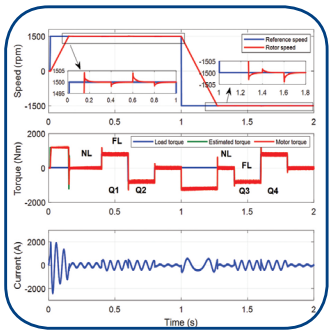
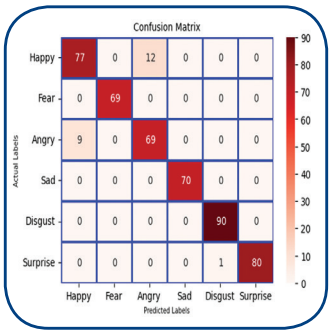
Volume 16, Number 1, 2025 ISSN 1847-6996

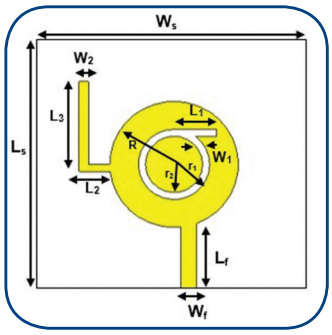
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY **OSIJEK**

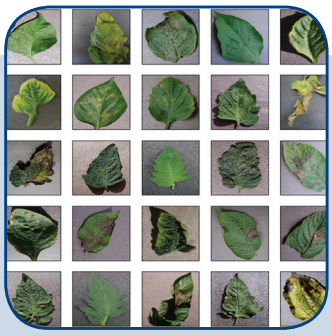
IJECES International Journal of Electrical and Computer Engineering Systems

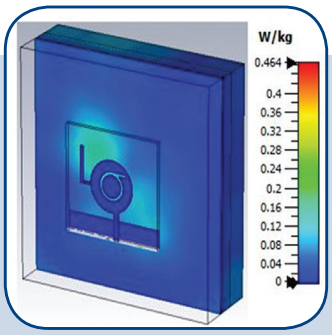
International Journal of Electrical and Computer Engineering Systems

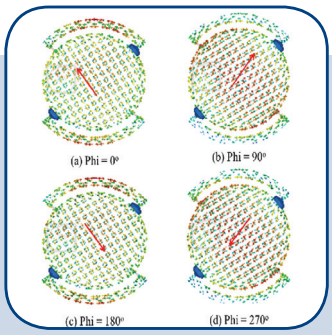












ISSN: 1847-6996

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Croatia

Osijek, Croatia | Volume 16, Number 1, 2025 | Pages 1 - 85

The International Journal of Electrical and Computer Engineering Systems is published with the financial support of the Ministry of Science and Education of the Republic of Croatia

CONTACT

International Journal of Electrical and Computer Engineering Systems (IJECES)

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b, 31000 Osijek, Croatia
Phone: +38531224600, Fax: +38531224605
e-mail: ijeces@ferit.hr

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.
Giro account: 2390001 - 1100016777, Croatian Postal Bank

Journal is referred in:

- Scopus
- Web of Science Core Collection (Emerging Sources Citation Index - ESCI)
- Google Scholar
- CiteFactor
- Genamics
- Hrčak
- Ulrichweb
- Reaxys
- Embase
- Engineering Village

Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003
Published: quarterly
Circulation: 300

IJECES online

https://ijeces.ferit.hr

TABLE OF CONTENTS

Polarization Reconfigurable Patch Antenna Using Parasitic Elements for Sub–6 GHz Applications

Original Scientific Paper

Kanniyappan Vinayagam
Vellore Institute of Technology, School of Electronics Engineering, Vellore, Tamilnadu, India
vinayagam.k2020@vitstudent.ac.in

Rajesh Natarajan*
Vellore Institute of Technology, School of Electronics Engineering, Vellore, Tamilnadu, India
rajesh.natarajan@vit.ac.in

*Corresponding author

Abstract *– Polarization reconfigurable antenna using parasitic elements are designed for sub-6 GHz applications. A circular patch antenna is designed along with two semicircular arc - elements attached to the radiator with four diodes. By controlling the ON and OFF states of the diodes, the polarization of the antenna can be switched between LHCP and RHCP. Parasitic elements are characterized and placed around the conducting patch to enhance the gain of the antenna. The antenna exhibits a better gain of 5 dBi in both the polarization states. The prototype antenna is fabricated on a FR-4 substrate with full ground plane and tested for reflection coefficient, radiation pattern and polarization conversion ratio. The results are compared with the simulated one and they are having highest correlation between them.*

Keywords*: parasitic elements, axial ratio, polarization conversion ratio, LHCP and RHCP*

Received: July 18, 2024; Received in revised form: September 6, 2024; Accepted: October 2, 2024

1. INTRODUCTION

With the advent of adaptive and cognitive antennas, the research on reconfigurable antennas is still on with new directions. Meta-surfaces are extensively studied in literature with various forms and configurations to design reconfigurable antennas. By the same way, parasitic patches are also used for the reconfigurable antennas. This antenna can steer the primary lobe in three distinct directions by reconfiguring a different frequency. By enabling the parasitic elements to behave like a reflector or director, PIN diode switches make pattern reconfiguration easier when adjusting their electrical length [1]. It consists of four L-shaped patches, two modified trapezoid-microstrip lines, a pair of vacant-quarter feeding loops, and four grounded inverted L-shaped strips. The antenna gain is enhanced by insertion of grounded inverted strips below the patches, enhancing its bandwidth and wider CP operation [2]. For a filtering antenna design, dipole antennas are used with an asymmetric parasitic element; utilizing it's odd and even modes [3]. Metasurfaces are implemented for reflect array in [4] and performance enhancement of monopole antenna for few application-oriented implementations. A simple feeding network with 3×3 metasurface antenna designed in [5] has the capability to have LP, CP and also beam switching between 0^0 and $+30^0$. Square patch with diagonal slots on the radiator in [6], provides higher gain with the help of parasitic elements. Liquid metal-based polarizer is designed with two layers of micro fluidic channels in [7]. Circularly polarized antenna is proposed with the rectangular unit cells with truncated corners [8]. Conventional patch antenna is replaced with meta-surfaces as radiating elements in [9] gives low in band RCS. Here, operating frequency is reconfigured by loading the Varactor diodes. When the channels are empty it acts as the reflector and when the channels are filled with fluidic elements, it acts as the linear to circular polarization converter. To generate a wideband antenna consisting of one parasitic element placed between the dipole arms and two orthogonal bowties are used [10]. A broadband CP antenna is proposed with rotated parasitic patches in [11].

Non uniform meta-surfaces are embedded on the patch antenna gives wideband circular polarization in [12]. Corner truncated patch meta-surface is used to transform the linearly polarized wave into a circular polarized wave [13]. By rotating the meta-surface layer

different polarizations are obtained. Dual band polarization reconfigurable antenna designed with double layer meta-surface [14], meta-surface embedded design for low RCS antenna in [15] used mechanical rotation of the substrate to aid the reconfigurable nature. Wideband CP antenna with high gain is obtained using mushroom cells [16]. Two-layerfabry-perot cavity antenna designed with parasitic element to exhibit beam steering in [17]. He substrate of this antenna is a bi-stable composite shell, which has two states for stretched and coiled-up [18]. Physical rotation of the meta-surfaces with the PIN diodes enabled the multi band operation from the antenna. A reconfigurable antenna array is made by using frequency selective surface (FSS) with PIN diodes in [19]. Mushroom type meta-surfaces exhibit penta-polarization in [20] and wide axial bandwidth in [21] are designed with two layers. Metamaterials are utilized to improve antenna gain and reduce side lobes, adjusted for the same frequency depending on research parameters such as size, bandwidth, and gain [22]. The antenna is circularly polarized, with four radiating slots fed by the dual coaxial probe [23]. In [24], a microstrip patch antenna printed on the ground plane loaded with dumbbell meta-atoms and loaded with stubs used to attain better performance and characteristics. A single truncated corner square patch, incorporating a corner cut square slot increases design flexibility by allowing wide bandwidth in circular polarization [25]. The antenna designed with reduced size and enhanced gain in [26, 27]. The metasurface based pattern reconfigurable antenna allows beam steering in the desired directions using four different slots placed from different positions [28].

The literature offers wide range of reconfigurable implementations with multiple layers or with complex geometries. The objective of the work is to design a compact polarization reconfigurable patch antenna with minimum number of diodes. A circular patch antenna is designed with four diodes which operate on 4.7 GHz. Section 2 talk about the design of parasitic element and the integration of antenna with parasitic elements. Section 3 discussed the performance of the antenna and the last section presented the conclusion.

2. DESIGN OF ANTENNA AND PARASITIC ELEMENTS

2.1. DESIGN OF PARASITIC ELEMENTS

Set of two square shaped elements are infused like dumbbell shape to form the unit cell which is shown in Fig.1(a). The unit cell is characterized by the reflection coefficient (S11) to act as the parallel layer to the antenna element. The unit cell reflects all the incoming waves at 4.7 GHz with a bandwidth of 100 MHz. The selected geometry is an un-symmetric structure hence the TE and TM mode response has a small variation between S11 and S21 is presented in Fig.1(b). This unit cell is replicated on the radiating surface of the antenna to enhance the reflection from the antenna surface.

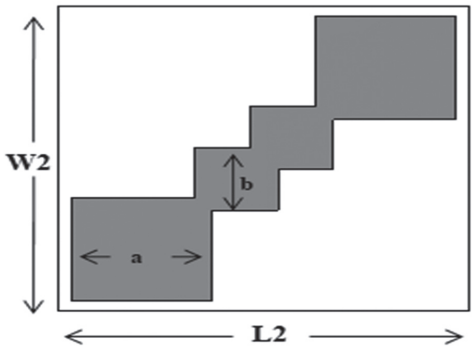


Fig.1.(a). Parasitic element with the dimensions are a=3.5 mm, b= 1.5 mm, W2= 10.5=L2

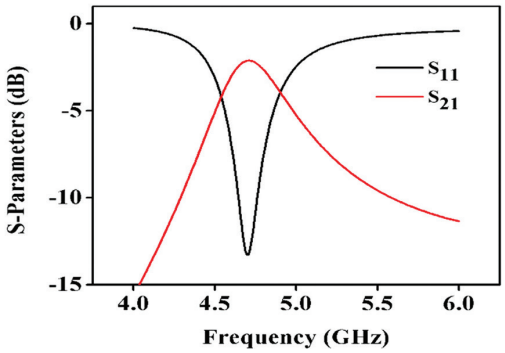
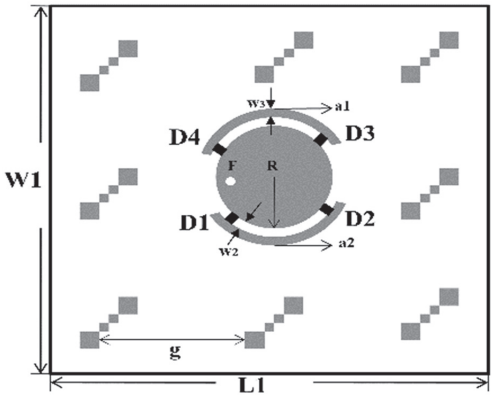
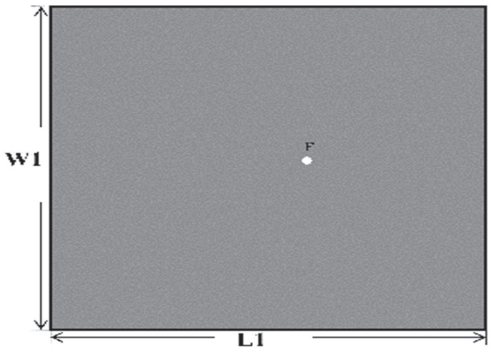


Fig.1.(b). Unit cell Characteristics (parasitic element)

2.2. PROPOSED ANTENNA DESIGN



(a)



(b)

Fig.2. (a) Front view (L1=W1=60mm, w2=1.4mm, w3=1mm, g=20mm) **(b)** Back view

The antenna design starts with the circular patch with full ground plane. The dimensions of the antenna are 60 mm* 60 mm * 1.6 mm. Coaxial feed is used to excite the antenna and it resonates at 5 GHz. The location of the feed is chosen such a way that it induces the orthogonal modes. To enable the current rotation on either side, two arc shaped conductors are placed along the circular patch. Four diodes are placed on each corner of the arc to establish the contact between the patch and arc. The diodes are named as D1, D2, D3 and D4 which are marked on Fig. 2.

To improve the gain of the antenna, parasitic elements are arranged on the radiating plane with the inter element spacing (g) = 20 mm. This improved the gain of the antenna further. The S_{11} of the antenna undergoes a small shift due to the loading of parasitic elements.

2.3. WORKING PRINCIPLE OF THE ANTENNA

Surface current is observed on the radiator for different diode states. When D1 and D2 diodes are ON, the diagonally opposite arc radiators are connected with the disc radiator. Current rotation for different time incident is presented for Phi= 0^0, 90^0, 180^0 and 270^0 in Fig. 3 (a) & (b). The rotation is towards the right hand side which ensures the RHCP operation of the antenna. The surface current direction is marked on the circular patch for better understanding. When D3 and D4 diodes are ON, then the current rotation is towards the left hand side which ensures LHCP of the antenna.

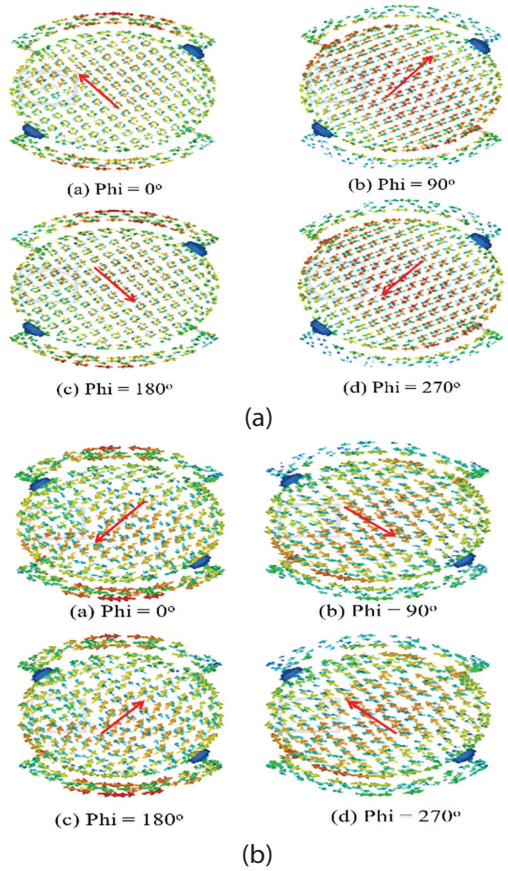
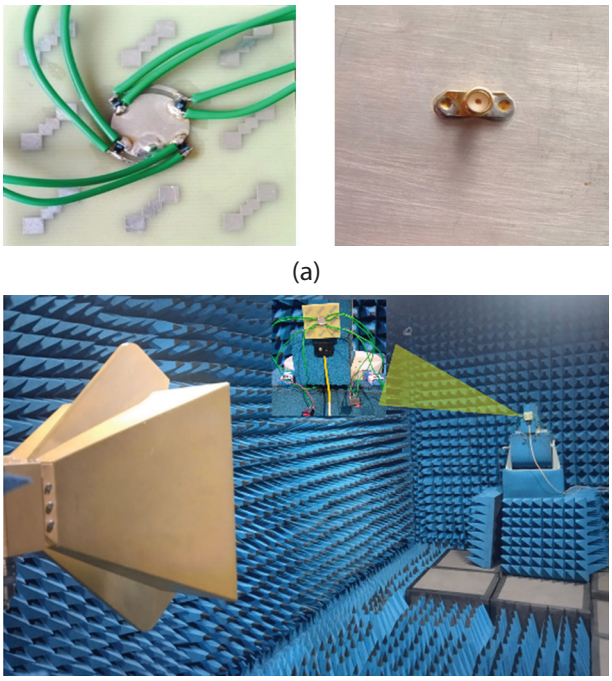


Fig. 3. (a) RHCP Rotation **(b)** LHCP Rotation

3. PERFORMANCE ANALYSIS AND RESULTS DISCUSSION

Prototype antenna is fabricated and tested for different metrics as reflection coefficient, radiation pattern and Polarization conversion ratio (PCR) values. Fabricated and chamber testing antenna is shown in Fig.4 (a) and (b). The diodes are biased and connected to the outer arc as shown in Fig. 4(a).



(a)

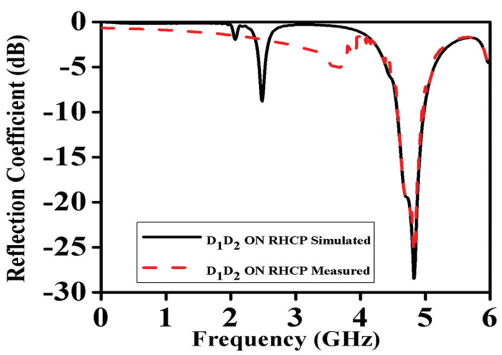
(b)

Fig.4. (a) Fabricated antenna
(b) Measurement setup at Champer

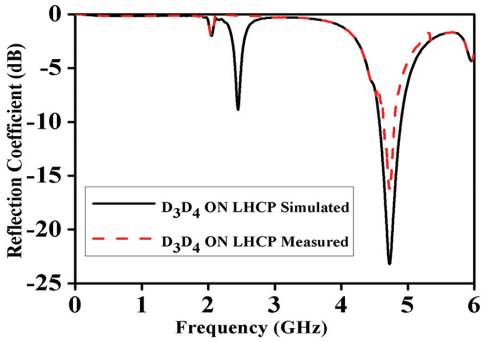
3.1. REFLECTION COEFFICIENT

Reflection coefficient characteristics of the antenna is taken from simulated and measured in both operating modes. The results are presented in Fig. 5(a) and 5(b) respectively. The antenna resonates at 4.7 GHz with bandwidth ranging from 4.47 GHz to 4.89 GHz. The S11 for the antenna for RHCP and LHCP shows high correlation with the simulated results.

The simulated VSWR for both LHCP and RHCP are depicted in Fig.6.



(a)



(b)

Fig. 5. Measured vs simulated S11 Characteristics

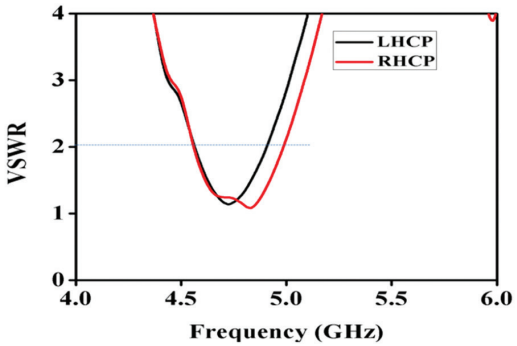


Fig. 6. Simulated VSWR Characteristics

3.2. POLARIZATION CONVERSION RATIO:

Polarization conversion ratio (PCR) is one of the metric to verify the suitability of polarization converters. The parasitic elements are verified for this metrics and are calculated from equation (1).

$$PCR = \frac{|R_{YX}|^2}{|R_{YX}|^2 + |R_{XX}|^2} \quad (1)$$

where $|RYX|$ - magnitute of cross-correlation coeffiecient between Y and X.

$|RXX|$- magnitute of auto-correlation coeffiecient of X

The co and cross polarization results are obtained and the PCR is calculated and presented in Fig. 7. A very good conversion will have the value of 1 or nearer to that. The proposed parasitic elements are exhibiting the PCR of 0.9 in the working band. This is suitable for this antenna design and conveniently assisting for the circular polarizations from the antenna.

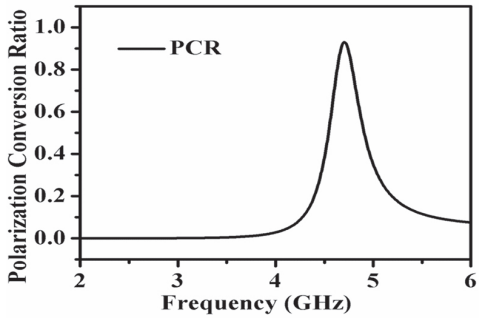
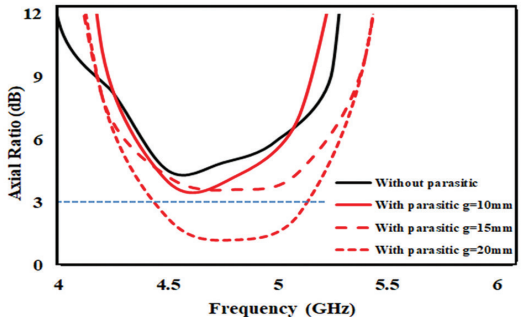


Fig. 7. Calculated (PCR)

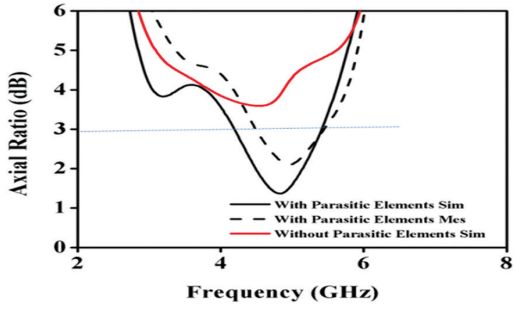
3.3. AXIAL RATIO

When parasitic elements are added to the antenna plane, they helped in improving the axial ratio through coupling Effects. Parasitic elements are strategically placed to create additional coupling between the elements of the antenna, which can help balance the electric field components. This balance is crucial for achieving good circular polarization. The parasitic elements can introduce phase shifts that compensate for any imbalances in the original antenna design. This helps in aligning the orthogonal field components more effectively, leading to a lower axial ratio.

Axial ratio of the antenna is presented for different inter element spacing (g). The radiator without the parasitic element gives the axial ratio greater than 5 dB and the plot approached less than 3 dB for the value of g=20 mm. The parasitic elements are acting as polarization convertors and resulted in circular polarization with better axial ratio. It is clearly understood from Fig. 8 (a) and (b). that the 3-dB axial ratio of the antenna covers the impedance bandwidth of 320 MHz which is sufficient for sub 6 GHz applications. Measured gain and efficiency of the antenna is presented in Fig. 9. The gain of the antenna is improved by adding more number of parasitic elements with optimized inter element spacing. The maximum gain is observed to be 8.2 dBi compared with the references listed in Table 1. Form the table, the gain is maximum when the antenna size is larger and the proposed antenna is better than the references [6] and [11] in terms of size and gain. Radiation pattern of the antenna is measured in the working frequency and are stable irrespective of the biasing conditions which is shown in Fig. 10.



(a)



(b)

Fig. 8. (a) Variations of parasitic element with different gap **(b)** Measured Axial ratio of the antenna g=20 mm

Gain improvement of the antenna is due to the parasitic elements primarily because these elements can help direct and focus the radiation pattern more effectively. Parasitic elements act as the reflectors, guiding the electromagnetic waves in a more focused direction. This increased the directivity results in higher gain because most of the radiated power is concentrated in the desired direction. Properly designed parasitic elements can enhance the overall radiation efficiency by reducing losses due to impedance mismatches or unwanted radiation in non-preferred directions.

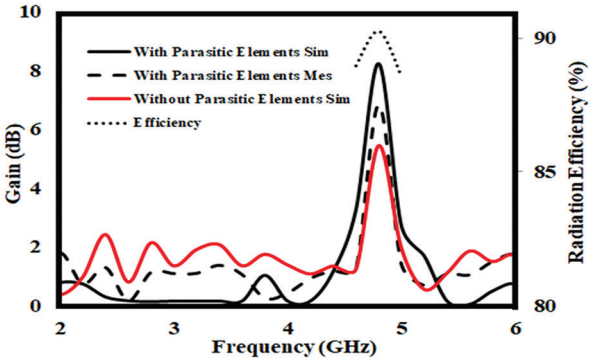
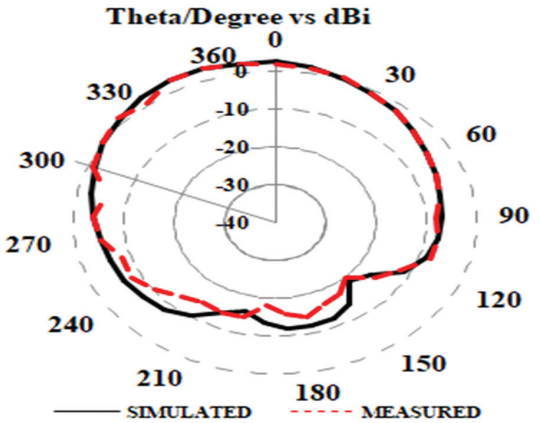
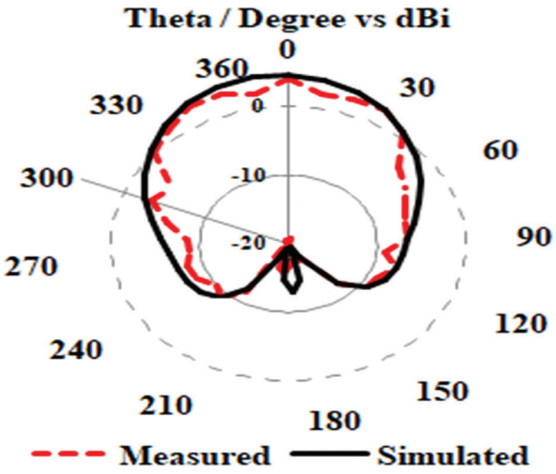


Fig. 9. Measured antenna gain and efficiency



(a)



(b)

Fig.10. Radiation Pattern **(a)** 4.8 GHz (During LHCP) **(b)** 4.8 GHz (During RHCP)

Table 1. Comparison of Proposed Work with Existing Antenna from the Literature

S.No	Antenna Size (mm)	Frequency (GHz)	No of Printing Layer	No of pol.	Gain (dB)
[1]	38×40	2.42, 2.43, 3.5, 3.29	1	NA	5.3, 3.82, 2.77, 2.2
[2]	180×180	1.85,2.8	1	2	10.8
[3]	140×140	1.75-2	2	2	8.0
[6]	40×39	4.52-7.42	2	1	6.5
[8]	82×82	1.95-3.85	1	2	8.1
[10]	130×130	2.32-2.95	2	2	8.0
[11]	42×42	3.75-6.67	1	2	7.0
[22]	50×50	3.5	1	NA	6
[25]	39.4×39.4	5.6,6.2	1	1	6.6
[27]	78×78	3.5	1	2	7.5
[28]	55×55	4.9,5.2	1	NA	3.9,5.6
Pro. Work	60×60	4.8	1	2	8.2

4. CONCLUSION

The paper presented the parasitic elements loaded patch antenna for sub 6 GHz application. Polarization reconfigurable states of the antenna are verified with 4 PIN diodes and the antenna exhibits LHCP and RHCP rotation. The axial ratio bandwidth covered the C band between 4.5 GHz and 4.8 GHz. Working principle of the antenna is validated through current distribution and the measured results are in good agreement with the simulated results. The inclusion of parasitic elements improved the gain of the antenna significantly.

5. REFERENCES

[1] K. Karthika, K. Kavitha, "Design and development of parasitic elements loaded quadband frequency and pattern reconfigurable antenna", International Journal of RF and Microwave Computer-Aided Engineering, 2023, p. 4034241.

[2] D. Yang, L. Hao, L. Wang, "A Compact Broad Circularly Polarized Cross-Dipole Antenna with Grounded Parasitic Elements", International Journal of Antennas and Propagation, 2023, p. 9359671.

[3] C. F. Ding, Y. Zeng, M. Yu. "Compact Dual-Polarized Filtering Dipole Antenna by Using Asymmetric Parasitic Element", IEEE Transactions on Antennas and Propagation, Vol. 70, No. 7, 2023, pp. 5941-5946.

[4] N. Zhang et al. "A dual-polarized reconfigurable reflect array antenna based on dual-channel programmable metasurface", IEEE Transactions on Antennas and Propagation, Vol. 70, No. 9, 2022, pp. 7403-7412.

[5] W. Li, Y. M. Wang, Y. Hei, B. Li, X. Shi, "A compact low-profile reconfigurable metasurface antenna with polarization and pattern diversities", IEEE An-

tennas and Wireless Propagation Letters, Vol. 20, No. 7, 2021, pp. 1170-1174.

[6] N. Hussain, H. H. Tran, T. T. Le, "Single-layer wideband high-gain circularly polarized patch antenna with parasitic elements", AEU-International Journal of Electronics and Communications, Vol. 113, 2020, p. 152992.

[7] O. M. Sanusi, Y. Wang, L. Roy, "Reconfigurable polarization converter using liquid metal based metasurface", IEEE Transactions on Antennas and Propagation, Vol. 70, No. 4, 2021, pp. 2801-2810.

[8] T. Nakamura, T. Fukusako, "Broadband design of circularly polarized microstrip patch antenna using artificial ground structure with rectangular unit cells", IEEE Transactions on Antennas and Propagation, Vol. 59, No. 6, 2011, pp. 2103-2110.

[9] H. Yang et al. "Low in-band-RCS antennas based on anisotropic metasurface using a novel integration method", IEEE Transactions on Antennas and Propagation, Vol. 69, No. 3, 2020, pp. 1239-1248.

[10] H. H. Tran, I. Park, T. K. Nguyen, "Circularly polarized bandwidth-enhanced crossed dipole antenna with a simple single parasitic element", IEEE Antennas and Wireless Propagation Letters, Vol. 16, 2017, pp. 1776-1779.

[11] J. Wu, Y. Yin, Z. Wang, R. Lian, "Broadband circularly polarized patch antenna with parasitic strips", IEEE Antennas and Wireless Propagation Letters, Vol. 14, 2014, pp. 559-562.

[12] H. H. Tran, C. D. Bui, N. Nguyen-Trong, T. K. Nguyen, "A wideband non-uniform metasurface-based circularly polarized reconfigurable antenna", IEEE Access, Vol. 9, 2021, pp. 42325-42332.

[13] C. H. S. Nkimbeng, H. Wang, I. Park, "Low-profile wideband unidirectional circularly polarized metasurface-based bowtie slot antenna", IEEE Access, Vol. 9, 2021, pp. 134743-134752.

[14] X. Chen, Y. Zhao, "Dual-band polarization and frequency reconfigurable antenna using double layer metasurface", IEEE Transactions on Antennas and Propagation, Vol. 95, No. 1, 2018, pp. 82-87.

[15] K. Kandasamy, B. Majumder, J. Mukherjee, K. P. Ray, "Low-RCS and polarization- reconfigurable antenna using cross-slot-based metasurface", IEEE Antennas and Wireless Propagation Letters, Vol. 14, 2015, pp. 1638-1641.

[16] Q. Chen, H. Zhang, Y. J. Shao, T. Zhong, "Bandwidth and gain improvement of an L-shaped slot antenna with metamaterial loading", IEEE Antennas and Wireless Propagation Letters, Vol. 17, No. 8, 2018, pp. 1411-1415.

[17] P. Xie, G. Wang, T. Cai, H. Li, J. Liang, "Novel fabry-pérot cavity antenna with enhanced beam steering property using reconfigurable meta-surface", Applied Physics A, Vol. 123, 2017, pp. 1-6.

[18] Y. Zhang, S. Lin, Y. Li, J. Cui, F. Dai, J. Jiao, A. Denisov, "Wideband pattern-and polarization-reconfigurable antenna based on bistable composite cylindrical shells", IEEE Access, Vol. 8, 2020, pp. 66777-66787.

[19] W. Li, Y. Wang, S. Sun, X. Shi, "An FSS-backed reflection/transmission reconfigurable array antenna", IEEE Access, Vol. 8, 2020, pp. 23904-23911.

[20] P. Liu, W. Jiang, S. Sun, Y. Xi, S. Gong, "Broadband and low-profile penta-polarization reconfigurable metamaterial antenna", IEEE Access, Vol. 8, 2020, pp. 21823-21831.

[21] S. Liu, D. Yang, J. Pan, "A low-profile circularly polarized metasurface antenna with wide axial-ratio beamwidth", IEEE Antennas and Wireless Propagation Letters, Vol. 18, No. 7, 2019, pp. 1438-1442.

[22] M. Al-Abbasi, T. Abdul Latef, "Wideband circularly polarized fractal antenna with SSRR metasurface for 5G applications", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 1, 2024, pp. 89-98.

[23] T. Apparao, G. Karunakar, "A Four Slot Dual Feed and Dual Band Reconfigurable Antenna for Fixed Satellite Service Applications", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 10, 2023, pp. 1165-1171.

