathrm{U}}/(\Omega_h \rho_{\mathrm{S}})) \sum_{k=1}^{M-1} \frac{(k-1)!(-1/\Omega_h \rho_{\mathrm{S}})^{M-k-1}}{(M-1)!(u_{\mathrm{U}}+\Psi)^k} - \frac{(-1/\Omega_h \rho_{\mathrm{S}})^{M-1} \exp \Psi \ (\Omega \rho_{\mathrm{S}})}{(M-1)!} Ei(-(u_{\mathrm{U}} + \Psi) \ (\Omega_h \rho_{\mathrm{S}}))$.

Proof: The probability density functions (PDFs) of the random variables $X = |h|^2$ and $\kappa = \sum_{m=1}^{M} |g_{\mathrm{I_m U}}|^2$ are $f_X(x) = 1/(\Omega_h) \exp(-x/\Omega_h)$ and $f_\kappa(x) = (1/\bar{\Omega}_g) x^{M-1}/(M-1)! \exp(-x/\bar{\Omega}_g)$, respectively.

Firstly, the CDFs of Λ_{U} given in (2) can be expressed as

$$\begin{aligned} F_{\Lambda_{\mathrm{U}}}(z) &= \Pr\Big(\frac{\rho_{\mathrm{S}} X}{\rho_{\mathrm{I}} \kappa + 1} \le z\Big) \\ &= \int_0^\infty F_X\Big(z\left(\frac{\rho_{\mathrm{I}} y + 1}{\rho_{\mathrm{S}}}\right)\Big) f_\kappa(y) dy. \end{aligned} \tag{7}$$

With help of Eq. 3.351.3 in [20], we have

$$F_{\Lambda_{\mathrm{U}}}(z) = 1 - \left(\frac{\Psi}{z + \Psi}\right)^M \exp\left(\frac{-z}{\Omega_h \rho_{\mathrm{S}}}\right). \tag{8}$$

Then, putting (8) in (5) and using Eq. 3.353.1 in [20], we obtain the equation as in (6).

IV. Simulation and numerical results

We set the parameters as following $\mathcal{N} = 80$(bits), $l = 100$ channel uses (CUs), $d_{\mathrm{SR}} = 40\,(\mathrm{m})$; $\nu = 2$ denotes the path loss exponent [21]. The interference sources are placed positions which have random distances between them and the destination and in range [50, 100 m].

Fig. 2 shows the average BLER of the destination changing with the the source's transmit power and with different transmit power at the interference source. It is clearly that the average BLER decreases as the transmit power of the source increases. The curses presenting theory match well

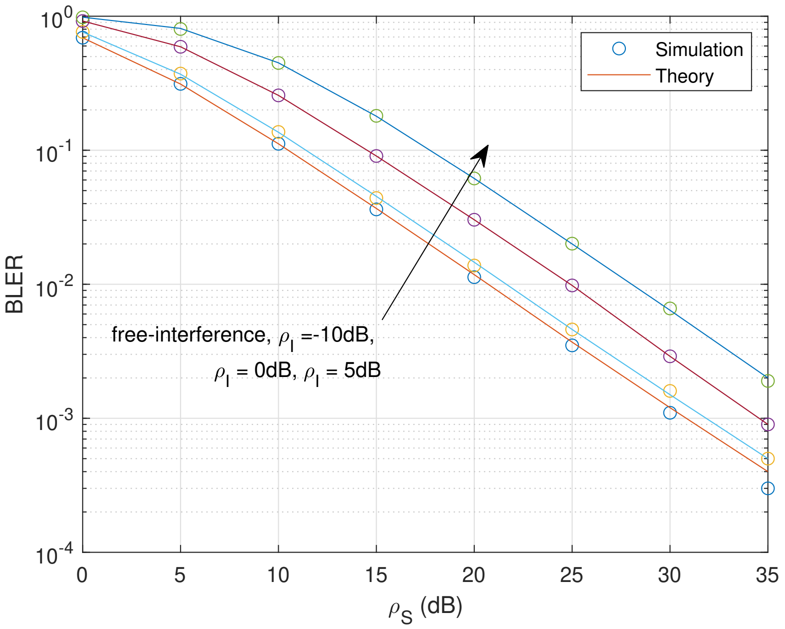


Fig. 2. Average BLER versus ρ_S with $M = 3$.

with that presenting simulation. Moreover, it is clear that the interference source's transmit powers affect significantly on the the performance of the system with regards to the BLER.

Fig. 3 indicates the average BLER with a change of the number of interference sources. Obviously, increase in the number of interference sources results in a reduction of the BLER of the destination. This is because increase in co-channel interference makes decoding at the destination worse.

To evaluate affect of the number of channel use on the performance, we provide Fig. 4 where the BLER is changed with the number of channel uses. We can see that the increase in the amount of channel uses leads to better performance because the longer the length of blockcode is, the more successful the decoding at the destination is.

V. Conclusions

In this paper, an investigation SPC in a single-hop communication system with the presence of the co-channel interference was presented. The BLER of the destination for was derived. The results based on evaluating the transmit power of the inference source and the number of interference sources affecting on the system performance that the two parameters make the performance of the system worse. In the future works, some solutions such as deploying multiple antenna at the source or at the destination to reduce affect of the co-channel interference; and the number of hop with the assistance of relay nodes can be considered to improved the system performance.

References

[1] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[2] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[3] Y. Yu, H. Chen, Y. Li, Z. Ding, and B. Vucetic, "On the performance of non-orthogonal multiple access in short-packet communications," *IEEE Communications Letters*, vol. 22, no. 3, pp. 590–593, 2018.

[4] X. Lai, Q. Zhang, and J. Qin, "Cooperative noma short-packet communications in flat rayleigh fading channels," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6182–6186, 2019.

[5] T.-H. Vu, T.-V. Nguyen, T.-T. Nguyen, V. N. Q. Bao, and S. Kim, "Short-packet communications in noma-cdrt iot networks with cochannel interference and imperfect sic," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5552–5557, 2022.

[6] J. Zheng, Q. Zhang, and J. Qin, "Average achievable rate and average bler analyses for mimo short-packet communication systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 238–12 242, 2021.

[7] L. Yuan, N. Yang, F. Fang, and Z. Ding, "Performance analysis of uav-assisted short-packet cooperative communications," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4471–4476, 2022.

[8] T.-T. Nguyen and S. Q. Nguyen, "Short packet communications for cooperative uav-noma-based iot systems with sic imperfections," *Computer Communications*, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422003693

[9] F. Salehi, N. Neda, M.-H. Majidi, and H. Ahmadi, "Cooperative noma-based user pairing for urllc: A max–min fairness approach," *IEEE Systems Journal*, vol. 16, no. 3, pp. 3833–3843, 2022.

[10] T.-H. Vu, T.-V. Nguyen, T.-T. Nguyen, and S. Kim, "Performance analysis and deep learning design of wireless powered cognitive noma iot short-packet communications with imperfect csi and sic," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 10 464–10 479, 2022.

[11] P. N. Son, T. T. Duy, P. V. Tuan, and T.-P. Huynh, "Short packet communication in underlay cognitive network assisted by an intelligent reflecting surface," *ETRI Journal*, vol. n/a, no. n/a. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.2021-0435

[12] S. Q. Nguyen and H. Y. Kong, "Exact outage analysis of the effect of co-channel interference on secured multi-hop relaying networks," *International Journal of Electronics*, vol. 103, no. 11, pp. 1822–1838, 2016. [Online]. Available: https://doi.org/10.1080/00207217.2016.1138534

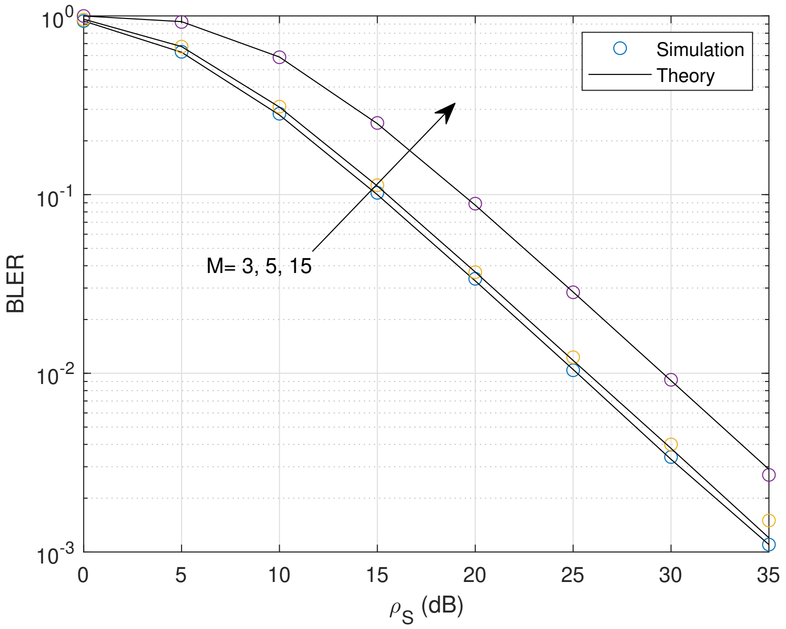


Fig. 3. Average BLER versus the number of interference sources with $\rho_{\mathrm{S}} = \{0, 35\}$ (dB), $\rho_{\mathrm{I}} = 0$dB.

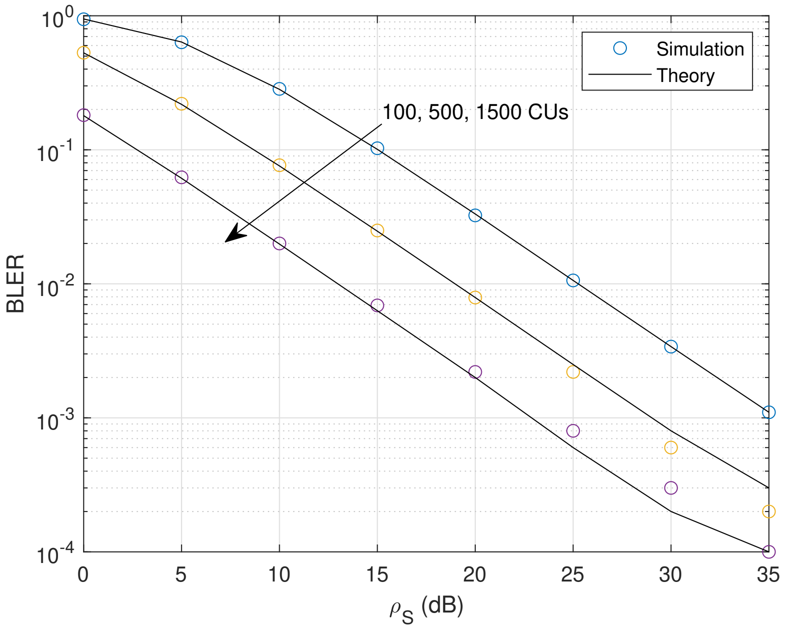


Fig. 4. Average BLER versus the number of channel uses with $M = 3$, $\rho_I = 0dB$.

[13] D.-B. Ha, S. Q. Nguyen, and H. T. Nguyen, "Cooperative cognitive non-orthgonal multiple access under unreliable backhaul connections," *Mobile Networks and Applications*, vol. 24, no. 2, pp. 596–617, Apr 2019. [Online]. Available: https://doi.org/10.1007/s11036-018-1161-5

[14] S. Q. Nguyen and H. Y. Kong, "Generalized diversity combining of energy harvesting multiple antenna relay networks: outage and throughput performance analysis," *Annals of Telecommunications*, vol. 71, no. 5, pp. 265–277, Jun 2016. [Online]. Available: https://doi.org/10.1007/s12243-016-0508-9

[15] L. T. TU, P. L. Tung, T. Van Chien, T. T. Duy, and N. T. Hoa, "Performance evaluation of incremental relaying in underlay cognitive radio networks with imperfect csi," in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*, 2021, pp. 472–477.

[16] P. T. D. Ngoc, T. T. Duy, and H. V. Khuong, "Outage performance of cooperative cognitive radio networks under joint constraints of co-channel interference, intercept probability and hardware imperfection," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 6, no. 19, 6 2019.

[17] S. Q. Nguyen and H. Y. Kong, "Secrecy enhancement in two-hop df relaying system under hardware impairment," *International Journal of Electronics*, vol. 104, no. 3, pp. 442–461, 2017. [Online]. Available: https://doi.org/10.1080/00207217.2016.1218063

[18] P. Tran Tin, T. N. Nguyen, N. Q. Sang, T. Trung Duy, P. T. Tran, and M. Voznak, "Rateless codes-based secure communication employing transmit antenna selection and harvest-to-jam under joint effect of interference and hardware impairments," *Entropy*, vol. 21, no. 7, 2019. [Online]. Available: https://www.mdpi.com/1099-4300/21/7/700

[19] P. Viet Tuan, P. Ngoc Son, T. Trung Duy, S. Q. Nguyen, V. Q. B. Ngo, D. Vinh Quang, and I. Koo, "Optimizing a secure two-way network with non-linear swipt, channel uncertainty, and a hidden eavesdropper," *Electronics*, vol. 9, no. 8, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/8/1222

[20] A. Jeffrey and D. Zwillinger, *Table of integrals, series, and products*. Elsevier, 2007.

[21] O. L. Alcaraz López, E. M. G. Fernández, R. D. Souza, and H. Alves, "Ultra-reliable cooperative short-packet communications with wireless energy transfer," *IEEE Sen. J.*, vol. 18, no. 5, pp. 2161–2177, 2018.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 273–278 DOI: 10.15439/2022R20 ISSN 2300-5963 ACSIS, Vol. 33

Design of scheduling algorithms for UAVs to Detect Air Pollution Sources from Chimneys in Industrial Area

Le Van Vinh, Ngo Thanh Huyen
Faculty of Information Technology, Hung Yen University of Technology and Education, Hungyen, Vietnam
Email: vinhlv.utehy@gmail.com, nthuyenster@gmail.com

***Abstract*—With the rapid development of Unmanned Aerial Vehicle (UAV), many related applications using UAVs is to monitor air quality in urban, rural or industrial areas. They often focus on how to monitor the propagation of air pollution, provided the pollution sources should be positioned with permanently placed wireless sensors. However, it is hard and time-consuming to identify pollution sources due to a number of chimneys in industrial areas. Therefore, to air pollution source detection in the minimum search time from the chimneys with fixed locations in an industrial park using one or more UAVs. In this paper, we propose two heuristics algorithms for air-pollution-source detection by UAVs including Interference-Graph- Based Algorithm (IGBA), and Extended Interference-Graph-Based Algorithm (EIGBA). As a result, the detection time by these proposed algorithms compared with that by the Traveling Salesman Problem (TSP) algorithm air pollution source detection time is significantly reduced.**

***Index Terms*—Gaussian Plume Model, Unmanned Aerial Vehicle, Air Pollution.**

I. Introduction

Today with the advancement of science and technology, human civilized life is increasingly prosperous, it is followed by the impact on human health and damage to the earth ecosystem far beyond the previous level [1]. Among them, air pollution has become an issue of global concern, in addition to the serious impact on the environment, it also affects human due to deteriorating air quality that humans have never faced before. Therefore, the issue of air quality has become a difficult problem that people face and measures need to be taken to solve today's environmental problems.

To monitor the air pollution source, in [2] they set up air quality monitoring stations, by placing sensors in the monitoring area and transmitting information about the base station for inspection and analysis. Traditional methods for air pollution measurement are expensive and have a spatial constraint. With these limitations, air pollution monitoring in a broader area is not feasible, the problem of using the modern low-cost multi-sensor node for air pollution measurement in conjunction with wireless sensor network (WSN) is to collect and aggregate real-time data from different locations and provide detailed pollution map proposed in [3]. This urban air pollution monitoring system is based on static locations, so this problem is limited by mobility. To overcome this limitation. In [4], that uses a combination of portable mobile sensor units, smart-phones, cloud computing, and mobile apps to measure air pollution information for individuals.

In recent years, UAV development and application have made great progress such as mobility, low cost, lightweight, and ease of use [5]. First, the UAV application is used in the military, with the aim of replacing large aircraft and humans with small UAVs for reconnaissance missions, which not only reduces the probability of casualties for humans but also increases the ability to conceal the quest [6]. Next, due to low cost, reduced job execution time and ease of use [7], package delivery services [8]. In the field of air environment detection, adding sensors to UAVs replaces permanently placed traditional wireless sensors, for collecting, and sending the collected data to the remote acquisition point [9]. That is useful for comprehending the impact of the local environment caused by air pollution.

The problem of adding sensors to the UAV, in addition to portability it also reduces the cost of environmental monitoring [10]. In [11], UAVs are equipped with off-the-shelf sensors to perform air pollution monitoring tasks, after transmitting data to the base station, it can draw pollution maps of the monitoring area. In critical areas [12], use UAVs to monitor air pollution in inconvenient traffic areas by focusing on areas with high air pollution to redraw the map polluted. . we noticed that, previous studies on air pollution focused on environmental monitoring, proposing solutions and improving, there have been no studies on pollution sources, and most pollution sources are diverse such as pollution sources in urban areas, rural areas, industrial areas, etc. In order to improve the air environment quality in the monitored area, it is necessary and timely to find the fastest and most accurate source of air pollution by the UAVs.

From the above problems. In this paper, we propose a model for air pollution source detection with reduced time in industrial areas by UAV. First, we propose TSP-based algorithm to solve flight scheduling problem. Second, we use the K-means clustering algorithm to divide the area into the number of sub-areas corresponding to the number of UAVs. Then we rely on the interference graph relationship to remove the positions that are not in the interference graph. The main contributions of this paper are as follows:

1) We propose the task allocation method for UAV based on clustering algorithm and mTSP-based flight scheduling

method.

2) We propose solution and experimental to the problem of optimizing the air pollution source detection time by the UAVs with the relationship of the interference graph.
3) Improved UAV features such as mobility, ease of maintenance, low cost, and being able to apply artificial intelligence (AI).

II. System model and problem definition

In this paper, our goal is to minimum search time air pollution source detection by UAVs. In Section II-A, we will talk about how the system model. In Section II-B, we generalize the Gaussian plume dispersion model. In Section II-C, we introduce a model of polluted chimney detection. In Section II-D, we will describe the problem definition of this paper.

A. System model

As illustrated in Fig. 1, We consider a system based on actual environmental conditions in a geographic area and it consists of a base station (BS), a set $U = \{U_1, U_2, ..., U_m\}$ of UAVs and a set $P = \{P_1, P_2, ..., P_n\}$ of the chimneys diffuse pollutant plume are distributed on the ground has a fixed position. The BS has responsibilities including air pollution source detection in the entire monitored surrounding area, providing chimney location coordinate information for UAVs, replenishing energy for UAVs, and planning a flight route for the UAV. The UAVs are equipped with air pollution sensors and have limited battery capacity. In our investigation, the UAV flight schedule is provided by the base station and to land at the initial position. When the BS air pollution source detection in the monitored area, the UAV will take-off at the initial position with the cruising speed u_v and the concentration value of the detected source of pollution u_c, assume the value of pollutant concentration threshold ξ. Initially, the UAV flew according to schedule to each chimney location to air pollution source detection. During the flying of the UAV, when the detected u_c concentration is found to be higher than ξ, a chimney search disperses pollution is performed. To shorten the search time for polluting chimneys. the UAVs will fly in search of chimneys in the wind direction, the chimneys that are not in the wind direction will be removed from the original schedule. When the UAV detects a pollution chimney location, the UAV will skip the remaining chimney locations and to return initial position.