[24] S. Angadi, K. Viswanadha, R. Chinthaginjala C. Dhanamjayulu, K. Tai-Hoon, S. Kumar, "Meta-atom loaded circularly polarized triple band patch antenna for Wi-Fi, ISM and X-band communications", Heliyon, Vol. 10, No. 7, 2024.

[25] Y. Chi, "A new wideband CP antenna with a single-layer metasurface", Electromagnetics, Vol. 42, No. 8, 2022, pp. 616-623.

[26] S. V. Pande, D. P. Patil, "Frequency-reconfigurable variode enabled metasurface reflector loaded antenna", Physica Scripta, Vol. 98, No. 12, 2023, p. 125515.

[27] H. L. Zhu, S. W. Cheung, X. H. Liu, T. I. Yuk, "Design of polarization reconfigurable antenna using metasurface", IEEE Transactions on Antennas and Propagation, Vol. 62, No. 6, 2014, pp. 2891-2898.

[28] H. H. Tran, T. T. Le, "A metasurface based low-profile reconfigurable antenna with pattern diversity", AEU-International Journal of Electronics and Communications, Vol. 115, 2020, p. 153037.

Comparative Analysis of Deepfake Detection Models on Diverse GAN-Generated Images

Original Scientific Paper

Medha Wyawahare*
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
medha.wyawahare@vit.edu

Vrinda Parkhi
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
vrinda.parkhi@vit.edu

Siddharth Bhorge
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
siddharth.bhorge@vit.edu

Mayank Jha
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
mayank.jha20@vit.edu

Milind Rane
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
milind.rane@vit.edu

Narendra Muhal
Vishwakarma Institute of Technology,
Department of Electronics and Telecommunication
Pune, Maharashtra, India
narendra.muhal20@vit.edu

*Corresponding author

Abstract *– Advancement in Artificial intelligence has resulted in evolvement of various Deepfake generation methods. This subsequently leads to spread of fake information which needs to be restricted. Deepfake detection methods offer solution to this problem. However, a particular Deepfake detection method which gives best results for a set of Deepfake images (generated by a particular generation method) fails to detect another set of Deepfake images (generated by another method). In this work various Deepfake detection methods were tested for their suitability to decipher Deepfake images generated by various generation methods.*

We have used VGG16, ResNet50, VGG19, and MobileNetV2 for deepfake detection and pre-trained models of StyleGAN2, StyleGAN3, and ProGAN for fake generation. The training dataset comprised of 200000 images, 50 % of which were real and 50% were fake. The best performing Deepfake detection model was VGG19 with more than 96 percent accuracy for StyleGAN2, StyleGAN3, and ProGAN-generated fakes.

Keywords*: CNN, GAN, VGG19, StyleGAN3, Deepfake*

Received: March 22, 2024; Received in revised form: August 20, 2024; Accepted: August 21, 2024

1. INTRODUCTION

The deepfake image synthesis and detection field has attracted significant research due to the convergence of computer vision and artificial intelligence. This multidisciplinary area, which attracts academicians, researchers and business experts, focuses on the automated creation and recognition of modified visual content. Deepfake image generation and detection have practical repercussions in a variety of fields, from digital forensics and content verification to maintaining user confidence in computer-human interactions. Furthermore, it has the power to fundamentally alter how society interacts with visual information. The creation of models that can not only create but also recognize real images from altered ones is at the core of this endeavor. Similar to how image captioning seeks to describe scenes, the main goal in this case is to create material that fools or imitates reality. This technology offers inventive ways to create digital content but also presents difficulties that call for strict safeguards against misuse and false information.

Modern deep learning techniques serve as inspiration for the architecture supporting deepfake image production and detection. Modern methods usually

use the encoder-decoder paradigm, which is appropriate for both facets of this topic. To encode the source's unique features, the encoder must convert them into little feature vectors. The decoder then makes use of these vectors to create them or determine their legitimacy. The core of the encoder component is a convolutional neural network (CNN). The use of well-known CNN architectures like VGG, ResNet, and MobileNet, is common in the encoder. The model's capacity to spot subtle patterns in real and altered images is aided by this larger viewpoint.

The use of four specialized detection models demonstrates our commitment to excellence in deepfake image production and detection. Every model has been painstakingly designed to handle particular aspects of deepfake identification, improving the model's overall accuracy and adaptability. We explore the world of Generative Adversarial Networks (GANs), a powerful method for producing deepfake content, as a complement to these detection attempts. We seek to advance the authenticity and realism of the created images by utilizing adversarial training, adding to the continuing arms race between creation and detection.

In this paper, we conduct a comprehensive evaluation of eight different CNN models, such as VGG16, ResNet50, VGG19, and Xception, among others, for deepfake detection. Initially, we train these CNN models on the OpenForensics dataset, a widely used benchmark dataset in the field of deepfake detection. To evaluate these models performance and generalizability, we test them using a recently created dataset that contains a wide variety of deepfakes. Furthermore, to enhance the robustness of our evaluation, we augment the OpenForensics dataset by incorporating our own generated data, thereby expanding the dataset's diversity and realism. Subsequently, we retrain the CNN models on this augmented dataset, leveraging the enriched data to improve the model's performance.

Finally, we rigorously evaluate the trained models by testing them on three distinct sets of GAN-generated data: ProGAN, StyleGAN2, and StyleGAN3 [1]. The newly generated dataset from these GANs is available on Kaggle for public access. We hope to shed light on how well CNN models perform in identifying deepfakes using a variety of datasets and GAN architectures by using this thorough approach.

2. LITERATURE REVIEW

In 2019, Yadav et al. [2] put forth that deepfake images are man-made media, especially edited videos or photos, produced by cutting-edge machine learning algorithms that can accurately replicate real human expressions and activities. They examine many strategies, from conventional GAN-based techniques to more complex variants, like conditional GANs and cycle-consistent GANs. To create extremely realistic facial forgeries, the proposed deepfake generation model uses a conditional GAN architecture, where the generator is conditioned on both the input and the target identity. To remove the potential misuse of deepfake, they added watermarks on the deepfakes. Sanjana et al. [3] gave a thorough analysis of the current deepfake detection methods to stop the spread of false information and protect the integrity of multimedia content. Detection techniques like CNNs and GANs can spot deepfake face swapping, in which one person's face is swapped out for the face of another. To increase the effectiveness of deepfake detection, transfer learning techniques that use pre-trained models for related tasks (e.g., facial recognition) are used.

Malik et al. [4] provided a thorough analysis of the various techniques and procedures employed for deepfake detection. They look at a variety of strategies, computer vision approaches, and deep learning-based solutions in particular. The survey examines the advantages and disadvantages of various detection techniques and covers both text-based and video-based deepfake. Rana et al. [5] and Paul et al. [6] show significant progress in developing robust deepfake detectors capable of fending off ever-more complex manipulation techniques using GANs for adversarial training. The study produces encouraging results in audio-based deepfake detection using recurrent neural networks, concentrating on minute acoustic artifacts created during speech synthesis to distinguish altered audio from real recordings.

Nguyen et al. [7] investigated various deep learning architectures, such as GANs, autoencoders, and others, that are used to produce deepfake content. To maintain visual integrity, autoencoders, a form of unsupervised deep learning model, have been used for deepfake production. By training on small samples of recently emerging deepfake content, one-shot learning algorithms have demonstrated potential for identifying unique deepfake variants. Datasets like Face Forensics++ and the deepfake Detection Dataset (DFDC) have been significant in advancing research and testing performance in the deepfake detection field. Shen et al. [8] have investigated the technical aspects of how GANs are utilized to create deepfakes, including training the networks, selecting suitable datasets, and fine-tuning the models to produce realistic results, which are probably covered fully in the study. To improve convergence and generation quality, the Wasserstein GAN algorithm variation with a deep convolutional architecture is used to train the generator network. They use the Structural Similarity Index (SSIM) metric and Peak Signal-to-Noise Ratio (PSNR) to objectively analyze the similarity between the created content and the ground truth data to assess the performance of our deepfake generation. Giudice et al. [9] outline a technique to spot abnormalities in the Discrete Cosine Transform (DCT) domain of GAN-generated. This work focuses on preventing deep fakes. The DCT is frequently used for image compression, including JPEG encoding, and GANs

also employ it for creation. By examining anomalies in the DCT coefficients of images created using GANs, the authors of this research take a fresh approach to deepfake identification. They make a distinction between real content and content that has been altered by using statistical metrics and machine learning classifiers to identify specific DCT artifacts connected to deepfake image.

Shad et al. [10] conducted a comparative analysis of the performance of eight CNN architectures. They have used images from the Flickr dataset for training the models. Fake images for training were generated using StyleGAN. They evaluated these models on five different evaluation metrics, such as accuracy, precision, recall, etc. VGGFace and ResNet50 performed best with an accuracy of 99% and 97%, respectively. Saxena et al. [11] gave a thorough introduction to GANs, noting the difficulties in training and using them, suggesting different ways to solve these problems, and providing suggestions for possible future research paths. The study contributes to a deeper knowledge of GANs and acts as an invaluable resource for scholars and practitioners in the field of artificial intelligence by carefully examining existing research and methodologies.

Ali et al. [12] tested the generalization of the fake face detection methods. Two types of fake face detection methods have been tested in this paper. The first is texture-based Local Binary Patterns (LBP), and the second is using different CNN architectures such as Alexnet, VGG19, ResNet50, etc. These methods are tested on known and unknown data, and the results show that their performance drops for the unknowns. These results indicate the lack of transferability of the learned classifiers to the general face-forgery classification cases. Patashnik et al. [13] proposed StyleCLIP, a powerful framework for manipulating s generated by StyleGAN using natural language descriptions. By aligning the CLIP model's textual embeddings with StyleGAN's latent space, users can apply targeted modifications to generated text simply by providing descriptive text. StyleCLIP allows users to create diverse and specific visual outputs, offering an exciting approach to interactive and intuitive synthesis and manipulation.

Kumar et al. [14] reviewed various techniques for implementing and detecting deepfake images, focusing on Deep Convolution-based GAN models. A comparative analysis of the proposed GAN model with existing models is performed using parameters such as Inception Score (IS) and Fréchet Inception Distance (FID). Deepfake images present a substantial threat to biometric security and facilitate counterfeiting and fraudulent activities.

Tiwari et al. [15] discuss the use of GANs in creating highly realistic deepfakes and their role in both generating and detecting fake content through discriminator networks. CNNs are highlighted for their effectiveness in classification and detecting subtle anomalies in images and videos, making them a primary method for deepfake detection. RNNs and LSTMs are noted for their capability to handle sequential data, making them suitable for analyzing video content and identifying temporal inconsistencies indicative of deepfakes. Recent advancements in attention mechanisms and transformers show promise for improving deepfake detection accuracy through sophisticated feature extraction and analysis. The authors evaluated deepfake detection models using the Inception Score (IS) and Fréchet Inception Distance (FID) to quantify the quality of the generated data and the effectiveness of detection algorithms.

Nowroozi et al. [16] describe the application of GAN-based CNN models for deepfake detection, highlighting their effectiveness in distinguishing real from artificial faces. The effectiveness of the CNN models, which are Cross-Co-Net and Co-Net, was compared to alternative approaches. It showed superior accuracy, which underscores the robustness of combining GAN-generated data with CNN for deepfake.

Sharma et al. [17] presented the effectiveness of GANs for deepfake detection, leveraging a GAN-based CNN model. Using the Indian actor's dataset, demonstrates how GANs may be used to expand training datasets, hence improving the robustness of the model. The suggested approach outperforms existing techniques and demonstrates its potential for useful applications in digital forensics and image recognition. Sergi et al. [18] investigated the human ability to identify deepfakes created using the StyleGAN2 algorithm. Three intervention tactics were tested for their efficacy in detecting deepfakes through an online poll that attracted 280 participants. Following the evaluation of twenty images, the participant's accuracy score ranged from 60% to 64% depending on the situation, indicating that deepfake images produced by StyleGAN2 are difficult for humans to detect. Notably, interventions did not significantly improve detection accuracy. The findings highlight the difficulty in detecting deepfake images and underscore the urgent need for enhanced detection methods and public awareness.

3. RESEARCH GAP

There has been a lot of intensive research and development in this field in the last few years as a result of the rise of Artificial Intelligence (AI) and Deep Learning (DL) technologies. In the literature that we reviewed, there are several gaps in the current state of deepfake detection research. While the majority of research to date has focused on GAN-based methods and specific designs, there are noticeably few comprehensive comparative studies that look at a larger variety of GAN variants. Moreover, reliance on established datasets, such as Face Forensics++ and DFDC, restricts the understanding of model effectiveness across diverse data sources, indicating a need for research that examines model's performance on more varied and less curated datasets.

Furthermore, the challenge of generalization persists, with many models demonstrating effectiveness on known data but struggling to maintain accuracy in the face of new deepfake techniques or unknown data. The field lacks sufficient exploration of the robustness of detection models against adversarial attacks, highlighting a critical gap in ensuring the reliability of detection methods in real-world scenarios. The scarcity of work comparing different detection models on fakes generated using different GANs is evident.

An organized research for identifying the most robust detection algorithm capable of performing well on all types of deepfakes across various GAN architectures is essential. The lack of established evaluation metrics and benchmarks makes it difficult to compare detection models consistently, which emphasizes the necessity of research projects targeted at creating accurate and consistent evaluations of model performance.

4. PROBLEM STATEMENT

The objectives of our research are to

- Generate deepfake images using three different GANs, so that we have diverse fake images to test our detection methods.
- Train eight different CNN models on the dataset to detect fake images, which will be our detection models.
- Compare the performance of deepfake detection models when they are tested on the diverse fake images that are generated different GANs in order to suggest the best performing deepfake detection method.

5. METHODOLOGY

Fig. 3 shows the general preprocessing and detection flow that the model is going through.

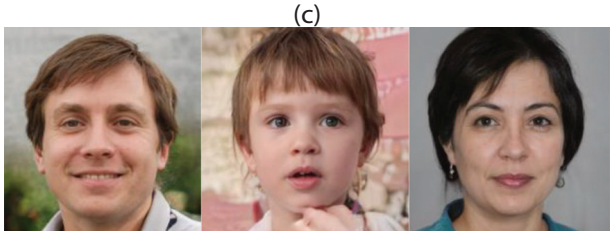
5.1. DATASET

The dataset used comprised of both real and fake images Fig. 1 shows sample images, real and fake, along with the fake images generated using ProGAN, StyleGAN2, and StyleGAN3.

(a)

(b)

(c)



(d)

(e)

Fig. 1. Dataset containing (a) Real , Fake from (b) OpenForensics, (c) ProGAN, (d) StyleGAN2, and (e) StyleGAN3. [19]

We used the Openforensics dataset [19] which is an open dataset and contains approximately 200,000 images. It was split in the ratio 70:20:10 (70% training, 20% validation, and 10% testing). The quantity of images used in the datasets for training, testing, and validation is displayed in Table 1.

A dataset of 15,000 fake images was generated from the three pre-trained GAN models, five thousand from each. We added 5,000 and 1,400 fake generated images from each GAN model for training, testing and validation. This increased the robustness, diversity, and overall quality of the dataset before it was used for training.

Table 1. The Dataset Utilized

Datasets	Number of images		
	Train	Validation	Test
Real	70001	19787	5413
Fake	70001	19641	5492

5.2. GENERATION OF DEEPFAKES

Generative Adversarial networks (GANs) are mostly used to generate fake media. A GAN consists of two parts. The first is the generator, which generates the fakes. It starts with a random vector and keeps improving until it generates an image of the desired quality. The discriminator in the second section determines whether the data produced by the generator is real or fake based on real training data. If the discriminator correctly classifies the generator's fake as fake, then the generator updates its model weights to generate better fakes, and

if the discriminator fails to recognize the fake of the generator, then the discriminator updates its model weights to better distinguish between real and fake. Both the generator and discriminator keep updating their models in a loop until the generator can generate fake images good enough to fool the discriminator. Fig. 2 depicts the GAN architecture nicely in a pictorial manner. It shows how the two parts work together, as mentioned above.

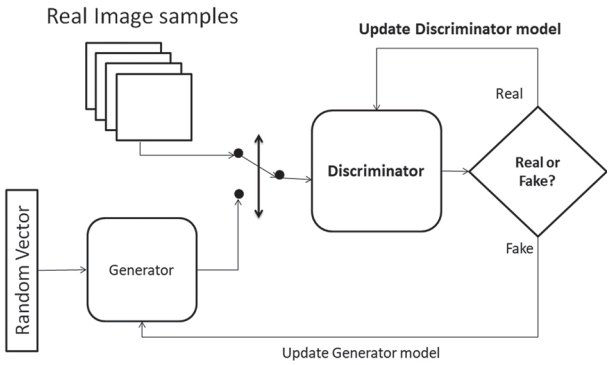


Fig. 2. Block diagram of GAN

For generating fakes, we used three types of pre-trained GAN models: StyleGAN2, StyleGAN3, and ProGAN. A total of 5000 tests were generated for each GAN model to test its detection methods. ProGAN, short for Progressive GAN, was trained on the 'CelebA' dataset and produced images with a resolution of 128x128 pixels [20]. Its progressive training approach starts with low-resolution and gradually increases the resolution, allowing it to capture fine details as it progresses.

In contrast, StyleGAN2 and StyleGAN3 are both high-resolution GANs. With a 256x256 model, they were trained on the 'FFHQ' dataset [21], which includes human faces and is known for generating exceptionally high-quality images. Since SyleGAN3 is advancement over StyleGAN2, it generated the best fake images of them all.

Algorithm 1 outlines the process of generating fake images using pre-trained GAN models. Initially, the algorithm loads the pre-trained GAN model from a specified file and extracts the generator network responsible for generation. Afterward, it sets parameters such as the number of fakes to generate and the truncation factor for controlling quality. Through a loop, the algorithm generates each fake image by creating a random latent space vector, feeding it into the generator network, and converting the output into a recognizable format. These generated fake images are then saved to a designated directory. By systematically iterating through these steps, the algorithm efficiently produces a set of fake images, leveraging the capabilities of pre-trained GAN models.

Algorithm 1 - Deepfake generation using GAN

Input:

- Pretrained model
- Truncation factor (truncation_psi) or latent dimension for controlling quality

Output:

- Fake generated images stored in specified directory.

Load Pretrained GAN Model:

- Load the pre-trained GAN model from the specified file.
- Extract the generator network (G) from the loaded model.

Create Output Directory

Generate fake images:

- Loop for each :

a. Generate a random latent space vector using torch. randn.

b. Generate an using the generator network (G) with the specific latent space and conditioning.

c. Convert the PyTorch tensor to a PIL .

d. Save the generated in the output directory with a unique filename.

End

5.3. DETECTION OF DEEPFAKES

CNN models are frequently used for detection tasks and usually use an encoder-decoder design. The CNN encoder creates a condensed feature vector after processing the inputs. The desired output is then produced by a CNN decoder using this feature representation. In this system, a CNN model is used for training the datasets and has the best accuracy.

The goal is to ascertain the relative performance of each model in identifying deepfake content. Models such as ResNet50V2, DenseNet121, VGG16, VGG19, InceptionNetV3, InceptionResNetV2, Xception, and MobileNetV2 are being examined in greater detail. In this manner, we may truly learn about their distinct advantages and disadvantages in terms of spotting deepfakes.

We may choose the model or combination of models that works best for our deepfake detection task by evaluating each model's accuracy independently. By using this technique, we can improve the deepfake detection system and make it more dependable and capable of handling the rapidly changing deepfake technology landscape.

The basis for constructing and optimizing the eight different CNNs is our training dataset, which consists of more than 140,000 images.

Hyperparameter and Training Settings:

- Learning Rate: 0.0001

- Activation Method: ReLu
- Optimizer: Adam optimizer
- Batch Size: 64
- Number of Epochs: 10

The model was trained with a learning rate of 0.0001 and the Adam optimizer, which combines the benefits of AdaGrad and RMSProp. With a batch size of 64 to fit GPU memory, the model was trained for 10 epochs to balance training time and performance.

After the training phase, we use a testing dataset of about 10,000 images to thoroughly evaluate the model's performance. Each model is tasked with determining whether a given image is real or fake throughout this review. After testing the model, we predict whether it is real or fake and then calculate the accuracy of the model.

Algorithm 2 outlines the steps involved in building, training, and evaluating a deepfake detection model using a generic CNN architecture. The flexibility of using CNN allows for customization based on specific requirements and facilitates the development of an effective deepfake detection system.

Algorithm 2: Deepfake Detection using CNN

Input:

- datasets for training, validation, and testing (real and fake images)
- Hyperparameters for the CNN model
- Number of training iterations

Output:

- Trained CNN model

Start

- Import necessary libraries and modules.
- Set the base path for the dataset.

Prepare the Dataset

- Load and preprocess the training, validation, and testing datasets.
- Visualize a sample of s from the training set.

Build the Model

- Define the architecture for the CNN model.
- Utilizing the Adam optimizer and categorical cross-entropy loss, compile the model.

Define Callback for Evaluation

- Create a custom callback (Prediction Callback) to evaluate the model on the validation set after each epoch.

Train the Model

- Set the number of training steps and validation steps based on batch size and dataset size.
- Train the model using the training and validation datasets.
- Utilize the custom callback for evaluating the model's performance on the validation set.

Save the Model

- Save the trained CNN model for future use.

End

Fig.3. shows the block diagram of our detection model, where it shows how we train the model and then preprocess the dataset, after preprocessing the model, it is trained on the different CNN architectures. After training the model has been exported and tested on the test dataset which consists of 10,000 images containing both real and fake. Then the accuracy of the model has been calculated.

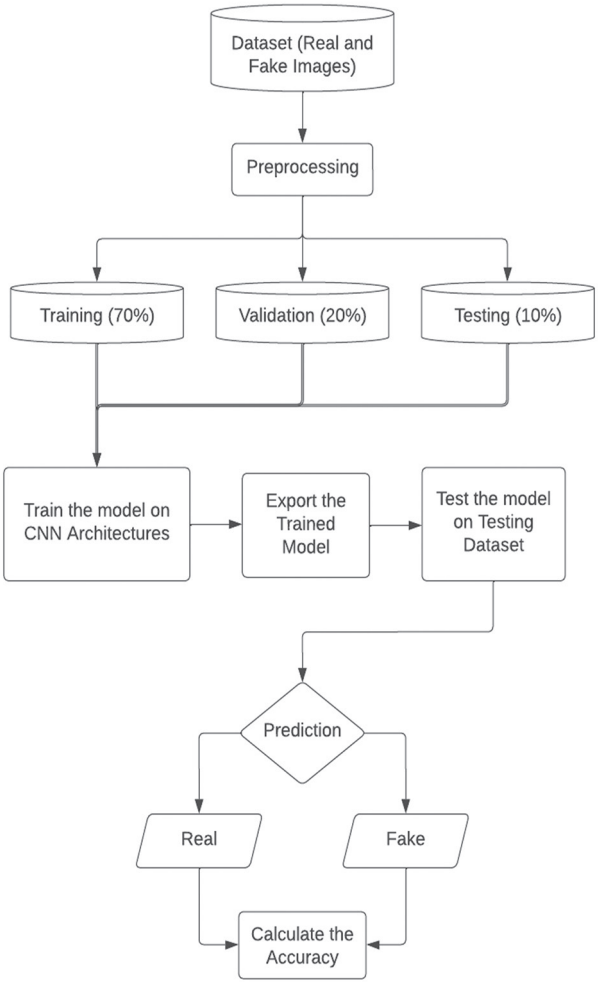


Fig. 3. Block diagram of detection model

Our approach to deepfake detection follows a structured and well-thought-out flow. It all starts with the data collection of both real and fake images that form the basis of our system. To make this data useful, we take a step called preprocessing, where we divide it

into three key parts: the training set, the validation set, and the test set. We distribute them in a balanced way, with 70% for training, 20% for validation, and 10% for testing. CNNs are an effective technique that we utilize to train the model using the training dataset.

We used exported models for deepfake detection in the testing set. This indicated the true effectiveness of our deepfake detecting algorithm. It served as performance evaluation of the system, demonstrating its dependability and efficiency in exposing misleading material across a range of contexts.

6. RESULT AND DISCUSSION

Training and testing of the models was done on cloud infrastructure. It featured dual Intel Xeon Silver 4114 CPUs with 40 cores, 128GB of DDR4-2666 ECC Memory, and an Nvidia Tesla V100 GPU with 32GB VRAM. With 4TB of HDD storage and a 100 Mb/s Ethernet interface, it was well-equipped for demanding Deep learning tasks.

A dataset containing 200,000 distinct real and fake images were used to train the model. Random images were fed into the testing process to determine whether or not they were real.

Despite initial success with the OpenForensics dataset, testing on deepfake images from StyleGAN2, StyleGAN3, and ProGAN revealed underwhelming accuracies. Table 2. Shows the obtained accuracies, which ranged from 20% to 50%, these accuracies are noticeably low across multiple CNN architectures. These results tell us about how well the model could distinguish between real and fake.

Table 2. Comparison of Detection Accuracies of CNN models tested on various GANs

Models	Comparison of Detection Accuracies of CNN models tested on various GANs		
	Style_GAN_2_ FFHQ_256	Style_GAN_3_ FFHQ_256	ProGAN_ CelebA_128
VGG16	29.620%	21.342%	35.305%
VGG19	35.343%	26.355%	42.397%
DenseNet121	30.650%	23.165%	38.525%
MobileNetV2	29.420%	20.270%	33.447%

In the initial stage when we evaluated the performance, we realized that the model needed to be trained on all of the datasets, including ProGAN, StyleGAN2, and StyleGAN3. Then a calculated choice was made to add more images to the training dataset created by each of the three GAN models—Style_GAN_3, Style_GAN_2, and ProGAN to improve accuracy.

The objective of this addition was to ensure that the model was exposed to better quality fake images and a wider variety of fake images by adding more diversity and balance to the dataset. So, to address the initial low accuracy rates, the newly generated fake images were subsequently included in the training and validation sets of the dataset.

After making this modification, the model's performance was significantly improved. Across all three GAN datasets, the re-trained models showed a notable increase in accuracy after being trained on better fake images. Experiments with different activation strategies and learning rates were conducted to achieve even better results. ReLU activation and 0.0001 learning rate were found to work best for the model.

Fig.4. shows the graph of loss in training and loss in validation vs the epochs. Graphs of all the eight models have been included in the figure. Optimal configurations of 64 batch sizes and 10 epochs were determined through systematic testing for all CNN models.

In Fig.4 it can be observed that VGG16 has the best training-validation loss curve as it has good training loss convergence and the validation loss also converges close to training loss with little fluctuations. Some of the model's validation graphs were smooth with less fluctuation, and good convergence and some had spikes and variation as compared to training loss. This indicates how the models performed on unseen data as compared to seen data and help determine which model works best on unseen data. Some models do not have a good training validation graph, it is because the model has not generalized well that is it has not performed well on unseen data, its accuracy is bad and the other reason is that sometimes the validation data differs in quality to that of the training data. The models that have better validation graphs have generalized well on unseen data.

Table 3. Shows the accuracy of summarizing different CNNs on the difficult job of detecting manipulated images part of our thorough review of deepfake detection approaches.

Table 3. Comparison of Detection Accuracy for Various GAN Models Using Different CNNs

Models	Comparison of Detection Accuracy for Various GAN Models Using Different Convolutional Neural Networks (CNNs)		
	Style_ GAN_2_ FFHQ_256	Style_ GAN_3_ FFHQ_256	ProGAN_ CelebA_128
VGG16	95.038%	94.901%	94.987%
VGG19	97.983%	96.744%	97.397%

DenseNet121	95.650%	95.045%	95.525%
InceptionResNetV2	96.495%	96.380%	96.447%
InceptionV3	97.052%	96.668%	96.831%
MobileNetV2	95.707%	95.592%	95.659%
ResNet50V2	95.150%	93.230%	95.304%
Xception	96.956%	96.908%	96.898%

In Table 3. Three different deepfake datasets are used to thoroughly evaluate each CNN's performance in differentiating between real and fake content: Style_GAN_2_FFHQ_256, Style_GAN_3_FFHQ_256, and ProGAN_CelebA_128. The accuracy in percentage values of the models under consideration—VGG16, VGG19, DenseNet121, InceptionResNetV2, InceptionV3, MobileNetV2, ResNet50V2, and Xception is achieved.

VGG19 emerges as a strong contender with the highest accuracy of 97.983% on the Style_GAN_2_FFHQ_256 dataset and 97.397% on ProGAN_CelebA_128. Xception demonstrates its robustness on the Style_GAN_3_FFHQ_256 dataset by attaining the highest accuracy of 96.908%. Some of the models like InceptionResNetV2 and InceptionV3 had accuracy above 96% in all of the datasets and they were consistent along with the other models.

Overall, for all the models the accuracy ranged between 94% and 98%. Models gave the least accuracy on StyleGAN3 images because they were of the best quality amongst the images created by three GANs. ProGAN and StyleGAN2 images were comparatively detected with greater accuracy.

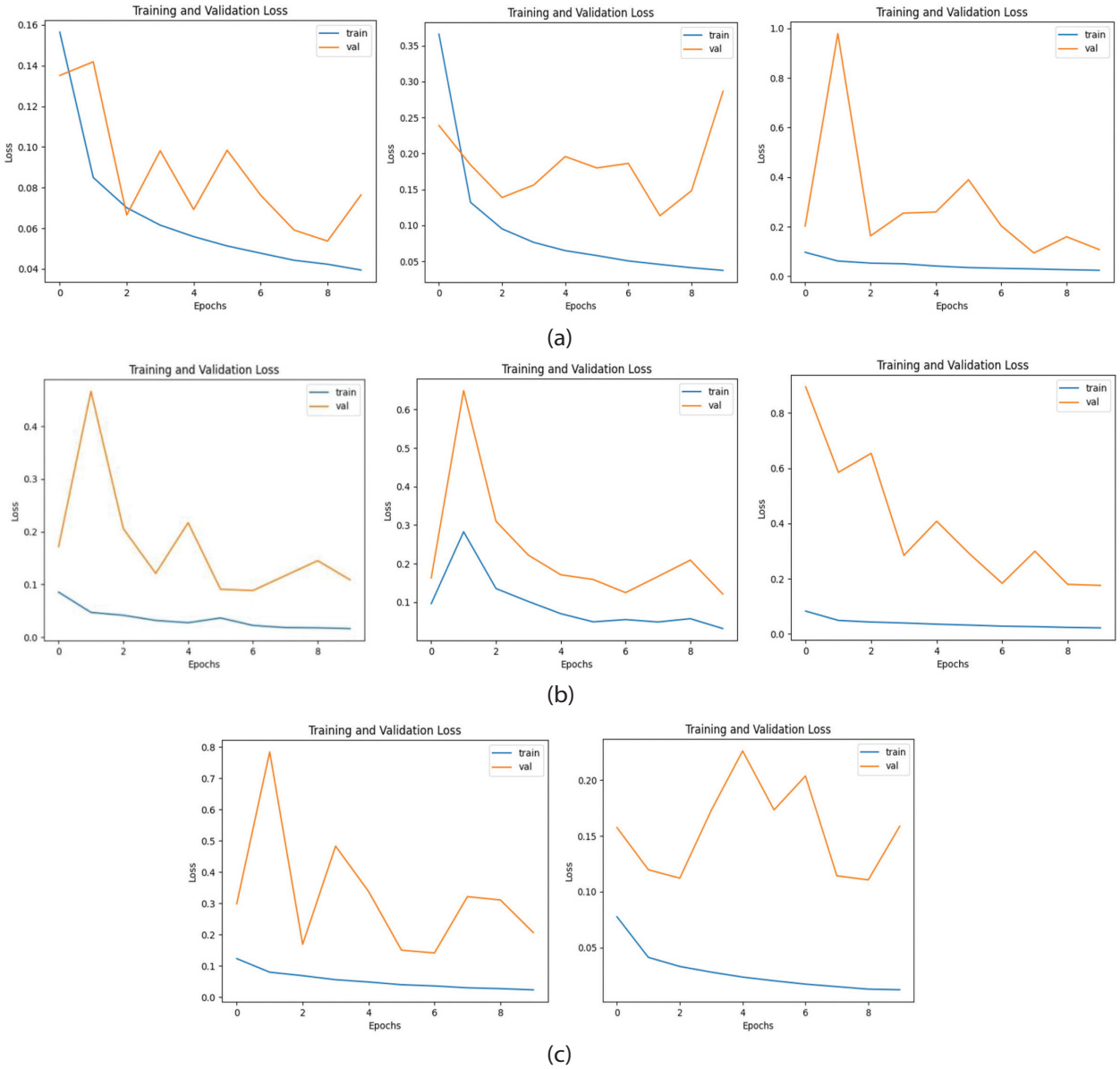


Fig. 4. Loss in training and validation where **(a)** VGG16, VGG19, and DenseNet121, **(b)** InceptionResNetV2, InceptionV3, and MobileNetV2, and **(c)** ResNet50V2 and Xception

7. CONCLUSION

Our study shows that different models work well in different situations for spotting deepfake images. We tested the eight CNN models against fake images from three GANs: StyleGAN2, StyleGAN3, and ProGAN. VGG19 and VGG16 do great in some cases, while InceptionV3 and Xception are consistently good giving an accuracy above 96.6% for all three GANs. The best-performing model however is VGG19 since it has the best overall accuracy across the three GANs. So our study based on the performances of the CNN models concludes that VGG19 is the better alternative to detect deepfake images coming from various sources.

With more powerful GPUs and CPUs, we can generate and detect deepfakes more efficiently. Advanced systems enable the use of models like EfficientNet, a highly effective CNN architecture, further enhancing our deepfake detection capabilities.

With the rise of artificial intelligence, the quality of deepfakeimages is only going to increase thus making their detection a continuous research topic. Our goal was to find a model that works well on fake s generated through diverse sources thus making it a reliable tool for countering the ever-evolving deepfake creation.

8. REFERENCES:

[1] M. Kumar, N. Muhal, “Fake Face s Generated From Different GANs”, https://www.kaggle.com/datasets/mayankjha146025/fake-face-s-generated-from-different-gans (accessed: 2024)

[2] Y. Digvijay, S. Salmani, "Deepfake: A survey on facial forgery technique using the generative adversarial network", Proceedings of the International Conference on Intelligent Computing and Control Systems, Madurai, India, 15-17 May 2019, pp. 852-857.

[3] S. Sanjan, P. Thushara, P. C. Karthik, M. P. A. Vijayan, A. Wilson, “Review of Deepfake Detection Techniques”, International Journal of Engineering Research & Technology, Vol. 10, No. 5, 2021, pp. 813-816.

[4] A. Malik, M. Kuribayashi, S. M. Abdullahi, A. N. Khan, "DeepFake detection for human faces and videos: A survey", IEEE Access, Vol. 10, 2022, pp. 18757-18775.

[5] M. S. Rana, M. N. Nobi, B. Murali, A. H. Sung, "Deepfake Detection: A Systematic Literature Review", IEEE Access, Vol. 10, 2022, pp. 25494-25513.

[6] O. A. Paul, "Deepfakes Generated by Generative Adversarial Networks", Georgia Southern University, Honors College Theses, 2021.

[7] T. T. Nguyen, Q. V. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q. V. Pham, C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey", Computer Vision and Understanding, Vol. 223, 2022, p. 103525.

[8] T. Shen, R. Liu, J. Bai, Z. Li. "'Deep fakes' using generative adversarial networks (GAN)", Noiselab, University of California, San Diego, 2018, Report 16.

[9] O. Giudice, L. Guarnera, S. Battiato, "Fighting deepfakes by detecting GAN DCT anomalies", Journal of Imaging, Vol. 7, No. 8, 2021, p. 128.

[10] H. S. Shad, M. M. Rizvee, N. T. Roza, S. M. Hoq, M. M. Khan, A. Singh, A. Zaguia, S. Bourouis, "Comparative analysis of deepfake detection method using convolutional neural network", Computational Intelligence and Neuroscience, Vol. 2021, 2021.

[11] D. Saxena, J. Cao, "Generative adversarial networks (GANs) challenges, solutions, and future directions", ACM Computing Surveys, Vol. 54, No. 3, 2021, pp. 1-42.

[12] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, C. Busch, "Fake face detection methods: Can they be generalized?", Proceedings of the International conference of the biometrics special interest group, Darmstadt, Germany, 26-28 September 2018, pp. 1-6.

[13] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, D. Lischinski, "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery", Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10-17 October 2021, pp. 2065-2074.

[14] M. Kumar, H.K. Sharma, "A GAN-based model of deepfake detection in social media", Procedia Computer Science, Vol. 218, 2023, pp. 2153-2162.

[15] A. Tiwari, R. Dave, M. Vanamala. "Leveraging deep learning approaches for deepfake detection: A review", Proceedings of the 7th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, 2023, pp. 12-19. 2023.

[16] E. Nowroozi, Y. Mekdad. "Detecting high-quality GAN-generated faces using neural networks", Big

Data Analytics and Intelligent Systems for Cyber Threat Intelligence, River Publishers, 2023, pp. 235-252.

[17] S. Preeti, M. Kumar, H. K. Sharma. "Robust GAN-Based CNN Model as Generative AI Application for Deepfake Detection", EAI Endorsed Transactions on Internet of Things, Vol. 10, 2024.

[18] B. D. Sergi, S. D. Johnson, B. Kleinberg, "Testing human ability to detect 'deepfake' s of human faces", Journal of Cybersecurity, Vol. 9, No. 1, 2023.

[19] T. N. Le, H. H. Nguyen, J. Yamagishi, I. Echizen, "Openforensics: Multi-face Forgery Detection and Segmentation In-the-wild Dataset [V.1.0.0]", https://doi.org/10.5281/zenodo.5528418 (accessed: 2023)

[20] CelebA-Dataset, "Large-scale CelebFaces Attributes (CelebA) Dataset", https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html (accessed: 2024)

[21] FFHQ, "NVlabs/ffhq-dataset: Flickr-Faces-HQ Dataset (FFHQ)", https://github.com/NVlabs/ffhq-dataset (accessed: 2024)

High-Performance Graph Storage and Mutation for Graph Processing and Streaming: A Review

Review Paper

Soukaina Firmli*
Mohammed V University in Rabat
Ecole Mohammadia d'Ingénieurs, SIP Research Team
Rabat, Morocco
soukaina.firmli@gmail.com

Dalila Chiadmi
Mohammed V University in Rabat
Ecole Mohammadia d'Ingénieurs, SIP Research Team
Rabat, Morocco
chiadmi@emi.ac.ma

*Corresponding author

Abstract *– The growing need for managing extensive dynamic datasets has propelled graph processing and streaming to the forefront of the data processing community. Given the irregularity of graph workloads and the large scale of real-world graphs, researchers face numerous challenges when designing high-performance graph processing and streaming systems, due to the sheer volume, intricacy, and continual evolution of graph data. In this paper, we highlight the challenges related to two vital aspects within Graph Processing Systems that significantly impact the overall system performance: 1) the graph storage, encompassing the data structures storing vertices and edges, and 2) graph mutation protocols, referring to the ingestion and storage of new graph updates, such as additions of edges and vertices. Our paper provides a practical taxonomy of techniques designed to improve the efficiency of graph storage and mutation, by reviewing state-of-the-art systems and highlighting the challenges they face in offering a good performance tradeoff for read, write, and memory consumption. Consequently, this enables us to highlight overlooked aspects of performance, that are essential for real-world applications, such as the lack of mutation protocols for graph properties and auxiliary graph data, lack of configurability and cross-platform evaluation of solutions for graph processing and streaming.*

Keywords*: Graph Processing; Graph Streaming; Data Structures; Mutation; Performance*

Received: May 9, 2024; Received in revised form: September 13, 2024; Accepted: September 16, 2024

1. INTRODUCTION

Graph processing technology is continually advancing and thriving due to the distinctive capacity of graphs to model intricate relationships and dependencies within data, making them ideal for various contemporary applications. Notable graph applications include Knowledge Graphs (KG) [1] used in search engines, personal assistants, and recommendation systems; Graph Neural Networks (GNNs) [2] employed in AI tasks such as node classification, link prediction, and graph classification; and real-time graph analysis for streaming data from platforms like Twitter and financial transactions.

Given the extensive use of graphs, there is an increasing demand for efficient management and analysis. This demand has spurred the creation of graph processing systems and databases like Neo4j [3], which are adept at storing, analyzing, and streaming large graph datasets due to their scalability, real-time processing capabilities, and proficiency in handling complex relationships.

An ideal graph processing system should excel in analytics performance, provide fast mutations, and exhibit low memory consumption, regardless of mutation operations. However, these systems encounter several challenges inherent to graph processing and streaming. These challenges include the varying graph characteristics, the memory-intensive graph algorithms, the access patterns that cause latency, and the continuous evolution of graph topology and properties [4].

To understand these challenges, it is important to recognize that among the crucial software design elements are the graph data structure, which is responsi-

ble for storing vertices and edges, and the graph mutation protocols, such as additions or deletions of vertices and edges. While most of the optimization techniques in the literature stem from the data structures research community, there is a lack of research on their practical applicability within graph workloads.

Our research contributes by providing a practical taxonomy of techniques aimed at enhancing the efficiency of graph storage and mutation. These techniques include the optimization of the memory layout of graph data structures, compression, partitioning, batching techniques, changeset-based updates with delta maps and multi-versioning to improve the update-friendliness of classic data structures such as the Compressed Sparse Row (CSR) [5]. Furthermore, our analysis delves into the performance claims of existing literature on high performance, unveiling persistent challenges in read, write, and memory performance. This allows us to shed light on research gaps and overlooked aspects of performance crucial for real-world scenarios.

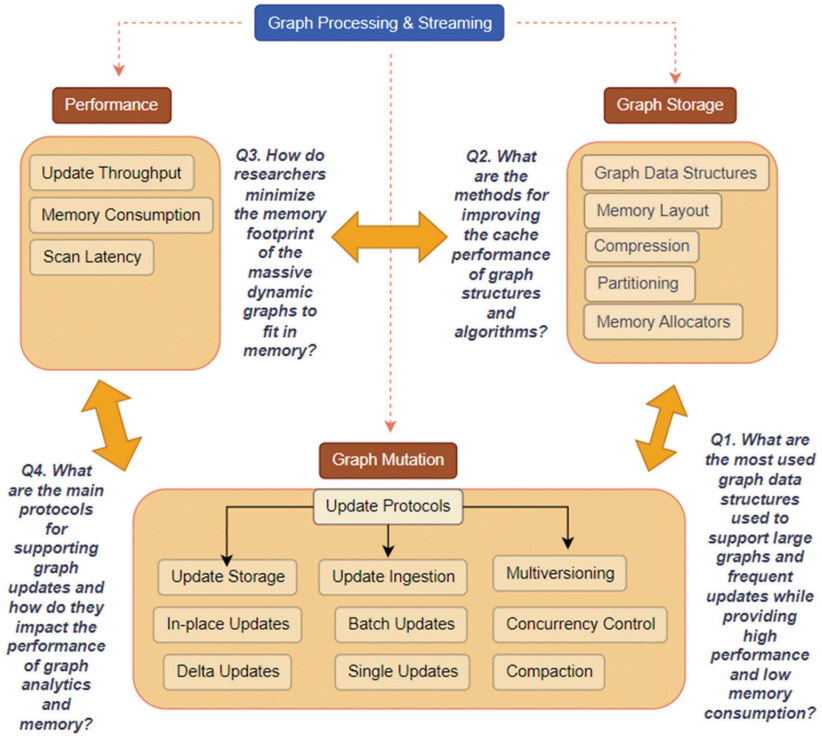


Fig. 1. Conceptual framework of our review

We present the background in Section 2 of our study. Second, we discuss our research methodology and research questions in Section 3 before presenting our review and analysis in sections 4 and 5. Finally, we discuss our findings in section 6, present the related work in section 7, and then present our future work and conclude in section 8.

2. BACKGROUND & CONTEXTUALIZATION

A graph is defined as a mathematical representation comprising vertices (nodes) and edges, which represent entities and the relationships between them, respectively. The volume, velocity, and variety of big data [6-8] that come from storing and processing large graph data, pose unique challenges in the field of computer science and data processing.

Graph technology, including graph processing systems and graph databases like Neo4j [3], emerges as a suitable approach for storing, analyzing and streaming big graph data, due to its scalability, real-time processing capabilities, and ability to handle complex relationships.

These GPS store graph data (i.e., graph topology and properties) in containers using data structures, such as adjacency lists, edge lists, or matrices [5]. They also provide algorithms for running analytic workloads and queries as well as updating graphs. These systems use combinations of high-performance data structures and update protocols to achieve their target performance.

They take advantage of hardware resources such as parallelism and Distributed machines to address the scalability challenge of processing and streaming large graphs efficiently.

From our research, it became evident that graph processing and streaming systems encounter three major challenges in efficiently storing and updating graphs which are as follows.

Challenge 1. Many real-world situations can be understood via the lens of scale-free networks. These graphs have a power-law distribution of degrees and a low density [9], with many vertices having very few or even zero degrees. The Internet and other social networks are two common examples of such graphs.

When designing systems, however, it might be difficult to account for the skew (degree variation) of these graphs. In reality, additional memory and processing power (RAM, CPU) may be needed to process vertices with higher degrees.

Challenge 2. The large size of graph data and access patterns of graph algorithms are important factors in the practical applicability of graph systems in production environments. Essentially, the memory-intensive nature of graph algorithms and their access patterns is one of the primary causes of the latency in graph computations. For instance, the PageRank algorithm [9] requires a large amount of input/output (I/O) operations to main memory and random accesses when iterating over vertices and edges in the graph, causing a lot of cache misses [10] and more slowdown

Challenge 3. The velocity characteristic of big data makes processing data more challenging where large amounts of graph data are generated rapidly and need to be added to graphs as new relationships in real-time. This is handled by stream processing systems, closely associated with real-time processing, involving processing data as it is created [11].

In summary, big data's volume, velocity, and variety pose challenges for traditional data processing methods including the storage, mutation and processing of graph data sets.

3. METHODOLOGY

There is a growing body of literature in the context of processing and streaming big dynamic graphs. In our paper, we aim to give a global overview of the different techniques used by researchers to improve the performance of graph processing systems and streaming.

In an attempt to give this overview, we narrowed the scope of this review to cover literature published over the past 16 years. We define a set of strings derived from keywords related to our research. The initial keywords are Graph, Analytics, Processing, Storage, Streaming and Mutation. We then use combinations to form strings to search for relevant papers on IEEE Xplore, ACM Digital Library and Google Scholar. We compile our database of around 97 prominent publications, we exclude some papers as they are out of our scope (e.g., incremental computation and graph database systems [12]).

With the collected papers, we note the following techniques that are used generally in the literature, which include:

- The optimization of graph data structures for the storage of dynamic graphs
- The design of parallel algorithms to execute graph analytics and queries efficiently on dynamic graphs
- The design of high-level graph languages to express and execute graph queries
- The implementation of algorithms for Distributed processing of graphs
- The design of efficient graph mutation protocols for fast graph updates

We focus on the graph representation in memory and the algorithms for graph updates as shown in Fig. 1 and Table 1.

Table 1. Analysis dimensions and their corresponding sections in this paper

Dimension	Description	RQ	RQ Section
Graph Representation	The data structures to store the graphs	RQ1	Sec. 4
Memory Consumption	The memory footprint of the graph represented in physical memory	RQ2, RQ3	Sec. 4 & 5
Graph Mutation	The implementation of graph mutations: in-place, delta maps, snapshots	RQ4	Sec. 5
Performance Optimization	The process of modifying a system to improve its functionality, thus making it more efficient in read and updating workloads	RQ2, RQ3,	Sec. 4 & 5
Parallelism & Hardware Resources	The underlying architecture of systems such as multi-core, CPU Cache and Distributed Systems	RQ2, RQ3,	Sec. 4 & 5

4. TAXONOMY OF TECHNIQUES FOR EFFICIENT GRAPH REPRESENTATION

In this section, we give an overview of techniques for optimizing classical graph data structures for high analytic and update performance as well as minimizing the memory footprint. We organize this section as the following. First, we discuss the performance of classical data structures, to identify their limitations. Then we analyze the different techniques available for researchers to optimize them, namely: optimizing the memory layout of the data structures (Section 4.2), compression (Section 4.3), using memory allocator software (Section 4.4) and partitioning (Section 4.5). Finally, we present a summary in Table 2.

4.1. REPRESENTATIVE GRAPH CONTAINER

In the following, We provide descriptions of classical graph data structures and we discuss the costs of performing graph mutations on each structure.

Adjacency Matrix

It holds a square matrix M with dimensions VxV, where V stands for the graph's vertex count. To indicate a directed edge from a source vertex vs to a destination vertex vd, the cell M[vs][vd] must be assigned a non-zero value. While this method simplifies edge manipulation, it's inefficient for sparse graphs due to high memory usage and suboptimal analytics performance. Moreover, adding or removing vertices requires completely recreating the matrix.

Adjacency Lists

This structure stores vertex information within a node list, with each element pointing to a list of its neighbours. It consumes less memory compared to an adjacency matrix because it only stores existing edges. The typical approach involves using linked lists for these connections, yet there are more efficient alternatives designed for better caching. For instance, variants like Blocked Adjacency Lists use simpler arrays for representing adjacencies [13] or utilize linked lists with fixed-size edge-containing buckets.

CSR (Compressed Sparse Row)

This representation, widely used for sparse graphs, condenses adjacencies into primarily two arrays: an edge array holding indices of destination vertices from the node array. The latter contains offsets to identify the beginning and end of the neighbours' list. To find the degree of node i, we compute Node_Array[i+1] – Node_ Array[i]. However, while this format is efficient in many cases, it still faces limitations, particularly in contexts where frequent updates occur. Several variations of CSR (Compressed Sparse Row) have been suggested—such as CSR++ [14-16] —aiming to enhance support for quicker structural updates. Further details on this topic are discussed in subsequent sections.

4.2. MEMORY LAYOUT AND CACHE AWARENESS

A good memory layout for a graph representation refers to the optimal way to arrange the graph data in memory to enhance its performance [10], by using hardware optimizations such as caching and prefetching. Moreover, knowing the access patterns of graph algorithms and storing graph entities in a specific layout can guarantee both spatial and temporal locality for graph data structures and algorithms [14, 17, 18], hence better performance.

By considering these factors, researchers in the graph community propose new variants of data structures by changing their memory layouts [9, 14, 16, 18, 19] to optimize the performance of graph analytics, queries and streaming, and improve their overall efficiency.

Essentially, to remediate the poor cache locality of AL, many researchers [13, 20] use bucketing technique where buckets are used to group edges from the same source vertex together, or use linked lists to group edges from different source vertices together (we elaborate this on Section 5. As we note from the evaluation results of works in the literature, there are only a few works that provide a thorough sensitivity analysis of different variations of their solutions like [19, 21]. There is a consensus on the size of the buckets which shouldn't be too huge, as it would slow down update performance, or too small, as that would cause cache misses. Apart from [16] as a pioneer solution for dynamic graph storage, we think there is a lack of configurability in systems in the literature and therefore a lot of opportunities to tweak existing designs in favour of new ones and for different types of workloads.

Furthermore, since CSR is known for its high cache performance and slow update performance, many researchers [15, 16, 19] opt for it as a main data structure for graph analytics and queries, then use an extra data structure to store the updates. For instance, to support fast updates, LLAMA stores multiple versions of the graph in CSR structures.

Essentially, it implements two variants of CSR, namely performance-optimized (PO) and space-optimized (SO), where the former keeps a complete list of edges of the same vertex in each version of the graph, and the latter only stores fragments of the edge lists for every vertex in different snapshots. As the names suggest, the PO provides high performance since all the edges are stored contiguously but the memory suffers from multiple copies of the edge list.

The SO saves on memory; however, it is slower since the edge lists are not stored contiguously, and the system needs to reconstruct the full adjacency of a vertex for read queries.

Finally, while memory layout techniques predominantly focus on graph topology, we emphasise the criticality of aligning memory layout with storing graph properties and auxiliary graph data (e.g., user keys), especially given the prevalence of graph algorithms dealing with weighted graphs.

4.3. COMPRESSION FOR DYNAMIC GRAPHS

Compression allows for a reduction of the memory needed to store data while maintaining its essential properties and functionality [22]. It is more efficient in graph representation since it allows for better cache performance since more data can be loaded in the cache and accessed at once by the CPU.

However, despite being a classic technique, there appears to be a scarcity of research exploring the application of compression techniques within the realm of graph updates. Notably, we observed a prevailing reliance on Ligra+ [23] within existing systems, attributable to its user-friendly interfaces. Among the notable systems claiming high performance in this context are Aspen [18], and SSTGraph [24], suggesting a potential avenue for further investigation into optimizing graph update procedures.

Aspen stores graph data in compressed purely functional trees, a form of persistent data structure for storing large graphs. By compressing trees, Aspen solves the issue of storing massive graphs (up to 200B edges) in machines with just 1TB of RAM by using compression. Given that Aspen stores edges as integers, it uses difference encoding to reduce the size of the edge arrays. However, while this method can significantly reduce memory consumption, it still increases the price

of encoding and decoding processes, and a penalty may be incurred when executing queries or updating the graph.

To remediate this cost, SSTGraph a parallel framework designed for the storage and analysis of dynamic graphs, is based on the tinyset parallel dynamic set data structure, which implements set membership using sorted packed memory arrays. This allows for logarithmic time access and updates, as well as optimal linear time scanning. Compared to systems that use data compression, tinyset achieves comparable space efficiency without the computational and serialization overhead.

Finally, there is a notable absence of exploration into advanced compression algorithms within this domain. We propose that the graph processing community delve deeper into the challenges associated with graph workloads to assess the feasibility and potential benefits of implementing sophisticated compression algorithms.

4.4. MEMORY ALLOCATORS IN GRAPH STREAMING

A memory allocator is a software component that manages the allocation and deallocation of memory in a computer program [25]. GPS either develop their memory allocators or use out-of-the-box allocator libraries to manage their memory allocations and reduce memory fragmentation [26].

The first approach is used by some systems [20, 21, 27] where they develop built-in memory managers that facilitate the speedy allocation of memory needed for applying mutations. For instance, to efficiently perform memory reclamation and manage space, Hornet's [27] internal memory management uses a B+ tree for insertions and deletions to keep track of the available blocks of edges. Moreover, when data is duplicated, the system uses a load-balancing mechanism to locate the freed memory for later usage.

However, our findings underscore both the lack and the potential for smart predictive allocation techniques [28], particularly concerning updates within graph processing systems. Notably, while reallocation of edges commonly employs a predefined factor in existing systems, this often results in unnecessary allocation of extra space. We posit the feasibility of implementing smarter allocators leveraging machine learning methodologies to predict the optimal reallocation factor, thus enhancing memory utilization efficiency within graph processing frameworks.

The second approach uses memory allocators such as Jemalloc [29] and TCMalloc [30], which are widely used for their parallel support of memory allocation which helps with providing high update throughput. Moreover, these allocators use highly efficient algorithms to limit memory fragmentation, which leads to better cache locality and lower memory footprint.

We note a particular system called Metall [31] which is a persistent memory allocator that uses the copy-on-write technique for graph workloads, stores and manipulates large graphs at the exascale (billions of billions of operations per second), by employing smart allocation algorithms like those found in TCMalloc [30]. Essentially, Metall employs the use of mmap system calls to create memory-mapped files. With mmap, one may essentially access the files as if they were RAM, since it redirects the data to a virtual memory region. Therefore, to offer lightweight multi-versioning, Metall makes use of copy-on-write by taking snapshots of the graph after ingesting a batch of updates and employing a file copy method in the filesystems called reflink, which permits copy-on-write of data.

4.5. DYNAMIC DATA STRUCTURE PARTITIONING

Graph data partitioning is the process of dividing a large graph into smaller subgraphs [7], called partitions, to enable parallel processing of the graph on multiple machines or processors [8].

Some GPS systems [32, 33] use partitioning to optimize their performance by introducing several novel techniques to scale graph processing on a distributed cluster, including partitioning for sequential storage access, random distribution of data across the cluster, and work stealing for load balancing. These techniques enable GPS to handle graphs with trillions of edges, representing up to 16 TB of input data. However, the research on graph mutations using partitioning in a distributed system is still premature, and a very small number of articles address the challenges that come with it.

Finally, a single previous study provided an overview of graph update types, dynamic graph partitioning, and associated challenges [34]. Additionally, [35] did not address the performance consequences of partitioning on updates, memory, and read operations. Hence, it could be beneficial to investigate this aspect further.

Table 2. Summary of the characterization of systems included in this study based on their techniques for efficient graph storage

Dimension	Impact on Perf.	Implementations	Graph Processing & Streaming Systems
Memory Layout	Cache-friendly data structures	Blocking, CSR, PMA	[15, 17, 19, 27, 36]
Data Compression	Small memory footprint and better cache locality	Difference Encoding, Bit Indexing	[18, 24, 33]
Partitioning	Distributed processing and load balancing	Edge-cut, Vertex-cut	[32, 37]
Memory Allocators	Parallel allocation, low memory fragmentation, high cache performance	Jemalloc, TCMalloc, B+ Trees	[14, 16, 20, 27, 31]

5. TAXONOMY OF UPDATE PROTOCOLS FOR EFFICIENT GRAPH MUTATION

In addition to the representation of graphs in memory for graph queries and analytics, a wide range of GPS [16, 18, 20, 38] support graph mutation by allowing modifications to the graph's topology by adding or removing edges and vertices, as well as modification to the graph properties.

Essentially, to achieve that, we identify emerging patterns in the literature where GPS implement different techniques that we refer to as update protocols, which are different approaches for the ingestion and storage of new incoming graph data.

In the following, and in an attempt to answer our research question, we define the update protocols and analyze the different techniques in the literature for implementing graph updates and discuss their performance implications and limitations. Finally, we present a summary in Table 3.

5.1. OVERVIEW OF UPDATE PROTOCOLS

a) Update Ingestion

We categorize the update ingestions depending on two criteria: i) How a stream of incoming updates is handled right before storing them in the system? (Section 5.2), and ii) how a stream of incoming updates is handled in the presence of analytical queries (e.g., algorithms, pattern matching, scans ...) (Section 5.3).

First, there are two approaches to ingesting the updates in the GPS: i) Single update queries refer to the insertion or removal of a single edge or vertex at a time, while ii) batch updates refer to the grouping of the updates in a batch before applying them all at once. Moreover, we extrapolated two modes researchers are exploring for ingesting updates depending on how the graph analytics and queries are executed: in bulk or concurrently. Essentially, in the bulk mode, updates and graph algorithms are executed sequentially "in phases". On the other hand, in the concurrent mode, updates and graph analytics are processed simultaneously [13, 20, 27].

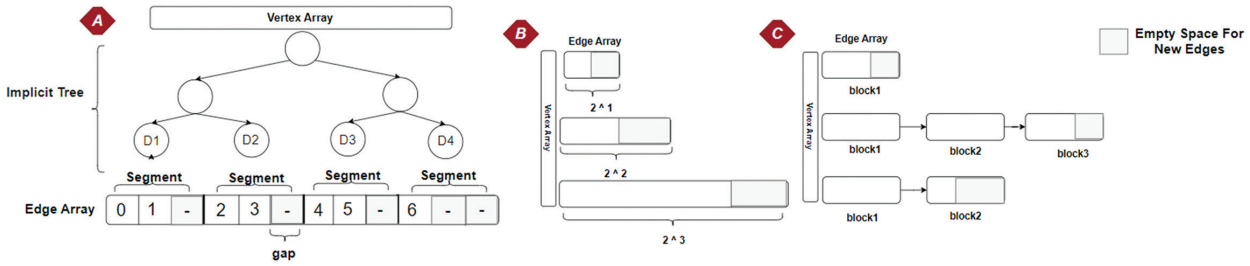


Fig. 2. Dynamic graph data structures. **A)** CSR representation of a graph using PMA to store the edge array. **B)** Adjacencies of a graph are stored in growable arrays with factor x2. **C)** Adjacencies of a graph are stored in a linked list of blocks

b) Update Storage

When applying the mutations, changes can be applied 1) in-place (i.e., incorporated into the main structure) or 2) stored in additional data structures called deltas [16]. In-place update is a technique where systems augment the traditional graph data structures, by permitting in-place storage of updates without the costly rebuilding of the whole graph data structure. As for the Delta approach, GPS use update-friendly data structures such as Adjacency List or Edge List to quickly store the updates, with the additional cost of merging these updates into the main read-friendly structure such as CSR. We elaborate on these techniques in Sections 5.4 and 5.5.

5.2. UPDATE INGESTION: SINGLE VS BATCH UPDATES

First, single updates are challenging to support for two main reasons. First, in most cases [14, 19, 20] and especially in deletion workloads, the system needs to perform a search over the neighbours of a vertex upon every edge insertion, which is not possible to do in parallel. This makes the system's performance very slow.

Second, depending on the availability of memory, systems need to allocate more memory to store the new edges [14]. Consequently, the frequent checks for memory availability and reallocations cause a large overhead, making the single updates very slow.

To remediate the slow single update performance, systems [14, 16, 18, 38] opt for batch updates where the batch of edge updates is pre-processed.

For instance, the sorting allows grouping all the edge updates of a specific vertex, separating deletions from insertions, which allows running edge updates in parallel for separate vertices. This improves the rate of update ingestions. This finding is supported by prior research [9, 17, 20] highlighting the effectiveness of sorting in optimizing the processing of edge updates.

Moreover, another technique used in batch updates is partitioning, which refers to splitting the updates into partitions that can be handled in parallel by multiple threads simultaneously [32]. This allows for better load balancing between parallel threads, especially for skewed graphs.

Unfortunately, the techniques mentioned above still incur large latency overhead as measured by GPS in litera-

ture [14, 16, 19], and most systems do not offer both single updates and batch updates, which is necessary for some real-world scenarios where updates are not frequent.

5.3. UPDATE INGESTION & MULTI-VERSIONING: BULK VS CONCURRENT UPDATES

The approach used by systems [13, 20, 27], to implement updates using the bulk mode, is a sequential approach where updates are held back until queries are completed, allowing updates to modify the graph while keeping data consistency [39], which ensures that the returned results accurately represent the current state of the data.

On the other hand, in the concurrent mode, systems may process updates and queries simultaneously. In this case, maintaining query consistency can be challenging.

Systems [14, 17, 20] in the bulk mode, mostly focus on supporting high update rates since updates don't have to be delayed by the queries. For instance, STINGER achieves an update rate of over 1.8 million updates per second on single multi-core machines, by executing updates in batches and running them in parallel without being concerned about concurrent reads.

Despite the high update throughput, research [18] shows that systems that employ the bulk mode have limited usage, since in real-world scenarios, graph users are constantly updating and running analytic queries concurrently. In the case where update/read happens in phases, this can introduce delays and decrease the system's overall performance.

On the other hand, in the concurrent mode, systems may process updates and queries simultaneously. In this case, maintaining query consistency can be challenging . In the following, we review different approaches used in practice, to allow concurrent updates and queries and discuss the challenges researchers face and potential research areas for the future.

Hybrid Store

A lot of systems [16, 18, 19, 40] implement protocols to maintain data coherence and consistency between multiple readers and writers through different isolation levels, which is similar to traditional database systems.

One way to achieve concurrent analytic workloads and update workloads is by creating a hybrid graph representation that uses separate data structures [16], [19]: one usually referred to as the write-optimised (WO Storage) data structure for the incoming updates and another read-optimised (RO Storage) structure for storing the main graph and can be accessed concurrently. This way, updates and read workloads would operate in parallel on different structures.

In this category, we cite a notable system LLAMA [16], which creates a new delta, a.k.a., snapshot, every time the user runs a batch of updates as shown in Fig. 4 B). This technique enables readers to have parallel access to the previously created snapshots and run analytics and queries on the RO store without interfering with the newer update queries performed on the WO store. Another example is GraphOne [40], which implements a hybrid store for snapshots using an adjacency lists (AL) store and an edge list (EL) as shown in Fig. 3 A). The AL keeps track of a linked list of vertex degrees at various points in time using timestamps.

Concurrency Control

Concurrency Control (CC) is a notable approach we extracted from the literature for processing concurrent reads and writes in GPS literature. For instance, a popular model is the Multiversion Concurrency Control (MVCC) [19] used by transactional systems such as Teseo [19] to achieve Snapshot Isolation. Teseo uses timestamps and a reversed chain of images to store the original copies of data, showing the items as they were before any changes were made (from newest to oldest). These versions are temporarily kept in the transaction's undo buffers whilst they are being rolled back, and then they are garbage collected as soon as the transaction is no longer valid (i.e. version pruning).

However, Teseo and many other systems face challenges with garbage collection, necessitating its execution without disrupting ongoing queries, thereby imposing additional costs on performance. We observed the lack of systems addressing the performance implications of executing compaction. We deem this a critical performance concern as both the execution of version pruning and the required resources may be hindered by the granularity and frequency of updates.

5.4. UPDATE STORAGE: IN-PLACE UPDATES

Systems [9, 13, 14, 19, 20] implement in-place updates by designing data structures that are suitable for graph updates, where the new entities (vertices or edges) can be directly stored in the data structures without requiring to reallocate the main data structure or to store them in extra data structures.