B. Pollution source model

When we look at the atmosphere, we see that there are many factors that influence the behavior of a pollutant plume: wind speed, atmospheric stability, the occurrence of temperature inversion, plume temperature, plume exit speed, landscape, etc. To simulate the diffusion of the pollutant source, it is necessary to calculate quickly and accurately. In [13], [14] the Gaussian plume model can predict the surrounding concentration of a pollutant source. Assuming that the wind speed and direction are constant as shown in Fig. 2, we can observe a pollutant plume emitted continuously from the chimney mouth. When pollutant plume stain rises to a certain height, it will not continue to increase but it will remain motionless. If there is wind, from the chimney mouth the plume rise caused by the momentum of the plume as it leaves the stack or by the buoyancy as hot plumes are lighter than ambient air and stretches across the horizontal plane in the wind direction. Hence, in different weather conditions, then the instantaneous concentration a pollutant in a plume will be irregular. Set the air pollution concentration at the detection position to $C(x, y, z)$, the equation for pollutant concentrations in Gaussian plumes is as follows:

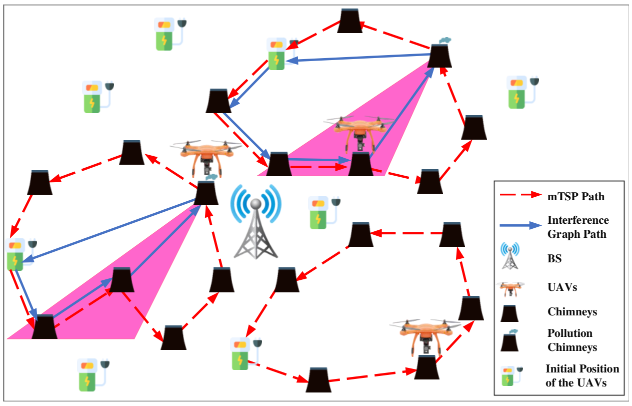


Fig. 1: System model to the air pollution source detection

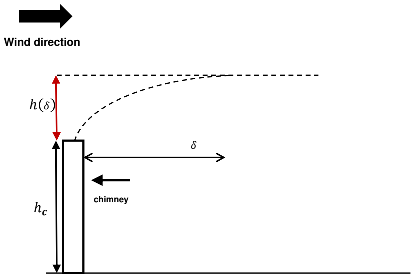


Fig. 2: The Gaussian plume dispersion model.

$$C(x,y,z) = \frac{Q}{V}\frac{1}{2\pi\sigma_y\sigma_z}\exp(-\frac{1}{2}\frac{y^2}{\sigma_y^2})\exp(-\frac{z^2}{2\sigma_z^2}) \quad (1)$$

$$\sigma_y = \frac{ax}{(1+bx)^{0.5}} \quad (2)$$

$$\sigma_z = \frac{dx}{(1+ex)^{0.5}} \quad (3)$$

where

C = concentration at a given point (gm^3)
Q = emission rate (gs^1)
V = wind speed (ms^1)

σ_y = dispersion parameter in the horizontal (lateral) direction (m)

σ_z = dispersion parameter in the vertical direction (m)

The coordinate axes in Gaussian plume dispersion modeling are illustrated in Fig. 3 as follows:

x = direction of the wind (x = 0 at the source; x >0 downwind).

y = horizontal direction perpendicular to the wind (y = 0 at the center of the plume; positive on your left when you look downwind).

z = vertical direction (z = 0 at the surface and positive above the surface).

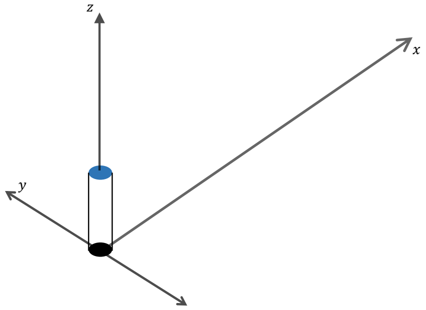


Fig. 3: Coordinate system in simple Gaussian dispersion model

From Eq 1 and Fig. 4, the wind speed depends on height, the wind speed at the effective source height h should be used for V and the spread parameters depend on the distance from the source and on weather conditions.

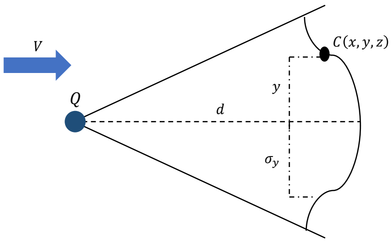


Fig. 4: Horizontal plume dispersion in the wind direction

C. Model of detecting pollution chimney

To detect whether a chimney is a source of air pollution, by easiest intuition is that the UAVs fly to the chimney to check, but the chimney emits stack of pollutant at very high temperatures can lead to the UAV malfunctioning. Hence, we set up the UAV to fly about an δ distance away from the chimney mouth to ensure that the UAV will not fail. As shown in Fig. 2, effective source height h of the UAV is the sum of source height h_c and plume rise $h(\delta)$. The chimney diffuses the plume do not rise indefinitely but stabilize at a certain height, the final plume rise height. The height equation of plume rise $h(\delta)$ achieved at a distance δ from the source is as follows:

$$h(\delta) = \frac{1.6F_b^{\frac{1}{3}}\delta^{\frac{2}{3}}}{V} \tag{4}$$

where, F_b is the floating force parameter, this parameter is related to the gravitational acceleration and the temperature difference.

As shown in Fig. 5, when the UAV flies to the chimneys to detect the source of air pollution, the UAV flies in a $semicircle$ from point $p^{'}$ to point $p^{''}$. If the concentration values of the pollutant plume diffuse u_c are found greater than the value of the concentration threshold δ, it means that the chimney being detected is a source of pollution, otherwise, the chimney being detected is not a source of pollution.

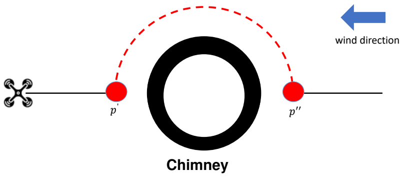


Fig 5: Semicircle diagram

D. Problem definition

In this section, the main issue discussed is that when the UAV gets the necessary information from the BS provided. The ideal situation is that the UAV can air pollution source detection and flies back to its initial position with excess energy. Assuming the search time is T, our goal is to minimize the search time for T as short as possible. Therefore, the problem of flight planning and flight time of the UAV to air pollution source detection is carried out. The search time of multiple UAVs is calculated based on the calculation of the first UAV that air pollution source detection and flies back to its initial position. Accordingly, Minimum-SearchTime Air Pollution Source Detection Problem (MSTAPSD) is proposed.

III. Problem Statement

A. Schedule

In this phase, since all chimney locations are known. If the expected flight path is given in advance for the UAV, the UAV can avoid having to calculate the flight path by itself, which will reduce power consumption and computation time for the UAV. Hence, UAV path planning can be divided into two different formats.

In the first one, uses single UAV to air pollution source detection in the minimum search time. First, the UAV take-off from initial position, then the UAV flies to the positions of all the chimneys by the shortest path, at each position of the UAV chimney flying in a semicircle. Obviously, the minimum flight time problem is similar to Traveling Salesman Problem (TSP) [15]. TSP forms the basis of the problem we're facing and considered as one of the most popular NP-complete problems where TSP can be defined as visiting all cites for the UAV in the environment once in a time.

The second format uses multiple UAVs to air pollution source detection in the minimum search time. Hence we cut down the area of responsibility for the UAVs using the K-means clustering algorithm [16], in each cluster has a UAV which corresponds to air pollution source detection. Definition 2 and Definition 3 in section II-D is a variation of multiple traveling salesman problems (mTSP), which is to minimize the maximum distance traveled by any UAV (min-max mTSP) [17], [18] , Compared with minimizing the total distance by all the UAVs.

B. Interference Graph

During the flight schedule of the UAV to the air pollution source detection. When the UAV hovered along the flight path and detected the pollution location. Then the UAV must know which source of air pollution is coming from the chimney. Hence, our goal at this stage is to establish a relationship diagram for the chimney locations, i.e. in which direction the UAV will fly, which fly to the next chimney location, and remove which chimney positions were in the original flight schedule. We use the following two as requests during this phase. 1) Wind direction: Know the wind direction that the source of pollution can affect; 2) Pollution source formula: Knowing the spread of the pollution source and being able to know which source of pollution affects the chimney positions as shown in Fig. 6.

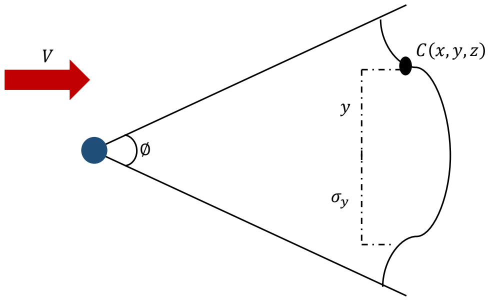


Fig 6: The air pollution source interference graph

IV. PROPOSED ALGORITHMS

In this section, algorithms that optimize the air pollution source detection time are based on flight schedules, interference graph. The details of this algorithm are as follows.

A. Interference-Graph-Based Algorithm (IGBA)

Combining the two problems presented in Part III including the UAV flight schedule and the interference map. First, the UAV take-off at the initial position and flies according to a flight schedule based on the TSP algorithm in Section III-A with using all chimney positions. Then during the flight of the UAV, the UAV detects the pollutant plume in the air. According to the air pollution source interference graph mentioned in Section III-B, the UAV determines where the chimney locations can cause the source of air pollution and deletes the chimney locations that do not cause pollution in the interference map. Therefore, the time to find the air pollution source of UAV will decrease compared to the original schedule.

Algorithm 1 IGBA(C,u,p, θ, δ)

```
$(V_{G_r}, E_{G_r}) \leftarrow$ constructInterferenceGraph $(C)$
Let $R$ be the routing path $(u, r_1, r_2, \ldots, r_n, u)$ constructed
    by the Ant Algorithm [17, 18] to visit $\{u\} \cup C$
$psrc \leftarrow NULL$
Remove the first node from $R$
while $R \neq \emptyset$ do
    Let $x$ be the node pop up from the first item of $R$
    Let $pos'$ and $pos''$ be the checkpoints obtained by $x$,
    $\theta$, $\delta$
    if the UAV detects the air pollution at $pos'$ then
        Move UAV to $pos''$
        if the UAV detects the air pollution at $pos''$ then
            Let $Y$ be the set of nodes $y$ for all $(y, x) \in E_{G_r}$
            for each $r$ in $R$ do
                if $r \notin Y$ then
                    pop up $r$ from $R$
                end if
            end for
        else
            $psrc \leftarrow x$
            $R \leftarrow (u)$
        end if
    end if
end while
return $psrc$
```

B. Extended Interference-Graph-Based Algorithm (EIGBA)

In this algorithm, the implementation problems are similar to the IGBA algorithm in Section IV-A. But the difference is that more UAVs are used to air pollution source detection, each UAV is responsible for a corresponding area. The first stage is clustering, we based on the number of UAVs to divide into the number of corresponding clusters using K-mean algorithm. Then, we scheduled the flight using the multiple traveling salesman problem (mTSP). Finally, the relationship diagram is implemented similar to the IGBA algorithm after mTSP has been computed. EIGBA proceeds as follow.

Algorithm 2 EIGBA(C,U,p, θ, δ)

```
Let $\mathcal{C} = \{C_1, C_2, \ldots, C_{|U|}\}$ be the set of groups constructed by K-means, where $C = C_1 \cup C_2 \cup \ldots \cup C_{|U|}$ [16]
$psrc \leftarrow NULL$
for each $u \in U$ do
    Let $c$ be the chimney closest to $u$
    Let $C_i$ be the group that contains $c$
    $psrc \leftarrow$ IGBA($C_i$,$u$,$p$, $\theta$, $\delta$)
    if $psrc$ then
        return $psrc$
    end if
    Remove $C_i$ from $\mathcal{C}$
end for
```

V. SIMULATION RESULTS

In this section, we present simulation results to demonstrate performance of our Minimum-SearchTime Air Pollution Source Detection algorithms. We consider a UAV-enabled model air pollution source detection in industrial area that consists of UAVs, chimneys, and wind direction. The position of the chimneys is randomly placed in a squareshaped area from 800m^2 × 800m^2 to 1800m^2 × 1800m^2, the number of UAVs and chimneys will increase or decrease according to the simulation scenario, and the number of clusters corresponds to the number of UAVs. In each flight schedule, the UAVs were dispatched to air pollution source detection along the flight trajectory with the velocity v set as 10m/s.

The (TSP, mTSP)-based algorithm optimizes the air pollution source detection time proposed and compared with the IGBA, EIGBA algorithms subverted by the original (TSP, mTSP) algorithm.

A. Comparison of search problems with single UAV

In this subsection, to air pollution source detection with single UAV, the TSP-based algorithm is compared with the IGBA algorithm.

1) **Comparison the number of the chimneys** In Fig. 7, we compare the air pollution source detection time of the UAV in each flight schedule corresponding to the number of chimneys increased from 10 to 80. We noticed that as the number of chimneys increased, the TSP-based algorithm with UAV hovering around all the chimney locations. Therefore, UAVs perform a longer flight schedule, because many chimney positions are not required for the UAV to fly over. Our proposed IGBA algorithm is combined with the interference graph. Hence it has the relationship of obtaining information from the air pollution source information and it is used to remove unnecessary chimney positions, resulting in a significantly reduced flight time of the UAV on a schedule. So the IGBA algorithm results in shorter flight times than the TSP-based algorithm.

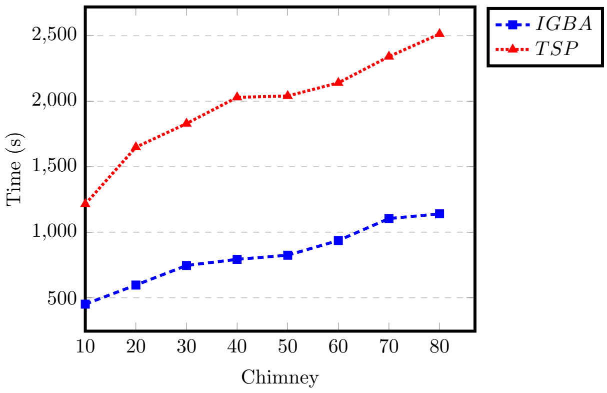


Fig 7: Compare the air pollution source detection time with single UAV when the number of chimneys increases.

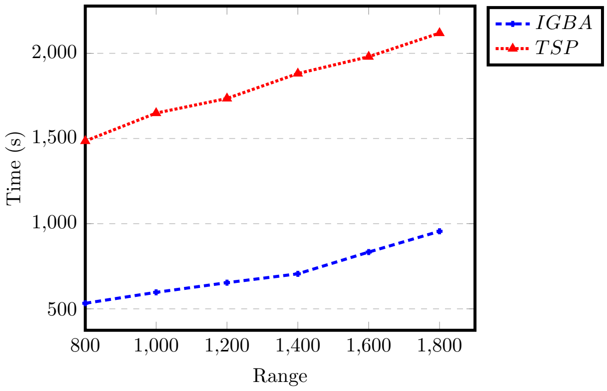


Fig 8: Compare the air pollution source detection time with single UAV when changing area.

2) **Comparison areas of change in size** With the number of chimneys maintained at 50. we noticed that the search time increases gradually as the range expands from 800m^2 to 1800m^2. When the range is enlarged, the positions of the chimneys will be further apart, so the distance between the chimneys will increase. Since the maximum diffusion range of the air pollution source is constant, the diffuse pollution source is only detected within a certain range. The results are shown in Fig. 8, the air pollution source detection time of the IGBA algorithm is shorter than that of the TSP-based algorithm.

B. Comparison of search problems with multiple UAVs

In this subsection, to air pollution source detection with multiple UAV, the mTSP-based algorithm is compared with the EIGBA algorithm.

1) **Comparison areas of change in size** With the number of chimneys maintained at 50. we noticed that the

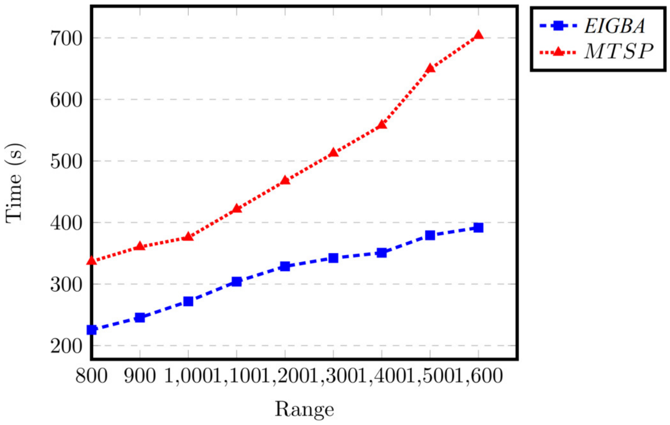


Fig 9: Compare the air pollution source detection time with multiple UAV when changing area.

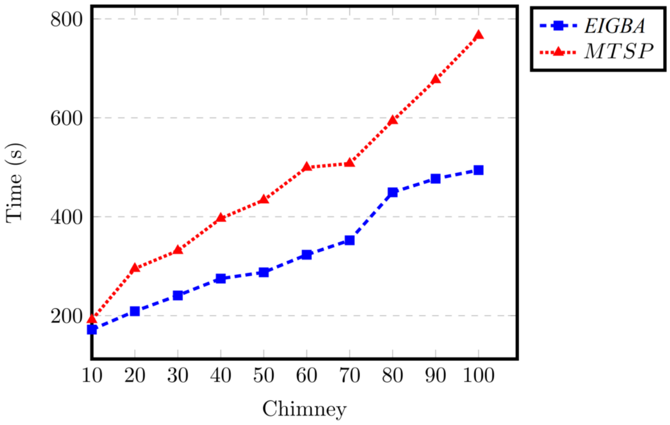


Fig 10: Comparison of the number of chimneys that minimize the detection problem of multiple UAV.

air pollution source detection time of the EIGBA and TSP algorithms gradually increases as the area increases from 800m^2 to 1600m^2. Due to increased position and distance between chimneys. In Fig. 9, the air pollution source detection time of EIGBA algorithm is shorter than that of mTSP algorithm.

2) **Comparison the number of the chimneys** In this subsection we compare the change in the number of chimneys shown in Fig. 10. As the number of chimneys increased, leading to more and more chimney positions in the flight path, so the flight distance of the UAVs also increased. Since the EIGBA algorithm benefits from the air pollution source interference graph reflected at this time, the flight path removes unnecessary chimney locations. Therefore, the air pollution source detection time of the EIGBA algorithm has a shorter flight schedule than the mTSP algorithm.

VI. CONCLUSIONS

In applications using sensors to detect air quality is the mainstream. These applications often focus on monitoring the spread of air pollution when the source of the pollution is known, resulting in deficiencies and failure to meet actual needs. To overcome the above limitations. In this paper, we use UAVs to the air pollution source detection with low cost and high efficiency. First, we flight schedule for the UAVs to monitor respective areas. Next, based on the position where the UAV detected the pollutant plume to construct the interference graph. Finally, remove chimney locations that are not in the interference graph. Through simulation experiments, with the use of the interference graph method. Our proposed algorithms have a shorter air pollution source detection than the (TSP, mTSP)-based algorithm.

REFERENCES

[1] P. W. Birnie and A. E. Boyle, *International law and the environment*, 1994.

[2] K. K. Khedo, R. Perseedoss, and A. Mungur, “A wireless sensor network air pollution monitoring system,” *arXiv preprint arXiv:1005.1737*, 2010.

[3] B. Bathiya, S. Srivastava, and B. Mishra, “Air pollution monitoring using wireless sensor network,” in *2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 2016, pp. 112–117.

[4] K. Hu, V. Sivaraman, B. G. Luxan, and A. Rahman, “Design and evaluation of a metropolitan air pollution sensing system,” *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1448–1459, 2015.

[5] Z. Qiu, X. Chu, C. Calvo-Ramirez, C. Briso, and X. Yin, “Low altitude uav air-to-ground channel measurement and modeling in semiurban environments,” *Wireless Communications and Mobile Computing*, 2017.

[6] M. Benjamin, *Drone warfare: Killing by remote control*. Verso Books, 2013.

[7] S. Pochwała, A. Gardecki, P. Lewandowski, V. Somogyi, and S. Anweiler, “Developing of low-cost air pollution sensor—measurements with the unmanned aerial vehicles in poland,” *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3582

[8] S. Ito, K. Akaiwa, Y. Funabashi, H. Nishikawa, X. Kong, I. Taniguchi, and H. Tomiyama, “Load and wind aware routing of delivery drones,” *Drones*, vol. 6, no. 2, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/2/50

[9] C. Zhan, Y. Zeng, and R. Zhang, “Energy-efficient data collection in uav enabled wireless sensor network,” *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2017.

[10] Q. Gu, D. R Michanowicz, and C. Jia, “Developing a modular unmanned aerial vehicle (uav) platform for air pollution profiling,” *Sensors*, vol. 18, no. 12, p. 4363, 2018.

[11] O. Alvear, N. R. Zema, E. Natalizio, and C. T. Calafate, “Using uav-based systems to monitor air pollution in areas with poor accessibility,” *Journal of Advanced Transportation*, vol. 2017, 2017.

[12] D. Ni, G. Yu, and S. Rathinam, “Unmanned aircraft system and its applications in transportation,” *Journal of Advanced Transportation*, vol. 2017, 2017.

[13] A. De Visscher, *Air dispersion modeling: foundations and applications*. John Wiley & Sons, 2013.

[14] A. Green, R. Singhal, and R. Venkateswar, “Analytic extensions of the gaussian plume model,” *Journal of the Air Pollution Control Association*, vol. 30, no. 7, pp. 773–776, 1980.

[15] G. Reinelt, “Tsplib—a traveling salesman problem library,” *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991.

[16] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[17] E. Benavent and A. Martínez, “Multi-depot multiple tsp: a polyhedral study and computational results,” *Annals of Operations Research*, vol. 207, no. 1, pp. 7–25, 2013.

[18] M. Latah, “Solving multiple tsp problem by k-means and crossover based modified aco algorithm.” *International Journal of Engineering Research and Technology*, vol. 5, pp. 430–434, 2016.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 279–284 DOI: 10.15439/2022R05 ISSN 2300-5963 ACSIS, Vol. 33

Safety-Assisted Driving Technology Based on Artificial Intelligence and Machine Learning for Moving Vehicles in Vietnam

Hong-Son Vu[1,*] , Van-Hien Nguyen[1]

[1]Faculty of Electrical and Electronics Engineering, Hung Yen University of Technology and Education

* Email: hongson.ute@gmail.com

***Abstract*—ADAS (Advanced Driver Assistance Systems) plays an important role in building a safe and modern traffic system. For these systems, precise detection performance and response speed are critical. However, the detection of mobile vehicles is facing many difficulties due to the density of vehicles, the complex background scene in the city, etc. In addition, the detection and identification requirements respond in real time is also a challenge for current systems. This paper proposes a model using deep learning algorithms and artificial intelligence to increase accuracy and improve response speed for intelligent driving assistance systems. Accordingly, this paper proposes the YOLO (You Only Look One) model together with a sample data set collected and classified separately suitable for Vietnam traffic and our training algorithm. The experimental results were then performed on an NVIDIA Jetson TX2 embedded computer. The experimental results show that, the proposed method has increased the speed by at least 1.5 times with the detection rate reaching 79% for the static camera system; and speed up at least 1.5x with a detection rate of 89% for the dynamic camera system at 1280x720px high resolution images.**

***Index Terms*—ADAS Advanced Driver Assistance Systems, YOLO model, Deep learning algorithm and Artificial Intelligence.**

1. Introduction

ADAS - Advanced Driver Assistance Systems plays an important role in building a safe and modern traffic system. In addition to building infrastructure, the development of new technologies for smart mobile vehicles is one of the most important elements that make up this system. Detecting and recognizing surrounding vehicles and giving warnings to users will help make driving safer. Today, with the hardware platform are embedded computers with compact size, high performance, integrated with many powerful supporting technologies for image processing, along with machine learning models developed by leading technology companies to train deep learning neural networks, this makes it easy for researchers to choose the right machine learning model to test the proposed methods.

The research results of applying deep learning algorithms and artificial intelligence to detect and recognize moving vehicles for ADAS intelligent driving assistance systems have been pointed out by the authors in [1] [2] [3]. This paper proposes a model using deep learning algorithms and artificial intelligence to increase accuracy and improve response speed for Advanced Driver Assistance Systems. Accordingly, we first propose the YOLO model [4] along with a sample data set collected and classified separately suitable for Vietnamese traffic and our training algorithm. The experimental results were then performed on an NVIDIA Jetson TX2 embedded computer. The experimental results show that, the proposed method has increased the speed by at least 1.5 times with the detection rate reaching 90% for the static camera system; and speed up at least 1.5x with a detection rate of 67% for dynamic camera systems at 1280x720px high resolution images.

2. Related research

2.1. YOLO model

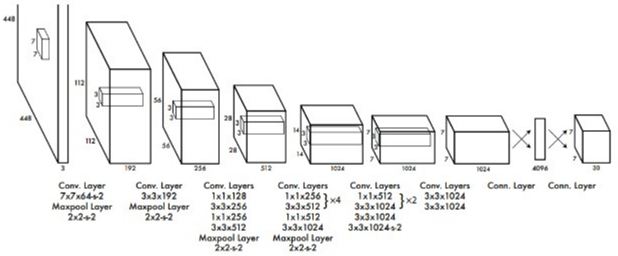


Figure 1: YOLO algorithm model.

YOLO [5] is a convolutional neural network (CNN) model that enables object detection and recognition. YOLO is created from a combination of convolutional layers and connected layers. In which the convolutional layers will extract the features of the image, while the full-connected layers will predict 8the probability and coordinates of the object.

In algorithms, indexes are always evaluated against the ground truth (which is predefined from the dataset via coordinates (cx, cy, w, h) to help identify the object). Ground truth is the location of the provided objects in the training, validated and tested dataset. For object detection systems, the ground truth includes the image, the classes of objects in it, and the frames surrounding the ground truth (ground truth box) of each object in the image.

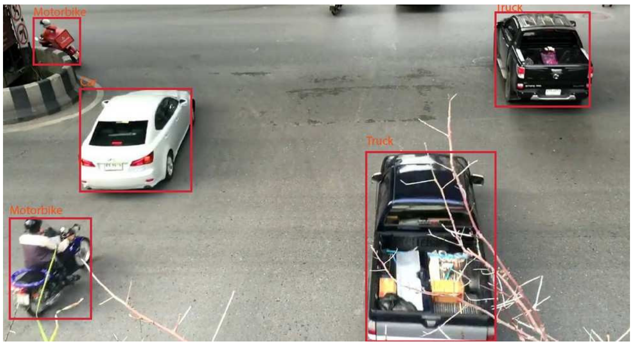


Figure 2: Define ground truth for objects

As shown in Figure 2, the ground truth boxes defined on the trained and tested images. Assume the original image and the caption for the ground truth are the same as the image above, training and test data all images are annotated in the same way, the model will return a lot of predictions, but most of them have very low confidence, so only those predictions above a certain confidence are considered. The original image through the model and the object detection algorithm will return the results of the location of the object in the image according to a threshold of confidence.

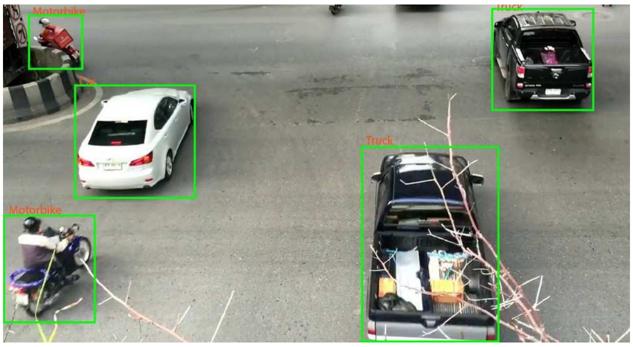


Figure 3: The model's prediction results

To evaluate the accuracy, it is first necessary to evaluate the accuracy of each prediction on the image. To calculate the accuracy of a bounding box, it is necessary to use the IoU measure - Intersection over Union: a ratio that measures the degree of intersection between two frames (usually the prediction frame and the ground truth frame) to aim determine if 2 frames overlap or not.

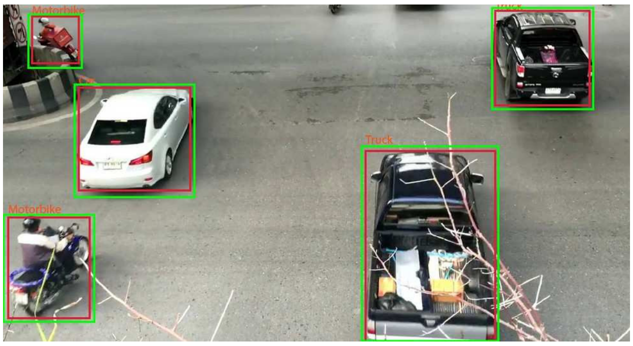


Figure 4: Compare the prediction results with the object's ground truth.

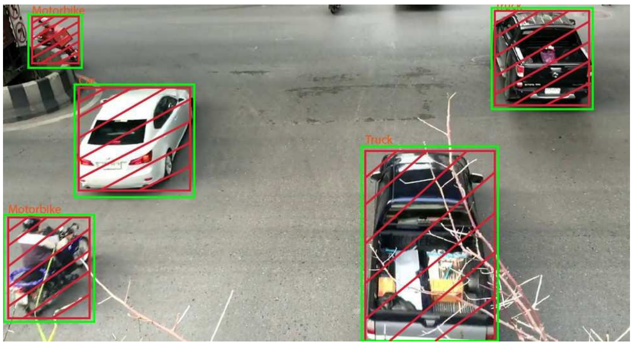


Figure 5: The area of intersection between the prediction result and the ground truth.

The IoU will be calculated as follows:

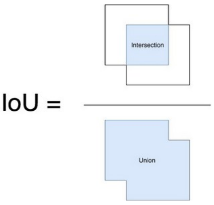


Figure 6: IoU

This ratio is calculated based on the area of intersection between 2 frames with the total area of intersection and non-intersection between them.

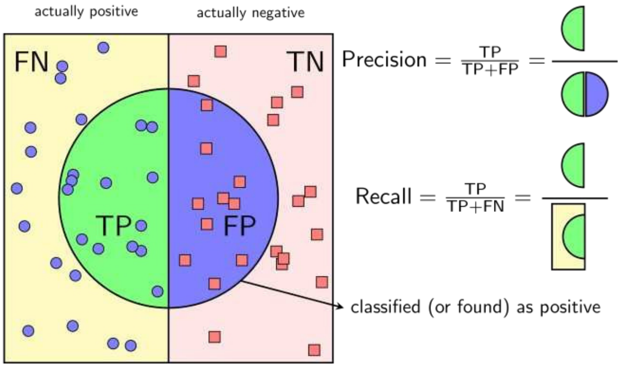


Figure 7: Precision and Recall

For classification problems where the data sets of the classes differ greatly (unbalanced), the Precision-Recall evaluation method will often be used [6].

Then, Precision is defined as the ratio of the number of positive points the model correctly predicts to the total number of points the model predicts is positive. The higher the precision, the higher the number of points the model predicts is positive. Precision = 1, that is, all the points that the model predicts are positive are correct, or there are no points labeled as negative that the model incorrectly predicts as positive.

$$Precision = \frac{TP}{TP+FP} = \frac{TP}{Total\ prediction} \tag{1}$$

Recall is defined as the ratio of the number of positives the model correctly predicted to the total number of points that are actually positive (or the total number of points labeled as positive initially). The higher the recall, the lower the number of missed positives. Recall = 1, that is, all points labeled as positive are recognized by the model [6].

$$Recall = \frac{TP}{TP+FN} = \frac{TP}{ground\ truth} \tag{2}$$

2.2. Training methods

YOLO predicts multiple bounding boxes for each grid cell. To calculate the error for correct predictions, it is necessary to define one of the bounding boxes responsible for the object. For this purpose, choose the one with the highest IoU with a true bounding box. This strategy leads to specialization in bounding box prediction. It is better to predict certain sizes and aspect ratios. YOLO uses the Sum-Squared Error function between the prediction and the desired value to calculate the loss [5].

The YOLO loss function has the form:

$$\begin{aligned}L=\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2]\\ +\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2]\\ +\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}(C_i-\hat{C}_i)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{noobj}(C_i-\hat{C}_i)^2\\ +\sum_{i=0}^{S^2} l_{ij}^{obj}\sum_{c\,\in\,Classes}[p_i(c)-\hat{p}_i(c)]^2\end{aligned} \tag{3}$$

The first part of the loss calculation equation deals with the location of the prediction bounding box and the ground truth bounding box based on the coordinates *(xcenter, ycenter)*.

$l_{ij}^{obj}$is equal to 1 if the object appears inside the predictor bounding box j^th^ in the j^th^ cell, and 0 otherwise. The bounding prediction box will be responsible for predicting an object based on the current highest IoU prediction.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2. \tag{4}$$

The second part of the loss function, YOLO calculates the error in predicting the width and height. However, the magnitude of the error in the large boxes affects the equation in the small boxes. Since both width and height are normalized between 0 and 1, their square root increases more than the difference for small and large values. From here on, the square root of the bounding box's width and height is used instead of the direct width and height.

$$\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}(C_i-\hat{C}_i)^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{noobj}(C_i-\hat{C}_i)^2 \tag{5}$$

The loss of confidence score is calculated in both cases whether the subject is present in the bounding box or not. The loss function only evaluates the reliability of the object if that predictor is responsible for the ground truth box. $l_{ij}^{obj}$is equal to 1 when there is an object in the cell and 0 otherwise. $l_{ij}^{noobj}$is the opposite.

$$\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{obj}[(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+(\sqrt{h_i}-\sqrt{\hat{h}_i})^2] \tag{6}$$

The last part of the loss function is similar to the normal loss function. This term is used because YOLO evaluates the classification error even if no objects appear in cell [7].

2.3. Data preparation and processing

The dataset used during model training and test runs is extracted from car dash cams and fixed cameras at intersections. With the model training dataset, the videos are converted into 1280x720 px images. Includes 500 images, of which 450 images are used for training and 50 images for testing the accuracy of the newly trained model.

The author uses the retraining method from the training model yolov5s.pt and yolov7-tiny.pt

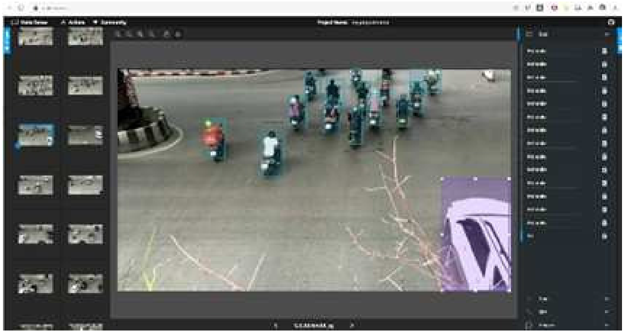


Figure 8: Create a database using makesense.ai.

The model training image dataset is preprocessed with makesense.ai and attached parameters include: [class number] [x coordinate of object center] [y-coordinate of object center] icon] [width] [height]. The data is stored as a .txt file for each image. In which ID for the classes in turn: 0-Car, 1-Motorbike, 2-Truck, 3-Bus.

2.4. Model training

Model training is a program that works continuously, consuming a lot of resources such as RAM, GPU, CPU, so to ensure fast and accurate training based on available hardware platforms, we have using Google's virtual server called Google Colab [8]. Colab offers 3 types of configurations:

Table 1: Technical Specifications of Google Colab

CPU	GPU	TPU
- Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)	- Up to Tesla K80 12 GB of GDDR5 VRAM, Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)	- Cloud TPU with 180 teraflops of computation, Intel Xeon Processor with two core @ 2.30 Ghz and 13GB RAM - OS: Ubuntu 18.04.2 LTS - Total size of Disk: 78.0 GB (48.0 GB Used)

We choose the GPU configuration to train the model as well as run the test of the mobile vehicle detection and recognition program using the YOLO model.

The YOLO project is hosted on Github and to use it we downloaded the project, then ran the train.py file so that we could start training the removable vehicle recognition model.

2.5. Installation and testing

2.5.1. Test environment

The test platform is Ubuntu 18.04.2 LTS operating system along with NVIDIA Jetson TX2 hardware with the following configuration:

Table 2: Technical Specifications of NVIDIA Jetson TX2

GPU:	256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores
CPU:	Dual-Core NVIDIA Denver 2 64-Bit CPU & Quad-Core ARM® Cortex®-A57 MPCore
Memory :	8GB 128-bit LPDDR4
Storage:	32GB eMMC 5.1

With GPU graphics hardware that supports CUDA cores, NVIDIA Jetson TX2 delivers powerful and specialized performance in handling AI & ML related tasks.

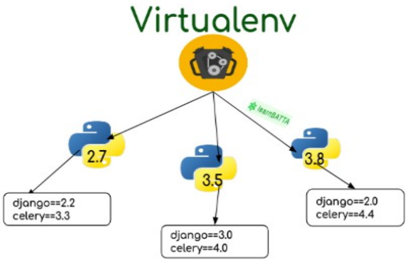


Figure 9: Python Virtual Environment.

First, we set up a virtual environment using Python Virtual Environment [9]. Virtual environments are used to isolate the environments of projects from each other. The virtual environment allows installation and management of installation packages separately and does not conflict with the system-wide installation package manager.

Library packages to install: Python 2.6.9, Pytorch 1.8.0, Torchvision 0.9.0, OpenCv 4.5.4.60, Matplotlib, Pillow, Pyyaml, Tensorboard, Tqdm, Scipy, Pandas, Seaborn, Numpy

2.5.2. Model of a mobile vehicle detection and identification system

The steps of the proposed method are as follows: First the video input data is split into frames and converted to a resolution of 1280x720 px, which is the optimal resolution for speed as well as enough quality to determine object definition. The extracted data will be compared with the pre-trained model and fed into YOLO's object recognition algorithm. Output data includes object coordinates, object ID will be zoned and labeled accordingly. The output of the system is the video displayed in real time along with which will be stored as .mp4.

We run tests on two versions of YOLO v5 and YOLO v7 to compare, evaluate and select the best version for recognizing removable media objects.

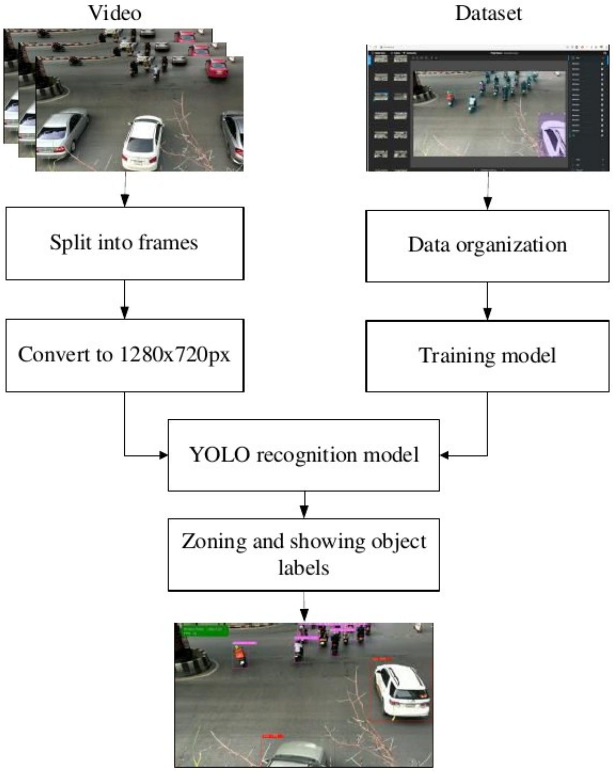


Figure 10: Model of the detection and identification system.