The main idea is to leave some space in the data structure for the new incoming entities, and in case there is not enough space, the data structure should allocate extra space suitable for that new entity as shown in Fig. 2. In the following, we discuss the popular approaches to achieve in-place updates.

a) Dynamic Arrays

A lot of systems use growable dynamic [9, 13, 14] where edge insertions are performed by directly storing the new edges in dynamic growable edge arrays. Systems employing Adjacency Lists are more prone to use this technique as shown in Figs. 2 B) and C).

NetworKit [13] performs edge insertions by directly storing the new edges in dynamic growable edge arrays and reallocating twice the initial array size when

there is no memory space available. The same method is employed by Madduri et. al. [9] except that the size of the new edge array is defined in terms of a customizable factor rather than a fixed factor of 2.

Subsequently, when employing dynamic arrays, the amortised cost for updates is O(1) for insertions and O(deg V) for deletions. However, the memory footprint can be quite substantial as the reallocations leave unused space, when there are no updates. As mentioned above, using smart memory managers can help keep track of unused space and perform a better strategy for pre-allocation while maintaining a memory footprint comparable to static graph data structures. We also highlight the importance of knowing the pattern of the streams, to estimate the size of the reallocation. We predict that machine learning techniques are a possible venue for research that the community of graph processing and streaming should explore to propose innovative mechanisms to tackle these challenges in real-world graphs. For instance, by learning the patterns of the creation of relationships in social networks, e.g., a post that goes viral on social media might bring many followers in a short amount of time.

b) PMAs

Packed Memory Arrays (PMAs) are another classic technique in data structures that allows in-place updates by maintaining dynamic sets of sorted elements. It is based on storing an array of sizes larger than the number of elements N and leaving gaps between the elements to allow for new insertions with a moderate cost of $O(lg^2(N))$. However, PMA relies on extra computation using an implicit balanced tree that keeps track of gaps within regions of the array as shown in Fig. 2 A). Essentially, when the number of gaps is too small or too large, systems are required to perform a re-balancing of the tree to rearrange the gaps in the array. This may slow down the update performance and delay the analytic workloads.

For instance, [17] developed a variant of CSR based on PMAs called PCSR, which provides efficient single-threaded mutations by leaving space at the end of each adjacency list. Moreover, multi-versioning systems like Teseo use PMAs to store multiple versions of the graph. By using PMAs, Teseo updates graph data in place, hence preserving the contiguous memory placement of the data. Ultimately, PMAs have gained traction within dynamic graph representations [19, 41, 42]. However, the expected improvement in read performance associated with PMAs does not justify the overhead incurred by tree rebalancing. Systems employing dynamic arrays [21] outperform PMAs in terms of update performance while supporting in-place updates.

c) Deletions

Regarding the deletions of graph entities, there are two main approaches: physical deletions and logical deletions. In the latter, systems [14, 16] enhance vertex and edge data with flags that can be set if they are removed, which allows for logical deletions of entities. The disadvantage of this method is that it requires changing the graph algorithms to account for deleted entities during traversals, which can make them slower, due to extra branches in the algorithm.

On the other hand, when entities are deleted physically from memory, the corresponding slots in the data structure are left unfilled, which results in a higher storage cost than logical deletion. Systems [16, 19, 40] employ compaction, which means reconstructing the graph without assigning space for the removed entities, to decrease unnecessary space after numerous physical deletions. This operation is very costly as it requires a whole rebuild of the graph, however, it helps reduce memory fragmentation and, therefore better read performance.

5.5. UPDATE STORAGE: DELTA UPDATES

One way to extend CSR to support fast updates is by employing deltas, which are separate structures to store only the new changes in vertex/edge logs. For instance, edge lists [16, 40] are particularly well suited for storing updates as deltas, as we can append new edges in O(1) time and space. They also help to maintain the temporality/history of updates, since the edges are stored in the order they arrive.

In practice, to support deltas, systems [16, 18, 19, 40] tend to design separate structures as read-optimized stores (usually CSR-based) and write-optimized stores which practically refer to a delta. However, the downside of storing updates in delta maps or edge lists is two-fold. First, maintaining separate structures for each batch of updates increases the system's memory requirements, as a frequent stream of graph updates results in many deltas. Second, analytics performance is degraded because they need to read from both the original structure and the deltas and reconcile them.

For example, LLAMA creates a new delta (i.e., snapshot) once the write-optimized store has been "flushed" into the read-optimized store. This might be practical for multi-versioning, but creating new deltas too frequently often results in out-of-memory errors as shown in [21] which can be fatal for real-time systems.

Finally, to remediate this problem, these systems perform compaction on the frequently created data structure into one main structure. For this purpose, several authors [16, 19, 40] have attempted to design their version of compaction. Throughout our research, we came across a multitude of references to the same operation, such as "archiving," "merging," and "building."

GraphOne [40] stores newly added edges in a circular log and uses adjacency lists as the main read-optimized store, which provides a coarse-grained snapshot method as shown in Fig. 3 A). First, GraphOne establishes an edge log cutoff for the "transfer" of the edges from the buffer to the base structure to preserve the

separate store for the update operations and achieve good analytic performance. Second, in some cases, some data may be kept in both stores for a predetermined amount of time to maximize efficiency, this is known as data overlap. However, when there is a lot of duplication, it may use more RAM than usual.

It is shown in [19] that GraphOne's iterator architecture is to blame for its subpar performance in reading workloads. Point lookups are not possible because of the high space and time costs associated with analytic performance, which are incurred whenever the system iterates over vertices or edges and copies neighbours into an intermediate buffer.

In summary, compaction is computationally demanding, often zeroing the mutability performance benefits of these structures. For users who wish to operate on the most up-to-date version of the graph data, we believe that designing a system for in-place graph mutations is a better option to achieve higher analytics and update performance with lower memory requirements.

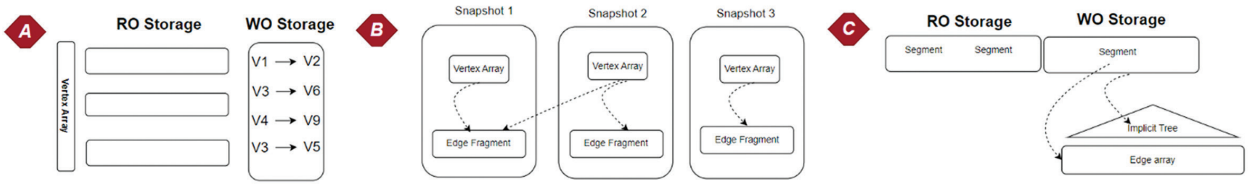


Fig. 3. Delta Updates, **A)** Hybrid storage using AL main storage and Edge List as delta. **B)** Snapshot mechanism as seen in LLAMA to access multiple versions of the graph. **C)** Delta structures stored in PMAs used in Teseo to support concurrent read and writes

Table 3. Summary of the characterization of systems included in this study based on their techniques for efficient graph updates

Technique	Category	Advantages	Systems
Single Upd.	Ingestion	Fine granularity of updates & Requires more CPU and memory	[14, 17, 20]
Batch Upd.	Ingestion	Fast update, load balancing Requires extra pre-processing	[14, 16, 18, 40]
Upd. in Bulk	Ingestion	Fast update throughput, fast analytics & No concurrency	[9, 14, 17, 20, 27]
Concurrent Upd.	Ingestion	Parallel updates and reads& High memory footprint, slow updates	[16, 18, 19, 40]
In-place Upd.	Storage	Low memory footprint, fast analytics & Slow updates	[9, 13, 14, 17, 19, 42]
Delta Upd.	Storage	Multi-versioning, fewer resources & Slow analytics, high memory footprint slow updates	[16, 18, 40]

6. RESULTS & DISCUSSION

In the following, we summarize key insights about the 50 research on high-efficiency graph representation and 51 update protocols. Table 4 presents a classification of the most popular graph systems reviewed in our paper, based on the dimensions discussed previously.

a) Configurations and Hybrid Representations

Based on the findings presented in Table 4, it is evident that numerous systems heavily rely on the Compressed Sparse Row (CSR) format due to its cache-friendly nature and reduced memory footprint. However, these systems often incorporate optimizations such as hardware enhancements or distributed computing frameworks [37], and supplementary data structures to manage updates efficiently, but there are limited designs for inherently fast data structures that support fast in-place and single updates while maintaining low memory footprint and high read performance.

Researchers can take advantage of the power of machine learning, to have more adaptable configurations depending on the workload. For instance, in the case of social networks, we can use learning algorithms to estimate the factor by which we can grow our dynamic arrays. This means that for a celebrity account, there are higher chances of gaining more followers than a normal account, therefore we can choose a higher factor (x4, x5) to pre-allocate the edge lists while keeping an x2 factor for normal accounts to maintain a low memory footprint of our dynamic graph.

b) Cross-platform implementation and evaluation

We highlight the prevalence of shared-memory systems, underscoring the ongoing necessity for advancements in this domain to attain comparable performance levels to distributed systems. Moreover, within the realm of distributed systems, there exists a notable lack of research on graph streaming, particularly concerning protocols tailored for update ingestion and storage in distributed infrastructures, or at least evaluation of the data structures in Distributed environments to showcase the advances in that area.

This observation implies that while the examined systems may ensure performance within specific hardware configurations, they lack assurance regarding the portability of their data structures across different infrastructures. Thus, there arises a demand for solutions which can seamlessly transition between single-machine and distributed environments.

Table 4. Summary of reviewed systems. Infra.: the supported architecture either Single Machine (SM) or Distributed (Dist); DS: data structures (CSR, Adjacency List, Tree); SU: support for single updates; Batch: support for batch updates; MV: support for multi-versioning; Compact: support for compaction; Up. Store: the update storage, either In-place (IP) or Delta (D); Scans: performance in read workloads; Mem: memory consumption; Up. Perf.: performance in mutation

Systems	Infra.	Data Structure	SU	Batch	MV	Compact.	Up. Store	Scans	Mem	Up.Perf.
BGL[43]	SM	AL	+	-	-	+	IP	-	-	+
PGX.SM[36]	SM	CSR	-	+	+	-	D	+	+	-
Ligra[44]	SM	CSR	-	-	-	-	X	+	+	-
GraphLab[45]	Dist	CSR	-	-	-	-	X	-	+	-
PGX.D[37]	Dist	CSR	-	+	+	-	D	+	+	-
STINGER[20]	SM Dist	AL	+	+	-	-	IP	-	-	+
Hornet[27]	GPU	AL	+	+	-	-	IP	-	+	+
Madduri et. Al. [9]	SM	AL	-	+	-	+	IP	-	-	+
PCSR[42]	SM	CSR	+	-	-	+	IP	+	-	+
LLAMA[16]	SM	CSR	-	-	+	+	D	+	-	-
Metall[31]	SM	AL	-	+	+	+	D	-	+	+
GraphOne[40]	SM	AL+EL	+	+	+	+	D	-	+	+
Aspen[18]	SM	Tree	-	+	+	+	IP	-	+	+
Teseo[19]	SM	Tree	+	+	+	+	IP	+	-	+

c) Achieving optimal performance trade-offs

We have shown that the choice of the optimization techniques, in either graph representation or updates depends on the specific characteristics of the graphs, the types of workloads, and the constraints of the application environment. Most systems tend to improve on an aspect of performance, either read performance, updates throughput or memory consumption, and to barely achieve the best trade-off between all three of these aspects. For instance, the majority of the systems that we reviewed struggle to support several versions of the graph because of high memory cost, and graph compaction is necessary to reduce the amount of space needed for changes, however, this operation requires expensive computation and slows down the performance of the system.

d) Overlooked metrics and benchmarks

While memory layout techniques typically concentrate on graph topology, we stress the importance of aligning memory layout with the storage of additional graph data, such as user keys and properties. This alignment becomes particularly critical due to the widespread use of graph algorithms dealing with weighted graphs.

Furthermore, there is a notable emphasis on protocols for updating graph topology. However, it is infrequent to find comprehensive implementations and evaluations of performance when considering other graph data elements, such as graph properties, reverse edges, or intermediary results in graph algorithms.

7. RELATED WORK

The prevalence of graph processing and streaming in various domains has prompted researchers to conduct reviews to understand how graphs are used in practice [5], [35].

Graph Systems and Databases. Angles et al. [46] provide a comprehensive survey of graph database models, focusing on data structures, query languages, and integrity constraints. [5] review graph systems and classifies them based on their infrastructure. Furthermore, Besta et. al. [35] provide descriptions and analysis of different approaches for representing graphs in a streaming context. However, they do not analyze the performance trade-offs of different approaches for graph storage and mutation.

High-Performance Graph Representations. [47] propose theoretical frameworks for the study of data structure designs and the generation of new structures to better serve specific types of workloads. They highlight the applicability of their abstractions for key-value stores, however, they do not study the compositions for the graph model.

Moreover, Wheatman et al. [42] review existing graph representations such as CSR and adjacency lists. They provide a theoretical study of the time complexity of graph access operations as well as update operations on such data structures. However, they do not review how systems in the literature aim to optimize the performance of these graph data structures.

8. CONCLUSION

In this paper, we discuss the increasing focus within the literature on optimizing classical graph data structures to achieve high analytical and update performance while minimising memory usage. We provide an overview of techniques available to researchers and developers, highlighting their roles in enhancing graph processing and streaming. Additionally, we examine update protocols and the performance implications of different designs. Our main findings reveal challenges faced by systems offering multi-versioning in terms of compaction costs, the inadequate evaluation of auxiliary data structures like reverse edges and multiple graph properties, and the need for more advanced partitioning and compression algorithms for dynamic graph data structures. Furthermore, our survey leads us to suggest that systems cannot provide high performance for scans, updates and memory consumption simultaneously. As a future scope, we aim to develop a framework for graph processing and streaming that would address the gaps in the research literature, especially tackling the expensive cost of compaction for multi-versioning. Furthermore, we aim to extend our study by delving into algorithmic details of compres-

sion and partitioning for graph mutations and, performing quantitative analysis and benchmarks.

9. REFERENCES:

[1] C. Peng, F. Xia, M. Naseriparsa, F. Osborne, "Knowledge graphs: Opportunities and challenges", Artificial Intelligence Review, Vol. 56, No. 11, 2023, pp. 13071-13102.

[2] C. Gao et al., "A survey of graph neural networks for recommender systems: Challenges, methods, and directions", ACM Transactions on Recommender Systems, Vol. 1, No. 1, 2023, pp. 1-51.

[3] M. Saad, Y. Zhang, J. Tian, J. Jia, "A graph database for life cycle inventory using Neo4j", Journal of Cleaner Production, Vol. 393, 2023, p. 136344.

[4] S. Firmli et al. "CSR++: a Fast, Scalable, Update-Friendly Graph Data Structure", Proceedings of the 24th International Conference on Principles of Distributed Systems, Leibniz, Germany, 2021, pp. 1-16.

[5] S. Firmli, D. Chiadmi, "A Review of Engines for Graph Storage and Mutations", Innovation in Information Systems and Technologies to Support Learning Research: Proceedings of EMENA-ISTL 2019, 2020, pp. 214-223.

[6] Y. Cui, S. Kara, K. C. Chan, "Manufacturing big data ecosystem: A systematic literature review", Robotics and Computer-Integrated Manufacturing, Vol. 62, 2020, p. 101861.

[7] S. Phansalkar and S. Ahirrao, "Survey of data partitioning algorithms for big data stores", Proceedings of the Fourth International Conference on Parallel, Distributed and Grid Computing, Waknaghat, India, 22-24 December 2016, pp. 163-168.

[8] R. Dindokar, N. Choudhury, Y. Simmhan, "A meta-graph approach to analyze subgraph-centric distributed programming models", Proceedings of the IEEE International Conference on Big Data, Washington, DC, USA, 5-8 December 2016, pp. 37-47.

[9] K. Madduri, D. A. Bader, "Compact Graph Representations And Parallel Connectivity Algorithms For Massive Dynamic Network Analysis", Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, Rome, Italy, 23-29 May 2009.

[10] U. Drepper, "What every programmer should know about memory", Red Hat, Inc, Vol. 11, No. 2007, 2007.

[11] A. Bifet, R. Gavalda, G. Holmes, B. Pfahringer, "Machine learning for data streams: with practical examples in MOA", MIT press, 2023.

[12] W. Ju, J. Li, W. Yu, R. Zhang, "iGraph: an incremental data processing system for dynamic graph", Frontiers of Computer Science, Vol. 10, No. 3, 2016, pp. 462-476.

[13] C. L. Staudt, A. Sazonovs, H. Meyerhenke, "NetworKit: a Tool Suite For Large-Scale Complex Network Analysis", Network Science, Vol. 4, No. 4, 2016, pp. 508-530.

[14] S. Firmli et al. "CSR++: A Fast, Scalable, Update-Friendly Graph Data Structure", Proceedings of the 24th International Conference on Principles of Distributed Systems, 2020.

[15] R. Raman, O. van Rest, S. Hong, Z. Wu, H. Chafi, J. Banerjee, "PGX.ISO: parallel And Efficient In-memory Engine For Subgraph Isomorphism", Proceedings of Workshop on GRAph Data management Experiences and Systems, Snowbird, UT, USA, 22-27 June 2014.

[16] P. Macko, V. J. Marathe, D. W. Margo, M. I. Seltzer, "Llama: Efficient graph analytics using large multiversioned arrays", Proceedings of the IEEE 31st International Conference on Data Engineering, Seoul, Korea, 13-17 April 2015, pp. 363-374.

[17] B. Wheatman, H. Xu, "A Parallel Packed Memory Array to Store Dynamic Graphs", Proceedings of the Proceedings of the Symposium on Algorithm Engineering and Experiments, pp. 31-45.

[18] L. Dhulipala, G. E. Blelloch, J. Shun, "Low-Latency Graph Streaming Using Compressed Purely-Functional Trees", Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, New York, NY, USA, 2019, pp. 918-934.

[19] D. De Leo, P. Boncz, "Teseo and the Analysis of Structural Dynamic Graphs", Proceedings of the VLDB Endowment, Vol. 14, No. 6, 2021, pp. 1053-1066.

[20] D. Ediger, R. McColl, J. Riedy, D. A. Bader, "Stinger: High performance data structure for streaming graphs", Proceedings of the IEEE Conference on High Performance Extreme Computing, Waltham, MA, USA, 10-12 September 2012, pp. 1-5.

[21] S. Firmli, D. Chiadmi, "A Scalable Data Structure for Efficient Graph Analytics and In-Place Mutations", Data, Vol. 8, No. 11, 2023.

[22] P. Boldi, S. Vigna, "The Webgraph Framework I: Compression Techniques", Proceedings of the 13th International Conference on World Wide Web, New York, NY, USA, 2004, pp. 595-602.

[23] J. Shun, L. Dhulipala, G. E. Blelloch, "Smaller and Faster: Parallel Processing of Compressed Graphs with Ligra+", Proceedings of the Data Compression Conference, 2015, pp. 403-412.

[24] B. Wheatman, R. Burns, "Streaming sparse graphs using efficient dynamic sets", Proceedings of the IEEE Interna-

tional Conference on Big Data, Orlando, FL, USA, 15-18 December 2021, pp. 284-294.

[25] E. D. Berger, K. S. McKinley, R. D. Blumofe, P. R. Wilson, "Hoard: A scalable memory allocator for multithreaded applications", ACM Sigplan Notices, Vol. 35, No. 11, 2000, pp. 117-128.

[26] M. S. Johnstone, P. R. Wilson, "The memory fragmentation problem: Solved?", ACM Sigplan Notices, Vol. 34, No. 3, 1998, pp. 26-36.

[27] F. Busato, O. Green, N. Bombieri, D. A. Bader, "Hornet: An Efficient Data Structure for Dynamic Sparse Graphs and Matrices on GPUs", Proceedings of hte IEEE High Performance extreme Computing Conference, Waltham, MA, USA, 25-27 September 2018, pp. 1-7.

[28] I. U. Akgun, A. S. Aydin, A. Shaikh, L. Velikov, E. Zadok, "A machine learning framework to improve storage system performance", Proceedings of the 13th ACM Workshop on Hot Topics in Storage and File Systems, 2021, pp. 94-102.

[29] D. Durner, V. Leis, T. Neumann, "On the Impact of Memory Allocation on High-Performance Query Processing", Proceedings of the 15th International Workshop on Data Management on New Hardware, New York, NY, USA, July 2019, pp. 1-3.

[30] S. Lee, T. Johnson, E. Raman, "Feedback directed optimization of TCMalloc", Proceedings of the workshop on Memory Systems Performance and Correctness, New York, NY, USA, June 2014, pp. 1-8.

[31] K. Iwabuchi, K. Youssef, K. Velusamy, M. Gokhale, R. Pearce, "Metall: A persistent memory allocator for data-centric analytics", Parallel Computing, Vol. 111, 2022, p. 102905.

[32] A. Kyrola, G. Blelloch, C. Guestrin, "GraphChi: large-Scale Graph Computation on Just a PC", Proceedings of the 10th USENIX Symposium on Operating Systems Design and Implementation, 2012.

[33] A. Roy, L. Bindschaedler, J. Malicevic, W. Zwaenepoel, "Chaos: Scale-out graph processing from secondary storage", Proceedings of the 25th Symposium on Operating Systems Principles, 2015, pp. 410-424.

[34] L. M. Vaquero, F. Cuadrado, M. Ripeanu, "Systems for near real-time analysis of large-scale dynamic graphs", arXiv:1410.1903, 2014.

[35] M. Besta, M. Fischer, V. Kalavri, M. Kapralov, T. Hoefler, "Practice of streaming and dynamic graphs: Concepts, models, systems, and parallelism", arXiv:1912.12740, 2020.

[36] S. Hong, H. Chafi, E. Sedlar, K. Olukotun, "Green-Marl: a DSL For Easy And Efficient Graph Analysis", ACM SIGPLAN Notices, Vol. 47, No. 4, 2012, pp. 349-362.

[37] N. P. Roth et al. "PGX.D/Async: a Scalable Distributed Graph Pattern Matching Engine", Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems, Chicago, IL, USA, 19 May 2017.

[38] M. Haubenschild, M. Then, S. Hong, H. Chafi, "Asgraph: a mutable multi-versioned graph container with high analytical performance", Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems, Redwood Shores, CA, USA, 24 June 2016, pp. 1-6.

[39] A. Adya, B. Liskov, P. O'Neil, "Generalized isolation level definitions", Proceedings of 16th International Conference on Data Engineering, San Diego, CA, USA, 29 February - 3 March 2000, pp. 67-78.

[40] P. Kumar, H. H. Huang, "GraphOne: a Data Store for Real-Time Analytics on Evolving Graphs", ACM Transactions on Storage, Vol. 15, No. 4, 2020.

[41] M. A. Bender, H. Hu, "An Adaptive Packed-Memory Array", ACM Transactions on Database Systems, Vol. 32, No. 4, 2007, pp. 26-es.

[42] B. Wheatman, H. Xu, "Packed Compressed Sparse Row: a Dynamic Graph Representation", Proceedings of the IEEE High Performance extreme Computing Conference, Waltham, MA, USA, 25-27 September 2018.

[43] "The Boost Graph Library - 1.75.0.", https://www.boost.org/doc/libs/1_75_0/libs/graph/doc/index.html (accessed: 2024)

[44] J. Shun, G. E. Blelloch, "Ligra: a lightweight graph processing framework for shared memory", Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming, 2013, pp. 135-146.

[45] Y. Low, J. E. Gonzalez, A. Kyrola, D. Bickson, C. E. Guestrin, J. Hellerstein, "Graphlab: A new framework for parallel machine learning", arXiv:1408.2041, 2014.

[46] R. Angles, C. Gutierrez, "An Introduction to Graph Data Management", Data-Centric Systems and Applications, Springer International Publishing, 2018, pp. 1-32.

[47] S. Idreos et al. "The Periodic Table of Data Structures", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 41, No. 3, 2018, pp. 64-75.

Vector Control of the Induction Motor Based on Whale Optimization Algorithm

Original Scientific Paper

Fadhil A. Hasan*
University of Technology- Iraq,
Department of Electrical Engineering
Baghdad, Iraq
fadhil.a.hasan@uotechnology.edu.iq

*Corresponding author

Abstract *– This paper presents the Whale Optimizing Algorithm (WOA) to improve the performance of the induction motors through vector control (VC). The optimization algorithm is utilized to tune the proportional-integral (PI) controllers in both the outer and inner controlling loops. The parameters of these controllers are crucial components of the control system. The WOA is inspired by the social behavior of humpback whales, which is a powerful meta-heuristic algorithm as compared to other techniques. The controlling system and the WOA are implemented using MATLAB-SIMULINK environments. Simulation results demonstrate that this approach significantly improves both dynamic and steady-state responses of the induction motor compared to other optimization techniques. Simultaneously, the success of the WOA in reaching the global optimal parameters can be realized by the significant reduction in computation time and iterations as compared to other methods. The results show a considerable enhancement of about 2% in rise time, 30% in overshoot, and 60% in settling time in accelerating mode in conjunction with a reference case. Also, it gives an improvement of about 9% in rise time, 11% in overshoot, and 64% in settling time in step response. This research contributes to the field of motor control by providing an efficient and reliable optimization method for enhancing the performance of induction motors for various industrial applications.*

Keywords: *Field-oriented control, Whale optimization, Induction motor, Optimum control*

Received: February 19, 2024; Received in revised form: August 28, 2024; Accepted: September 17, 2024

1. INTRODUCTION

Induction motors (IM) have many technical features such as; low cost, minimum maintenance, high efficiency, durable build, and extended operating life for medium and high loads. Accordingly, they were employed in wide fields of industrial applications. Controlling strategies for electrical machines utilize several techniques depending on the machine's type and performance requirements. To achieve high-performance characteristics with these control approaches, various machine and controller parameters must be optimized [1]. These factors vary from one operation status to the other, such as temperature, saturation, frequency, and skin effect which impact on those parameters. As a result, the influence of all parameter variations in field-oriented control (FOC) for an induction motor at constant flux, field weakening regions, starting mode, and full-load condition is important in this article. The induction motor with field-oriented control or vector control (VC) emulates the direct current motor, and the stator and rotor flux components are vertical, so FOC reserves high dynamic operation for IM controllers. This controlling method keeps the percentage of voltage and flux in the airgap at its rated level. The motor's physical components like flux, current, and voltage are changed as space vectors [1, 2].

Driving the IM for higher than the rated speeds, the applied voltage is maintained constant as the frequency rises, as in Scalar Control (SC). Even though vector control gives excellent performance and ensures high dynamic response, this behavior may be lost if the actual parameters are mismatched with the estimated parameters that are utilized in the controller. Hamdy [2], looks at how different parameters affect how well an induction motor with FOC at start-up and full load performs.

Vector control is a control strategy used to regulate the torque and flux of IM independently. Proportional-integral (PI) controllers are widely used in these control systems. The parameters of a PI controller need to be tuned to achieve the desired system performance. Tuning these parameters can be a challenging task,

for this goal, there are several techniques presented in the literature; Hasan et al. [3], present a combination of Kharitonov's theorem and particle swarm optimization (PSO) which improves the dynamic performance due to parameter variation. Dhaouia et al. [4], used the sliding mode based on a MARS controller with intelligent ANFIS which improves the transient response. Salih et al. [5], present a PI controller based on an artificial neural network to control the synchronous machine, this gives high dynamic response and stability.

In addition to such methods, a variety of optimization techniques have been used to tune PI controllers. The majority of optimization methods are classified as "computational intelligence algorithms," with stochastic gradient descent as the primary method of computation [6]. To produce intelligent programs, the strategy must incorporate principles of training, adaptability, and evolution. One of the most important features of these methods is that they help to find the best solution to complex optimization problems faster than standard optimization methods [7]. Mehedi et al. [8], present a comprehensive analysis of improved FOC incorporating intelligent controllers, which have led to improved performance metrics compared to traditional control methods. Shaija Daniel [9], utilizes two nature-inspired optimization algorithms, Gray Wolf Optimization (GWO) and Teaching-Learning-Based Optimization (TLBO), for the optimal tuning of PI controllers. The results show an improvement in the performance of the IM drives in terms of speed control, efficiency, and dynamic response. Albalawi et al. [10], present the application of ant colony optimization (ACO) to enhance the the steady-state and dynamic response. Tiacharoen et al. [11], propose applying the Bee optimization technique to the design of the FOC and the development of an intelligent control system. The results demonstrate improved efficiency and effectiveness in the control system, showcasing, the potential of Bee optimization as an achievable optimization method. Mohamed et al. [12], present genetic and PSO algorithms in the context of direct torque control of the IM. This method leads to improving the torque, speed, and torque ripple. Employing those optimization techniques is simple, able to avoid local optima, and compatible with many application problems in various fields, thus those algorithms have become popular.

Mirjalili et al. [13], present a new heuristic method called whale optimization algorithm that emulates the hunting behavior of the whales. this method is population-based and utilizes the seeking modes of exploration and exploitation. The WOA shows a competitive performance in terms of convergence, accuracy, and simplicity. The authors emphasize the effectiveness of the WOA in various domains, including engineering and software applications. The results demonstrate the significance of the WOA as a valuable addition to the group of optimization algorithms, presenting a nature-inspired approach to solving complex optimization problems. The comprehensive evaluation and positive outcomes make WOA a noteworthy choice for researchers and practitioners in optimization-related fields.

In this work, we investigate the utilization of the whale optimization algorithm to tuning the PI controllers in the vector control strategy, for the induction motor. The main goal emphasizes searching for the optimum values of the controller parameters that enhance both the steady-state and dynamic response. The evaluation criteria of the motor performance will be the integral time square error fitness function. This approach is the contribution of this work, which is characterized by simplicity compared to the complex and sophisticated optimization methods presented in previous works. The effectiveness of the proposed method is demonstrated by comparing it with the method used PSO technique presented in [3]. The challenge is achieving a balance between dynamic performance (rise time, overshoot, and settling time) and system stability, which traditional methods often fail to adequately resolve. The WOA's helical searching path is particularly effective in avoiding local optima, which helps prevent the algorithm from diverging and oscillating. This characteristic significantly reduces the computation time and number of iterations over other optimization techniques these are the main contribution of this work.

The paper is organized as follows; Section 2 presents the principles of vector control; Section 3 provides the adopted methodology in detail. Different types of fitness functions of the optimization algorithm are discussed in Sections 4; Sections 5 and 6 present the simulation results and conclusions.

2. VECTOR CONTROL

The induction motor may be operated with excellent transient behavior using vector or field-oriented control (FOC). It converts the IM's dynamic structure to an independent excitation DC machine [14]. The magnetic field is a function of the excitation current in a DC motor. Therefore, if the magnetic field is considered to maintain invariant and autonomous of the armature current, then the torque can be directly proportion to the magnetic field and armature current as [14, 15]:

$$T_e = k_m \cdot I_f \cdot I_a \tag{1}$$

Where; k_m is the machine constant, I_f is the field current and I_a is the armature current. It appears like an independently stimulated D.C motor when the induction motor is converted to the d-q plane. The (FOC) method separates the stator current into two parts: the first provides air gap magnetic flux and the second generates torque. The characteristics of these current components are linear [15], and they allow independent flux and torque control. Before returning to the rotor, these components are moved to the stator frame. The two components correspond to the field current on the

d-axis and the armature current on the q-axis of a separately activated DC motor [14]. As indicated in the phasor diagram in Fig. 1, the rotor field vector can be aligned along the d-axis. Fig. 2 shows the principles of vector control realization. The concept is represented in a synchronously rotating reference frame, and the inverter generates the supply voltages (v_a, v_b, v_c) proportional to the reference control signals (v_a^*,v_b^*,v_c^*) The stator current components (flux and torque) are employed as system controlling signals i_{ds}^{s*} & i_{qs}^{s*} respectively, which are reconverted to three-phase reference currents (i_a^*, i_b^*, i_c^*) and then converted to reference voltages (v_a^*,v_b^*,v_c^*) meanwhile the PI controllers [14, 15]. FOC can be utilized in one of two ways: directly or indirectly. The main difference is how they estimate the vectors ($cos\theta_e$ and $sin\theta_e$).