3. Research results and discussion

3.1. Object Recognition from Fixed Camera

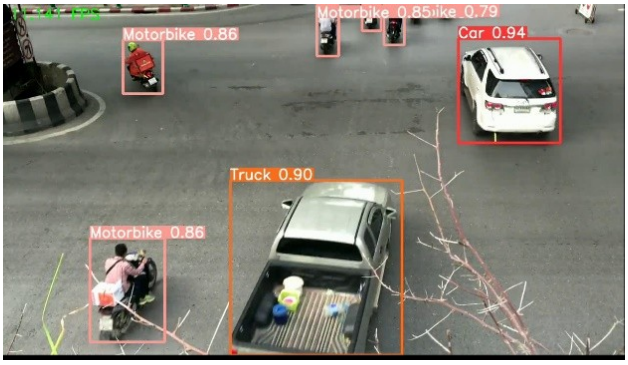


Figure 11: Identification results using fixed camera YOLO v5 model.

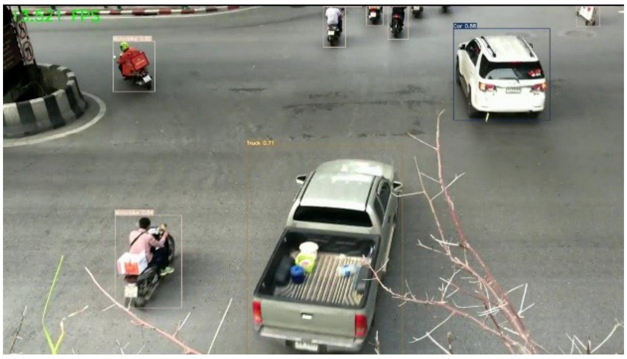


Figure 12: Identification results using fixed camera YOLO v7 model.

Experimental results show that, in the environment of the intersection under the bridge, many vehicles passing through both model versions fully recognize the objects. With the YOLO v5 model, the accuracy reaches from 79%, the average frame rate is about 11 FPS at 1280x720 video resolution. With the YOLO v7 model, the accuracy is from 71% but has a higher average frame rate of about 13FPS. The system shows stability and there is no overlapping of the identified frame or subject.

3.2 Object recognition from Dashcam mounted on cars

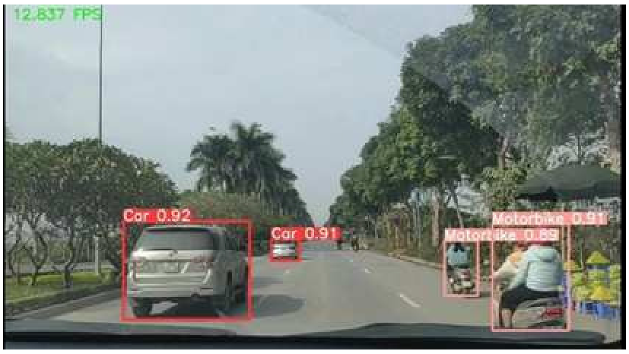


Figure 13: Identification results using Dashcam YOLO v5 model.

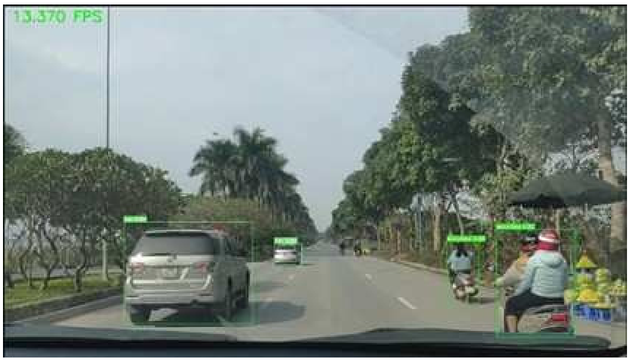


Figure 14: Identification results using Dashcam YOLO v7 model.

Stay in a moving vehicle in a more complex environment with many noisy objects such as trees, buildings, roadside foreign objects. The system gives relatively good results. With the YOLO v5 model, the accuracy is 89% or higher, the average frame rate is about 12 FPS at 1280x720 px resolution. With the YOLO v7 model, the accuracy is 82% or higher, the average frame rate is more than 13 FPS at 1280x720 px resolution. The system shows stability and there is no overlapping of the identified frame or subject.

The recognition system achieves the display accuracy rate of > 51%. Partly due to camera quality, the rest is due to small and blurry objects so there are still unrecognizable vehicles or low % accuracy. During actual driving, objects located on the opposite side of the road and at a distance do not usually cause an accident to the driver so the current results are acceptable.

3.3. Comparison and evaluation

Conduct a detailed test and compare 2 identification models YOLO v5 and YOLO v7 with a training model built specifically for Vietnam's traffic system including: Cars, motorbikes, trucks, buses.

Table 3: Compare recognition models.

Data	Total number of frames	Resolution	Model	Detection Rate (%)	Miss rate (%)	False detection rate (%)	FPS
Fixed camera	513	1280x720	YOLOv5	87.5	12.5	0	11
		1280x720	YOLOv7	87.5	12.5	0	13
Dashcam	1575	1280x720	YOLOv5	66.7	33.3	0	12
		1280x720	YOLOv7	66.7	33.3	0	13

The comparison table shows that, with the recognition model YOLO v5 and YOLO v7, both have a relatively high rate of detecting and correctly identifying objects.

The YOLO v7 model shows an improvement in processing speed compared to the old model YOLO v5 both in terms of fixed and mobile cameras.

4. Conclusion

Preliminary results obtained when using YOLO model in training and object recognition yield relatively good results, showing great potential in building a general intelligent traffic model and assistance system. ADAS advanced driving in particular. Moreover, the successful test on NVIDIA Jetson TX2 device, this opens a new approach in real-time recognition of different objects executed directly devices on camera devices that are available on the mobile system. Currently, the model training data is limited, increasing the number of images will bring better accuracy. In addition, along with the development of hardware technology, the model will be able to improve in processing speed.

Acknowledgment

This work was supported by Ministry of Education and Training, Vietnam, under Grant MOET B2020-SKH-02.

References

[1] A. F. Agarap, Deep Learning using Rectified Linear Units (ReLU), arXiv:1803.08375, 2018.

[2] G. S. W. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, ISBN 978-0-8053-4780-7, 26 July 2020.

[3] M. Galvani, History and future of driver assistance," IEEE Instrumentation Measurement Magazine, ISSN 1941-0123, 2019.

[4] Ultralytics, "YOLOv5 Documentation," [Trực tuyến]. Available: https://docs.ultralytics.com/.

[5] S. D. R. G. A. F. Joseph Redmon, You Only Look Once: Unified, Real-Time Object Detection, arXiv:1506.02640 [cs.CV], 8 Jun 2015.

[6] M. Schumann, A Book about Colab: (and related activities), ISBN 978-0-89439-085-2, 2015.

[7] GeeksforGeeks, "Python Virtual Environment | Introduction," 2020. [Trực tuyến]. Available: https://www.geeksforgeeks.

[8] C. H. Thuc, "Precision, Recall và F1-score là gì?," 23 02 2020. [Trực tuyến]. Available: https://caihuuthuc.wordpress.com/2020/02/23/precision-recall-va-f1-score-la-gi/.

[9] D. Thuan, Evolution of YOLO Algorithm and YOLOv5: The State-of-the-art Object Detection, Bachelor thesis (3.092Mt), Spring 2021.

[10] H.-S. Vu, J.-X. Guo, K.-H. Chen, S.-J. Hsieh và D.-S. Chen, A real-time moving objects detection and classification approach for static cameras, IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2016.

[11] V. T. D. T. D. N. Hong Son Vu, MỘt Phương Pháp Phát Hiện Điểm Mù Với Độ Tin Cậy Cao Và Thời Gian Thực Cho Các Hệ Thống Hỗ Trợ Lái Xe Thông Minh, Moet B2020-SKH-02, October 2020.

[12] V. H. Son, A high dynamic range imaging algorithm: implementation and evaluation, Engineering and Technology - Research article, Aug 7, 2019.

DOI: 10.15439/2022R44
ISSN 2300-5963 ACSIS, Vol. 33

A New Approach of Question Answering based on Knowledge Graph in Traditional Medicine

Pham Van Duong[†], Tien-Dat Trinh[†], Hai Van Pham[†], Tran Manh Tuan[‡], Le Hoang Son[§]
Huy-The Vu[¶], Minh-Tien Nguyen[¶], Pham Minh Chuan[¶]*
[†]School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, VietNam
[‡]Faculty of Computer Science and Enginering, Thuyloi University, Hanoi, VietNam
[§]VNU Information Technology Institute, Vietnam National University, Hanoi, Vietnam
[¶]Faculty of Information Technology, Hung Yen University of Technology and Education, Hung Yen, Vietnam
{duong.pv4w, tdtrinh11}@gmail.com; haipv@soict.hust.edu.vn; tmtuan@tlu.edu.vn;
sonlh@vnu.edu.vn; {thevh, tiennm, chuanpm}@utehy.edu.vn

***Abstract*—In recent years, it has been great interest for Question Answering (QA) systems applied to many areas placing a high value on the community. The study and development of such QA systems through chatbot tools in medicine raise great needs for clinicians in their daily activities. Chatbots use the knowledge that could be retrieved from a database, but with limited inference capability. In this paper, we propose a new QA system based on Knowledge Graph (knowledge graph) for Traditional Medicine. Data of the knowledge graph is obtained from two sources including those from diagnostic of treatment diagrams and those collected on well-known medical websites through the Internet. The knowledge graph is then formed by combining the entities and relationships using the Named Entity Recognition (NER) model. Diagnosis is made via the node similarity algorithm in the knowledge graph for symptom identification. The effectiveness of the system is demonstrated through theoretical analysis and real-world experimental outcomes.**

***Index Terms*—Knowledge Graph; Traditional Medicine; Node Similarity.**

I. Introduction

Progress in the field of Question Answering (QA) is propelling humanity to innovative technical heights, particularly in the medical field, to benefit health workers in solving medical issues. QA systems can be defined as the task whereby an automated machine (such as a computer) answers arbitrary questions formulated in natural language. Realizing the above benefits, in response to medical concerns, healthcare practitioners can use QA systems to retrieve brief phrases or paragraphs. The advantage of such systems is that they may generate responses and provide clues in seconds. These systems have partly solved difficulties such as consulting and answering patients' concerns about health and medicine. This helps reduce the pressure on the health system quite effectively, especially during the recent Covid-19 epidemic.

Due to the variety of medical textual data sources, the QA structures are just as diverse. Many researchers have done some work for it. For instance, in building a knowledge graph, [1], [2] introduces the problem of entity name recognition to extract entities and relationships in the knowledge graph automatically. [3] proposed to use deep learning focus on CBOW and BiLSTM + CRF for address the above problem. Unlike other studies, we study the construction of a medical knowledge graph in Vietnamese from many data sources by applying deep learning models to the entity extraction problem as well as building information extraction rules with data sources from the diagnostic of disease diagrams. In addition, we apply the latest technology to build our system and propose algorithms for retrieving answers from the medical knowledge graph. To summarize, our contributions are as follows:

- We construct a medical knowledge graph in Vietnamese from two data sources: data from diagnostic of disease diagrams and data collected on medical websites.
- We propose to apply a deep learning model to address the name entity recognition for data collected on medical websites. Therefore, it will reduce the construction time of the knowledge graph as well as increase the efficiency of extracting information from many different sources.
- We build the QA system by using the Rasa framework, which is one of the best frameworks for building and implementing QA systems. Besides, we also improves the search algorithm in the disease diagnosis by combining the node similarity algorithm and cypher query in neo4j to increase the accuracy.

II. Proposed Method

A. Knowledge Graph Construction

1) Data Acquisition: Data acquisition is the first step to establishing a knowledge graph. In this study, the data source of the knowledge graph is divided into two types: data from diagnostic of disease diagrams and data collected on medical websites such as vinmec.com[1], nhathuoclongchau[2], etc. As for the data source from the diagnostic of disease diagram, which is a highly reputable data source, this study will apply manual conversion of the data source from images to text.

*Corresponding Author.

[1] https://www.vinmec.com/vi/benh/
[2] https://nhathuoclongchau.com/benh

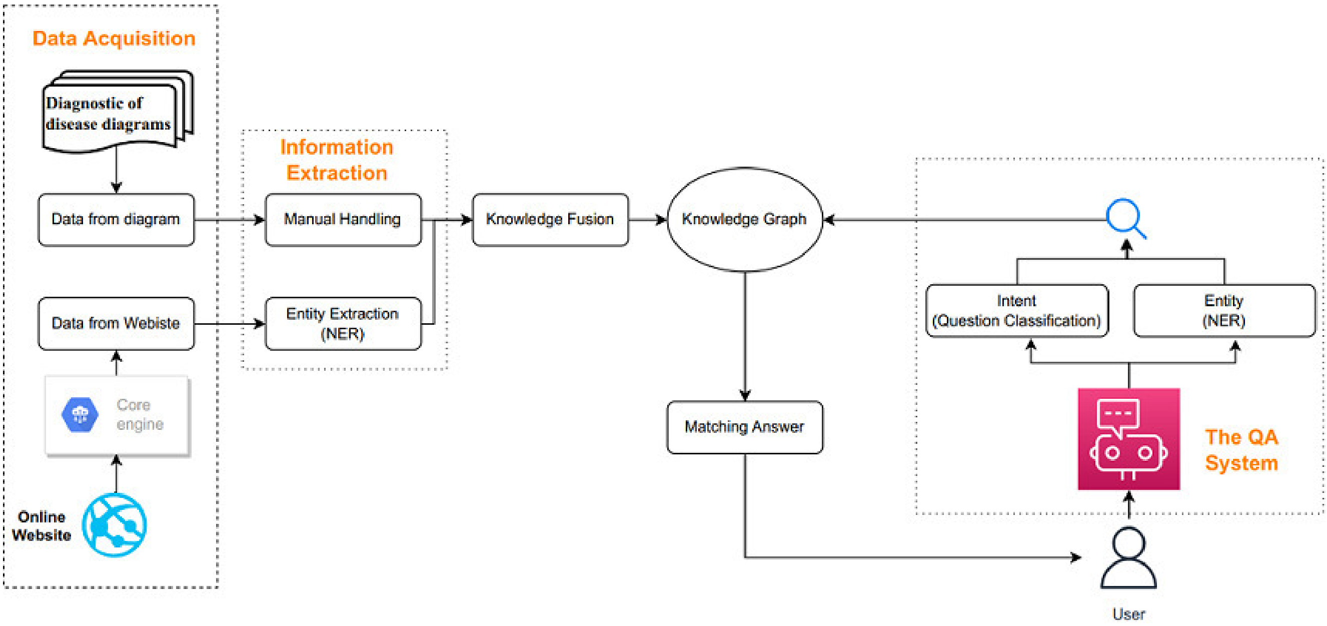


Fig. 1. Architecture of QA system based on Knowledge Graph

2) Information Extraction: Information extraction is the second step in establishing a knowledge graph. We extract structured information including Entity Extraction about symptoms and disease names while Relation Extraction and Attribute Extraction have been processed in the first step with the core engine.

In this work, we manually built a dataset based on collected data on medical websites with 2 labels (disease, symptom) containing 2,104 sentences and 26,106 labels. After collecting information about symptoms of diseases on medical websites, we preprocess the data such as removing HTML code in data, normalizing Vietnamese Unicode, removing unnecessary characters, etc. To ensure the accuracy and assurance of the construction data source, we calculate the percentage of correctly labeled sentences and correctly labeled sentences per 400 sentences by 3 independent annotators. The corresponding result is 90.5% and 96%. In addition, we calculate the Cohen's Kappa[4] measure and the result is 0.81. That proves, the dataset has quite high confidence and can serve the models well

3) Knowledge Fusion: The third step in constructing a knowledge graph is knowledge fusion. Due to the complexity of information extraction sources, the absence of hierarchy and logicality in knowledge relationships, redundancy of non-homologous knowledge, unequal knowledge expression, and other issues, this procedure requires a number of processes such as Entity Disambiguation and Entity Linking. The core application objects in this system are the atlas triple's entities, attributes, and relationships. We begin by defining the entities and their relationships, then use the similarity matching algorithm to find similar entities collectively, and then generate any attributes that exist in that entity. The entities and attributes are combined and fed into the knowledge graph through neo4j [5] database (this is a graph database used by many researchers) to store data about nodes and relationships. Nodes have 4 labels including typical_symptom, symptom, disease, and diagram corresponding to which relationships include HAS_TYPICAL_SYMPTOM, HAS_TYPICAL_SYMPTOM, HAS_SYMPTOM, YES, NO, HAS, HAS_NO. For the disease node, the knowledge graph provides additional attributes including an overview, diagnostic_measures, disease_cause, preventive_measures, treatment_measures. The constructed medical data knowledge graph includes 13,311 nodes and 16,638 relationships. The detailed description of the graph is shown in Table I and Table II. As we can see in Table I and II, the number of symptom nodes is the largest with 10,520 node and the "HAS_SYMPTOM" relationship is the largest with 15,247 relationships. Because the knowledge graph is mainly built to aid in the diagnosis of disease, the number of symptom nodes and relationships should be the majority.

TABLE I
ENTITY TYPES IN THE MEDICAL KNOWLEDGE GRAPH

Entity Type	Number of Entities
typical_symptom	62
symptom	10,520
disease	2,666
diagram	63
Total	13,311

B. QA System based on Knowledge Graph

The content of this section will present the technologies and algorithms that we use to build a QA system from the knowledge graph built from the above section.

TABLE II
RELATIONSHIP TYPES IN THE MEDICAL KNOWLEDGE GRAPH

Relationship Type	Number of Relationships
HAS_TYPICAL_SYMPTOM	63
HAS_FIRST_SYMPTOM	124
HAS_SYMPTOM	15,247
YES	122
NO	422
HAS	521
HAS_NO	139
Total	16,638

1) Rasa Framework: In this paper, we use the Rasa Framework in building Q&A scenarios by initializing user intents in domain.yml file and setting bot responses for each intent type in actions file. actions.py Rasa[6] is an open source machine learning framework for automated text and voice-based conversations. In addition, we use rasa-x to deploy the system. It is a tool for Conversation-Driven Development (CDD), the process of listening to your users and using those insights to improve your AI assistant.

2) Question Classification: For the initial requirements of building a chatbot and based on the actual types of questions that users can ask with a medical chatbot system, the article divides the types of questions from users into 6 types of questions include: asking about disease diagnosis, getting an overview of the disease, asking about the causes of the disease, asking about the treatment measures, asking about the disease diagnosis and prevention measures. Taking advantage of the rasa framework's support for user intent determination, the article customizes some changes in rasa's nlu[7] intent detector with Vietnamese characteristics. We initialize possible user intents, then for each intent we provide 10 to 20 sample questions. Because the question and answer system is specific to Vietnamese, so we customize in the config file of rasa the nlp model in the pipeline to ConveRTTokenizer and ConveRTFeaturize, this is the language model that supports Vietnamese

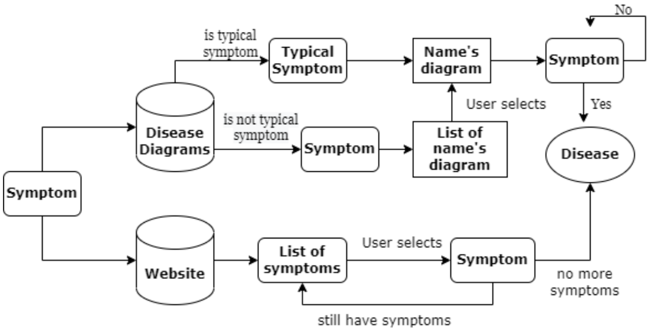


Fig. 2. Diagram about Query Processing for finding the best answer for the question of disease diagnosis

3) Query Processing for finding the best answer: Fig. 2 describes diagram about Query Processing for finding the best answer for the question of disease diagnosis. Our work uses two sources of data, because data from diagnostic of disease diagrams has a higher reputation so that when the patient provides symptoms, it will be prioritized to search symptoms in the diagram. The system will first check for that symptom in the typical symptoms of the disease diagrams, if it exists, it will use that disease diagram. Ask the user for the symptoms in turn until the name of the diagnosis is found. If that symptom is not included in any typical symptom, the system will check all disease diagrams with that symptom and give that list to the user. Then, the system will be performed as above when the disease diagram is determined. When the symptoms do not belong to the diagram in the knowledge graph, the system will switch to searching for symptoms collected from the website. In a real situation, when the patient provides symptoms, the doctor will ask and give other symptoms to determine the patient's likelihood of being sick. Based on that experiment, the study proposes to combine with the node similarity algorithm [8] to find the symptoms that frequently occur simultaneously with the symptoms provided by the patient.

4) Matching Answer: When determining the question type from the user and the entities and attributes searched from the above step, we use a mechanism to match the answers corresponding to the question to follow the originally defined template and return it to the user.

III. EXPERIMENTS AND RESULTS

In this section, we will evaluate the performance of deep learning models for the NER problem and the medical knowledge graph in Vietnamese as well as demo the QA system based on the knowledge graph.

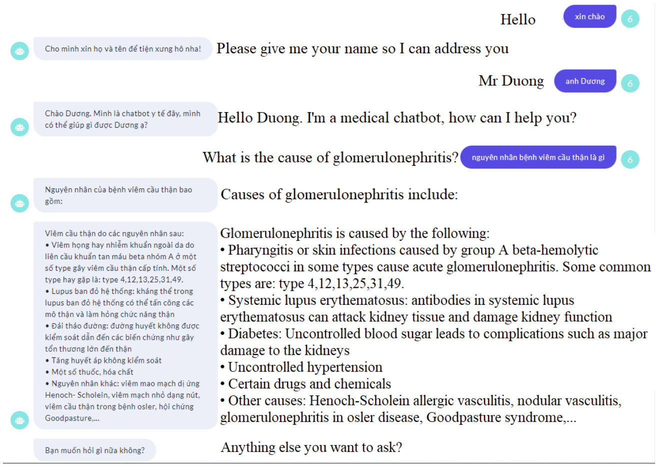
1) Evaluation of NER Model: In order to find the best performance of the deep learning model with the NER problem, we survey the most used and most effective models in recent times to put into practice the construction of knowledge graphs. The five chosen models include ACE (Automated Concatenation of Embeddings) [9], BERT [10], PhoBERT model [11], XLM-R-large [12].

Table III describes the F1 score of five NER models. For the ACE, XLM modelsachieving the highest efficiency on the 2 data sets ShARe/CLEF, NCBI-disease. However, on the dataset of the proposed paper, the model's performance is worse than the Fine-tuning PhoBERT model. Because the dataset is in Vietnamese, being different from the above datasets in English and the Fine-tuning PhoBERT model achieves state-of-the-art for Vietnamese. From the above results, we decide to use the Fine-tuning PhoBERT model for the NER problem in building the knowledge graph and the QA system.

2) Evaluation of Knowledge Graph Experiment: To measure the trust of a knowledge graph from the data source on the website, we calculate its confidence through the link prediction in the graph. The main idea of knowledge graph evaluation is by hiding some of the relationships between entities in the graph, and then use association prediction problem-solving

TABLE III
F1 SCORE OF NER MODELS

Model	F1 Score
ACE - Embedding: XLM-R	65.7%
ACE - Embedding: Glove	70.3%
Fine-tuning PhoBERT	76.3%
Fine-tuning BERT	66.4%
Fine-tuning XLM-R-large	66.7%



Hello

xin chào

Cho mình xin họ và tên để tiện xưng hô nha!

Please give me your name so I can address you

Mr Duong

anh Dương

Chào Dương. Mình là chatbot y tế đây, mình có thể giúp gì được Dương ạ?

Hello Duong. I'm a medical chatbot, how can I help you?

What is the cause of glomerulonephritis?

nguyên nhân bệnh viêm cầu thận là gì

Nguyên nhân của bệnh viêm cầu thận bao gồm:

Causes of glomerulonephritis include:

Viêm cầu thận do các nguyên nhân sau:
• Viêm họng hay nhiễm khuẩn ngoài da do liên cầu khuẩn tan máu beta nhóm A ở một số type gây viêm cầu thận cấp tính. Một số type hay gặp là: type 4,12,13,25,31,49.
• Lupus ban đỏ hệ thống: kháng thể trong lupus ban đỏ hệ thống có thể tấn công các mô thận và làm hỏng chức năng thận
• Đái tháo đường: đường huyết không được kiểm soát dẫn đến các biến chứng như gây tổn thương lớn đến thận
• Tăng huyết áp không kiểm soát
• Một số thuốc, hóa chất
• Nguyên nhân khác: viêm mao mạch dị ứng Henoch- Scholein, viêm mạch nhỏ dạng nút, viêm cầu thận trong bệnh osler, hội chứng Goodpasture,...

Glomerulonephritis is caused by the following:
• Pharyngitis or skin infections caused by group A beta-hemolytic streptococci in some types cause acute glomerulonephritis. Some common types are: type 4,12,13,25,31,49.
• Systemic lupus erythematosus: antibodies in systemic lupus erythematosus can attack kidney tissue and damage kidney function
• Diabetes: Uncontrolled blood sugar leads to complications such as major damage to the kidneys
• Uncontrolled hypertension
• Certain drugs and chemicals
• Other causes: Henoch-Scholein allergic vasculitis, nodular vasculitis, glomerulonephritis in osler disease, Goodpasture syndrome,...

Bạn muốn hỏi gì nữa không?

Anything else you want to ask?

Fig. 3. Sample dialogue about disease investigation

models to determine if the relationship exists or not. Table IV describes the results of the models. The results demonstrated that the knowledge graph is excellent, with nodes in close proximity to one another.

TABLE IV
AUC_ROC SCORE OF MACHINE LEARNING MODEL FOR LINK PREDICTION IN KNOWLEDGE GRAPH

Model	AUC_ROC score
LR Model	74.46%
Random Forest Model	92.89%
LightGBM Model	98.26%

3) Demo System: In this paper, we have built a medical knowledge graph in Vietnamese from two data sources including data from diagnostic disease diagrams and data collected on medical websites. The data on the medical website is automatically collected as well as the extraction of information from this data is done by a deep learning model with the problem of name entity recognition to put into the knowledge graph. After building a knowledge graph, we built the QA system (Fig. 3) based on the rasa framework and applied it for finding the answer to each question. The Rasa framework supports the intent recognition of input questions, so the paper also uses this utility for identifying the type of questions that the user provides.

Fig 3 shows a short description of a sample dialogue about disease investigation.

IV. CONCLUSION AND FUTURE WORK

In this paper, we present the QA system based on the knowledge graph for the medical domain in Vietnamese. In this method, we create a knowledge graph that comprises medical entities, attributes of the entities, and relationships between the entities from two data sources (diagnostic of disease diagrams and an online website). The notion and entities of a given query are then recognized. An inference method is used to process the idea and entities in the inquiry. Finally, we apply the node similarity algorithm for finding symptoms that can best diagnose the disease. The outcomes of real-world experiments show that our method is effective. In the future, we will expand the dataset on more sources and consider increasing the knowledge graph's construction and retrieval speed. In addition, we will use multi-loop dialogue and complex intellectual reasoning to find the most suitable answer. This is a special feature of the system that is different from similar Q&A systems.

ACKNOWLEDGMENT

This research is funded by the Ministry of Education and Training under grant number B2022-SKH-01.

REFERENCES

[1] Z. Jiang, C. Chi, and Y. Zhan, "Research on medical question answering system based on knowledge graph," *IEEE Access*, vol. 9, pp. 21 094–21 101, 2021.

[2] D. N. Tien and H. P. Van, "Graph neural network combined knowledge graph for recommendation system," in *International Conference on Computational Data and Social Networks*. Springer, 2020, pp. 59–70.

[3] Y. Xie, "A tcm question and answer system based on medical records knowledge graph," in *2020 International Conference on Computing and Data Science (CDS)*. IEEE, 2020, pp. 373–376.

[4] M. Wirtz and M. Kutschmann, "Analyse der beurteilerübereinstimmung für kategoriale daten mittels cohens kappa und alternativer maße," *Die Rehabilitation*, vol. 46, no. 06, pp. 370–377, 2007.

[5] J. J. Miller, "Graph database applications and concepts with neo4j," in *Proceedings of the southern association for information systems conference, Atlanta, GA, USA*, vol. 2324, no. 36, 2013.

[6] R. K. Sharma and M. Joshi, "An analytical study and review of open source chatbot framework, rasa," *International Journal of Engineering Research and*, vol. 9, no. 06, 2020.

[7] A. Jiao, "An intelligent chatbot system based on entity extraction using rasa nlu and neural network," in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012014.

[8] W. Lu, J. C. M. Janssen, E. E. Milios, N. Japkowicz, and Y. Zhang, "Node similarity in the citation graph," *Knowledge and Information Systems*, vol. 11, pp. 105–129, 2006.

[9] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu, "Automated concatenation of embeddings for structured prediction," *arXiv preprint arXiv:2010.05006*, 2020.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[11] D. Q. Nguyen and A. T. Nguyen, "Phobert: Pre-trained language models for vietnamese," *arXiv preprint arXiv:2003.00744*, 2020.

[12] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," *arXiv preprint arXiv:1911.02116*, 2019.

DOI: 10.15439/2022R48
ISSN 2300-5963 ACSIS, Vol. 33

Designing and Manufacturing the AI Robot for Water Resources Management

Vu Ngoc Thuong
Faculty of Mechanical Engineering
Thuy Loi University
Ha Noi, Vietnam
vungocthuong77@gmail.com

Le Khac Binh
Science Management and International Collaboration Department
Vinh University of Technology Education
Vinh city, Vietnam
khacbinhvute@yahoo.com

***Abstract*—Information technology is nowadays developing strongly, especially the artificial intelligence field. These technologies are being applied to the manufacture of robots, the interdisciplinary combination has created more and more perfect, smarter robots which are applied in many different fields. Stemming from the actual demand in the irrigation industry in the field of water resources, we have researched, fabricated, and programmed artificial intelligence robots to manage water resources. Robots are designed with an interdisciplinary combination of the fields of Mechanics, Electronics and Information Technology. The robot has the main functions of being able to interact with verbal communication, and controlling peripheral devices of water pump devices. The successful robots manufacturing meets practical needs and can be applied in practice to help manage water resources scientifically and effectively.**

***Index Terms*—AI robots, intelligence, mechatronics, IT, water resources.**

I. Introduction

The current situation of water resource management in Vietnam is still limited, leading to the loss and risk of water depletion. There are many different tasks of water resource management, such as propagating policies and regulations to help people raise their awareness, managing infrastructure to regulate water sources, but these jobs are currently implemented quite simply without much innovation.

In order to solve the above problem for better water resources management, we have researched and applied advanced technologies to design and manufacture intelligent robots with functions that meet the requirements set out which are capable of communication, presentation, programmable to automatically open and close water pumping systems using IoT technology.

Water environment monitoring is attracting research attention. In [1], a system which is combined IoT and aquaculture. The monitoring system displays measured sensor values and optimizes energy consumption in sensor networks.

In [2], the author focuses on researching a monitoring system based on Raspberry Pi and Arduino to monitor the water environment. The system includes of wireless sensor nodes (WSNs) which is using the ZigBee protocol. These nodes send data to a webserver. The information is then accessed through handheld devices such as mobile phones or computers.

In this paper, a monitoring system to monitor and warning the pollution level of water will be presented. It is include the collecting sensor nodes and a central node which describing the model of the monitoring system, including: measuring station acts as sensor nodes to collect data, then transfer data to the central node through Lora wireless network. The parameters such as conductivity (EC), temperature, turbidity level, dissolved solids concentration (TDS), pH and DO levels could be measured by wireless sensor network nodes (measuring station). These data is transmitted through the wife internet to the robot for processing.

II. Research Objectives, Subjects and Methods

The research objective is to design, manufacture and program an artificial intelligence robot with both communication and information storage capabilities to automatically control water supply and drainage to serve the management of water resources.

The research subject is a robot with simple communication functions, presentation and information exchange. In order to achieve the stated objectives, the author has carried out research, design and manufacture at ThuyLoi University, where there are sufficient conditions for research and manufacturing.

Research Methods: combination of theoretical and experimental research, using expert method to get opinions of experts, using synthesis and analysis methods, verifying the research results by experiment on the product after manufacturing.

III. Research Results, Design and Manufacture Robots

Based on the goals set out, we have researched and used documents to look up technical specifications, calculate data to meet the technical requirements of the robot by using Solidworks software to design robots in sequence of work steps, research of the function of each part, detail assembly, then building system diagram, overall design robot, design each part of robots, assembly for simulation, programming, fabrication, testing.

A. Basic specifications of the robot

Dimension: height x width (1350x380) mm
Voltage: 12V DC
Weight: 30 kg
Possibility to charge 24 hours (continuously)
Capacity: 40W

B. Design and manufacture the mechanical part of the robot

Bases on requirements and goals set out, the author makes a research and design the details, parts, and mechanical structure of the robot. The robot is composed of 3 main parts:

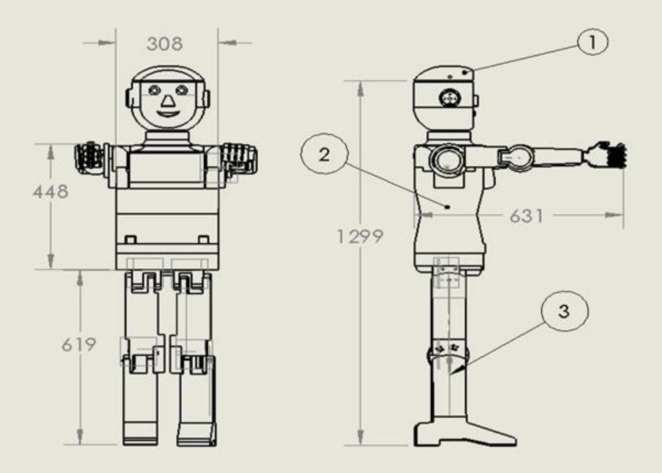


Fig. 1. Design of AI Robot for Water Resource Management

The face and head of robot where the robot's mouth can move.

The robot body which contains microchips and control circuits.

The robot legs are fitted with motors that help the robot move.

The solidworks software is used to design robots. The details of the robot after the design were made by 3D printing method, then fully assembled.

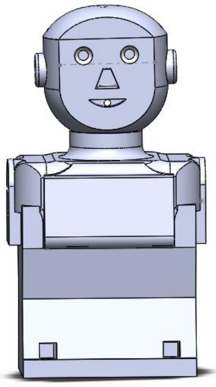


Fig. 2. Design head and body of the robot

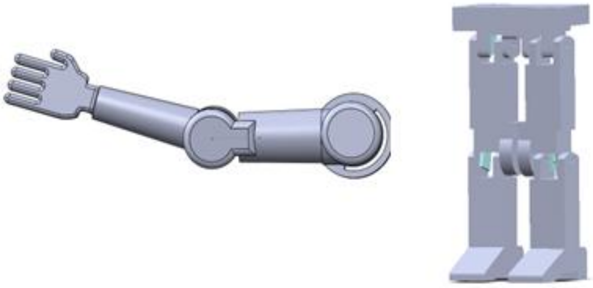


Fig. 3. Design arm and leg of the robot

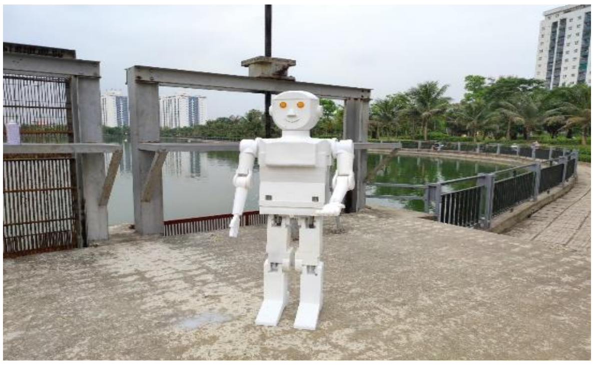


Fig. 4. Robot after fabrication and installation

C. Design of Control circuit design [3]

The robot uses Rasphberry Pi3 to control the robot's operation as well as store data, program the robot voice communication.

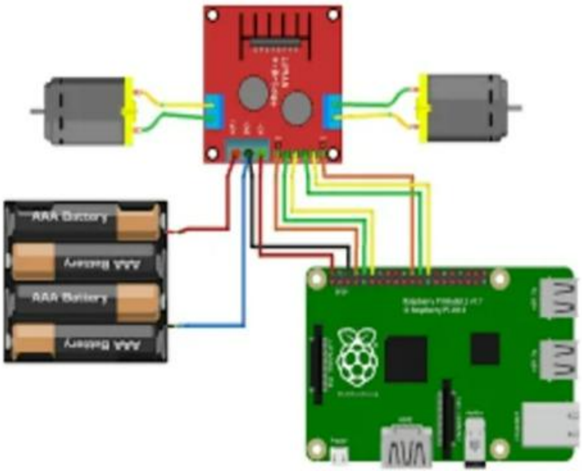


Fig. 5. Raspberry module for motor control

D. Programming communication control, voice presentation

The program to process data from the related peripherals is written in Python language and is executed on the Rasberry Pi embedded board. After being programmed, the robot can listen and receive audio signals and then convert to text files, the built-in program will find data and answer in text file format and text data will be converted to voice spech. Therefore, humans can communicate with robots and robots could accurately answer questions with pre-programmed data.

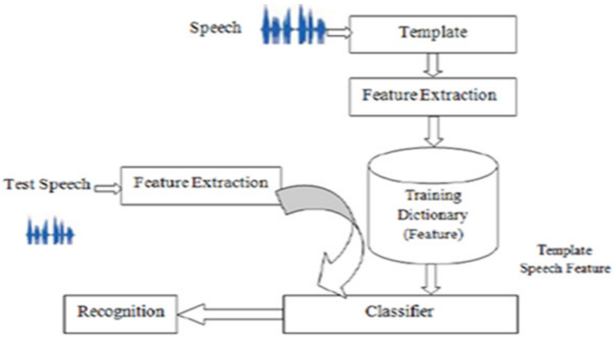
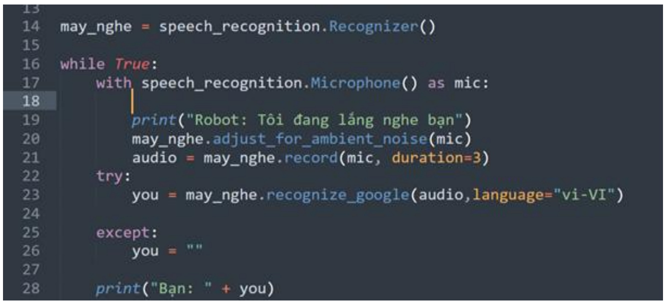


Fig. 6. Algorithm diagram of the program

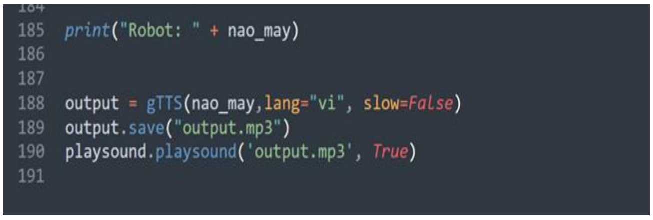
Data collection (listening)



```
may_nghe = speech_recognition.Recognizer()

while True:
    with speech_recognition.Microphone() as mic:

        print("Robot: Tôi đang lắng nghe bạn")
        may_nghe.adjust_for_ambient_noise(mic)
        audio = may_nghe.record(mic, duration=3)
    try:
        you = may_nghe.recognize_google(audio,language="vi-VI")

    except:
        you = ""

    print("Bạn: " + you)
```

Fig. 7. Code helps the Robot hear and convert data into writing

Send a reply (speech)



```
print("Robot: " + nao_may)


output = gTTS(nao_may,lang="vi", slow=False)
output.save("output.mp3")
playsound.playsound('output.mp3', True)

```

Fig. 8. Code converts the answer from text to audio

When someone appears in front of the robot at a distance of 3m, the robot will automatically recognize through the sensor, then the control circuit will control so that the robot emits a greeting voice, at the same time the lights in the robot's eyes will turn on, the robot hand will wave hello. Users can talk to the robot to find out information about water resources, the robot will answer based on stored data or expand the search for information related to water resources on wikipedia. All conversations are communicated in Vietnamese and English. The robot can connect to the internet to update information and communicate with management agencies related to water resources. The robot can also present itself with stored information.