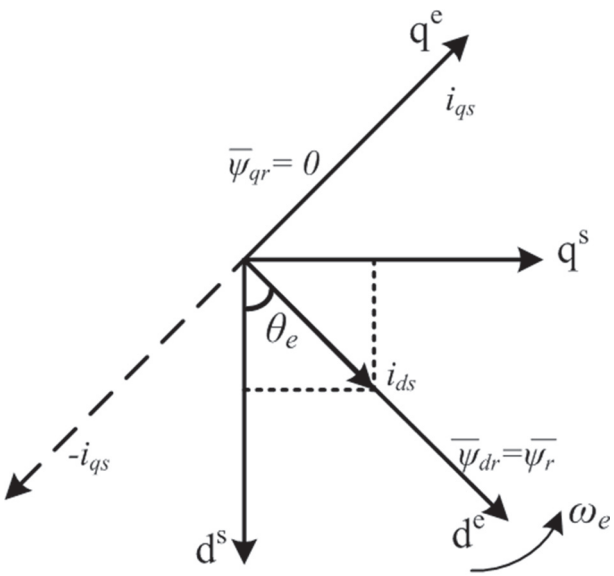


Fig. 1. Correct Rotor Flux Orientation

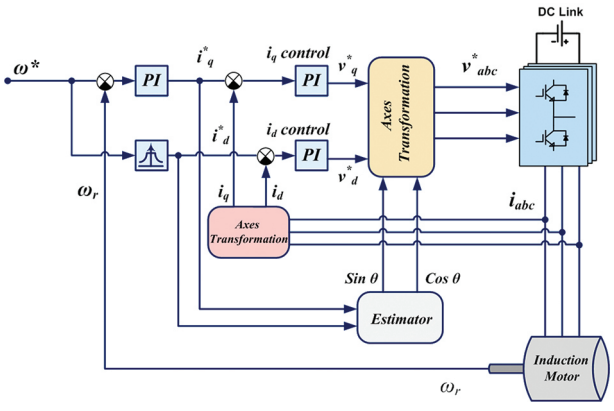


Fig. 2. Field Oriented Vector Control

3. WHALE OPTIMIZATION TECHNIQUE

Mirjalili and Lewis [13] proposed the whale optimization technique for solving numerical problems. The system mimics humpback whale intelligence hunting activity. This type of eating activity is known as "bubble-net feeding" and is only seen in humpback whales. While surrounding prey during hunting, the whales blow bubbles in a circular pattern. Simply put, bubble-net hunting techniques involve humpback whales diving down around 12 meters, creating a spiral-shaped bubble around their prey, and then swimming up to the surface, tracking the bubbles. With the view of achieving these optima, the helical bubble-net can be mathematically modeled for hunting activity as in the following [5, 10, 16, 17]:

3.1. ENCIRCLING PREY

Humpback whales can track down and surround their prey. The WOA evaluates the current adequate tracking agents' status to be the objective target or near the optimal position, while the rest of the track agents will attempt to modify their location about the best search agent. The following equations describe this behavior [18]:

$$\vec{S} = \left|\vec{C} \cdot \vec{Y}^*(k) - \vec{Y}(k)\right| \quad (2)$$

$$\vec{Y}(k+1) = \vec{Y}^*(k) - \vec{A} \cdot \vec{S} \quad (3)$$

Where; k is the iteration pointer, $\vec{Y}^*$ is the location vector of the best solution that has been found till the current iteration k, $\vec{Y}$ is the location vector of each agent. The factors $\vec{A}$ and $\vec{C}$ are determined as following:

$$\vec{A} = 2\vec{a} \cdot b - \vec{a} \quad (4)$$

$$\vec{C} = 2b \quad (5)$$

Where; $\vec{a}$ is reduced from 2 to 0 due to the iteration process, and b is a random value between 0 and 1.

3.2. ATTACKING WITH A BUBBLE-NET MECHANISM

The bubble-net technique is a mix of two different methods that can be modeled mathematically as the following [13]:

a. Shrunk encircling technique

By reducing the value of $\vec{a}$ in the equation, this behavior of whales may be emulated in Equation (4). It's worth noting that $\vec{a}$ reduces the fluctuation range $\vec{A}$. In other words, $\vec{A}$ is a random number in the interval [-a, a] where $\vec{a}$ is reduced from 2 to 0 during the duration of repetitions. The new location of a search agent can be defined anywhere between the initial location of the agent and the location of the current best agent by using random values for $\vec{A}$ in [-1, 1].

b. Upgrading the spiral location

To move like humpback whales, a spiral equation is set up between the location of the whale and the location of the prey, as shown:

$$\vec{S}' = \left|\vec{Y}^*(k) - \vec{Y}(k)\right| \quad (6)$$

$$\vec{Y}(k+1) = \vec{S}' \cdot e^{mn} \cdot cos(2\pi n) + \vec{Y}^*(k) \quad (7)$$

Where; $\vec{S}'$ is the distance between the whale and the target, m is a constant that specifies the logarithmic form, and n is random in the range [-1, 1].

Humpback whales do move in a spiral-shaped pattern while also swimming within a diminishing circle. Selecting the decreasing circular motion or the helical model tendency can be simulated throughout the rounds of the program by assuming a possibility of 50%. That is,

$$\vec{Y}(k+1) = \begin{cases} \vec{Y}^*(k) - \vec{A} \cdot \vec{S} & ;if \quad g < 0.5 \\ \vec{S}' \cdot e^{mn} \cdot cos(2\pi n) + \vec{Y}^*(k) & ;if \quad g \geq 0.5 \end{cases} \quad (8)$$

Where; g is a random number between 0 and 1.

3.3. SEARCHING FOR TARGETS

Most of the meta-heuristic techniques use random selection to get the best solution. Because the location of the optimum design in the bubble-net approach is unknown, whales explore for targets at random. In difference to the exploiting step, which is used $\vec{A}$ in the region between [-1, 1], this step uses $\vec{A}$ as a random values vector larger than or equal to -1. According to this assumption, a hunting agent can go a long distance far from the reference whale. In exchange, the location of the hunt agent is updated based on a random selection of search agents, rather than the best search agent discovered thus far. These two activities can be expressed as follows [13, 17, and 18]:

$$\vec{S} = \left| \vec{C} \cdot \vec{Y}_{rand} - \vec{Y} \right| \quad (9)$$

$$\vec{Y}(k+1) = \vec{Y}_{rand} - \vec{A} \cdot \vec{S} \quad (10)$$

Where; $\vec{Z}_{rand}$ represents a vector of random locations.

Whale optimization process begins with a set of randomly generated populations. The searching agents change their location according to the preceding reasons, at each iteration. WOA is an optimizer that works on a global scale. The WOA algorithm can quickly transition between exploration and exploitation due to adaptive change of the search vector $\vec{A}$. Furthermore, WOA only has two significant internal settings that may be changed. WOA's high exploration capabilities are due to the whales' position update system Equation (10). The effect of high exploitation and convergence is obtained from the derivation of Equations (7 and 8). Those equations demonstrate the WOA is very good at avoiding local optima and getting to the next solution quickly during each iteration.

3.4. WHALE OPTIMIZATION PROCEDURE

The algorithm can be explained as follows:

- Beginning by generating a collection of random agents for each variable.
- Extract a set of solutions and compare the current solutions with the best-obtained solution then update the agent's position accordingly.
- Reducing the factor $\vec{a}$ from 2 to 0 to perform the shrinking encircling technique.
- Inspect the value of g an exchange between the shrinking technique and the helical technique.
- If $A \geq 1$, select an arbitrary search agent and if $A < 1$, select the best solution to update the location of searching agents.
- Ending the process when the satisfied termination constraint is achieved.

4. FITNESS FUNCTION

The error criteria are used to determine the fitness functions. There are a lot of criteria for evaluating controller performance, and the Integral time of Absolute Error (ITAE) criterion is used in this study [19]. The equation gives the measure cumulative error of the motor performance at certain operation intervals:

$$FF_{ITAE} = \int_0^T t|e(t)|\,dt \quad (11)$$

The ITAE uses a time-weighted error weighting scale that gives more weight to error values at a certain time (T) to account for the predicted steady-state time.

The second performance metric is the Integral Square Errors (ISE) constraint.

$$FF_{ISE} = \int_0^T (e(t))^2\,dt \quad (12)$$

There are also absolute Errors (ITSE) compounded by the Integral of Time (T)

$$FF_{ITSE} = \int_0^T t(e(t))^2\,dt \quad (13)$$

The period spans from 0 to T, with T being the amount of interval for a unit step input to bring the system to a steady state.

As shown by the MSE (mean square error),

$$FF_{MSE} = \frac{1}{T}\int_0^T (e(t))^2\,dt \quad (14)$$

All these fitness functions represent a comprehensive evaluation both of dynamic and steady-state responses. It is worthwhile to say that when the advantage of a fast system needs minimum achievable values for the transient response, then the maximum overshoot, settling time, and rising time are typically regarded as crucial.

5. SIMULATION RESULTS

The presented method of developing the controllers of the vector control of the IM is recognized by being implemented using the MATLAB-SIMULINK program. The optimization approach was accomplished in offline mode to evaluate the controllers' parameters singly. The nameplate parameters of the investigated induction machine are listed in Table 1.

Table 1. Machine parameters

Motor Parameters	Value
Rated Torque	800 Nm
Frequency	50 Hz
Rated voltage	460 V (line-line)
Number of poles	4
Stator inductance	0.302 mH
Stator winding resistance	14.8 mΩ
Rotor inductance	0.303 mH
Rotor resistance	9.3 mΩ
Mutual inductance	10.5 H
Rotor inertia	3.1 kgm2
Friction coefficient	0.08 N.m.s
Nominal flux	0.73 Wb

The parameters of the WOA were adjusted to 100 inspection agents, 100 iterations, and two searching dimensions representing the two variables (k_p, k_i) of the three PI controllers in the FOC system. The optimization process was utilized by developing a MATLAB sub-routine in offline mode. The movement in the whale optimization searching agents during the 100 iterations is illustrated in Fig. 3. Obviously, the agents quickly collected toward the global optima in the helical tracking path as shown by the red solid curve. This helical trajectory path gives the strength of this technique by bypassing all local optimal points which prevent the algorithm from divergence and oscillation.The absolute locations of the agents after ending the process are listed in Table 2, which presents the founded optimum parameters for the three PI controllers in the system (speed, torque, and flux controllers).

Table 2. The optimum parameters

Controller	k_p	k_i
Speed	735.4	8970.8
Torque (iq)	2.3	11.6
Flux (id)	80.5	34.2

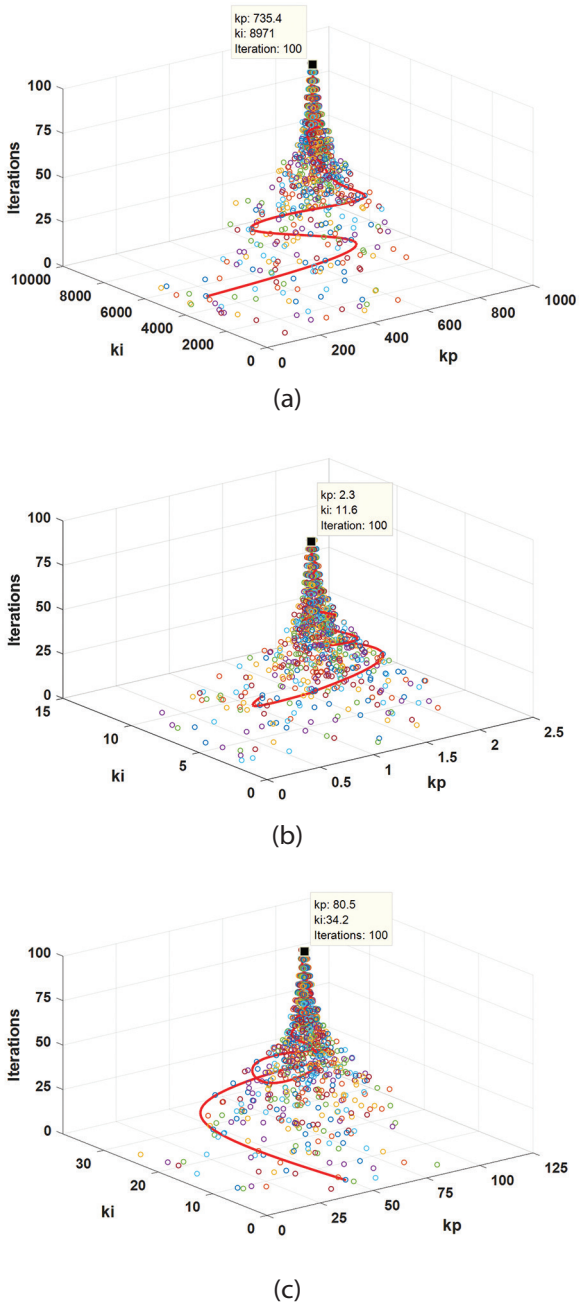


Fig. 3. Optimization agents' trajectories; (a) Speed controller parameters, (b) Torque controller parameters, (c) Flux controller parameters.

To investigate the effectiveness of the obtained parameters of the FOC controllers, the overall system must be tested by different operating modes. Firstly, the controller investigates the four-quadrant operation mode at the full and no-load conditions as depicted in Fig. 4. The figure illustrates the rotor speed, the shaft torque, and phase current. Obviously, from this, the response shows a fast starting time of about 0.17s, a small overshoot of approximately 1% over the nominal speed, and a steady-state time of 0.21s. Secondly, to realize the features of the proposed controller a comparison performance in conjunction with the FOC model presented in [3], in which a robust PI-PSO controller of the IM motor was presented. The accelerating and decelerating operation under full-load conditions is shown in Fig. 5. From this comparison one can note that the motor speed is perfectly tracking the reference speeds. Also, the estimated and actual electromagnetic torque is identical to the load torque. Moreover, the step response comparison between the proposed controller and reference case is shown in Fig. 6. The dynamic constraints obtained from these comparison cases are summarized in Table 3.

Finally, the reference speed-tracking performance for different speed commands, at full-load conditions, is illustrated in Fig. 7. The dynamic constraints obtained from these comparison cases are summarized in Table 3.

Table 3. Time domain constraints comparison

Constraints	Ramp response		Step response	
	Proposed controller	Matlab example	Proposed controller	Matlab example
Rise time (s)	0.455	0.465	0.145	0.16
Overshoot %	0.7	1	0.8	0.9
Shoot time (s)	0.46	0.466	0.154	0.17
Settling time (s)	0.5	0.8	0.18	0.5

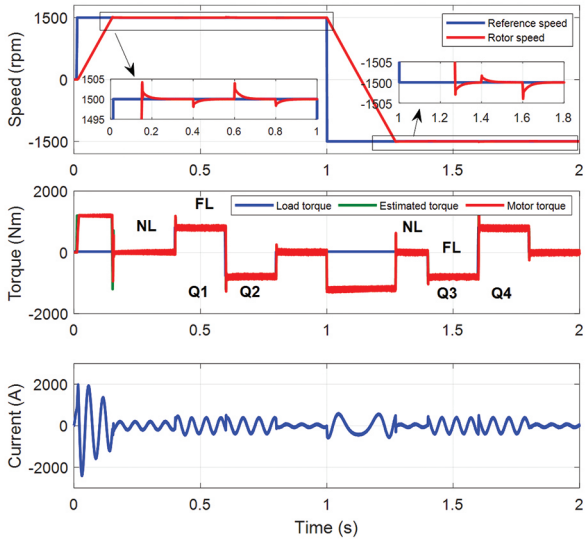


Fig. 4. Four-quadrant operating mode

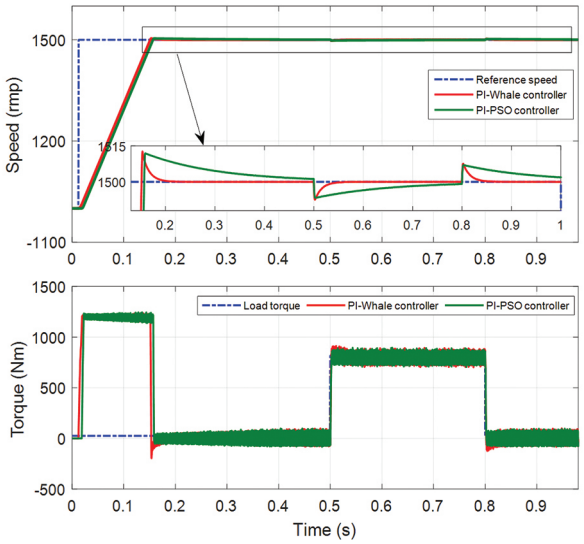


Fig. 6. Step response comparison

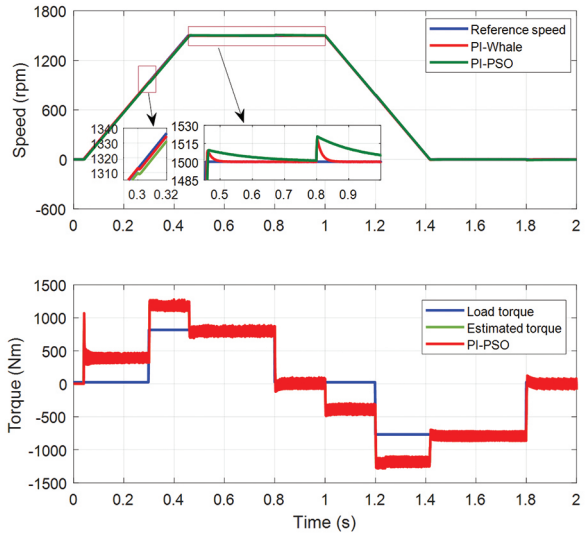


Fig. 5. Accelerating and decelerating comparison

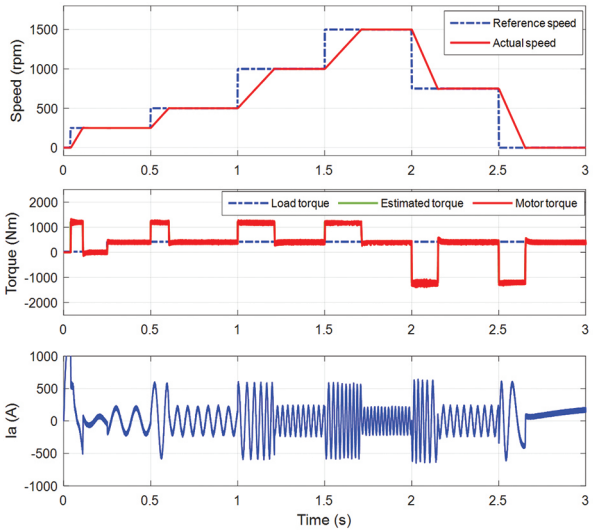


Fig. 7. Different command speed tracking

6. CONCLUSIONS

The paper proposes a metaheuristic approach for enhancing the performance of a three-phase induction motor. Particularly, the research interests are in applying the whale optimization method to control the IM utilizing the VC method. The optimized control strategy incorporates a searching process for the PI controller parameters. The WOA shows a unique behavior as compared with other optimization techniques, in terms of the number of iterations and fast computing time. The helical searching path of the population agents enables them to bypass the local optima regions, which prevents the algorithm from divergence and oscillation. The obtained PI controllers showcased enhanced performance in controlling motor speed. Compared to the PI-PSO, the proposed PI-WOA gives an improvement of about 2% in rise time, 30% in overshoot, and 60% in settling time in accelerating mode. Also, it gave an improvement of about 9% in rise time, 11% in overshoot, and 64% in settling time in step response in conjunction with the reference case. This suggests that the WOA method effectively fine-tuned the control parameters, leading to better dynamic response and overall motor performance.

7. REFERENCES

[1] R. Krishnan, A. S. Bharadwaj, "A Review of Parameter sensitivity and adaptation in Industrial vector controlled Induction motor drive system", IEEE Transactions on Power Electronics, Vol. 6, No. 4, 1991, pp. 219-225.

[2] H. M. Soliman, "Effect of the Parameters Variation for Induction Motor on its Performance Characteristics with Field Oriented Control Compared to Scalar Control", International Journal of Engineering Research & Science, Vol. 2, No. 8, 2016.

[3] F. A. Hasan, A. T. Humod, L. J. Rashad, "Robust decoupled controller of induction motor by combining PSO and Kharitonov's theorem", Engineering Science and Technology, an International Journal, Vol. 23, No. 6, 2020, pp. 1415-1424.

[4] A. Salih, A. Humod, F. A. Hasan, "Optimum Design for PID-ANN Controller for Automatic Voltage Regulator of Synchronous Generator", Proceedings of the 4th Scientific International Conference Najaf, Al-Najef, Iraq, 29-30 April 2019, pp. 74-79.

[5] N. H. M. Nezhad, M. G. Niasar, C. W. Hagen, P. Kruit, "Comparison of Different Optimization Techniques in Electron Lens Design", Proceedings of the 9th International Conference on Optimization and Applications, AbuDhabi, United Arab Emirates, 2023, pp. 1-5.

[6] A. Kumar, S. S. Badini, V. Verma, "Comparative Evaluation of Different Meta-heuristic Optimization Based Tuning of PI Controllers for Vector Controlled Induction Motor Drive", Proceedings of the 2nd International Conference on Emerging Frontiers in Electrical and Electronic Technologies, Patna, India, 24-25 June 2022, pp. 1-6.

[7] I. M. Mehedi, N. Saad, M. A. Magzoub, U. M. Al-Saggaf, A. H. Milyani, "Simulation Analysis and Experimental Evaluation of Improved Field-Oriented Controlled Induction Motors Incorporating Intelligent Controllers", IEEE Access, Vol. 10, 2022, pp. 18380-18394.

[8] P. J. Shaija, A. E. Daniel, "Optimal Tuning of PI Controllers for IM Drive using GWO and TLBO Algorithms", Proceedings of the Fifth International Conference on Electrical, Computer and Communication Technologies, Erode, India, 22-24 February 2023, pp. 1-9.

[9] H. Albalawi, S. A. Zaid, M. E. El-Shimy, A. M. Kassem, "Ant Colony Optimized Controller for Fast Direct Torque Control of Induction Motor", Sustainability, Vol. 15, No. 4, 2023, p. 3740.

[10] S. Tiacharoen, T. Chatchanayuenyong, "Design and development of an intelligent control by using bee colony optimization technique", American Journal of Applied Sciences, Vol. 9, No. 9, 2012, p. 1464.

[11] M. Elgbaily, F. Anayi, M. Packianather, "Genetic and particle swarm optimization algorithms based direct torque control for torque ripple attenuation of induction motor", Materials Today: Proceedings, Vol. 67, No. 4, 2022, pp. 577-590.

[12] S. Mirjalili, A. Lewis, "The Whale Optimization Algorithm", Advances in Engineering Software, Vol. 95, 2016, pp. 51-67.

[13] B. Wu, M. Narimani, "Control of Induction Motor Drives", High-Power Converters and AC Drives, Wiley-IEEE Press, 2017, pp. 321-352.

[14] P. Girovsky, J. Timko, J. Zilková, V. Fedák, "Neural Estimators for Shaft Sensorless FOC Control of Induction Motor", Proceedings of 14th International Power Electronics and Motion Control Conference, Ohrid, Macedonia, 6-8 September 2010, pp. T7-1-T7-5, 2010.

[15] A. M. Mosaad, M. A. Attia, A. Y. Abdelaziz, "Whale optimization algorithm to tune PID and PIDA controllers on AVR system", Ain Shams Engineering Journal, Vol. 10, No. 4, 2019, pp. 755-767.

[16] H. Kamil et al. "Design and Simulation of a Portable Copper Tubes Induction Brazing Tool for PV System Application", Journal of Physics: Conference Series, Vol. 1804, 2021.

[17] A. A. Kumar, S. M. G. Kumar, "Application of Whale Optimization Algorithm for tuning of a PID controller for a drilling machine", Proceedings of the International Conference on Advancements in Automation, Robotics & Sensing, Coimbatore, India, 13-15 December 2018.

[18] M. M. Mwembeshi, C. A. Kent, S. Salhi, "A genetic algorithm-based approach to intelligent modeling and control of pH in reactors", Computers & Chemical Engineering, Vol. 28, No. 9, 2004.

[19] F. A. Hasan, H. Q. Hameed, L. J. Rashad, "Robust field oriented control of PMSM using lyapunov theorem and particle swarm optimization", Journal Européen des Systèmes Automatisés, Vol. 57, No. 2, pp. 487-496.

Time Domain and Qualitative Analysis of a Compact Asymmetrically Fed Circular UWB Antenna for WBAN Scenarios

Original Scientific Paper

Venkatesh P*
Ramco Institute of Technology, Department of Electronics and Communication Engineering
Rajapalayam, India
pvenkatesh@ritrjpm.ac.in

Ponnrajakumari M
Velammal Engineering College, Department of Electronics and Communication Engineering
Chennai, India.
ponnrajakumari23@gmail.com

Narmadha T V
St.Joseph's College of Engineering, Department of Electronics and Communication Engineering
Chennai, India
hodeeelabaffairs@stjosephs.ac.in

Lavanya K
Velammal Engineering College, Department of Electronics and Communication Engineering
Chennai, India.
lavanya201180@gmail.com

*Corresponding author

Abstract *– This article presents about the design, time domain analysis and qualitative analysis of a circular monopole Ultra Wideband (UWB) antenna for Wireless Body Area Networks (WBAN) applications. The size of the proposed antenna is 30 x 30 x 1 mm3. The proposed antenna provides Ultra wide bandwidth from 2 – 10 GHz and also complies with IEEE C95.3 safety standards. The simulated and measured results are close to each other with minimum deviations. To ensure proper and less distorted communication in real time scenarios, time domain analysis was done for free space, on body and off body conditions. The magnitude and phase of transmission coefficient (S21) were found to be consistent. The group delay was analyzed under free space, on &off body conditions whose variations are less than 0.5 ns in the entire UWB range. Fidelity factor was also analyzed for flat and bent conditions to ensure pulse similarity. Also to ensure the communication link quality, the obtained minimum path loss of 51.10 dB and maximum Received Signal Power of 0.17 dBm were found to be satisfactory warranting a good transmission and reception characteristics of the proposed antenna.*

Keywords*: Group delay, Link budget, Link margin, On/Off body, Ultra Wideband*

Received: July 8, 2024; Received in revised form: September 6, 2024; Accepted: September 19, 2024

1. INTRODUCTION

Because of the fast moving life style of the people, Bio telemetry is creating remarkable footprints. To implement proper bio telemetry, a proactive health management is required. Wireless Body Area Network has gained a keen attention because of the major role played by it in health care systems. This is because it finds its application in various sectors like Bio health care, Telemedicine, sports, multimedia applications [1] and indoor data transmission [2]. WBAN communications can classified into 3 types on the basis of placing the antenna in/on the body and the method of communication with other devices. One is in body communication, where communication is established between the antenna located inside the body and another antenna located outside but on the body surface. Next is the on-body communication where two or more devices placed on the surface of the human body communicates with each other. Lastly, the off- the device located on the human body surface and an outward device or a network located away from the surface of the human body at a certain distance. Antennas that are capable of supporting both on and off body communications are of primary importance [3].

To implement such a network with good performance, the Ultra Wideband technology will be highly suitable. Ultra Wideband (UWB) antennas shall be one of the right candidates because of the advantages like compactness, low profile, large bandwidth, high data rate, low power consumption, less penetrating effects. Announcement from Federal Communication Commission (FCC) for utilizing the frequency range 3.1 – 10.6

GHz has led to such advantages. Though UWB antennas offer certain advantages, they also suffer from certain kind of short falls. Since UWB communications use short pulses with less time duration for transmission and reception, there are possibilities for those pulses being distorted easily.

Hence analyzing the transmission characteristics (time domain analysis) of an UWB antenna in terms of transmission coefficient, group delay, fidelity factor becomes inevitable to ensure good performance of UWB antennas [4].

Several UWB antennas have been used for WBAN applications, a few recent of such works are listed in the literature. Though the antennas reported in [5-13] satisfy the bandwidth and gain requirement requirements of UWB, not any parameters related to time domain analysis such as group delay, fidelity factor, path loss were analyzed. This becomes a major drawback where the UWB communication system may become less reliable. Most of the antennas reported are larger in dimensions which is also a major disadvantage in WBAN scenarios.

Also, most of the antennas support either on body communication or off body communication but not the both.

In the works reported from [14-17], group delay was analyzed under free space conditions. In [18], group delay was analyzed for free space, on body and off body conditions and comments were presented for the three conditions. The works reported in [19-22] have analyzed the transmission coefficient and group delay but it is done only under free space conditions. Fidelity analysis was done in the works [23-26] only under two conditions namely face to face and side to side, but not for face to side and side to face conditions.

In all of the above works done, the analysis was done only under free space condition. But in actual WBAN or wearable scenarios, communication shall happen in on body and off body scenarios also.

Hence, keeping in mind the requirements of an efficient antenna for WBAN and coming to know the short falls prevailing, a compact UWB antenna with ultra-wide bandwidth, good gain supporting both on and off body communications is proposed in this work. Also, to furnish the need of real time WBAN scenario and analyze the system's transmission performance, time domain analysis was carried out for free space, on body and off body conditions. Fidelity analysis is done in normal and bent conditions with four different configurations. Quality analysis was also done by analyzing the Received Signal power or Received Signal Strength (RSS) for free space, on body and off body conditions. The effects of the parameters that affect the RSS are also studied and the results are presented.

The manuscript is organized with the antenna design methodology in chapter II, achieved results in chapter III, followed by Time domain and qualitative analysis in chapter IV and V respectively.

2. ANTENNA DESIGN

This section focuses on the proposed UWB Antenna design. The entire design of the antenna shown in Fig. 1 is considered for further analysis. The substrate used is ROGERS 4003C whose dielectric constant is 3.5 & loss tangent is 0.0027. The total foot print of the antenna is 30x30x1 mm3. All the dimensions are given in Table 1.

The antenna is fed by a 50Ω micro strip line. The stage to stage evolution of the antenna design is depicted in Fig. 2 and the corresponding effects in the resonance characteristics in terms of return loss (S11) is depicted in Fig. 3. The basic shape of the proposed antenna is a circular patch. To design the circular patch antenna, the required radius was estimated using the conventional design equations in [27] which are represented in equations 1 and 2.

$$a = \frac{F}{\sqrt{1 + \frac{2h\left(\ln\left(\frac{\pi F}{2h}\right)\right) + 1.7726}{\epsilon \pi F}}} \quad (1)$$

$$F = \frac{8.791 \times 10^9}{\sqrt{\varepsilon_r}\, f_r} \quad (2)$$

Where,

- a-radius of the patch
- h-thickness of the substrate
- f_r-resonant frequency
- ε_r-dielectric constant of the substrate
- F-a constant

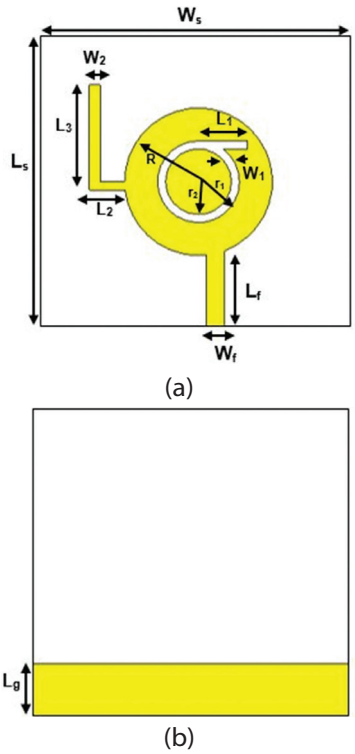


Fig. 1. Proposed antenna, (a) front view, (b) back view

The circular patch antenna designed in stage 1 resonates at 4.2 GHz with some additional harmonics at higher frequencies. In stage 2, a ring shaped slot is introduced slot is introduced in the patch to alter the current path and supress the higher order harmonics. The additional capacitive coupling introduced by the slot between the either sides of the slot help to supress the unnecessary higher order resonances.