The effectiveness of communication in Human-Robot Interaction will be greatly affected by noise in the surrounding environment. If a loud noise occurs, the robot cannot distinguish, recognize the voice, the volume of the subject that is speaking and it cannot listen and answer questions correctly.

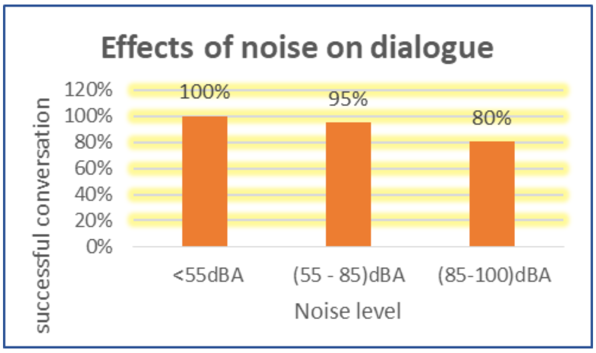


Fig. 9. Effects of noise on dialogue

After completion, the robot is operated and tested for question recognition and answer ability. Tests show that the robot will give the wrong questions when it gets incorrect identification. The robot will respond correctly to the scenario in the training set if it gets correct speech recognition. The author conducted the test by communicating with the robot through 200 questions in laboratory conditions. To evaluate the effectiveness of the robot's level of listening, receiving and processing the sounds of speech, collaborative authors have tested for robot communication at different noise levels with 1.5 m distance from the speaker to the robot position. The result is shown in the chart in Fig. 9.

E. Control the water pump system through wifi internet

To be able to regulate the irrigation system, the Robot is connected to peripheral devices, the relay opens and closes the water pump through a wifi device connected to the internet. The robot can automatically open and close the water pump motors in different places according to the pre-programmed program.

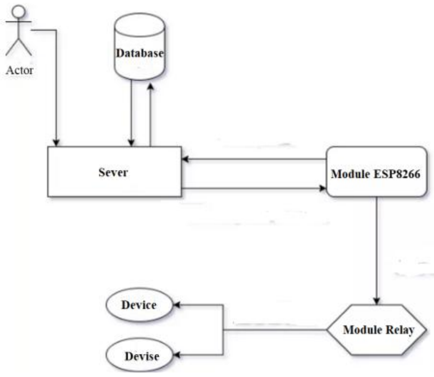


Fig. 10. Diagram of the IoT (Internet of things) control system

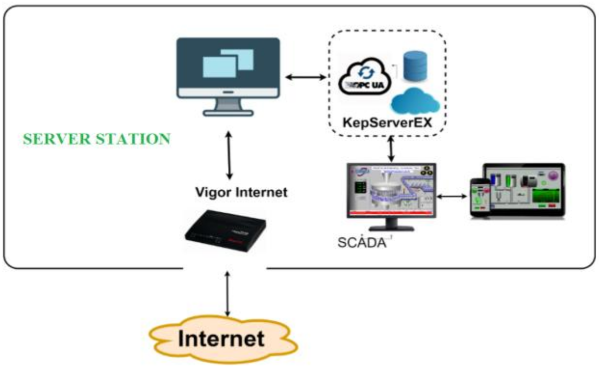


Fig. 11. Connection model at server

Connection configuration at unit station

At the units, each PLC or control device will be connected to the router to connect to the server. Fig. 12 shows the connection model at the units. In fact, the system can connect multiple unit stations. In the content of the research, the authors design a model consisting of 2 unit stations. The model will have 2 motors (pump in and pump out): The inlet pump will pump water from the reservoir or the underground water line into the tank, after the water is cleaned, the pump motor will go to the domestic water supply. … There are 2 sensors arranged at 2 positions of pumping in and pumping water out of the tank. At the pumping position, an inlet flow sensor is placed on the lid or mouth of the tank to measure the amount of water pumped in. In addition, the sensor can be used as a relay value when the pump motor is turned on.

At the pump in position, an inlet flow sensor is located on the lid or mouth of the tank to measure the amount of water injected. In addition, the sensor can also be used as a value relay when the motor is pumping in is enabled. At the pump out position, the pressure sensor is located behind the pump pipe of the pump out motor to measure the pumped water pressure. Sensor value for feedback tells the PLC whether water is being pumped out or not.

In one unit, because there are two motors which is used to pump out and pump in, so the control signal from the server will be the 2-ways signal. Therefore, at each unit, the router will be configured in the dial-in and dial-out way. In this system, 2 independent unit stations have the same function of pumping water out and in. So the configuration of two stations is the same, but the IP address for each units is different.

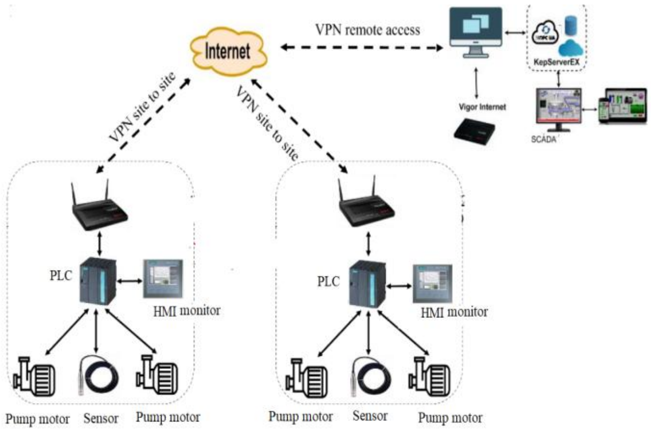


Fig. 12. Conection model in the unit station

Water flow measurement

The PLC receives the sensor values and displays them on the HMI control screen. Water flow and pressure are measured, so that the operator can operate the system through the control screen HMI. The flow meter sensor generates the pulse output signal. Accordingly, with one liter of water flowing through the sensor in 1 minute, the sensor generates 480 pulses. Or when one liter of water through the sensor in 1 minute, we get the frequency F = 8Hz, so the water flow can be calculated as the equation:

$$Q = \frac{F}{480} \tag{1}$$

Here: F - frequency (Hz)

Q - flow rate (litle per second).

Similarly, the out put of pressure sensor will be current in the range of 4 - 20 mA, corresponding to a pressure of 0 - 10 bar. According to Ohm's law in closed circuit, we have: $V = R.\ I$ with $R = 120\ \Omega$. Using the linear straight-line method, the pressure can be calculated as:

$$P = \frac{V - 0.48}{0.192} \tag{2}$$

Here: P - water pressure (bar)

V - sensor voltage (volt)

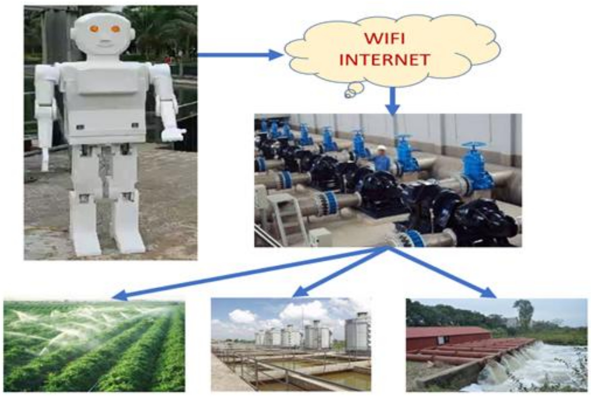


Fig. 13. Robots after being manufactured and put into use

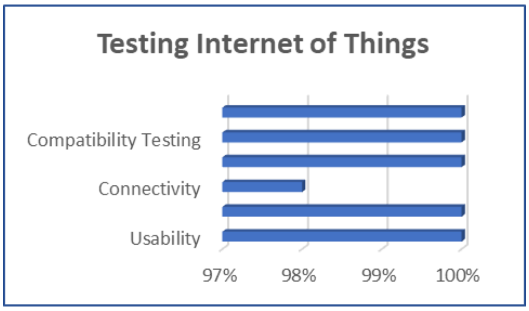
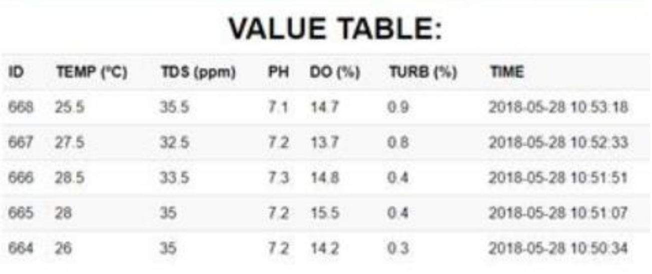


Fig. 14. Testing Internet of Things

As the results of testing water quality, the Temperature, TDS, pH, DO, TSS (Turb), TIME, Latitude, Longitude of the tested water sample is shown on the robot screen with the values as the Fig. 15.



VALUE TABLE:

ID	TEMP (°C)	TDS (ppm)	PH	DO (%)	TURB (%)	TIME
668	25.5	35.5	7.1	14.7	0.9	2018-05-28 10:53:18
667	27.5	32.5	7.2	13.7	0.8	2018-05-28 10:52:33
666	28.5	33.5	7.3	14.8	0.4	2018-05-28 10:51:51
665	28	35	7.2	15.5	0.4	2018-05-28 10:51:07
664	26	35	7.2	14.2	0.3	2018-05-28 10:50:34

Fig. 15. Data obtained from the monitoring system

The robot has been successfully tested with functions that work according to the set goals. The robot can give presentations and answer questions related to water resources, through the robot's computer system, which can be programmed to control water pumps and peripherals via wifi.

IV. CONCLUSION

The product of research results is an artificial intelligence robot applied in water resource management. Robots can also be applied in teaching and presenting topics on water resources. With new and intelligent features, the robot can be applied in water resources management agencies, supporting information search, asking and answering questions about water resources, regulating water sources at pumping stations through the Internet. This is also one of the first artificial intelligence robots applied in water resource management in Vietnam. Bases on these studies, it can be further developed to make the robot more complete and widely applied in the field of water resources management as well as environmental management. The water pollution monitoring system will be connected with the robot and wireless sensor network nodes operate based on Lora technology and process data transmitted to the robot. These parameters are very important in monitoring water quality which is comparing with standard values to show the pollution level of the water source. The analyzing, comparing, synthesizing data will bo calculated by robot and then issuing the warning to the person in charge.

References

[1] Hu, S. (2015, July). Dynamic monitoring based on wireless sensor networks of IoT. In Logistics, Informatics and Service Sciences (LISS), 2015 International Conference on (pp. 1-4). IEEE.

[2] Ferdoush, S., & Li, X. (2014). Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications. Procedia Computer Science, 34, 103-110.

[3] Le Hoai Quoc, Robotics Technology, Ho Chi Minh City National University Publishing House, 2005.

[4] Peter Marwedel, 2011. Embedded System Design: Embedded Systems Foundations of Cyber Physical Systems, 2nd Edition. Springer.

[5] Sung, W. T., Chen, J. H., Huang, D. C., & Ju, Y. H. (2014, October). Multisensors realtime data fusion optimization for IoT systems. In 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2299-2304). IEEE

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 295–298 DOI: 10.15439/2022R43 ISSN 2300-5963 ACSIS, Vol. 33

Bank Loan Analysis using Data Mining Techniques

Thi-Nhi Trinh
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
nhittcntt@gmail.com

Hoang-Diep Nguyen
Faculty of Information Technology
Hung Yen University of Technology and Education
Hung Yen, Vietnam
diep82003@gmail.com

***Abstract*—Nowadays, a bank loan can provide people with cash to fund home improvements or start a business. However, some customers who are accepted with a loan cannot repay or someone usually repays in a delayed time. Therefore, to minimize losses, examining loan applications is particularly evident for the bank. This paper study on bank loan analysis using data mining techniques. We use association rules mining, clustering, and classification techniques on the applicant's profile to help the bank quickly decide for a loan applicant.**

***Index Terms*—bank loan analysis, association rules mining, clustering, classification.**

I. Introduction

The bank has to decide whether to go ahead with the loan approval or not, this was based on the applicants' profiles. There are two kinds of situation risks for banks that are associated with the bank's decision: (1) if the applicants can repay the loan - good credit risk, the bank approves their application, (2) if the applicants are unable to repay the loan - bad credit risk, the bank does not approve this loan. Analyzing patterns from customer profiles in the dataset and making the right decisions based on patterns plays an extremely important role in the Bank's business. In this paper, we present the approaches for analyzing the credit data that contains the loan applicants. First, we use the Apriori algorithm to find associations given the rules to predict whether an applicant is 'good' or 'bad'. Second, we use K-Means clustering to group applicants with similar attributes into different groups which helps a bank manager to understand easily the characteristics of the group of applicants. Finally, we use classification algorithms including Naïve Bayes and K Nearest Neighbor to classify an applicant into 'good' or 'bad', which is an important task to help a bank manager quickly make a decision for a loan applicant.

II. Background

A. Mining Association Rules

We summarize the association rules issue as follows:

Let $I = \{i_1, i_2, \ldots, i_n\}$, set I includes the elements.

D: a set of all transactions where each transaction T is a set of elements such that $T \subseteq I$.

X, Y be a set of elements such that $X, Y \subseteq I$. An association rule implies the form $X \Longrightarrow Y$, where $X \subset I$, $Y \subset I$, $X \cap Y = \varnothing$.

Support

The rule $X \Longrightarrow Y$ holds with support ***s*** if ***s***% of transactions in D contains $X \cup Y$. Rules that have a ***s*** greater than user-specified support is said to have minimum support.

Confidence

The rule $X \Longrightarrow Y$ holds with confidence ***c*** if ***c***% of transactions in D contains $X \cup Y$. Rules that have a ***c*** greater than user-specified confidences are said to have minimum confidence.

B. K-Means Clustering

K-means algorithm performs the following steps:

1. Select *k* applicants from *S* to be used as cluster centroids (random)
2. Assign applicants to clusters according to their similarity to the cluster centroids.
3. For each cluster, recalculate the cluster centroid using the newly calculated cluster members.
4. Go to step 2 until the process converges.

C. Classification

Classification is a data mining technique used to predict class levels for data instances. In this paper, we are using two different classification techniques to predict the class level including Naïve Bays and K Nearest Neighbors.

Naive Bayes Algorithm

The Bayesian approach determines the class of document *x* as the one that maximizes the conditional probability P(C | *x*).

$$P(C \mid x) = \frac{P(x \mid C)\,P(C)}{P(x)} \qquad (1)$$

To compute P(x |C) we use equation as follows:

$$P(x \mid C) = P(x_1, x_2, \ldots, x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C) \qquad (2)$$

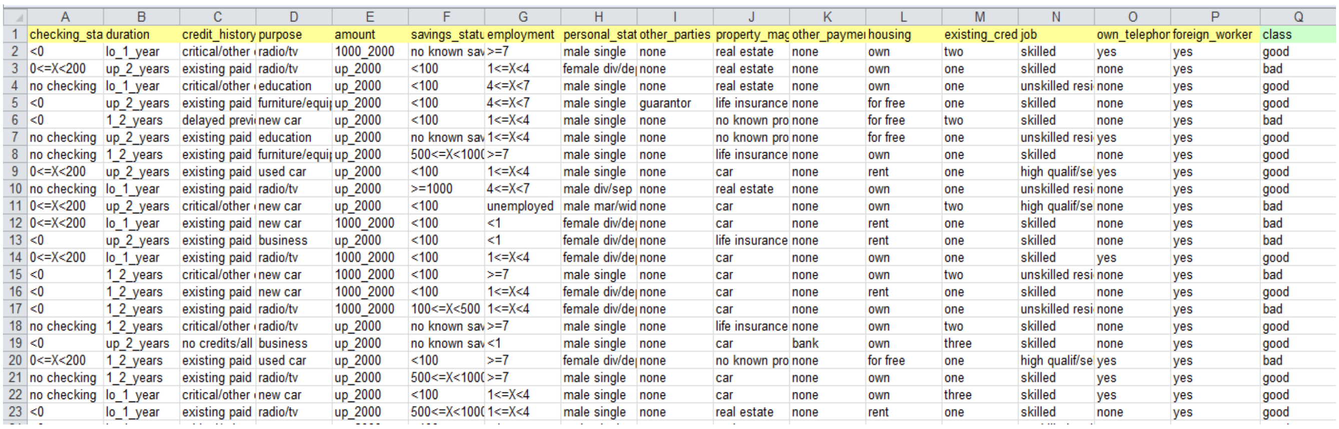
Where $P(x_i|C)$ is calculated as the proportion of items from class C that include attribute value x_i; P(C) is the probability of sampling for class C.

K Nearest Neighbor

K Nearest Neighbor is to predict the class of a new sample using the class label of the closest sample. We can summarize K-Nearest Neighbor algorithm as the following steps:

(1) Determine parameter K = number of nearest neighbors.
(2) Calculate the distance between the query instance and all the training samples.
(3) Sort the distance and determine the nearest neighbors based on the K-th minimum distance.
(4) Gather the category *Y* of the nearest neighbors.
(5) Use the simple majority of the category of nearest neighbors as the prediction value of the query instance.

III. EXPERIMENTS



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	checking_sta	duration	credit_history	purpose	amount	savings_statu	employment	personal_stat	other_parties	property_mag	other_paymen	housing	existing_cred	job	own_telephor	foreign_worker	class
2	<0	lo_1_year	critical/other	radio/tv	1000_2000	no known sav	>=7	male single	none	real estate	none	own	two	skilled	yes	yes	good
3	0<=X<200	up_2_years	existing paid	radio/tv	up_2000	<100	1<=X<4	female div/de	none	real estate	none	own	one	skilled	none	yes	bad
4	no checking	lo_1_year	critical/other	education	up_2000	<100	4<=X<7	male single	none	real estate	none	own	one	unskilled resi	none	yes	good
5	<0	up_2_years	existing paid	furniture/equi	up_2000	<100	4<=X<7	male single	guarantor	life insurance	none	for free	one	skilled	none	yes	good
6	<0	1_2_years	delayed previ	new car	up_2000	<100	1<=X<4	male single	none	no known pro	none	for free	two	skilled	none	yes	bad
7	no checking	up_2_years	existing paid	education	up_2000	no known sav	1<=X<4	male single	none	no known pro	none	for free	one	unskilled resi	yes	yes	good
8	no checking	1_2_years	existing paid	furniture/equi	up_2000	500<=X<100	>=7	male single	none	life insurance	none	own	one	skilled	none	yes	good
9	0<=X<200	up_2_years	existing paid	used car	up_2000	<100	1<=X<4	male single	none	car	none	rent	one	high qualif/se	yes	yes	good
10	no checking	lo_1_year	existing paid	radio/tv	up_2000	>=1000	4<=X<7	male div/sep	none	real estate	none	own	one	unskilled resi	none	yes	good
11	0<=X<200	up_2_years	critical/other	new car	up_2000	<100	unemployed	male mar/wid	none	car	none	own	two	high qualif/se	none	yes	bad
12	0<=X<200	lo_1_year	existing paid	new car	1000_2000	<100	<1	female div/de	none	car	none	rent	one	skilled	none	yes	bad
13	<0	up_2_years	existing paid	business	up_2000	<100	<1	female div/de	none	life insurance	none	rent	one	skilled	none	yes	bad
14	0<=X<200	lo_1_year	existing paid	radio/tv	1000_2000	<100	1<=X<4	female div/de	none	car	none	own	one	skilled	yes	yes	good
15	<0	1_2_years	critical/other	new car	1000_2000	<100	>=7	male single	none	car	none	own	two	unskilled resi	none	yes	bad
16	<0	1_2_years	existing paid	new car	1000_2000	<100	1<=X<4	female div/de	none	car	none	rent	one	skilled	none	yes	good
17	<0	1_2_years	existing paid	radio/tv	1000_2000	100<=X<500	1<=X<4	female div/de	none	car	none	own	one	unskilled resi	none	yes	bad
18	no checking	1_2_years	critical/other	radio/tv	up_2000	no known sav	>=7	male single	none	life insurance	none	own	two	skilled	none	yes	good
19	<0	up_2_years	no credits/all	business	up_2000	no known sav	<1	male single	none	car	bank	own	three	skilled	none	yes	good
20	0<=X<200	1_2_years	existing paid	used car	up_2000	<100	>=7	female div/de	none	no known pro	none	for free	one	high qualif/se	yes	yes	bad
21	no checking	1_2_years	existing paid	radio/tv	up_2000	500<=X<100	>=7	male single	none	car	none	own	one	skilled	yes	yes	good
22	no checking	lo_1_year	critical/other	new car	up_2000	<100	1<=X<4	male single	none	car	none	own	three	skilled	yes	yes	good
23	<0	lo_1_year	existing paid	radio/tv	up_2000	500<=X<100	1<=X<4	male single	none	real estate	none	rent	one	skilled	none	yes	good

Figure 1: The German Credit data set

A. *Dataset*

In this paper, we use the German Credit Data that contains data on 17 variables of 1000 past applicants for credit. Each applicant was rated as “good credit” (700 cases) or “bad credit” (300 cases). The attributes of each credit applicant are included as follows:

1. Status of account
2. Time in a month
3. History of credit
4. Purpose
5. Credit money
6. Savings bonds/account
7. Employment since
8. Personal status and sex
9. Other debtors / guarantors
10. Property
11. Payment plans
12. Housing
13. Number of existing credits
14. Job
15. Telephone
16. foreign worker
17. Class

B. *Experimental Results*

Weka is a software that helps people analyze data and build predictive models quickly and accurately. In this paper, we use Weka to run the Apriori algorithm, K-means clustering, Naïve Bays, and K Nearest Neighbor. We dynamically change the parameter to obtain a better result.

In this section, we present the evaluation of three data mining techniques for analyzing credit data. We run four algorithms in Weka including Apriori, K-means, Naïve Bays, and K-Nearest Neighbor, and analyze their results.

(1) Mining Association Rule with Apriori Algorithm

- **Run Apriori with default value of parameters by Weka**

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Set Minimum support is 0.6 (600 instances)

Set confidence is 0.9

Set number of executed is 8

Return best results

1. other_parties=none 907 ==> foreign_worker=yes 880 conf:(0.97)
2. job=skilled 630 ==> foreign_worker=yes 611 conf:(0.97)
3. other_parties=none other_payment_plans=none 742 ==> foreign_worker=yes 718 conf:(0.97)
4. other_parties=none housing=own 647 ==> foreign_worker=yes 625 conf:(0.97)
5. other_parties=none class=good 635 ==> foreign_worker=yes 611 conf:(0.96)
6. existing_credits=one 633 ==> foreign_worker=yes 609 conf:(0.96)
7. housing=own 713 ==> foreign_worker=yes 685 conf:(0.96)
8. other_payment_plans=none 814 ==> foreign_worker=yes 782 conf:(0.96)
9. class=good 700 ==> foreign_worker=yes 667 conf:(0.95)
10. other_payment_plans=none foreign_worker=yes 782 ==> other_parties=none 718 conf:(0.92)

- **Run Apriori with customization value of parameters**

In this case, we change parameter *car = ‘true’* to enable class association rules to be mined instead of (general) association rules. Then we use *classIndex = -1* to make the last attribute taken as a class attribute. Here, the ‘class’ attribute in our dataset is the last attribute that has a ‘good’ or ‘bad’ applicant. Then, we run Apriori again, and the results are as follows:

Set minimum support is 0.2 (200 instances)

Set confidence is 0.9

Set number of executed is 16

Return best results

1. checking_status=no checking other_parties=none other_payment_plans=none housing=own 244 ==> class=good 228 conf:(0.93)
2. checking_status=no checking other_parties=none other_payment_plans=none housing=own foreign_worker=yes 236 ==> class=good 220 conf:(0.93)

3. checking_status=no checking other_parties=none other_payment_plans=none job=skilled 217 ==> class=good 202 conf:(0.93)

4. checking_status=no checking other_payment_plans=none housing=own 256 ==> class=good 238 conf:(0.93)

5. checking_status=no checking other_payment_plans=none housing=own foreign_worker=yes 247 ==> class=good 229 conf:(0.93)

6. checking_status=no checking other_parties=none other_payment_plans=none 313 ==> class=good 290 conf:(0.93)

7. checking_status=no checking other_payment_plans=none job=skilled 230 ==> class=good 213 conf:(0.93)

8. checking_status=no checking other_parties=none other_payment_plans=none foreign_worker=yes 303 ==> class=good 280 conf:(0.92)

9. checking_status=no checking other_payment_plans=none job=skilled foreign_worker=yes 223 ==> class=good 206 conf:(0.92)

10. checking_status=no checking other_payment_plans=none 330 ==> class=good 303 conf:(0.92)

(2) K-Mean Clustering

We run K-means algorithm with various the number of clusters from 2 to 6. And we found the averaged Sum of Squared Error as shown in Figure 2.

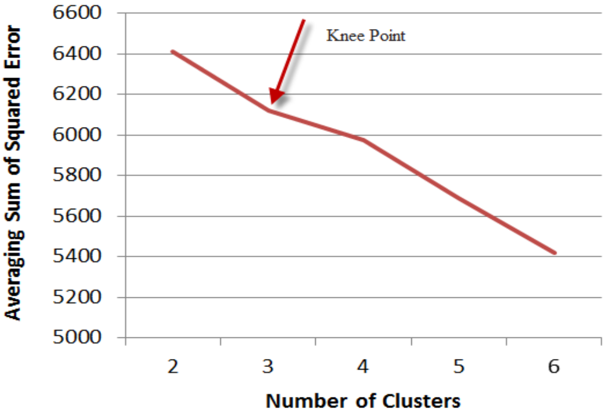
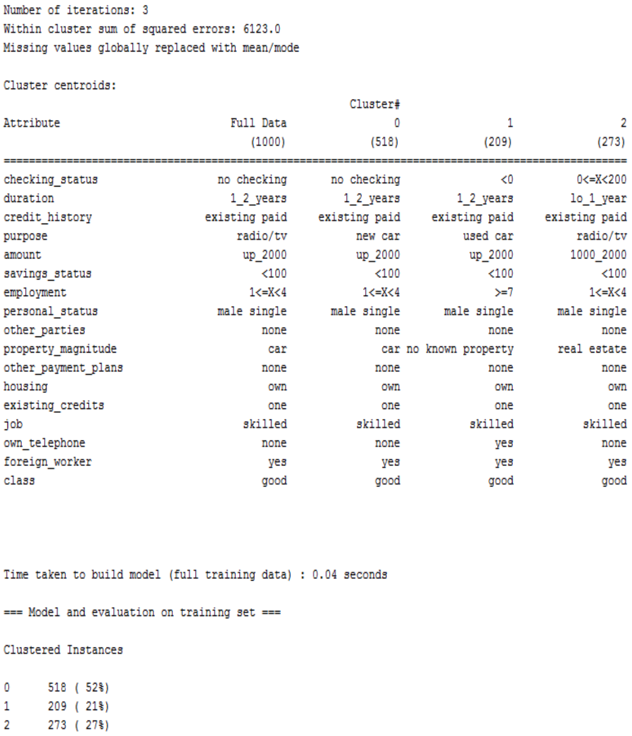


Figure 2: The averaged Sum of Squared Error

The knee point with k=3 shows that this data set should be grouped into 3 clusters. The results for running K-means with 3 clusters are illustrated in Figure 3 and Figure 4.



```
Number of iterations: 3
Within cluster sum of squared errors: 6123.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                         Cluster#
Attribute                    Full Data          0                1                2
                                (1000)      (518)            (209)            (273)
=====================================================================================
checking_status            no checking no checking              <0         0<=X<200
duration                     1_2_years  1_2_years        1_2_years        lo_1_year
credit_history           existing paid existing paid  existing paid    existing paid
purpose                       radio/tv    new car         used car         radio/tv
amount                         up_2000    up_2000          up_2000        1000_2000
savings_status                    <100       <100             <100             <100
employment                      1<=X<4     1<=X<4              >=7           1<=X<4
personal_status            male single male single     male single      male single
other_parties                     none       none             none             none
property_magnitude                 car        car no known property     real estate
other_payment_plans               none       none             none             none
housing                            own        own              own              own
existing_credits                   one        one              one              one
job                            skilled    skilled          skilled          skilled
own_telephone                     none       none              yes             none
foreign_worker                     yes        yes              yes              yes
class                             good       good             good             good



Time taken to build model (full training data) : 0.04 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       518 ( 52%)
1       209 ( 21%)
2       273 ( 27%)
```

Figure 3: The results run K-means with 3 clusters

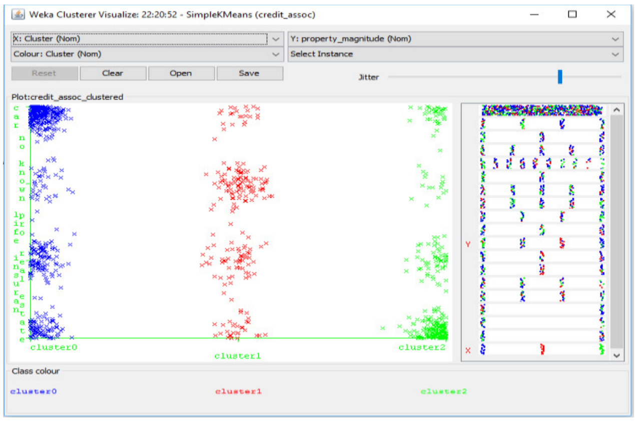


Figure 4: The results run K-means with 3 clusters and plot by Property Magnitude attribute

(3) Classification.

a) Data preparation

The data is separated into two *.csv files in which 70% data for training is stored in credit-train.csv and 30% data for testing is stored in credit -test.csv.

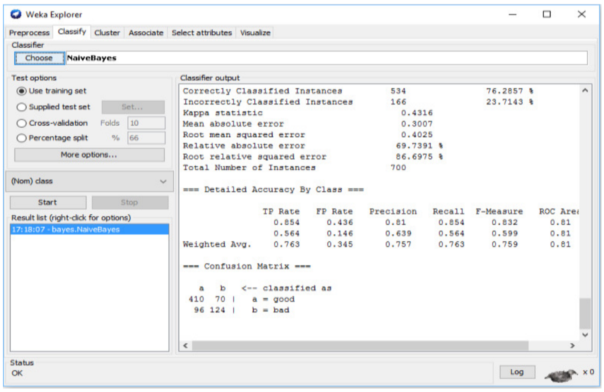
TABLE 1: THE DATASET SETTING FOR CLASSIFICATION

Class	Training data (number of items)	Testing data (number of items)
Good	480	220
Bad	220	80
Total	**700**	**300**

b) Classification with Weka

Run Naive Bayes Classify

- The results from training dataset



Classifier: NaiveBayes

Test options: ● Use training set ○ Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

Result list (right-click for options): 17:18:07 - bayes.NaiveBayes

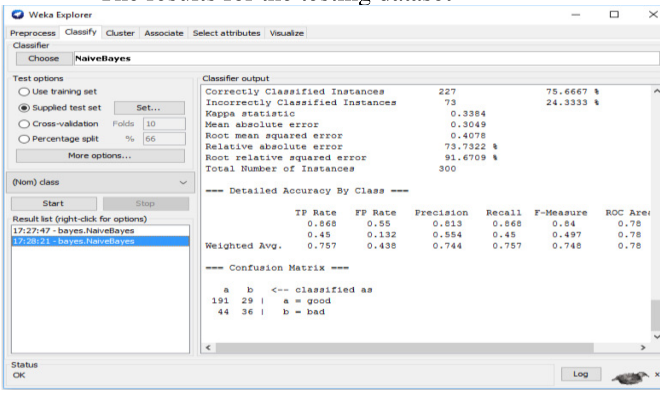
Status: OK

Classifier output

```
Correctly Classified Instances         534               76.2857 %
Incorrectly Classified Instances       166               23.7143 %
Kappa statistic                          0.4316
Mean absolute error                      0.3007
Root mean squared error                  0.4025
Relative absolute error                 69.7391 %
Root relative squared error             86.6975 %
Total Number of Instances              700

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Are
                 0.854     0.436     0.81      0.854     0.832      0.81
                 0.564     0.146     0.639     0.564     0.599      0.81
Weighted Avg.    0.763     0.345     0.757     0.763     0.759      0.81

=== Confusion Matrix ===

   a   b   <-- classified as
 410  70 |   a = good
  96 124 |   b = bad
```

Figure 5: Illustrates the results for training with Naive Bayes

- The results for the testing dataset



Classifier: NaiveBayes

Test options: ○ Use training set ● Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

Result list (right-click for options): 17:27:47 - bayes.NaiveBayes
17:28:21 - bayes.NaiveBayes

Status: OK

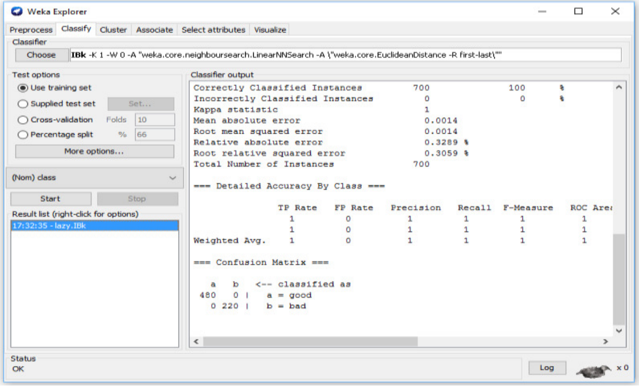
Classifier output

```
Correctly Classified Instances         227               75.6667 %
Incorrectly Classified Instances        73               24.3333 %
Kappa statistic                          0.3384
Mean absolute error                      0.3049
Root mean squared error                  0.4078
Relative absolute error                 73.7322 %
Root relative squared error             91.6709 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Are
                 0.868     0.55      0.813     0.868     0.84       0.78
                 0.45      0.132     0.554     0.45      0.497      0.78
Weighted Avg.    0.757     0.438     0.744     0.757     0.748      0.78

=== Confusion Matrix ===

   a   b   <-- classified as
 191  29 |   a = good
  44  36 |   b = bad
```

Figure 6: Illustates the results for testing with Naive Bayes

Run K-Nearest Neighbor Classify

- The results for training dataset



Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: ● Use training set ○ Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

Result list (right-click for options): 17:32:35 - lazy.IBk

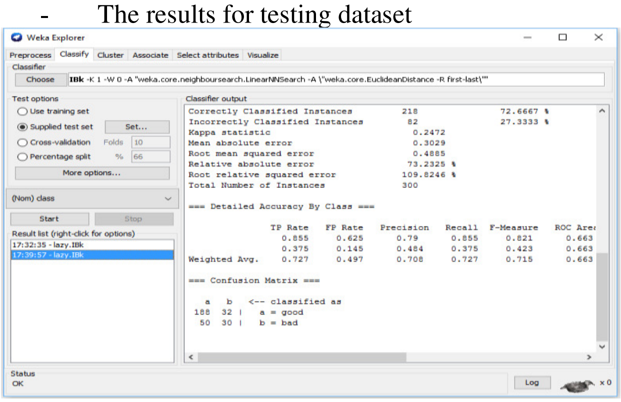
Status: OK

Classifier output

```
Correctly Classified Instances         700              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0.0014
Root mean squared error                  0.0014
Relative absolute error                  0.3289 %
Root relative squared error              0.3059 %
Total Number of Instances              700

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Are
                 1         0         1         1         1          1
                 1         0         1         1         1          1
Weighted Avg.    1         0         1         1         1          1

=== Confusion Matrix ===

   a   b   <-- classified as
 480   0 |   a = good
   0 220 |   b = bad
```

Figure 7: Illustates the results for training with KNN

- The results for testing dataset



Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: ○ Use training set ● Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

Result list (right-click for options): 17:32:35 - lazy.IBk
17:39:57 - lazy.IBk

Status: OK

Classifier output

```
Correctly Classified Instances         218               72.6667 %
Incorrectly Classified Instances        82               27.3333 %
Kappa statistic                          0.2472
Mean absolute error                      0.3029
Root mean squared error                  0.4885
Relative absolute error                 73.2325 %
Root relative squared error            109.8246 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Are
                 0.855     0.625     0.79      0.855     0.821      0.663
                 0.375     0.145     0.484     0.375     0.423      0.663
Weighted Avg.    0.727     0.497     0.708     0.727     0.715      0.663

=== Confusion Matrix ===

   a   b   <-- classified as
 188  32 |   a = good
  50  30 |   b = bad
```

Figure 8: Illustates the results for testing with KNN

Analysis. We can summarize the results as the tables below.

TABLE 2: SUMMARY OF THE RESULTS

Methods	Traning (Accuracy)	Testing (Accuracy)
Naïve Bayes	534/480	227/300
KNN	700/700	218/300

TABLE 3: CONFUSION MATRIX FOR RUNNING NAÏVE BASE

Assigned True	C1	C2	Total
C1	191	29	220
C2	44	36	80

TABLE 4: CONFUSION MATRIX FOR RUNNING KNN

Assigned True	C1	C2	Total
C1	188	32	220
C2	50	30	80

As the results, we can see that the trained KNN algorithm better than the Naïve Bayes algorithm for the German Credit dataset that is given above. However, for the results of the classification on the testing dataset, Naïve Bayes outperforms the KNN algorithm.

V. CONCLUSIONS

In this paper, we present approaches for analyzing the credit data and provide the information for the bank manager to make a decision regarding loan approval. Besides, we can conclude as follows. (1) For mining association rules, (a) some association rules have very high confidence, but it is not important in particular; (b) the number of rules depends on the confidence and minimum support; (c) the confidence of a rule does not depending on the minimum support. (2) It is better to classify the German Credit dataset into 3 clusters. (3) Even though the KNN algorithm trained better than the Naïve Bayes algorithm for the German Credit dataset, Naïve Bayes outperforms the KNN algorithm for classifying new applicants for loan service.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

Proceedings of the Seventh International Conference on Research in Intelligent and Computing in Engineering pp. 299–303 DOI: 10.15439/2022R39 ISSN 2300-5963 ACSIS, Vol. 33

Application of Machine Learning in Malicious IoT Classification and Detection on Fog-IoT Architecture

Duong Thi Van
Institute of Information Technology
Vietnam Academy of Science and Technology)
Hanoi, Viet Nam
dtvan@ioit.ac.vn

Tran Ba Hung
Institute of Information Technology
Vietnam Academy of Science and Technology)
Hanoi, Viet Nam
tbhung@ioit.ac.vn

Tran Duc Thang
Institute of Information Technology
Vietnam Academy of Science and Technology)
Hanoi, Viet Nam
thang@ioit.ac.vn

Nguyen Khac Giao
Institute of Information Technology
Vietnam Academy of Science and Technology)
Hanoi, Viet Nam
giaonk@ioit.ac.vn

***Abstract*—Due to the limitations in self-protection and information processing capabilities at IoT (Internet of Things) nodes, these nodes are susceptible to attacks, turning them into malicious nodes that cause damage or danger to the system. Early detection of these threats is essential to make timely recommendations and limit severe consequences for individuals and organizations. The study proposes applying a machine learning model to detect malicious traffic and IoT devices, which can be deployed and applied on the Fog IoT platform. This solution helps detect and early warn threats from IoT data before they are sent to the cloud. The model is evaluated on the IoT-23 dataset and gives good results.**

Keywords—Fog Computing; Security IoT; Malicious IoT Devices; Fog Node.