Table 1. Antenna Dimensions

Sl. No.	Dimensional parameter	Dimension name	Value (mm)
1.	W_s	Substrate width	30
2.	L_s	Substrate length	30
3.	R	Patch radius	9
4.	R_1	Radius of outer ring	6
5.	R_2	Radius of inner ring	5
6.	W_1	Width of ring slot	1
7.	W_2	Width of stub	1
8.	L_1	Length of slot	6
9.	L_2	Length of horizontal stub	4.5
10.	L_3	Length of vertical stub	12
11.	L_g	Length of ground plane	6.3
12.	L_f	Length of feed	10
13.	W_f	Width of feed	2

The outer radius of the ring slot was obtained using the following relation [18].

$$a_{slot} = \frac{7.15}{\sqrt{\varepsilon_r} f_r} \tag{3}$$

The inner radius was chosen in such a way to maintain better impedance matching and fixed to be 5mm. Once the higher order harmonics are supressed, now to improve the antenna's resonance at lower band, in stage 3, a horizontal slot was combined with the ring shaped slot due to which two resonances were created, one at 3.5 and the other at 3.7 GHz. But the S_{11} values appear very close to the threshold level of -10 dB. Hence to improve the S_{11}, the impedance matching has to be improved. In this notion, the feed location was placed asymmetrically which is the stage 4 and the S11 was improved to -19 dB from -11 dB. In stage 5, a horizontal stub is introduced in the patch to increase the electrical length of the antenna which in turn will increase the bandwidth. Seeking further improvement in bandwidth, another vertical stub was combined with the horizontal stub as it increases the inductance, which will improve the bandwidth at higher frequencies. The stub length was calculated using equations 4 and 5 as per the suggestions in [28]

$$f_r = \frac{c}{\sqrt{\varepsilon_r} 4 l_e} \tag{4}$$

Where *fr* is the frequency, ε_r is the permittivity, l_e is the effective length which is given by

$$l_e = l_{stub} + \frac{0.4 l_{stub}}{\sqrt{\varepsilon_r}} \tag{5}$$

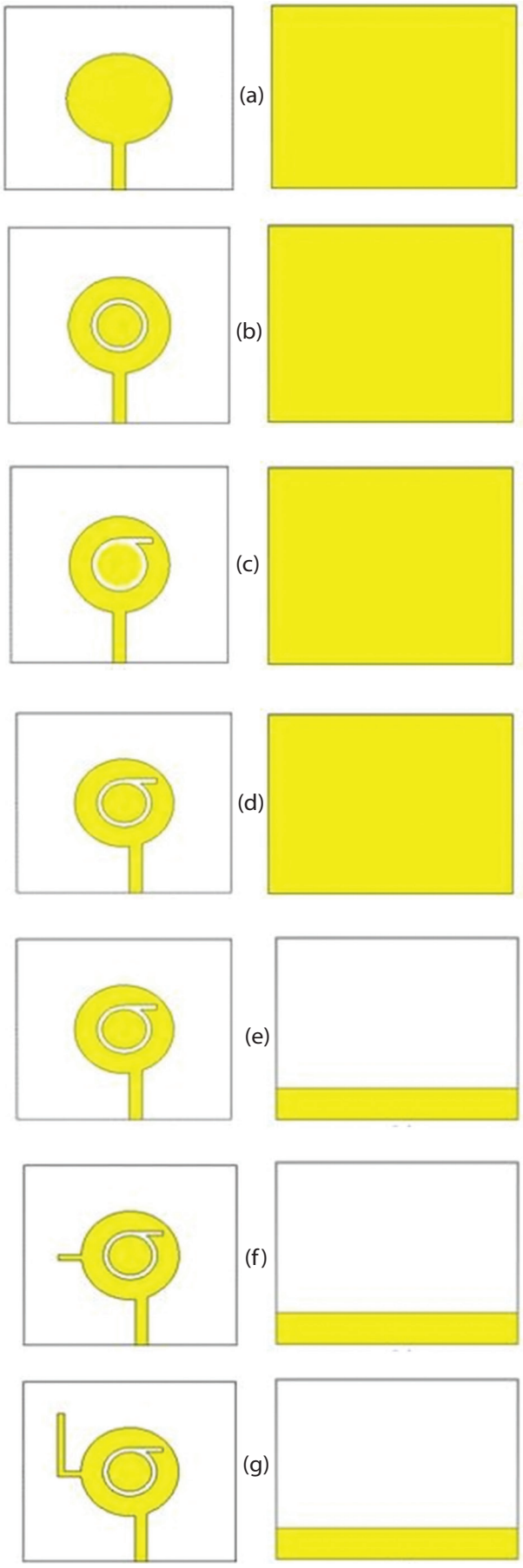


Fig. 2. Various stages of antenna design, (a) stage 1, (b) stage 2, (c) stage 3, (d) stage 4, (e) stage 5, (f) stage 6, (g) stage 7

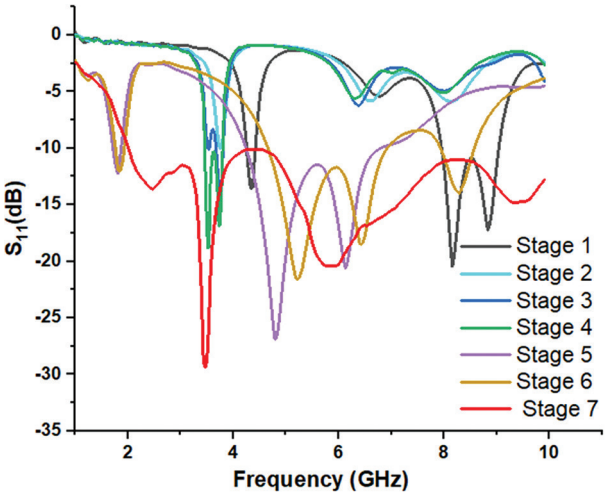


Fig. 3. S_{11} of various stages of antenna design

Initially, for a frequency of 10 GHz, the effective length was calculated using equation 4 and using that the stub length (lstub) was calculated using 5. But still the Ultra wide bandwidth of 7.5 GHz (3.1 – 10.6 GHz) was not achieved. Hence the length of the ground plane was reduced then to achieve UWB bandwidth. The reduced length of the ground plane was calculated using the following relation in equation 6.

$$L_{partial} = \frac{C}{1.45 f \sqrt{\varepsilon}} \quad (6)$$

Reducing the ground plane length results in a reduced Q factor which increases the bandwidth as per the relations given in equation 7 [28]

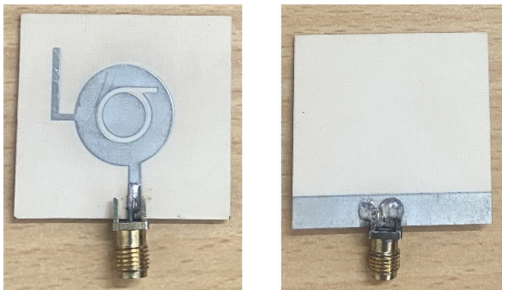
$$Bandwidth = \frac{f_r}{Q} \quad (7)$$

3. RESULTS AND DISCUSSION

The authors have focused on the analysis of basic results like S_{11}, bandwidth, gain etc. in brief while the time domain & qualitative analysis of the antenna were made in depth thereby justifying the manuscript title.

a) Return loss (S_{11}) and Bandwidth

The fabricated prototype is shown in Fig. 4. The simulated and measured S_{11} of the proposed antenna is shown in the Fig. 5.



(a) (b)

Fig. 4. Fabricated antenna, (a) front view, (b) back view

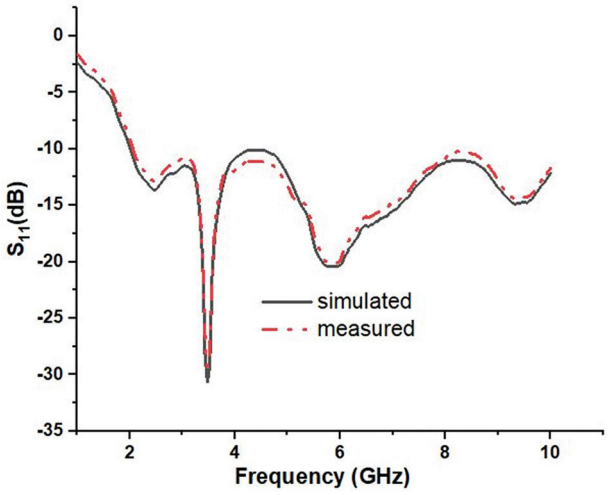


Fig. 5. S_{11} of the proposed antenna – Simulated and Measured

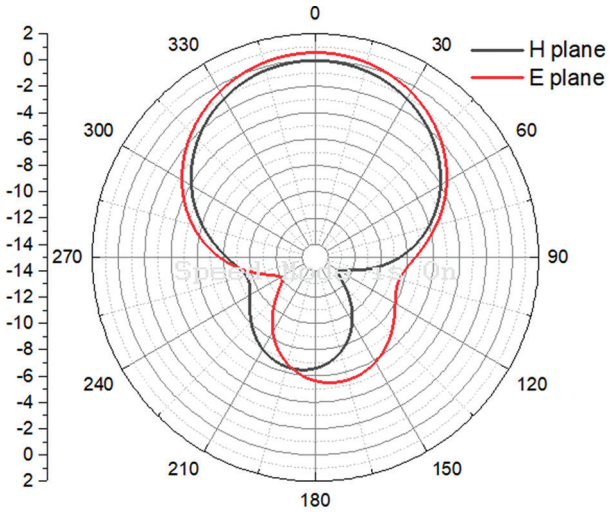
It is seen from the graph that the simulated and measured results agree well with each other with minimum deviations which are due to fabrication errors. The S_{11} of the antenna are -13 dB, -32 dB, -21dB and -14 dB at 2.4, 3.5, 5.8 and 9.3 GHz respectively. The absolute bandwidth of the antenna is 8 GHz ranging from 2-10 GHz.

b) Radiation pattern

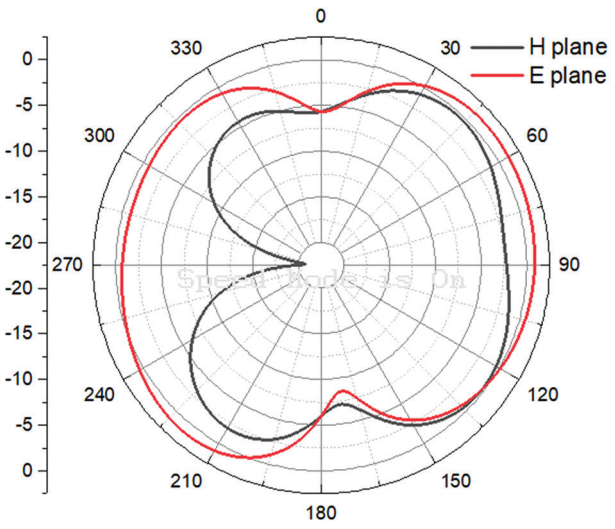
The measured radiation pattern of the antenna is depicted in Fig. 6. The antenna provides omnidirectional pattern at 5.8 GHz facilitating on body applications and directional pattern perpendicular to the ground at 2.4 GHz facilitating off body communication. Thus it supports both on and off body applications.

c) Gain

The measured gain of the antenna reported is shown in Fig. 7. The antenna affords the gain of 1.85 dB, 3.13 dB, 3.36 dB and 5.95 dB at 2.4, 3.5, 5.8 and 9.3 GHz respectively with a peak gain of 6.61 dB at 7 GHz. From the measured gain values, the calculated measured efficiencies are 86%, 89%, 92%, 94% at the respective frequencies.



(a)



(b)

Fig. 6. Radiation Pattern of proposed antenna (a) at 2.4 GHz (b) at 5.8 GHz

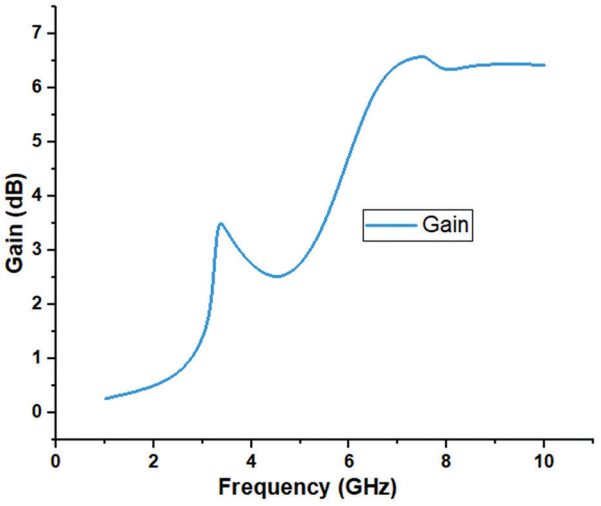


Fig. 7. Measured gain of the proposed antenna

d) SAR

The Specific Absorption Rate (SAR) of the proposed antenna is calculated in the presence of a three layered human tissue model whose setup is shown in the Fig. 8 [21]. The simulated SAR values are 0.579 W/Kg and 0.498 W/Kg at 2.4 and 5.8 GHz respectively. These values are much lower than the prescribed value of 1.6 W/Kg as per IEEE C95.3 standards which makes the proposed antenna more suitable for WBAN based applications.

e) Deformation analysis

Antennas deployed in real time WBAN scenarios undergo deformation in the form of bending. Hence, a bending based deformation analysis is done to study the antenna behaviour. The sample images of the bent antenna are shown in Fig. 10. Two orientations namely x and y orientations were considered for bending. The corresponding resonance behavior is depicted in the Fig. 11. From the results of y bend, it is observed that only the S_{11} values change from -30 dB to -25 and -22 dB. This change is because of the change in the impedance of the antenna's feed while bending it symmetrically. Still the antenna's S_{11} remains below -10 dB which the standard value for reference.

On analyzing the x bend results, there is slight shift of resonances. Nevertheless, the shift in resonance occurs only within the band of operation of the proposed antenna (2-10 GHz) and hence the results remain unaffected due to bending.

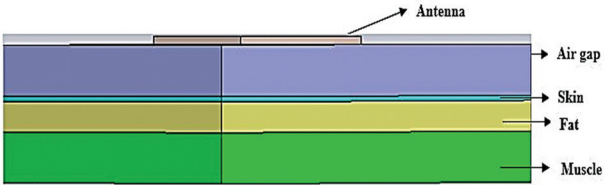


Fig. 8. SAR simulation setup of the proposed antenna

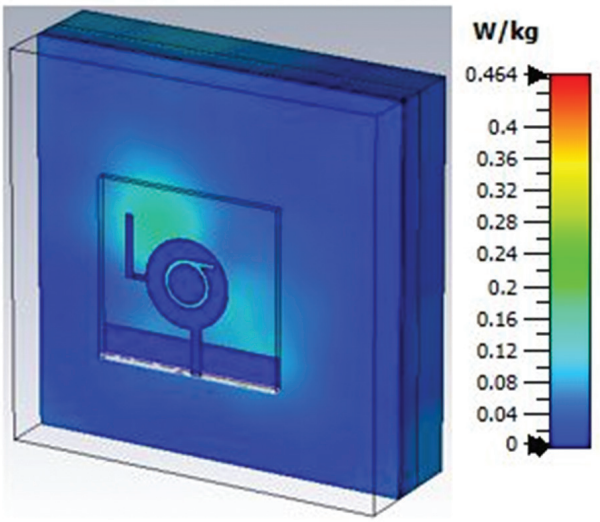
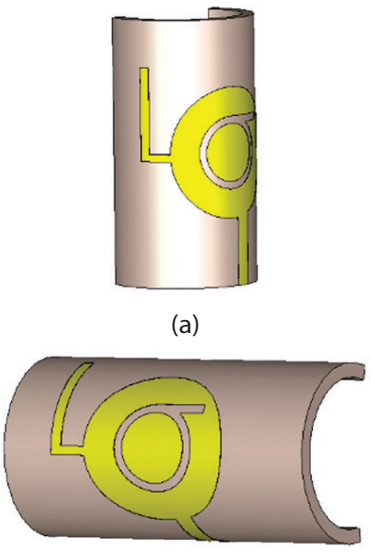


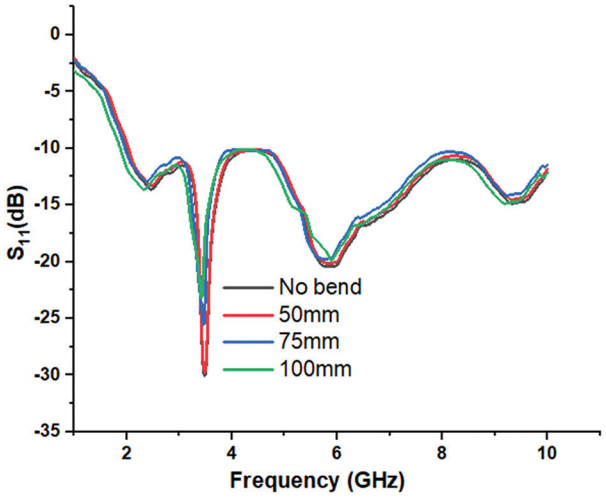
Fig. 9. SAR simulation of the proposed antenna



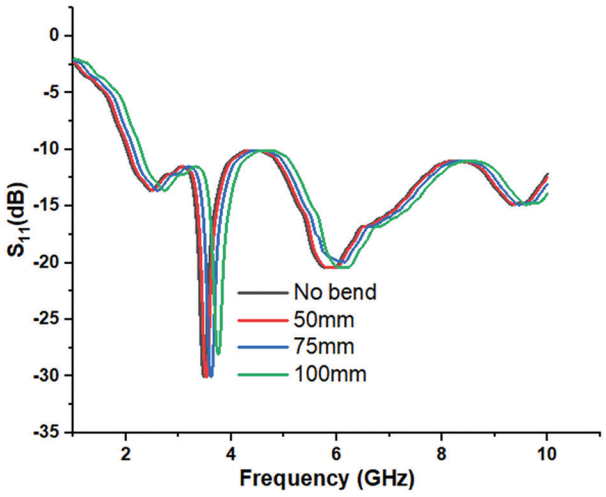
(a)

(b)

Fig. 10. Antenna bending (a) y bend (b) x bend



(a)



(b)

Fig. 11. Antenna bending results (a) y bend (b) x bend

From the bending results, it can be concluded that, the performance of the antenna undergoes only acceptable minimum change due to deformation which proves the proposed design's robustness.

4. TIME DOMAIN ANALYSI

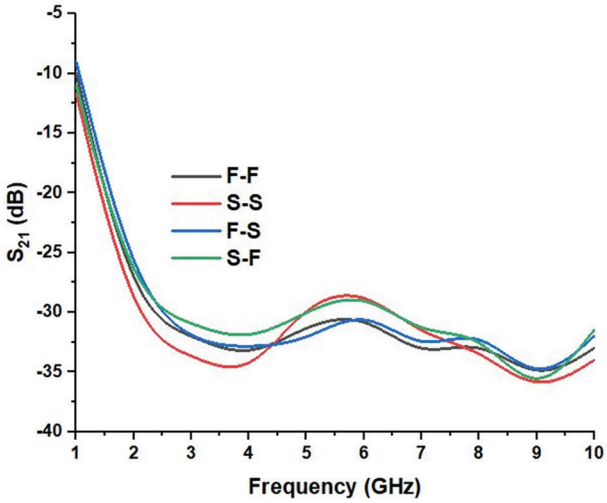
All the required results such as Reflection Coefficient, Bandwidth, Gain, Radiation Pattern, Efficiency, SAR were found to be satisfactory. Hence, the transient analysis alone is concentrated in depth in this section. Data transfer in UWB systems happens in the form of pulses which are of short duration and so there could be distortions in the communication. This can be verified by analyzing the system transfer function in terms of transmission coefficient S_{21}.

a) Free space condition

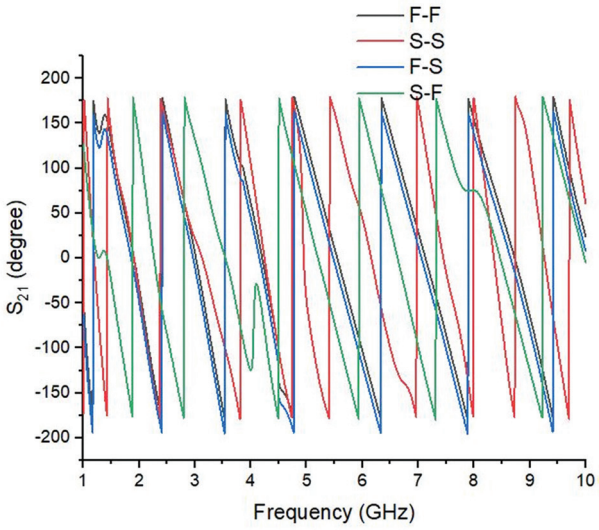
The measurement set up for evaluating the magnitude and phase of the system function S_{21} is presented in the Fig. 12 and the corresponding results are presented in Fig. 13. Four kinds of setups were used for analysis namely face to face (F-F), side to side (S-S), face to side (F-S) and side to face (S-F). These four scenarios shall be encountered in real time cases and hence they were chosen.

Fig. 15 a shows the magnitude of S_{21} and 15 b shows the phase of S_{21}. A communication with less distortion is attested by the minimum fluctuations in magnitude of S_{21} and linear phase variations in S_{21}.

Fig. 12. Free space measurement set up



(a)



(b)

Fig. 13. Free space Transmission response (S_{21}) performance under various setups (a) Magnitude of S_{21} (b) Phase of S_{21}

It can be seen that, the magnitude of S_{21} is almost consistent for all the cases and the variations in phase response is linear. This shows that the antenna performs well under free space conditions. However, a noticeable decrease in magnitude occurs between 1 and 2 GHz. Since the desired operational frequency range of the antenna is 2-10 GHz, it may not be taken into account.

b) On body condition

In WBAN scenarios, communication mainly exists between two nodes placed on the body surface. This scenario is on body scenario where the antenna is placed on the body surface with a suitable separation, preferably clothing [19]. Hence it is very much obligatory to study the performance of the antenna under this scenario. To make this study elaborate, two real time transmission scenarios were chosen which are Line of Sight (LOS) and Non – Line of Sight (NLOS). Antenna locations on the human body were chosen accordingly.

To establish LOS communication, two case studies were chosen. In the first case, one antenna which acts as the transmitting antenna is placed on the chest of the human body and the other antenna is placed on the abdomen of the same human. In the second case, one antenna is placed in the same chest location while the other is placed in the front portion of the arm (arm front).

Fig.14. Measurement set up for on body communication (same body)

To establish NLOS communication, two case studies were chosen. In the first case, one antenna is placed on the chest of the human body and the other antenna is placed on the back of the same human. In the second case, one antenna is placed in the same chest location while the other is placed in the back portion of the thighs (thighs back). All these measurement setups are shown in the Fig. 14.

The obtained S21 characteristics is depicted in Fig. 15. On observing figure 15 a, it can be seen that there is a slight variation in the magnitude and phase of S_{21} in chest to back and chest to thighs back cases. This is because change in magnitude occurs as both the antennas are placed facing opposite to each other. Also the presence of human body in between both the antennas influences the transmission and reception behavior.

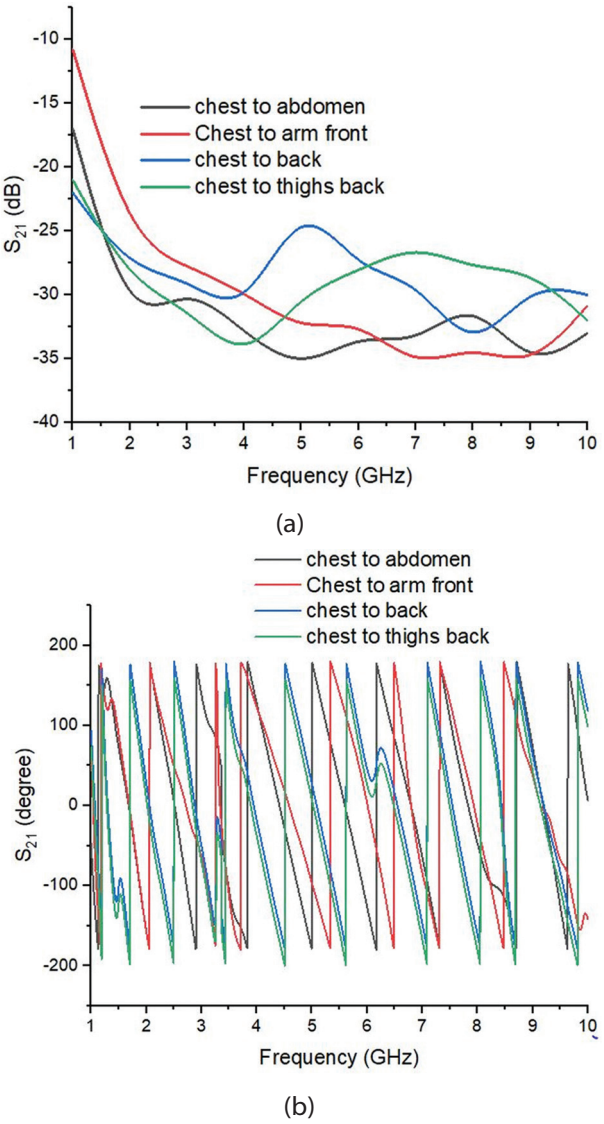


Fig. 15. On body Transmission response (S_{21}) under various setups (a) Magnitude of S_{21} (b) Phase of S_{21}

c) Off body condition

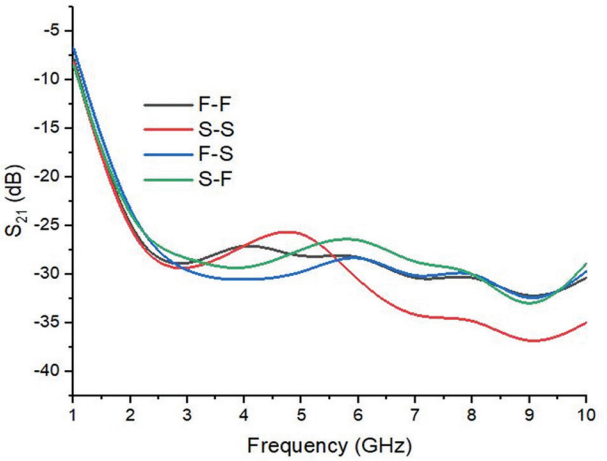
In off body condition, communication happens between the antenna in one human body to another antenna in another human body proximity or any other location. To encounter this scenario, one antenna is placed in the chest of one human volunteer while the other antenna is placed in the chest of another human volunteer of nearly same physic. Also to mimic real time scenarios, four real time cases such as F-F, S-S, F-S, S-F were also considered and the results were studied.

The measurement setup for off body condition is shown in the Fig. 16. The corresponding results are shown in the Fig. 17.

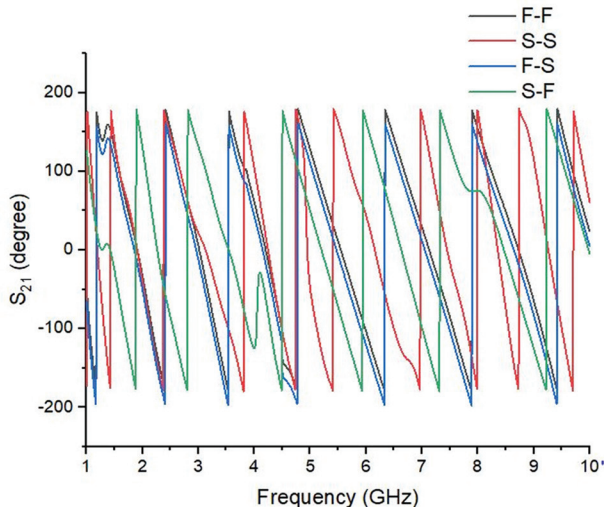
From the results, is can be understood that the magnitude and phase response of the proposed antenna are consistent and linear for F-F, S-S, F-S and S-F sce-

narios. So the proposed antenna exhibits acceptable transmission and reception characteristics under free space, on body and off body conditions.

Fig. 16. Measurement set up for off body communication (different bodies)



(a)



(b)

Fig. 17. Off body Transmission response (S_{21}) under various setups (a) Magnitude of S_{21} (b) Phase of S_{21}

d) Group Delay

To validate the phase distortions, a parameter called group delay is to be examined. It represents the distortion in phase response alone.

The group delay is given by the expression below.

$$T_g = \frac{-d\varphi}{d\omega} \tag{8}$$

Where φ is the phase factor and ω is the angular frequency. The group delay measurement was carried out with the measurement setups that were similar to the ones already depicted in Figs. 12, 14 and 16.

For free space and off body scenarios, group delay was measured for four cases F-F, S-S, F-S, S-F. The results are shown in the Figs. 18 and 19. From the figure it is seen that the group delay of the antenna under all the four cases are flat with variations being within the 0.5 ns.

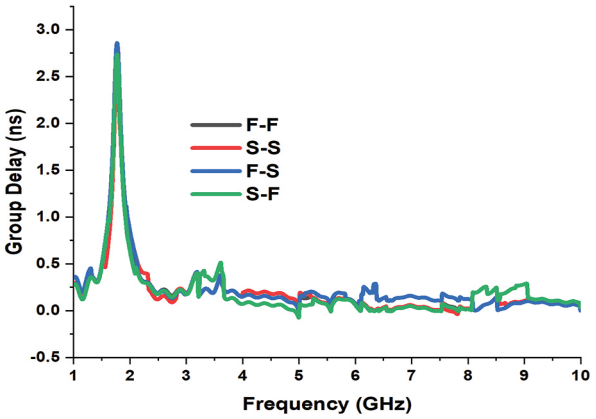


Fig. 18. Group delay (free space)

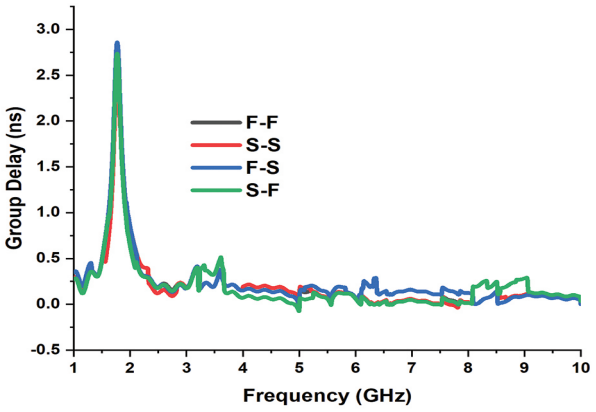


Fig. 19. Group delay (off body)

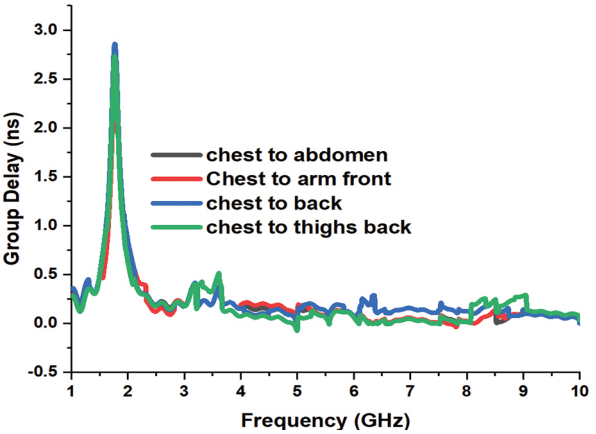


Fig. 20. Group delay (on body)

A flat response is found in all the cases which ensures minimum phase distortions. Some variations in the group delay is seen from 1.6 GHz to 1.9 GHz. But it may not be given more attention as it is not in the operational frequency range. The measured group delay of the antenna for on body scenario is shown below in Fig. 20. The group delay shows good results with flat response.

e) Fidelity Analysis

While establishing communication in UWB systems, the shape of the UWB pulses has to be preserved from the period of transmission to reception. Analyzing this pulse preserving capability is said to be fidelity analysis. The analysis term is called as fidelity factor (FF).

The expression used to evaluate the FF is as follows

$$FF = max \int_{-\infty}^{\infty} T_s(t)\, R_s(t+\tau)\, dt \quad (9)$$

To find out the FF of the proposed antenna, one antenna is used as a transmitter antenna and another as a receiving one. Fidelity factor was analyzed for four cases as already mentioned namely F-F, S-S, F-S and S-F. The transmitted and received signals were collected and applied in equation 4 for all the four cases. The obtained FFs are tabulated in Table 2.

Table 2. FF for Various Cases

Cases	F-F	S-S	F-S	S-F
FF (flat)	0.912	0.916	0.902	0.906
FF (x-bent)	0.853	0.863	0.852	0.866
FF (y-bent)	0.889	0.875	0.895	0.883

Also to analyze the fidelity performance in various angles of receiver, instead of using a similar antenna or another kind of antennas as receiver, field probes were setup with various angular separation. Analysis is also done for both flat, bent and on body conditions.

This simulation setup was developed using CST microwave studio and is shown in the Fig. 21 and the results are shown in Table 3.

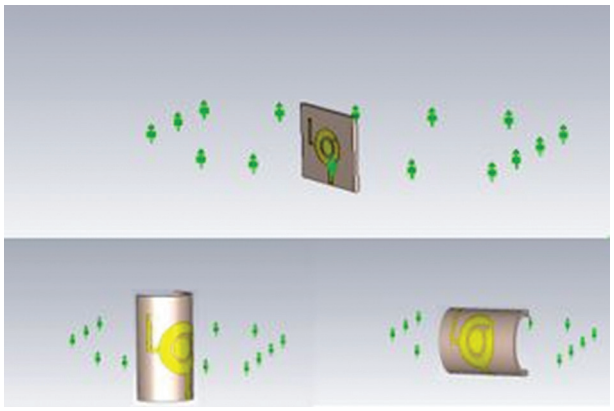


Fig. 21. Simulation set up for fidelity analysis

Table 3. FF for Various Angles

Angular separation (in degrees)	30	60	90	120	150	180
FF (flat)	0.92	0.91	0.91	0.91	0.90	0.90
FF (x-bent)	0.83	0.83	0.82	0.81	0.80	0.79
FF (y-bent)	0.83	0.82	0.81	0.81	0.81	0.79

A good correlation between the transmitted and received pulse exists if the FF value is equal or above 0.5 [29]. Looking into the values of FF in both table 2 and table 3, all the FF values are above 0.5 which ensures good correlation between the transmitted and received pulses.

5. QUALITATIVE ANALYSIS

To study the quality of the transmitted & received signal and also the proposed antenna, framing the link budget using the proposed device shall be certainly required. The link budget represents the received signal strength (RSS) of the antenna. With the conditions of acceptable impedance and polarization matching, the expression to evaluate the measured RSS is as follows [30].

$$P_{rx}(dBm) = P_{tx}(dBm) + G_{tx}(dB) + G_{rx}(dB) - P_L(dB) \quad (10)$$

Where, P_{tx} is the transmitted power, P_{rx} is the received power, G_{tx} is the transmitter antenna gain, G_{rx} is the receiver antenna gain, P_L is the path loss. The measured path loss of the antenna is calculated using the expression as follows [30]

$$P_L = -10 \log\left(\frac{G_{tx} G_{rx} \lambda^2}{(4\pi d)^2}\right) \quad (11)$$

Where, d is the distance between the transmitter and receiver and λ is the operating wavelength. The path loss and in turn the received power is calculated for free space, on body and off body conditions at varying distances and the results were studied.

For on body condition, the transmitter and receiver antennas were placed on the chest and belly locations with a measured separating distance of 200 mm (20 cm). This represents the LOS case. For NLOS case, the receiver antenna was placed at the same separating distance but exactly at the back portion to the belly. Similar to this, the P_L was calculated for free space and off body conditions. While evaluating P_L for off body condition, on body gain of the transmitter antenna and the free space gain of the receiving antenna was considered which resembles the off body case. For free space $P_{L'}$ the free space gain was considered for both the antennas.

All the path loss values are tabulated in table 4. With the obtained P_L values the link budget was analyzed by calculating the received power according equation 10 at two target frequencies 2.45 and 5.8 GHz. The transmitted power of the antenna is chosen to be 27 dBm

as per the suggestions in [31]. The obtained received signal power is tabulated in Table 5.

Table 4. Path Loss

Frequency (GHz)	PL (dB) for 100 mm			
	Free space	On body		Off body
		LOS	NLOS	
2.45	31.15	32.69	33.20	31.92
5.8	36.04	36.70	37.12	36.38
Frequency (GHz)	**PL (dB) for 200 mm**			
	Free space	On body		Off body
		LOS	NLOS	
2.45	45.02	46.55	47.02	45.79
5.8	49.93	50.56	51.10	50.24

Table 5. Received Signal Power

Frequency (GHz)	Prx (dBm) for 100 mm			
	Free space	On body		Off body
		LOS	NLOS	
2.45	0.17	-1.69	-2.20	-0.76
5.8	-1.04	-1.95	-2.37	-1.5
Frequency (GHz)	**Prx (dBm) for 200 mm**			
	Free space	On body		Off body
		LOS	NLOS	
2.45	-13.70	-15.55	-16.02	-14.63
5.8	-14.93	-15.81	-16.35	-15.36

Based on this received signal power, the communication link margin (LM) is calculated as [32]

$$LM\ (dB) = P_a - P_r \tag{12}$$

Where, P_a and P_r are the available and required power levels.

LM is the difference between the available power and required power. The available power level is given by

$$P_a = P_{rx} - N_0 \tag{13}$$

The required power level is given by

$$P_r = \frac{E_b}{N_0} + 10log_{10}(B_r) - G_c + G_d \tag{14}$$

The parameters in equations (13) & (14) are adhered with the methods and values in [32] and P_{rx} is taken from equation (10).

In order make a justified comparison with the available literatures, the distance between the transmitting and receiving antenna is taken to be 5m which was also considered for calculations in [33-36]. For this distance, the calculated LM under free space condition is 60 dB. This LM is comparatively better than the LM in [33-36].

A comparison of the proposed antenna's performance is shown in the Table 6.

On comparing with the existing works, the proposed antenna is lesser in size than [20, 23, 24, 32], more in gain than [30, 23-25, 32-35], higher in bandwidth than [25, 35], lesser in group delay than [20,24,25]. Also, better performance in terms of communication link budget is achieved on comparing with the link budget in [33-36] which is a notable feature of the antenna.

Table 6. Performance Comparison

Reference	Size (mm3)	Peak Gain (dB)	Bandwidth (GHz)	Group delay (ns)	Minimum Path Loss	Link margin (dB)
[20]	24 x 28 x 1.6	4.08	8.37	Up to 10 ns	NA	NA
[23]	51.5 x 31 x 1.6	5.97	10.39	up to 0.5 ns	NA	NA
[24]	35.9 x 29.8 x 1.6	6.5	9.2	up to 4 ns	NA	NA
[25]	12.5 x 12.5 x 1 mm^3	4.12	7.7	up to 4 ns	NA	NA
[26]	42 x 40 x 1.6	4.1	8.9	0.5	60	NA
[32]	36 x 36 x 9.5	6.4	NA	NA	96	-105
[33]	14.8 x 21.7 x 0.254	2.15	12.6	NA	NS	12.24
[34]	15 x 15 x 1.27	-31.4	0.261	NA	NS	20
[35]	26 x 11 x 1	-36	NA	NA	NS	6.89
[36]	10 x 10 x 1.27	-21	NA	NA	NS	50
Proposed	**30 x 30 x 1**	**6.61**	**8**	**0.5**	**109**	**60**

*NA – Not Analyzed, NS – Not Specified

6. CONCLUSION

The design, time domain and qualitative analysis of a dual mode UWB antenna has been discussed in this paper. The proposed antenna was able to provide 8 GHz bandwidth with sufficient gain and less SAR. Free space and body centric measurements were carried out for analyzing the parameters like the magnitude and phase of the transmission coefficient, group delay with various scenarios. The magnitude of transmission coefficient is almost constant and change in phase is also linear with slight variations in the NLOS case alone. The group delay of the antenna is also constant with less variations which ensures reduced amount of distortion in the desired operational range. Fidelity factor was calculated under flat and bent conditions for various cases and angles whose values are good enough to preserve the pulse during transmission and reception. Analysis of Link budget and Link margin ensured proper and quality communication link. Thus the proposed antenna's compliance with WBAN and UWB requirements makes it to be a promising candidate suitable for UWB based WBAN applications.

COMPETING INTERESTS

The authors declare hereby that there are no competing interests in publishing the paper in any form.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable as no data sets were generated

7. REFERENCES

[1] N. Chahat, M. Zhadobov, R. Sauleau, K. Ito, "A Compact UWB Antenna for On-Body Applications", IEEE Transactions on Antennas and Propagation, Vol. 59, No. 4, 2011, pp. 1123-1131.

[2] C. Briso, C. Calvo, Y. Xu, "UWB Propagation Measurements and Modelling in Large Indoor Environments", IEEE Access, Vol. 7, 2011, pp. 41913-41920.

[3] P. S. Hall, Y. Hao, "Antennas and propagation for body centric communications", Proceedings of the 1st European Conference on Antennas and Propagation, Nice, France, 6-10 November 2006, pp. 1-7.

[4] Federal Communications Commission Office of Engineering And Technology Policy and Rules Division, 2002.

[5] P. Chen, D. Wang, L. Liu, L.H. Wang, Y.M. Lin, "Design of UWB Wearable Conformal Antenna Based on Jean Material", International Journal of Antennas and Propagation, Vol. 4886844, 2022.

[6] M. Kanagasabai, P. Sambandam, M. G. N. Alsath, "Compact ultra-wideband pattern diversity antenna for body-centric communications", International Journal of Microwave and Wireless Technologies, Vol. 15, No. 2, 2023. pp. 245-254.

[7] P. Chen, D. Wang, Z. Gan, "Flexible and Small Textile Antenna for UWB Wireless Body Area Network", Micromachines. Vol. 14, No. 4, 2023.

[8] A. Yadav, P. Singh, R. K. Verma, V. K. Singh, "Design and comparative analysis of circuit theory model-based slot-loaded printed rectangular monopole antenna for UWB applications with notch band", International Journal of Communication Systems, Vol. 36, No. 3, 2023.

[9] M. D. Geyikoglu, "A novel UWB flexible antenna with dual notch bands for wearable biomedical devices", Analog Integrated Circuits and Signal Processing, Vol. 114, No. 3, 2023, pp. 439-450.

[10] A. K. Vallappi, B. A. Khawaja, M. K. A. Rahim, M. N. Iqbal, H. T. Chattha, M. F. Ali, "A Compact Triple-Band UWB Inverted Triangular Antenna with Dual-Notch Band Characteristics Using SSRR Metamaterial Structure for Use in Next-Generation Wireless Systems", Fractal and Fractional, Vol. 6, No. 8, 2022.

[11] P. Venkatesh, T. V. Narmadha, "Miniaturized Tribband Planar Monopole Antenna using Right Turned L-Shaped Stubs for Wireless Communications", Proceedings of 4th International Conference on Emerging Research in Electronics, Computer Science and Technology, Mandya, India, 26-27 December 2022, pp. 1-6.

[12] M. Mustaqim, B. A. Khawaja, H. T. Chattha, K. Shafique, M. J. Zafar, M. Jamil, "Ultra-wideband antenna for wearable Internet of Things devices and wireless body area network applications", International Journal of Numerical Modelling, Vol. 32, No. 6, 2019.

[13] D. Negi, R. Khanna, J. Kaur, "Design and performance analysis of a conformal CPW fed wideband antenna with Mu-Negative metamaterial for wearable applications", International Journal of Microwave and Wireless Technologies, Vol. 11, No. 8, 2019, pp. 806-820.

[14] N. Sharma, S. S. Bhatia, "Design of printed UWB antenna with CPW and microstrip-line-fed for

DCS/PCS/bluetooth/WLAN wireless applications", International Journal of RF & Microwave Computer Aided Engineering, Vol. 31, 2020.

[15] R. B. Rani, S. K. Pandey, "A CPW-fed circular patch antenna inspired by reduced ground plane and CSRR slot for UWB applications with notch band", Microwave and Optical Technology Letters, Vol. 59, No. 4, 2017, pp. 745-749.

[16] S. Parameswari, C. Chitra, "Textile UWB Antenna with Metamaterial for Healthcare Monitoring", International Journal of Antennas and Propagation, Vol. 5855626, 2021.

[17] S. S. Bhatia, N. Sharma, "A Compact Wideband Antenna Using Partial Ground Plane with Truncated Corners, L - Shaped Stubs and Inverted T-Shaped Slots", Progress In Electromagnetics, Vol. 97, 2020, pp. 133-144.

[18] P. Venkatesh, T. V. Narmadha, "Design of a compact sigma slotted dual-mode UWB antenna for wireless body area network applications", International Journal of Microwave and Wireless Technologies, 2023.

[19] D. Thangarasu, R. R. Thipparaju, S. K. Palaniswamy, M. Kanagasabai, M. G. N. Alsath, D. Potti, S. Kumar, "On the Design and Performance Analysis of Flexible Planar Monopole Ultra-Wideband Antennas for Wearable Wireless Applications", International Journal of Antennas and Propagation, Vol. 5049173, 2022.

[20] V. N. K. Rao Devana, A. M. Rao, "Design and Analysis of Dual Band-Notched UWB Antenna Using a Slot in Feed and Asymmetrical Parasitic stub", IETE Journal of Research, Vol. 69, No. 1, 2023, pp. 284-294.

[21] A. Yadav, V. K. Singh, A. K. Bhoi, G. Marques, B. Garcia-Zapirain, I. de la Torre Díez, "Wireless Body Area Networks: UWB Wearable Textile Antenna for Telemedicine and Mobile Health Systems", Micromachines, Vol. 11, No. 6, 2020.

[22] N. Mishra, S. Beg, "A Miniaturized Microstrip Antenna for Ultra-Wideband Applications", Advanced Electromagnetics, Vol. 11, No. 2, 2022, pp. 54-60.

[23] R. Kumar, R. Sinha, A. Choubey, S. K. Mahto, "A compact microstrip feedline printed antenna with perturbed partial ground plane for UWB applications", International Journal of RF & Microwave Computer Aided Engineering, Vol. 31, 2021.

[24] S. Park, K. Y. Jung, "Novel Compact UWB Planar Monopole Antenna Using a Ribbon-Shaped Slot", IEEE Access, Vol. 10, 2022, pp. 61951-61959.

[25] M. M. Alam, R. Azim, N. M. Sobahi, A. I. Khan, M. T. Islam, "A dual-band CPW-fed miniature planar antenna for S-, C-, WiMAX, WLAN, UWB, and X-band applications", Scientific Reports, Vol. 12, 2022.

[26] N. Ahuja, R. Khanna, J. Kaur, "Infinite length slotted ultra-wideband monopole antenna using step-feed with band notch characteristics", International Journal of Microwave and Wireless Technologies, Vol. 14, No. 9, 2022, pp. 1141-1148.

[27] C. A. Balanis, "Antenna theory analysis and design", 2nd Edition, John Wiley & Sons, 1997.

[28] V. P. Ravichandran, N. Velayudham, N. P. Velayudham, "On the design and analysis of a miniaturized ultra-wideband (UWB) antenna using quatrefoil-shaped nonuniform meta-surfaces for body-centric communications", International Journal of Communication Systems, Vol. 37, No. 6, 2024.

[29] G. Quintero, J. F. Zurcher, A. K. Skrivervik, "System Fidelity Factor: A New Method for Comparing UWB Antennas", IEEE Transactions on Antennas and Propagation. Vol. 59, No. 7, 2011, pp. 2502-2512.

[30] T. S. Rappaport, "Wireless Communications: Principles and Practice", Prentice-Hall, 2002.

[31] IEEE Recommended Practice for Measurements and Computations of Electric, Magnetic, and Electromagnetic Fields with Respect to Human Exposure to Such Fields, 0 Hz to 300 GHz. IEEE Std C95.3-2021 (Revision of IEEE Std C95.3-2002 and IEEE Std C95.3.1-2010): 1-240.

[32] R. Kangeyan, M. Karthikeyan, "Miniaturized meander-line dual-band implantable antenna for biotelemetry applications", ETRI Journal, Vol. 46, No. 3, 2024, pp. 413-420.

[33] R. Li, Y. Guo, "A Conformal UWB Dual-Polarized Antenna for Wireless Capsule Endoscope Systems", IEEE Antennas and Wireless Propagation Letters, Vol. 20, No. 4, 2021, pp. 483-487.

[34] A. Hamdi, A. Nahali, M. Harrabi, R. Brahem, "Optimized design and performance analysis of wearable antenna sensors for wireless body area network applications", Journal of Information and Telecommunication, Vol. 7, No. 2, 2023, pp. 155-175.

[35] H. Rajagopalan, Y. Rahmat-Samii, "Wireless Medical Telemetry Characterization for Ingestible Capsule Antenna Designs", IEEE Antennas and Wireless Propagation Letters, Vol. 11, 2011, pp. 1679-1682.

[36] X. Y. Zhang, H. Wong, T. Mo, Y. F. Cao, "Dual-Band Dual-Mode Button Antenna for On-Body and Off-Body Communications", IEEE Transactions on Biomedical Circuits and Systems, Vol. 11, No. 4, 2017, pp. 933-941.

Exploring Speech Emotion Recognition in Tribal Language with Deep Learning Techniques

Original Scientific Paper

Subrat Kumar Nayak*
Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
subratsilicon28@gmail.com

Prithviraj Mohanty
Department of Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
prithvirajmohanty@soa.ac.in

Ajit Kumar Nayak
Department of Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
ajitnayak@soa.ac.in

Nrusingha Tripathy
Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
nrusinghatripathy654@gmail.com

Smitaprava Mishra
Department of Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha, India
smitamishra@soa.ac.in

Kumar Surjeet Chaudhury
Department of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, Odisha, India
surjeet.chaudhuryfcs@kiit.ac.in

*Corresponding author

Abstract *– Emotion is fundamental to interpersonal interactions since it assists mutual understanding. Developing human-computer interactions and a related digital product depends heavily on emotion recognition. Due to the need for human-computer interaction applications, deep learning models for the voice recognition of emotions are an essential area of research. Most speech emotion recognition algorithms are only deployed in European and a few Asian languages. However, for a low-resource tribal language like KUI, the dataset is not available. So, we created the dataset and applied some augmentation techniques to increase the dataset size. Therefore, this study is based on speech emotion recognition using a low-resourced KUI speech dataset, and the results with and without augmentation of the dataset are compared. The dataset is created using a studio platform for better-quality speech data. They are labeled using six perceived emotions: ସଡାଙ୍ଗି (angry), ରେହା (happy), ଆଜି (fear), ବିକାଲି (sad), ବିଜାରି (disgust), and ଡ଼େକ୍ (surprise). Mel-frequency cepstral coefficient (MFCC) is used for feature extraction. The deep learning technique is an alternative to the traditional methods to recognize speech emotion. This study uses a hybrid architecture of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) as classification techniques for recognition. The results have been compared with existing benchmark models, with the experiments demonstrating that the proposed hybrid model achieved an accuracy of 96% without augmentation and 97% with augmentation.*

Keywords*: KUI Dataset, Speech Emotion Recognition, Deep Learning, Long Short-Term Memory, Data Augmentation*

Received: May 3, 2024; Received in revised form: September 23, 2024; Accepted: September 23, 2024

1. INTRODUCTION

Speech is the simplest, quickest, and most natural way to establish interaction between humans among the numerous forms of communication. A crucial component of regular human activity is emotion. Emotions support decision-making and help people understand one another. It facilitates communication in terms of safety and security. Human emotions can be recognized using a variety of modalities, including speech, writing, and facial expressions. Speech Emotion Recognition (SER) aims to identify emotions as they are communicated in spoken language. While speech communication between humans and machines is improving, it is still not

interactive, natural, or organic communication as all the machines are not fully equipped to understand human emotions, specifically in low-resource scenarios. This issue has given rise to a new area of study for researchers. Speech Emotion Recognition is the term for techniques that can successfully assist us in comprehending human emotions by identifying the speaker's emotional state from their speech. Speech can clearly express emotions, which may be utilized later to extract essential semantic information from the uttered words and enhance the effectiveness of speech recognition [1].

SER systems can be used in several applications that need human interaction, such as the caller's emotions in a call centre that tracks the problem. A device that could behave like a human is also thought to require adding emotion recognition features. Doctors can learn about the patient's psychological and physical condition, which is an excellent achievement in the case of speech-emotion recognition. As a result, many researchers are getting more interested in SER research to create a recognition model that performs better.

Most speech emotion datasets are available in German, English, and Spanish. Several SER studies have also been used in languages, including Odia, Tamil, Mandarin, and other European and Asian languages. For KUI, a Low Resourced Tribal language, there is a lack of speech emotion dataset even though it is one of the tribal languages of Odisha, spoken by over 10 million people in the Kandhamal District of Odisha and other states of India [2]. This work addresses this critical gap by developing a novel dataset specifically for the KUI language, thereby facilitating new research and development in underrepresented languages and advancing the field of Natural Language Processing (NLP). It introduces a hybrid model combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for emotion classification. This innovative approach leverages the strengths of both architectures: CNNs are effective for spatial feature extraction, while LSTMs excel at capturing temporal dependencies. By employing this hybrid model, we explore its performance using both original and augmented datasets, enhancing the diversity of training data through techniques such as noise injection, time shifting, random gain, and polarity inversion. Additionally, we detail the design and development process of our CNN-LSTM model, including parameter selection and feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs) [3]. This comprehensive approach contributes to understanding emotion recognition in the KUI language and is a foundation for future research in low-resource language processing [4].

The structure of this document is as follows. In Section II, relevant research is provided, and the findings are discussed. The procedure for creating the KUI dataset and the feature extraction technique is explained in Section III. The suggested model categorization is shown in Section IV, and the performance metrics are shown in Section V. Section VI provides the outcomes. The paper's conclusion, which includes a scope and future work, is included in Section VII.

2. LITERATURE SURVEY

Researchers now have more efficient opportunities to develop SER models using deep learning. Language, dataset properties, feature extractions, and various classifiers are all important factors in speech-emotion recognition systems [5]. Several researchers have applied machine learning and deep learning to detect emotions in English speech.

More research needs to be done on the KUI language. Some KUI commands are trained and tested using deep learning models, which yields significant results [6]. Several studies have been conducted to identify emotions in speech for a wide range of languages, but they have yet to be done for KUI speech, as shown in Table 1.

Jo et al. [7] presented a CNN-based transfer learning model for recognising emotions in Korean. Using Chosun University's Korean speech emotion database, they achieved an accuracy of 94.91%. This is a good performance, but accuracy is still required to improve in a tribal language.

Taj et al. [8] presented a 1D-CNN model for voice emotion identification in the URDU language. Although Urdu has fewer resources, the model still gives an accuracy of 97%. They used the URDU dataset for the experiments. In this research, the authors compare the results with existing work based on CNN and SVM.

Amjad et al. [9] proposed deep convolutional neural networks (DCCNs) with two layers of long-short-term memory (LSTM). They identify the emotions from the spontaneous speech. They used SAVEE, IEMOCAP, and BAUM-1 datasets for emotion recognition. They got an accuracy of 94.78% for speaker-independent with the raw SAVEE dataset.

Atila et al. [10] proposed an attention-guided 3D CNN-long short-term memory (LSTM) system for speech-based emotion identification. They used three datasets: RAVDESS, RML, and SAVEE, and a mixture of them. They compared the outcomes using the F1-score, sensitivity, specificity, and classification accuracy. The RML dataset, which consists of different languages, gave an accuracy of 93%.

Table 1. Emotion Recognition in different languages

Ref	Year of publication	Dataset	Language	Model Used
[11]	2023	RAVDESS	English	CNN-LSTM
[12]	2023	CREMA-D	English	LSTM
[13]	2023	EMO-DB SAVEE RAVDESS	German English English	CNN
[14]	2023	EMO-DB ENTERFACE05	German English	DCNN-GWO
[15]	2023	RAVDESS	English	Deep LSTM
[16]	2023	EMO-DB	German	DNN-SVM

[17]	2023	RAVDESS EMO-DB	English German	1-D DCNN
[18]	2023	RAVDESS TESS CREMA-D IEMOCAP	English English English English	CNN
[19]	2023	AVEC, AFEW	English	ASP-MTL
[20]	2022	EMO-DB URDU	German Urdu	CADCN
[21]	2021	NSSED	Sindhi	1D- CNN
[22]	2020	EMO-DB	German	CNN
[23]	2020	EMO-DB IEMOCAP RAVDESS	German English English	BiLSTM
[24]	2019	ARABIC	Arabic	CNN-LSTM
[25]	2019	EMIRATI	Arabic	GMM-DNN
[26]	2018	CUSTOM	Malayalam	DNN
[27]	2018	NCKU-ES	Chinese	CNN LSTM
[28]	2017	TELGU EMOTION TAMIL EMOTION	Telgu Tamil	ANN KNN ANN KNN
[29]	2013	MANDARIN	Chinese	HMM
[30]	2010	ODIA	Odia	K-Means

3. MATERIAL AND METHODS

3.1. DATASET

A KUI emotion speech dataset was generated from several speakers in the Kandhamal district of Odisha for this study. A platform has been developed for data preparation in the KUI language [31]. The platform snapshot is shown in Fig. 1. The manual dataset preparation takes more time. We have taken six common emotions in the dataset: ସଡାଙ୍ଗି, ରେହା, ଆଜି, ବିକାଲି, ବିଜାରି and ଡ଼େକ୍. The meaning of the KUI emotions is shown in Table 2.

Fig. 1. Speech emotion data collection platform

There are 2,383 expressions of 6 different emotions in our dataset. At a rate of 16 kHz, each file is saved in wave format as linear 16-bit single-channel Pulse Code Modulation (PCM) values. We used a Zoom audio recorder, a laptop, and a mobile device to capture the speech expressions. To reduce noise, it is recorded in a studio. Our collection's male-to-female voices are equal to avoid bias. After the completion of data collection, the sampling rate of all the recordings is checked. The pre-processing stage involved the removal of unwanted background noise. The total dataset includes 2383 different utterances.

Table 2. KUI Speech emotion and its meaning

Emotion (Kui)	Emotion (English)	Emotion (IPA)	#files		
			Original	Augmented	Total
ସଡାଙ୍ଗି	Angry	ɾagɔ	397	1588	1985
ରେହା	Happy	k^husi	400	1600	2000
ଆଜି	Fear	b^hɔjɔ	398	1592	1990
ବିକାଲି	Sad	d̪uhk^{h}ɔ	397	1588	1985
ବିଜାରି	Disgust	biɾɔkt̪i	396	1584	1980
ଡ଼େକ୍	Surprise	astʃɔrdʒjɔ	395	1580	1975

3.2. METHODOLOGY

The flow diagram of speech emotion recognition is shown in Fig. 2. KUI speech emotion recognition mainly depends on feature extraction methods. It mostly takes several features out of the audio streams. After feature extraction, the features are sent into the classifier, frequently referred to as input. The various emotions are identified using the inputs. The first and foremost stage of speech conversion is feature extraction. The vital goal of this process is to find the details of a speech signal. Over time, feature extraction needs to be consistent. In speaking, it must happen regularly and dynamically. There are several kinds of feature extraction techniques available [32]. The Mel-Frequency Cepstral Coefficient (MFCC) feature extraction approach is used in this paper.

3.2.1. DATA AUGMENTATION

The volume of data directly impacts the performance of deep learning. Deep learning is subject to overfitting when used with limited datasets. Typically, the first thing that is thought about is approaching this challenge from the data level [33].

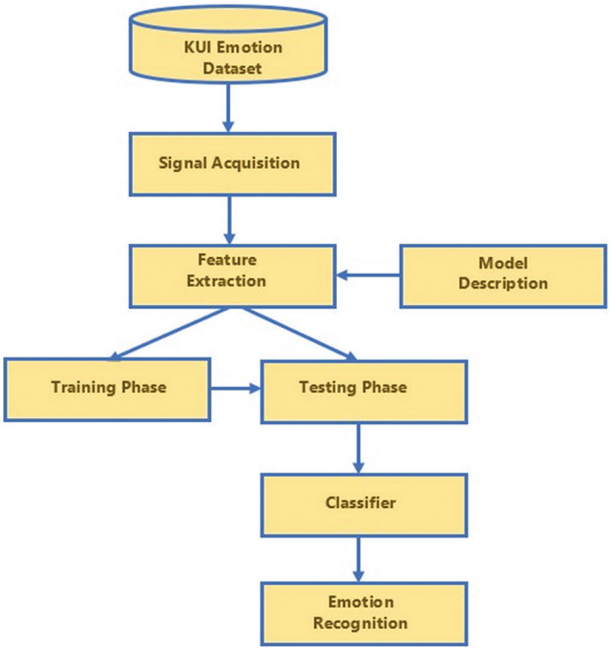


Fig. 2. Flow diagram of KUI emotion recognition

For low-resourced languages, gathering extra information can be challenging. Data augmentation aims to enhance the volume of data required for speech recognition system training. Data augmentation effectively increases current data availability and allows model training without requiring new data. We can present the audio data in two ways, i.e., raw audio and spectrogram [34]. Data augmentation, a regularisation technique, generates new, slightly modified samples from the original data to increase the training set. In this paper, four data augmentation strategies are considered as follows:

i. Noise injection: We can use different types of noises for data augmentations. Noise may be environmental, background, white, or thick. In our case, we inject white noise, adding a random value to the original data.

ii. Shifting time: It just moves the audio left or right. If there is not enough trailing silence, the audio will wrap around.

iii. Random gain: Random gain can change the amplitude. This method measures the loudness of the audio. The original audio signal is multiplied by a random factor, which converts into an augmented signal.

iv. Polarity: An audio time-frequency representation is independent of polarity. An audio data augmentation that reverses the phase of an audio stream might be beneficial for raw waveforms. All phases will cancel out when the phase-inverted signal is added to the original signal. In this case, the signal is multiplied by -1 for the phase insertion.

3.2.2. FEATURE EXTRACTION

Mel-Frequency Cepstral Coefficient (MFCC):

The European Telecommunications Standards Institute defines the MFCC algorithm. MFCC is an efficient method for feature extraction. It considers the frequencies of human perception sensitivity, which can be treated as one of the best tools for speech recognition. The block diagram of MFCC is described in Fig. 3. Various types of steps for finding the Mel-Frequency Cepstral Coefficient [35] are described in eqn 1 through 5.

Pre-emphasis: This step increases energy to higher frequencies, possibly related to vowels with more energy at low frequencies than at high frequencies. It improves the detection accuracy and uses a filter to increase higher frequencies.

$$y[n]=x[n]-\alpha\cdot x[n-1] \quad (1)$$

Where $x[n]$ is the input signal, and $y[n]$ is the pre-emphasized signal

Framing: In this step, the signal is split into small time frames where each frame can be independently analyzed and represented as a single feature vector. The frame time length of the extracted speech is 25-30 ms. The overlapping of the frames is very useful to reduce the loss of information. The advantage is not to do the Fourier transform across the entire signal.

$$x_f[i]=x[i.R:(i+1).R-1], i=0,1,\ldots,L-1 \quad (2)$$

Here, $x_f[i]$ represents the i-th frame. R is the frame length and L is the number of frames.

Windowing: It involves slicing the sound's waveform into different frames. But it cannot split at the boundary of the frame. For slicing, the audio signal amplitude should drop near the edge of a frame. Therefore, it is better to use Hamming windows to chop the signal.

$$x_w[i]=x_f[i]\cdot w[i], i=0,1,\ldots,R-1 \quad (3)$$

Here, $x_w[i]$ is the windowed frame.

Fast Fourier Transform (FFT): The frequency domain is created from the time domain using the Fast Fourier Transform (FFT) technique, as the time domain calculation is more complicated than the prevalence domain.

$$X[k]=\text{FFT}(x_w) \quad (4)$$

Where $X[k]$ is the spectrum of the frame.

Mel Filter Bank: The way of receiving the sound of our ears and the machine is different. We can differentiate easily if we hear sound at 10HZ and 30HZ, but it is not easy to distinguish if it becomes 1000HZ and 1020HZ. But the machine resolution is the same at all frequencies. Thus, the human hearing property will improve performance. Therefore, we use the mel scale to map the actual frequency. The power spectrum uses the filter bank to sum up the energies. This energy is applied with the algorithm to find the mel frequency co-efficient.

$$H_m[k]=\sum_{i=0}^{N-1} X[i]\cdot H_m[i,k],\ m=0,1,,,,,M-1 \quad (5)$$

Here, $H_m[k]$ is the output of the m-th filter bank. N is the number of FFT points and M is the number of mel filters.

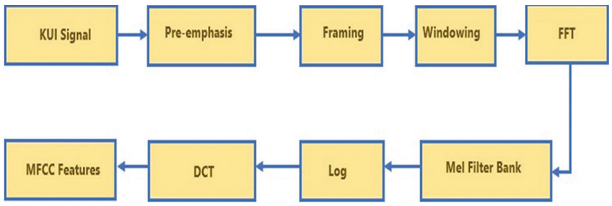


Fig. 3. Block diagram of Mel-Frequency Cepstral Coefficient

4. MODEL BUILDING

Many neural network approaches are utilized for voice emotion recognition. Speech emotion recognition in a low-resource KUI language has not yet been researched or developed. This work used the Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models. Taking the above two models, we propose a hybrid model and compare the model accuracy with performance metrics. We also compare the accuracy before data augmentation and after data augmentation. We used Python to implement our research work. The data were split into three different training and testing sets for classification, i.e., 90%-10%, 80%-20%, and 70%-30%. We also compare the results by taking the different epoch sizes.

4.1. CONVOLUTIONAL NEURAL NETWORKS (CNN) MODEL

As Convolutional Neural Networks (CNNs) are good at identifying local patterns in spectrogram representations of audio data, they have become essential tools in speech emotion recognition [13]. For this purpose, CNNs usually have convolutional layers and pooling layers, allowing them to extract discriminative features from raw audio input automatically. Through multiple layers of convolution and pooling, CNNs hierarchically extract features at different levels of abstraction, facilitating the identification of emotional cues such as pitch variations, spectral changes, and temporal dynamics in speech signals. After learning these representations, fully connected layers are provided for classification, where the model can predict the emotional state associated with the input speech segment. With appropriate training data and optimization strategies, CNNs have demonstrated promising performance in various emotion recognition tasks, offering robustness to noise and variability in speech signals while requiring minimal preprocessing of the input data [36]. Using convolutional layers, the Convolutional Neural Network (CNN) in KUI speech emotion recognition extracts hierarchical characteristics from the speech signal representations, as stated in equation 6.

$$CNN(x)=Conv1D(ReLU(BatchNorm(x))) \quad (6)$$

Where x represents the input feature of the speech signal, $Conv1D$ denotes the $1D$ convolutional operation, $ReLU$ is the rectified linear activation function and $BatchNorm$ represents batch normalization and accelerates training.