I. INTRODUCTION

The world is fighting to transition to the Internet of Things (IoT) society [1], where everything can be connected via the Internet. The help of smart devices, embedded devices, and sensors makes life and work more efficient and productive. More and more individuals and organizations use the services that IoT brings. Therefore, securing data and providing user privacy is of great interest.

IoT devices are not standing alone. They are often part of Cloud computing architecture (Cloud computing - CC) to provide on-demand services such as resources, storage, and services [2] ... Users only need to request services without regard to the construction, management, or installation behind it. As technology develops and the number of IoT devices increases rapidly. According to iot-analytics.com, about 50 billion IoT devices are connected to the Internet [3].

Unlike standard personal computers, IoT nodes are limited in processing capacity and storage capacity, so they need support from the Cloud server for analysis and decision-making. Data collected from IoT devices is continuously sent to the cloud. In some cases, the amount of data obtained is huge, fighting, leading to overload, and the response time from the cloud is not enough to meet many real-time applications. To overcome the above limitation, CISCO first proposed the concept of Fog Computing - FC [4]. Fog Computing is a new platform extending new cloud services closer to IoT devices. Fog computing has gained additional advantages over cloud computing, making it a remarkable platform: Utilizing device resources, low latency, distributed instead of centralized Centralized and supporting real-time applications. Therefore, fog computing helps to facilitate the deployment of many applications.

Besides inheriting the advantages of Cloud Computing, this technology has its limitations, so the number of cyber attacks still increases in frequency and intensity [5]. Traditional security solutions, such as firewalls and intrusion detection systems, are not enough to overcome this drawback. Applying machine learning to detect threats and attacks based on network traffic is still a topic of interest to researchers.

In this study, we use a new machine learning model compared to traditional models, XGBoost [6] to detect anomalies and provide early warning of malicious IoT risks. Ongoing damage/type of attack for timely warnings. We also assess malicious IoT devices based on the above results to find out which devices have been infected with malicious code or are the source of the attack. We deploy simulation on Fog-IoT architecture, collect data from the IoT layer, and aggregate it into analytical characteristics, thus predicting whether IoT node activity is benign or malicious.

The next structure of the paper is as follows: In Part II, we give an overview of the Fog-IoT architecture. Some related studies are presented in section III. The proposed method and experimental results are presented in sections IV and V, respectively. Finally, conclusions and development directions.

II. FOG-IOT ARCHITECTURE

A. Fog-IoT Architecture

Ants Fog - IoT architecture consists of three layers [7]:

- Device layer: Consists of IoT devices that are physically distributed, with limited computing power and storage resources. These devices often collect and send raw data to a cloud server for storage and processing.

- Fog layer: Includes devices such as switches, routers, and access points. This layer is located between the device layer and the cloud server layer, capable of processing and calculating data before uploading to the cloud. The fog layer provides various services and real-time analytics for data from IoT devices.

- Cloud layer: The cloud layer includes many high-performance servers and long-term data storage capabilities, providing a variety of applications and services. These services are designed to be accessible anywhere and anytime.

B. *Security issues in Fog –IoT*

Fog Computing has many advantages in improving service quality, but new problems related to security and privacy arise. Regarding computing power, it is difficult for FC to implement a complete set of security solutions to detect and prevent attacks. Additionally, due to its proximity to IoT devices, Fog is first vulnerable to malicious IoT attacks. Besides, since FC gets data from IoT devices and the cloud, it becomes a potential target.

There are some attack techniques on layers of IoT. On the perception layer, attackers can exploit eavesdropping, spoofing, or radio frequency jamming techniques. The Network layer can face attacks such as denial of service, malicious code injection, or man in the middle. On the support and application layers, hackers can execute denial-of-service attacks, malicious code injection, or eavesdrop on packets.

Since many IoT devices collect and exchange sensitive and private data, data security is a central issue. Several solutions to ensure the safety of IoT have been proposed. However, security threats are increasingly complex due to the heterogeneous diversity and increasing number of IoT. Existing solutions need to be continuously improved to deal with new security threats in IoT systems.

III. RELATED STUDIES

The rapid increase in data collected from IoT devices and different communication protocols has increased security risks, demonstrating the need for an effective IDS system. Researchers have focused on anomaly detection methods using new techniques such as machine learning and deep learning. In the past, some researchers have used the KDD99 or NSL-KDD dataset to identify malicious behaviors. The survey's main findings underscore the need for a legitimate and modern dataset to get accurate outputs.

In an IoT node attack early detection study, Y. Meidan et al. [10] performed a snapshot of the traffic behavior for every IoT device to extract the attributes. These attributes are used as input to deep learning techniques to detect anomalies. They trained an autoencoder (one encoder per device) to learn the normal operation of the IoT device. Using an autoencoder is learning complex patterns and minimizing false alarms.

A recent study by Layla Albdour et al . [8] uses a crawler that acts as a security checker to monitor IoT nodes and collect data streams to analyze the behavior of the nodes. Based on that, to put the fake IoT alert action. However, placing the behavior analyzer at all Fog nodes in the system (distributed) is not efficient in terms of time. In addition, the algorithm becomes complicated when it has to process simultaneously—several jobs in different places. In addition, if an attack occurs at a branch Fog system, it will be challenging to take action to respond on time.

Hasan, Islam, Zerif, et al. [12] have implemented machine learning algorithms to detect if the system is behaving abnormally. If so, they use algorithms to detect the type of attack that the system is not behaving device in progress. DS2OS dataset [11] was used for evaluation. The Random Forest algorithm is the best choice, with an accuracy of 99.4%.

In their research, Idrissi and his colleagues [14] learned about security vulnerabilities in IoT. They identify several vulnerabilities and threats on IoT devices, thus offering recommended solutions. They use a neural network-based intrusion detection technique to solve the problem and achieve satisfactory results.

In this paper, we propose to apply the XGBoost machine learning model to detect malicious network traffic from IoT devices. XGBoost is a model that previous studies have not fully considered. At the same time, we also tested a new job to detect malicious IoT devices based on the results of malicious traffic classification. Our contribution to detecting and removing malicious IoT devices in the Fog IoT network.

IV. PROPOSED METHODS

The number of IoT devices in the Fog-IoT network can be huge, including malicious and benign nodes. The research objective is to quickly detect network traffic generated by malicious nodes at the Fog layer. At the same time, scan and warn those malicious nodes to the administrator. Our proposed algorithm aims to solve the above two problems.

A. *Anomaly detection algorithm*

In this problem, we propose to apply a recently published algorithm model called XGBoost (Extreme Gradient Boosting). It is an algorithm developed based on gradient boosting, with many improvements in terms of algorithm optimization. XGBoost is suitable for many problems, such as regression, classification, and ranking.

The algorithm is as follows:

- Data: Where n is the number of training samples, m is the number of attributes, $D = \{(x_i, y_i)\}$ is the data set. With x_i is the th data i and y_i is the label corresponding to $1 \leq i \leq n$. We have:

$$|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R} \qquad (1)$$

- The tree structure includes: q is the structure of a tree, f_k is the structure of k independent trees, with w_i is the weight of the i-th leaf node.

We have the following learning model:

$$\widehat{y_i} = \emptyset(x_i) = \sum_{k=1}^{K} f_k\,(x_i), f_k \in F \qquad (2)$$

With K is the space of independent tree structures. The learning model determines the label of the value x_i based on the computation on each objective function in turn f_k. The final aggregate results help to find the label for the data.

- F is the objective function, expressed as follows:

$$F = \{f(x) = \omega_{q(x)}\} \tag{3}$$

Where $\omega_{q(x)}$ is the weight of the node $q_{(x)}$.

- L is a learning function, calculated by the following formula:

$$L(\emptyset) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{4}$$

The learning function L calculates the difference between the correct and predicted labels, determines the model weights, and evaluates the convergence of the model.

The XGBoost algorithm scales down the leaf nodes, improving the model's generality. Previous studies have shown the effectiveness of XGBoost in classification and regression problems.

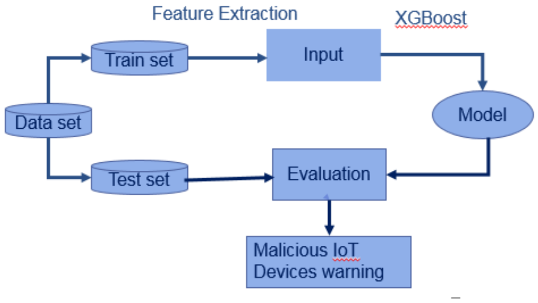


Fig. 1. Application model of the XGBoost algorithm

The flowchart of the proposed solution is described in Fig. 1. In which the Train Set data is attribute extracted and trained by XGBoost, the results are evaluated on the Test Set. Based on the labeling results, malicious IoT devices will be detected and alerted to users.

B. Deployment model on Fog-IoT

In this study, we propose a security control model on the Fog-IoT architecture. Adding a Server node behind the Fog layer monitors information flows when sending to the cloud, as shown in Fig. 2.

The Fog layer performs the function of processing information collected from the IoT layer. Data before going to the cloud will be analyzed and monitored at the server. Here, the server will do two things: (1) Detect malicious network traffic and stop them; (2) Report malicious IoT nodes.

V. EXPERIENCE AND ASSESSMENT

A. Experimental data set

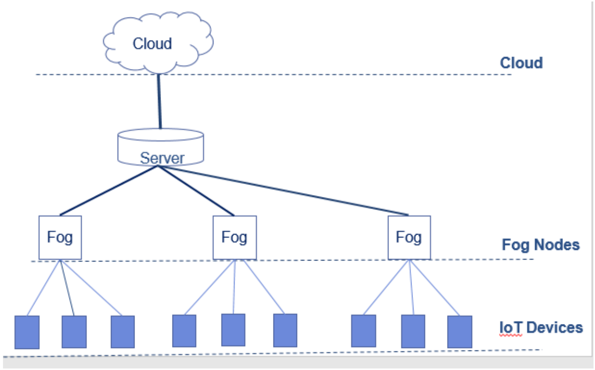


Fig. 2. Proposed security analysis model on Fog - IoT

This study uses the IoT-23 dataset [15], which Avast AIC lab generated. The dataset contains 20 types of malicious traffic logged from different IoT devices, which was collected from 2018 to 2019. The flow labels are the type of malicious traffic contained in the dataset, generated in the Stratosphere lab, and include the following: Attack, Benign, C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, Torii.

Collected network traffic for benign situations is obtained by collecting the network traffic of three IoT device types. These are actual deployed IoT devices, not simulators. This allows the demonstration of an IoT network model similar to reality. Evaluation results on this data set are also more reliable.

B. Experimental results

We perform our evaluations using the Python language, machine learning support libraries, and execution on the Google CoLab platform, Intel Xeon Processor 2.3GHz chip. The algorithm used is XGBoost.

The results of classification experiment on the IoT-23 dataset are given in Table I below:

TABLE I. CLASSIFICATION RESULTS OF NETWORK TRAFFIC

	Precision	**Recall**	**F_1-score**	**support**
Attack	0.99	0.98	0.98	783
Benign	*0.95*	*0.56*	*0.71*	*39951*
C&C	1.00	0.11	0.20	3020
C&C-FileDownload	0.57	0.89	0.70	9
C&C-HeartBeat	0.88	0.21	0.34	70
C&C-HeartBeat-FileDownload	0.00	0.00	0.00	2
C&C-Torri	0.00	0.00	0.00	6
DDOS	1.00	0.82	0.90	27755
FileDownload	1.00	0.50	0.67	2
Okiru	0.48	0.00	0.00	52538
PartOfAHorizontalPortScan	0.68	1.00	0.81	165188
Accuracy			**0.73**	289324

The evaluation results show that the XGBoost algorithm effectively detects various types of malicious traffic. Specifically, the Attack, C&C, DDOS, and FileDowload labels were detected with a very high rate, and Precision reached 0.99 or higher. However, the Recall rate is low in the case of C&C and FileDownload. Overall, the F1-score achieved is very good with the Attack and DDOS labels, with 0.98 and 0.90, respectively. It should also be noted that these labels have high Support. They make up most of the dataset.

Some other labels have low detection rates, such as Okiru (Precision is 0.48) and C&C's compound labels with malicious code on IoT. This could be explained by some of the labels compounded with C&C detected in the C&C label. At the same time, the number of samples for these labels is minimal, with the lowest being 02 and the highest being 70, resulting in the model lacking data to train. Overall, this low rate is acceptable because of its low specificity in the dataset.

The result correlation of the evaluation parameters of the network traffic labels is shown in Fig. 3 below.

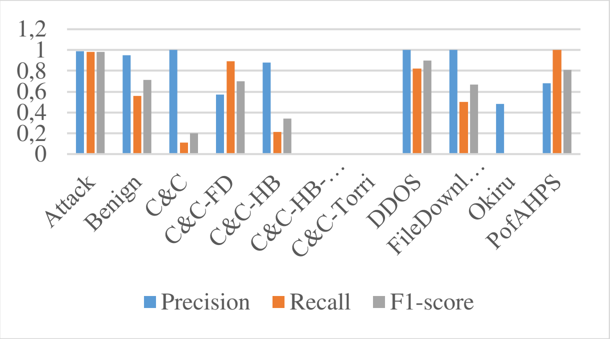


Fig. 3. Correlation of evaluation parameters of labels

The algorithm also performed well when correctly identifying benign data, labeled Benign, with Precision, Recall, and F1-scores reaching 0.95, 0.56, and 0.71, respectively. The support number is 39951.

We conduct a k-fold evaluation with k=5 to get a more comprehensive assessment of the overall accuracy of the XGBoost algorithm. The results per fold are given in Table II as follows:

TABLE II. RESULTS OF MODEL EVALUATION THROUGH EACH I-FOLD

i-fold	Precision	Recall	F1
0	0.7305	0.7305	0.7305
first	0.7304	0.7304	0.7304
2	0.7302	0.7302	0.7302
3	0.7304	0.7304	0.7304
4	0.7315	0.7315	0.7315
AVG	*0.7306*	*0.7306*	*0.7306*

Overall, the XGBoost model has an accuracy of 0.73. This is a pretty good detection rate in the case of the IoT-23 dataset.

We exclude benign network traffic with the Benign label to detect malicious IoT devices. The remaining traffic is identified as malicious, and look for the IPs of those malicious IoT devices. The ratio of malicious IPs to the total IPs is 72.53%. In which the detection results of malicious IoT devices are given in Table III as follows:

TABLE III. SEARCH RESULTS FOR MALICIOUS IP ADDRESSES

	0	first
0	<.>	57
first	4	<524>

Label 0 represents benign IoT devices, and label 1 represents malicious IoT and belongs to the group of IPs suggested by the algorithm. The evaluation parameters Precision, Recall, and F1-score have values of 0.90, 0.99, and 0.94, respectively. This result shows that the model effectively detects malicious IoT devices and allows suggestions to prevent malicious behavior from these devices.

C. Comparison with some other machine learning models

Comparing the evaluation results on the IoT23 dataset above with some other machine learning models, including NB, ANN, and SVM by Stoian et al. [16], the accuracy results are shown in Table IV:

TABLE IV. COMPARISON OF CLASSIFICATION RESULTS WITH SOME OTHER MACHINE LEARNING ALGORITHMS

	XGBoost	NB	ANN	SVM
Accuracy	0.73	0.25	0.52	0.59

It can be seen that XGBoost improves the accuracy a lot over models like NB, ANN, and SVM, with an accuracy of 0.73 compared to 0.25, 0.52, and 0.59, respectively. In general, XGBoost is an algorithm with many advantages over NB, ANN, and SVM algorithms.

VI. CONCLUSION AND DEVELOPMENT

Since IoT nodes collect sensitive information from users, securing Fog-IoT systems is a matter of life and death. Furthermore, the increasing demand for use due to IoT's conveniences leads to increased security vulnerabilities. Early identification of an IoT node as malicious so that appropriate action can be taken before an attack occurs is a method of constant concern.

The study applied a new machine learning model, XGBoost, to detect malicious network traffic, blocking them at the Fog layer before sending them to the cloud. The algorithm also allows for finding IoT devices with malicious behavior, which is a hint for system administrators to review and handle them. Overall, the XGBoost model gave positive results with 73% accuracy when detecting malicious network traffic while finding malicious IoT devices with 90% accuracy.

In the future, this study needs to be applied experimentally on large data sets and more diverse. From there, it is possible to develop an appropriate security policy for the Fog - IoT system, improve safety, and ensure customer service quality.

VII. ACKNOWLEDGMENT

This research was partly supported by the project CS21.12 managed by the Institute of Information Technology, Vietnam Academy of Science and Technology.

REFERENCES

[1] M. S. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, “Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions,” Comput. Commun., vol. 139, pp. 32–57, 2019, doi: 10.1016/j.comcom.2019.03.009.

[2] H. Sabireen and V. Neelanarayanan, “A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges,” ICT Express, vol. 7, no. 2, pp. 162–176, 2021, doi: 10.1016/j.icte.2021.05.004.

[3] K. L. Lueth, “IoT market analysis : Sizing the opportunity,” IoT Anal., no. March, pp. 0–12, 2015, [Online]. Available: http://iot-analytics.com/iot-market-forecasts-overview/.

[4] S. Khanagha, S. Ansari, S. Paroutis, and L. Oviedo, “Mutualism and the dynamics of new platform creation: A study of Cisco and fog computing,” Strateg. Manag. J., vol. 43, no. 3, pp. 476–506, 2022, doi: 10.1002/smj.3147.

[5] A. Aljumah and T. A. Ahanger, “Fog computing and security issues: A review,” 2018 7th Int. Conf. Comput. Commun. Control. ICCCC 2018 - Proc., pp. 237–239, 2018, doi: 10.1109/ICCCC.2018.8390464.

[6] T. Chen, T. He, and M. Benesty, “XGBoost : eXtreme Gradient Boosting,” R Packag. version 0.71-2, pp. 1–4, 2018.

[7] S. P. Ahuja and N. Wheeler, “Architecture of Fog-Enabled and Cloud-Enhanced Internet of Things Applications,” Int. J. Cloud Appl. Comput., vol. 10, no. 1, pp. 1–10, 2019, doi: 10.4018/ijcac.2020010101.

[8] L. Albdour, S. Manaseer, and A. Sharieh, “IoT crawler with behavior analyzer at fog layer for detecting malicious nodes,” Int. J. Commun. Networks Inf. Secur., vol. 12, no. 1, pp. 83–94, 2020, doi: 10.17762/ijcnis.v12i1.4459.

[9] M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, “Attack and anomaly detection in IoT sensors in IoT sites using machine

learning approaches," Internet of Things (Netherlands), vol. 7, 2019, doi: 10.1016/j.iot.2019.100059.

[10] M.-O. Pahl and F.-X. Aubet, "DS2OS traffic traces | Kaggle," [Online]. Available: https://www.kaggle.com/francoisxa/ds2ostraffictraces.

[11] I. Idrissi, M. Azizi, and O. Moussaoui, "IoT security with Deep Learning-based Intrusion Detection Systems: A systematic literature review," 4th Int. Conf. Intell. Comput. Data Sci. ICDS 2020, 2020, doi: 10.1109/ICDS50568.2020.9268713.

[12] Q. Tian, J. Li, and H. Liu, "A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning," IEEE Access, vol. 7, pp. 38688–38695, 2019, doi: 10.1109/ACCESS.2019.2905754.

[13] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," Zenodo., 2020, [Online]. Available: http://doi.org/10.5281/zenodo.4743746.

[14] N. A. Stoian, "Machine Learning for Anomaly Detection in IoT networks: Malware analysis on the IoT-23 Data set," Univ. Twente, 2020.

Author Index