4.2. LONG SHORT-TERM MEMORY (LSTM)

Emotion recognition has demonstrated the significant efficacy of Long Short-Term Memory (LSTM) models, especially when evaluating sequential input like speech. Recurrent Neural Networks, or RNNs, are particularly good at recognizing the temporal dynamics and long-range dependencies in voice signals. By processing speech input over time through recurrent connections with gated memory cells, LSTMs can learn intricate patterns and contextual information crucial for discerning emotional states [27]. This capability allows them to capture nuanced features like prosody, intonation, and subtle variations in speech, which indicate different emotions. Furthermore, LSTMs can handle variable-length input sequences, making them suitable for analyzing speech segments of varying durations by integrating LSTMs with additional layers, such as attention mechanisms or combining them with other architectures like CNNs. Researchers have demonstrated the accuracy of LSTM models in capturing the temporal dynamics and complex nuances inherent in emotional speech by achieving state-of-the-art performance in speech emotion recognition [37]. It is possible to build an LSTM model for emotion recognition, as stated in eqn 7.

$$LSTM(x)=LSTM(x_m, h_{m-1}, c_{m-1}) \quad (7)$$

Where x_m symbolizes the input at that moment m, h_{m-1} denotes the previous hidden state, a time step m-1. c_{m-1} represents the previous cell state at the moment m-1 and LSTM denotes the cell operation.

4.3. HYBRID MODEL

We present a hybrid model that includes CNN and LSTM. The convolution layer in our HYBRID model shows the feature map sequence and recognizes significant areas and varied length utterances. In the activation layer, a non-linear activation function is used. The Rectified Linear Unit (ReLU) has been utilized. When a layer is dense, the Softmax activation function is applied. The design, development, and assessment of the model, parameter selection, and feature selection for the MFCC in the HYBRID model for emotion classification are the primary contributions of this study.

The model uses multiple Conv1D layers with various filter and kernel sizes (3,5), extracting significant features from the input data. The hierarchical extraction captures local and global patterns in the data, which is crucial for tasks such as emotion classification, where subtle differences in the features matter. The LSTMs employed capture temporal dependencies within the sequential data, while the CNN is excellent for spatial feature extraction, making the proposed hybrid architecture leverage the strengths of both architectures. After the feature extraction and sequence learning phases, the model flattens the output from the LSTM layer and feeds it into dense layers. The 512-unit dense layer with ReLU activation and batch normalization processes the features before the final classification layer, passing it to the output layer. It uses a softmax activation function for multi-class classification, making the model suitable for categorizing emotions into distinct classes. The KUI dataset with limited resources is used in this research. The hybrid model can be described as a sequence of operations performed on the input data, as stated below.

1. Input:
- Input data shape:
  (batch_size, sequence_length, 1)
2. Convolutional Layers:
- $Conv1D$ with 512 filters, kernel size 5, strides 1, and ReLU activations, followed by batch normalization and max pooling:

$$Conv1D \text{ 512 (x)}$$
$$\rightarrow BatchNormalization(x)$$
$$\rightarrow MaxPool1D(x)$$

- Output shape:
  (batch_size, sequence_length/2, 512)
3. Convolutional Layers:
- Conv1D with 512 filters, kernel size 5, strides 1, and ReLU activations, followed by batch normalization and max pooling:

$$Conv1D_{512}(x) \rightarrow BatchNormalization(x) \rightarrow MaxPool1D(x)$$

- Output shape: (batch_size, sequence_length/4, 512)

4. Convolutional Layers:

- $Conv1D$ with 256 filters, kernel size 5, strides 1, and ReLU activations, followed by batch normalization and max pooling:

$$Conv1D_{256}(x) \rightarrow BatchNormalization(x) \rightarrow MaxPool1D(x)$$

- Output shape: (batch_size, sequence_length/8, 256)

5. Convolutional Layers:

- $Conv1D$ with 256 filters, kernel size 3, strides 1, and ReLU activations, followed by batch normalization and max pooling:

$$Conv1D_{256}(x) \rightarrow BatchNormalization(x) \rightarrow MaxPool1D(x)$$

- Output shape: (batch_size, sequence_length/16, 256)

6. Convolutional Layers:

- $Conv1D$ with 128 filters, kernel size 3, strides 1, and ReLU activations, followed by batch normalization and max pooling:

$$Conv1D_{128}(x) \rightarrow BatchNormalization(x) \rightarrow MaxPool1D(x)$$

- Output shape: (batch_size, sequence_length/32, 128)

7. LSTM Layer:

- LSTM layer with 256 units, returning sequences.

8. Flatten Layer:

- Flatten the output of the LSTM layer to prepare it for the dense layers.

9. Dense Layers:

- ReLU activation and 512-unit dense layer are followed by batch normalization.
- Output shape: (batch_size, 512)

10. Output Layer:

- Dense layer with 6 units and softmax activation for multi-classification.
- Output shape: (batch_size, 6)

5. PERFORMANCE MEASURE OF HYBRID DEEP LEARNING MODELS

Performance metrics in speech emotion recognition are crucial for evaluating the effectiveness of models in classifying emotional states from speech signals. Commonly used metrics include accuracy, precision, recall, and F1-score [38]. Mathematically, all the metrics are represented in eqn 8 through 11.

- **Precision**: Voice emotion recognition systems must achieve high precision for practical uses in sentiment analysis, customer service, human-computer interaction, and psychology research. This guarantees that the system can accurately recognize and react to spoken language's emotional content, which is essential for offering suitable and efficient communication interfaces and services.

$$Precision = TP/(TP+FP) \quad (8)$$

Where TP denotes the number of positive instances correctly classified by the model and FP denotes the number of negative instances incorrectly classified as positive by the model.

- **Recall**: Recall in speech emotion recognition refers to a system's or model's capacity to accurately identify every occurrence of an emotion class from the dataset's total number of instances of that emotion. It measures the system's ability to capture all relevant instances of a specific emotion without missing any.

$$Recall = TP/(TP+FN) \quad (9)$$

Where TP are the instances correctly identified as positive and FN denotes instances incorrectly identified as negative.

- **Accuracy**: Accuracy in speech emotion recognition refers to the degree to which a system can correctly identify the emotional state conveyed by human speech. This is typically measured as the percentage of correctly identified emotions out of the total number of emotions analyzed.

$$Accuracy = (TP+TN)/(TP+TN+FP+FN) \quad (10)$$

Where TN denotes the number of correctly predicted negative instances.

- **F1 Score**: An often-used statistic in speech emotion identification is the F1 score, which assesses how well emotion categorization models perform. The F1 score is a model accuracy metric considering the model's recall and precision.

$$F1\ Score = 2*(Recall*Precision)/(Recall+Precision) \quad (11)$$

6. RESULTS AND ANALYSIS

The study presented in this article compares prominent deep-learning algorithms to identify the emotion from KUI speech. We have considered six emotions ସଡାଙ୍ଗି (angry), ଓରହା (happy), ଆଜି (fear), ବିକାଲି (sad), ବିଜାରି (disgust), and ଓଡ଼ିକ୍ (surprise). We have compared the results of CNN and LSTM with our proposed HYBRID model. The experiment was done for data without augmentation as well as with augmentation. As mentioned before, we test deep learning models using our KUI dataset. We use 90:10, 80:20, and 70:30 ratios for training and testing. During the training phase, the batch size is set to 32. First, we train the model without

data augmentation using different parameters for our dataset. We used 0.01 as the learning rate. Initially, we used an epoch size of 50 in our experiment and trained with various split ratios. The testing accuracy of CNN is 0.91, whereas LSTM gives 0.93, and our proposed model has a maximum accuracy of 0.94. Similarly, the training and testing procedure was done for different epoch sizes. The graphical representation of the testing accuracy of all models having a split ratio of 80:20 with varying sizes of epoch is shown in Fig. 4 through 7.

Next, we train the model with the augmented dataset. After the data augmentation methods, our dataset size increased to 11915, and we input it into our models. The graphical representation of testing accuracy of all models with data augmentation having a split ratio of 80:20 with different epoch sizes is shown in Fig. 8 through 9. Table 3 displays details of all tests' accuracy of both augmentation and without augmentation

It is evident in Fig. 4 to Fig. 9 that across all the epoch sizes taken, the hybrid model consistently achieves the highest accuracy, showing fewer fluctuations in accuracy, indicating a more stable training process than the individual CNN and LSTM models. It also shows the benefits of combining CNN and LSTM architectures. This is also shown in the classification report in Tables 3 and 4. All the models taken show convergence to a high accuracy over time, but the proposed hybrid model achieves the best accuracy faster and maintains it better.

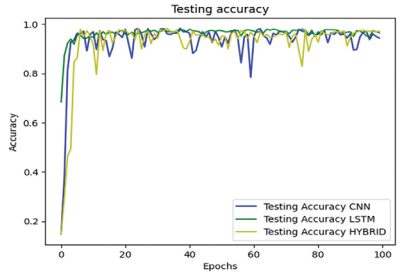


Fig. 4. Testing Accuracy without augmentation using epoch size 100

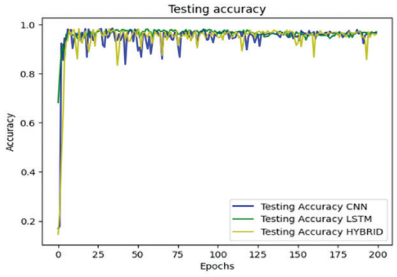


Fig. 5. Testing Accuracy without augmentation using epoch size 200

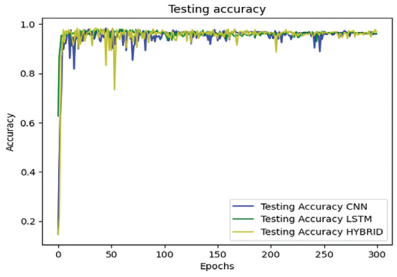


Fig. 6. Testing Accuracy without augmentation using epoch size 300

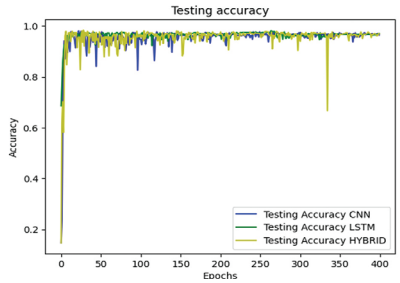


Fig. 7. Testing Accuracy without augmentation using epoch size 400

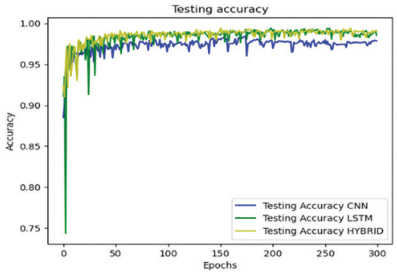


Fig. 8. Testing Accuracy with augmentation using epoch size 300

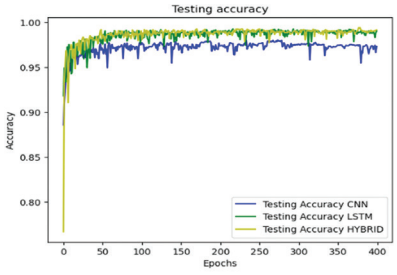


Fig. 9. Testing Accuracy with augmentation using epoch size 400

Table 3. Accuracy of different Models

No of epochs	split-ratio	Accuracy of different Models without augmentation			with augmentation		
		CNN	LSTM	HYBRID Model	CNN	LSTM	HYBRID Model
50	70-30	0.81	0.92	0.93	0.91	0.93	0.95
	80-20	**0.91**	**0.93**	**0.94**	**0.96**	**0.97**	**0.97**
	90-10	0.86	0.93	0.95	0.94	0.95	0.96
100	70-30	0.89	0.93	0.94	0.93	0.94	0.96
	80-20	**0.93**	**0.94**	**0.95**	**0.96**	**0.96**	**0.97**
	90-10	0.91	0.95	0.95	0.94	0.96	0.97
200	70-30	0.91	0.94	0.95	0.95	0.95	0.96
	80-20	**0.94**	**0.95**	**0.96**	**0.96**	**0.97**	**0.97**
	90-10	0.92	0.95	0.96	0.95	0.96	0.97
300	70-30	0.93	0.95	0.96	0.94	0.96	0.96
	80-20	**0.94**	**0.95**	**0.96**	**0.94**	**0.95**	**0.97**
	90-10	0.92	0.95	0.95	0.94	0.96	0.97
400	70-30	0.91	0.93	0.94	0.94	0.95	0.96
	80-20	**0.95**	**0.95**	**0.96**	**0.95**	**0.96**	**0.97**
	90-10	0.91	0.95	0.96	0.95	0.97	0.97

The classification metrics of all the models with different epoch sizes without augmentation are shown in Table 4. Our proposed model's performance matrix gives better results than others. The classification metrics of all the models with different epoch sizes with augmentation are also shown in Table 4. Our proposed model's performance matrix gives better results than others in the case of data augmentation.

Table 4. Comparing various performance metrics

Comparing various performance metrics																			
		without augmentation									with augmentation								
Epoch Size	Performance indicators	CNN			LSTM			HYBRID Model			CNN			LSTM			HYBRID Model		
		70-30	80-20	90-10	70-30	80-20	90-10	70-30	80-20	90-10	70-30	80-20	90-10	70-30	80-20	90-10	70-30	80-20	90-10
50	Precision	0.8	0.88	0.88	0.90	0.91	0.91	0.92	**0.92**	0.89	0.89	0.95	0.93	0.90	0.98	0.95	0.93	**0.91**	0.97
	Recall	0.84	0.89	0.82	0.91	0.95	0.93	0.93	**0.95**	0.94	0.91	0.97	0.96	0.95	0.97	0.94	0.95	**0.99**	0.94
	F1-Score	0.82	0.88	0.85	0.90	0.93	0.92	0.92	**0.93**	0.91	0.9	0.96	0.94	0.92	0.97	0.94	0.94	**0.95**	0.95
100	Precision	0.91	0.89	0.94	0.93	0.95	0.95	0.96	**0.95**	0.93	0.94	0.95	0.92	0.89	0.96	0.94	0.95	**0.97**	0.98
	Recall	0.9	0.94	0.91	0.94	0.93	0.96	0.93	**0.94**	0.98	0.91	0.93	0.94	0.94	0.94	0.98	0.96	**0.94**	0.95
	F1-Score	0.9	0.91	0.92	0.93	0.94	0.95	0.94	**0.94**	0.95	0.92	0.94	0.93	0.91	0.95	0.96	0.95	**0.95**	0.96
200	Precision	0.93	0.89	0.92	0.92	0.92	0.94	0.95	**0.93**	0.96	0.95	0.97	0.93	0.97	0.98	0.93	0.94	**0.98**	0.95
	Recall	0.9	0.92	0.93	0.95	0.93	0.93	0.96	**0.96**	0.97	0.93	0.94	0.96	0.98	0.95	0.96	0.98	**0.97**	0.98
	F1-Score	0.91	0.9	0.92	0.93	0.92	0.94	0.95	**0.94**	0.96	0.94	0.95	0.94	0.97	0.96	0.94	0.96	**0.97**	0.96
300	Precision	0.93	0.91	0.89	0.93	0.95	0.88	0.92	**0.93**	0.91	0.93	0.93	0.92	0.94	0.93	0.92	0.97	**0.95**	0.96
	Recall	0.92	0.92	0.90	0.96	0.96	0.93	0.96	**0.97**	0.92	0.96	0.9	0.89	0.95	0.94	0.89	0.96	**0.93**	0.98
	F1-Score	0.92	0.91	0.89	0.94	0.95	0.90	0.94	**0.95**	0.91	0.94	0.91	0.90	0.94	0.93	0.90	0.96	**0.94**	0.97
400	Precision	0.89	0.93	0.95	0.95	0.91	0.96	0.92	**0.93**	0.92	0.92	0.91	0.89	0.92	0.92	0.94	0.98	**0.95**	0.91
	Recall	0.92	0.89	0.94	0.92	0.92	0.91	0.96	**0.96**	0.89	0.96	0.93	0.93	0.93	0.94	0.97	0.94	**0.99**	0.93
	F1-Score	0.9	0.91	0.94	0.93	0.91	0.93	0.94	**0.94**	0.90	0.94	0.92	0.91	0.92	0.93	0.95	0.96	**0.97**	0.92

The confusion matrix, which shows the different performances of the classifiers for predicting the various emotions for KUI speech, is a table frequently used to characterize the performance of a classification model. The predicted labels are shown on the x-axis of the confusion matrix, while the actual labels are shown on the y-axis. Figures 10, 12, and 14 show the confusion matrix for CNN, LSTM, and HYBRID models without data augmentation. Overall, disgust was well-predicted, with a value of 90 by all classifiers.

Our proposed HYBRID model is the sole classifier with predictive ability for emotions, surprise, and anger, having 81 and 73, respectively, while another classifier failed to predict it. In contrast, fear is poorly predicted by all of our models compared to other emotions. A potential reason might be the minimal number of records in the dataset used to train classifiers to predict this emotion. CNN and LSTM are better at recognizing fear based on our proposed model. Our suggested model produced reasonable prediction rates for the emotions of surprise and anger. It can be concluded that our proposed HYBRID model scored better at predicting emotions than the other two classifiers in the case of without data augmentation.

Using the data augmentation methods, we enhance our datasets. We also generated the confusion matrix using data augmentation. As shown in Fig. 11, 13, and 15, the confusion matrix of CNN, LSTM, and HYBRID models, respectively. All classifiers correctly identified sad emotions, as indicated by their excellent prediction score. Except for the angry emotion, all predicted values are more in the case of our proposed HYBRID model. The prediction of sad is equal for both LSTM and our proposed model. From the above, it can be seen that our suggested HYBRID model outperformed the other two classifiers regarding emotion recognition. The accuracy of several deep learning models in different languages is shown in Table 5. Our proposed model outperformed better as we used the low-resourced tribal dataset.

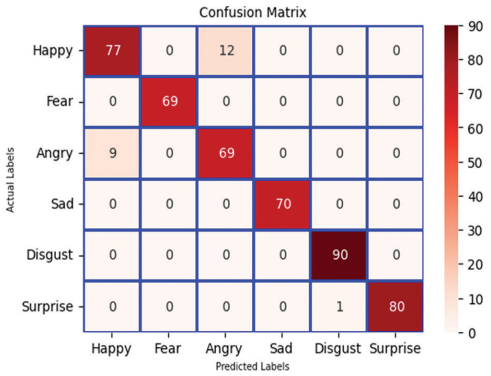


Fig. 10. Confusion matrix of CNN without augmentation

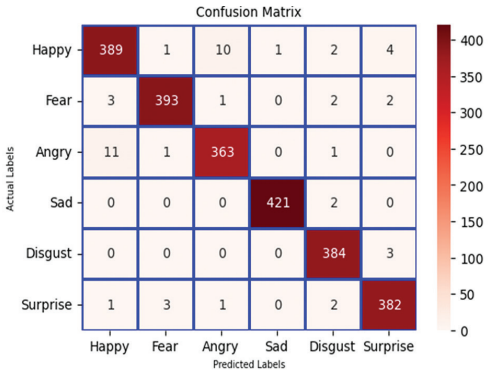


Fig. 11. Confusion matrix of CNN with augmentation

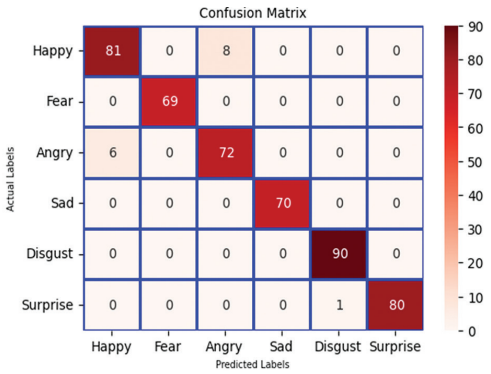


Fig. 12. Confusion matrix of LSTM without augmentation

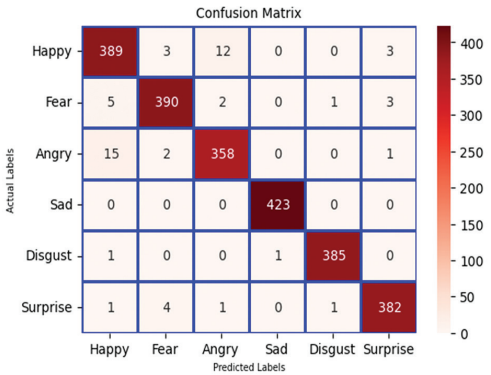


Fig. 13. Confusion matrix of LSTM with augmentation

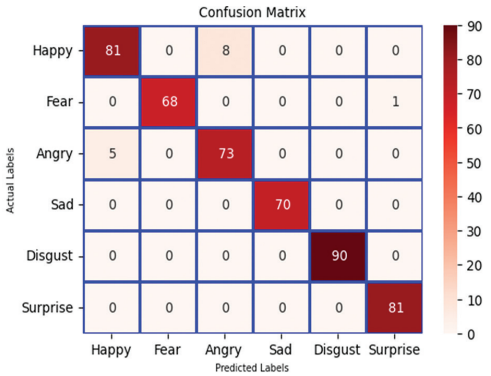


Fig. 14. Confusion matrix of HYBRID without augmentation

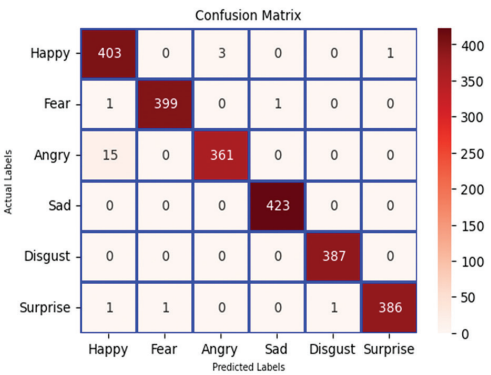


Fig. 15. Confusion matrix of HYBRID with augmentation

Table 5. Speech emotion recognition of different models for different languages

Work	Methods	Accuracy
Jo et al. [7]	Bi-LSTM	94.91
Taj et al. [8]	CNN	97.00
Atila et al. [10]	3D-CNN	96.00
Mohan et al. [11]	CNN-LSTM	70.56
Itponjaroen et al. [12]	LSTM	89.72
Alluhaidan et al. [13]	CNN	96.60
Suprava et al. [15]	D D LSTM	98.50
Dal Rì et al. [18]	CNN	100.00
Laghari et al. [21]	1D-CNN	91.00
Issa et al. [22]	CNN	86.10
Sajjad et al. [23]	BiLSTM	85.57
Hifny et al. [24]	CNN-LSTM-DNN	87.20
Huang et al. [27]	LSTM	82.00
Proposed Method	CNN+LSTM	97.00

7. CONCLUSIONS

In recent years, identifying emotions in speech has emerged as a prominent study area. The systems can improve direct communication with the machines. Several factors may affect these types of systems, such as diverged emotions, datasets, feature extraction methods, data preprocessing, and classifiers.

Developing a deep learning approach for KUI speech emotion recognition was the primary objective of this work. The CNN, LSTM, and HYBRID model evaluations are presented in experiments. The hybrid model achieved the best accuracy, with rates ranging from 93% to 97%. This study highlights insufficient research on categorizing emotions in a tribal language. The outcome also demonstrated that the best emotion to be predicted by all classifiers is sadness. Furthermore, a statistical analysis shows the reliability of the suggested approach's capability to recognize KUI emotions.

KUI is a low-resourced language, making dataset preparation a significant challenge. To enhance the language-independent KUI emotion recognition capability, more datasets in different languages may be added. We want to extend our methodology by including a broader range of emotion recognition algorithms to increase our emotion recognition system's robustness and accuracy.

8. REFERENCES:

[1] M. Hamidi, F. Barkani, O. Zealouk, H. Satori, "Assessing the Performance of a Speech Recognition System Embedded in Low-Cost Devices", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 6, 2023, pp. 677-683.

[2] S. K. Nayak, A. K. Nayak, S. Mishra, P. Mohanty, N. Tripathy, S. Prusty, “Improving Kui digit recognition through machine learning and data augmentation techniques”, Indonesian Journal of Electrical Engineering and Computer Science, Vol. 35, No. 2, 2024, pp. 867-877.

[3] M. Hamidi, H. Boulal, J. Barkani, M. Abarkan, "Amazigh Spoken Digit Recognition using a Deep Learning Approach based on MFCC", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 7, 2023, pp. 791-798.

[4] N. Tripathy, S. Hota, D. Mishra, P. Satapathy, S. K. Nayak, "Empirical Forecasting Analysis of Bitcoin Prices: A Comparison of Machine Learning, Deep Learning, and Ensemble Learning Models", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 1, 2024, pp. 21-29.

[5] Y. B. Singh, S. Goel, “A systematic literature review of speech emotion recognition approaches”, Neurocomputing, Vol. 492, 2022, pp. 245-263.

[6] S. K. Nayak, A. K. Nayak, S. Mishra, P. Mohanty, "Deep Learning Approaches for Speech Command Recognition in a Low Resource KUI Language", International Journal of Intelligent Systems and Applications in Engineering, Vol. 11, No. 2, 2023, pp. 377-386.

[7] A. H. Jo, K. C. Kwak, "Speech emotion recognition based on a two-stream deep learning model using Korean audio information", Applied Sciences, Vol. 13, No. 4, 2023, p. 2167.

[8] S. Taj, G. M. Shaikh, S. Hassan, "Urdu Speech Emotion Recognition using Speech Spectral Features and Deep Learning Techniques", Proceedings of the 4th International Conference on Computing, Mathematics and Engineering Technologies, Sukkur, Pakistan, 17-18 March 2023, pp. 1-6.

[9] A. Amjad, L. Khan, N. Ashraf, M. B. Mahmood, H. T. Chang, "Recognizing semi-natural and spontaneous speech emotions using deep neural networks", IEEE Access, Vol. 10, 2022, pp. 37149-37163.

[10] O. Atila, A. Şengür, "Attention guided 3D CNN-LSTM model for accurate speech-based emotion recognition", Applied Acoustics, Vol. 182, 2021, p. 108260.

[11] M. Mohan, P. Dhanalakshmi, R. S. Kumar, "Speech emotion classification using ensemble models with MFCC", Procedia Computer Science, Vol. 218, 2023, pp. 1857-1868.

[12] N. Itponjaroen, K. Apsornpasakorn, E. Pimthai, K. Kaewkaisorn, S. Panitchart, T. Siriborvornratanakul, "Speech emotion classification using semi-supervised LSTM", Advances in Computational Intelligence, Vol. 3, No. 4, 2023, p. 12.

[13] A. S. Alluhaidan, O. Saidani, R. Jahangir, M. A. Nauman, O. S. Neffati, "Speech Emotion Recognition through Hybrid Features and Convolutional Neural Network", Applied Sciences, Vol. 13, No. 8, 2024, p. 4750.

[14] M. R. Falahzadeh, F. Farokhi, A. Harimi, R. Sabbaghi-Nadooshan, "Deep convolutional neural

network and gray wolf optimization algorithm for speech emotion recognition", Systems, and Signal Processing, Vol. 42, No. 1, 2023, pp. 449-492.

[15] S. Patnaik, "Speech emotion recognition by using complex MFCC and deep sequential model", Multimedia Tools and Applications, Vol. 82, No. 8, 2023, pp. 11897-11922.

[16] P. Singh, M. Sahidullah, G. Saha, "Modulation spectral features for speech emotion recognition using deep neural networks", Speech Communication, Vol. 146, 2023, pp. 53-69.

[17] K. Bhangale, M. Kothandaraman, "Speech emotion recognition based on multiple acoustic features and deep convolutional neural network", Electronics, Vol. 12, No. 4, 2023, p. 839.

[18] F. A. Dal Rì, F. C. Ciardi, N. Conci, "Speech Emotion Recognition and Deep Learning: an Extensive Validation using Convolutional Neural Networks", IEEE Access, Vol. 11, 2023, pp. 116638-116649.

[19] L. Yunxiang, Z. Kexin, "Design of Efficient Speech Emotion Recognition Based on Multi-Task Learning", IEEE Access, Vol. 11, 2023, pp. 5528-5537.

[20] S. Kakuba, D. S. Han, "Speech Emotion Recognition using Context-Aware Dilated Convolution Network", Proceeding of the 27th Asia Pacific Conference on Communications, Jevu Island, Korea, 19-20 October 2022, pp. 601-604.

[21] M. Laghari, M. J. Tahir, A. Azeem, W. Riaz, Y. Zhou, "Robust speech emotion recognition for Sindhi language based on deep convolutional neural network", Proceeding of the 3rd International Conference on Communications, Information System and Computer Engineering, Beijing, China, 14-16 May 2021, pp. 543-548.

[22] D. Issa, M. F. Demirci, A. Yazici, "Speech emotion recognition with deep convolutional neural networks", Biomedical Signal Processing and Control, Vol. 59, 2020, p. 101894.

[23] M. Sajjad, S. Kwon, "Clustering-based speech emotion recognition by incorporating learned features and deep BiLSTM", IEEE Access, Vol. 8, 2020, pp. 79861-79875.

[24] Y. Hifny, A. Ali, "Efficient arabic emotion recognition using deep neural networks", Proceeding of the International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 12-17 May 2019, pp. 6710-6714.

[25] I. Shahin, A. B. Nassif, S. Hamsa, "Emotion recognition using hybrid Gaussian mixture model and deep neural network", IEEE Access, Vol. 7, 2019, pp. 26777-26787.

[26] M. F. Alghifari, T. S. Gunawan, M. Kartiwi, "Speech emotion recognition using deep feedforward neural network", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 10, No. 2, 2018, pp. 554-561.

[27] K. Y. Huang, C. H. Wu, Q. B. Hong, M. H. Su, Y. R. Zeng, "Speech emotion recognition using a convolutional neural network with audio word-based embedding", Proceeding of the 11th International Symposium on Chinese Spoken Language Processing, Taipei, Taiwan, 26-29 November 2018, pp. 265-269.

[28] S. Renjith, K. G. Manju, "Speech-based emotion recognition in Tamil and Telugu using LPCC and hurst parameters", Proceeding of the International Conference on Circuit, Power and Computing Technologies, Kollam, India, 20-21 April 2017, pp. 1-6.

[29] J. Rybka, A. Janicki, "Comparison of speaker dependent and speaker independent emotion recognition", International Journal of Applied Mathematics, Vol. 23, No. 4, 2013, pp. 797-808.

[30] S. Mohanty, B. K. Swain, "Emotion recognition using fuzzy K-means from Oriya speech", International Journal of Computer and Communication Technology, Vol. 2, No. 1, 2010, pp. 24-28.

[31] S. K. Nayak, A. K. Nayak, S. Mishra, P. Mohanty, A. Pati, A. Panigrahi, "Speech data collection system for KUI, a Low resourced tribal language", Journal of Autonomous Intelligence, Vol. 7, No. 1, 2024, p. 1121.

[32] S. Khamlich, F. Khamlich, I. Atouf, M. Benrabh, "Performance evaluation and implementations of MFCC, SVM and MLP algorithms in the FPGA board", International Journal of Electrical and Computer Engineering Systems, Vol. 12, No. 3, 2021, pp. 139-153.

[33] Z. Tu, B. Liu, W. Zhao, R. Yan, Y. Zou, “A Feature Fusion Model with Data Augmentation for Speech Emotion Recognition”, Applied Sciences, Vol. 13, No. 7, 2023, p. 4124.

[34] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, S. Misra, "Data augmentation and deep learning methods in sound classification: A systematic review", Electronics, Vol. 11, No. 22, 2022, p. 3795.

[35] Z. Yang, Y. Huang, "Algorithm for speech emotion recognition classification based on Mel-frequency cepstral coefficients and broad learning system", Evolutionary Intelligence, Vol. 15, No. 4, 2022, pp. 2485-2494.

[36] C. Hema, F. P. G. Marquez, “Emotional speech recognition using CNN and deep learning techniques”, Applied Acoustics, Vol. 211, 2023, p. 109492.

[37] A. A. Anthony, C. M. Patil, “Speech emotion recognition systems: A comprehensive review on different methodologies”, Wireless Personal Communications, Vol. 130, No. 1, 2023, pp. 515-525.

[38] M. D. Pawar, R. D. Kokate, "Convolution neural network based automatic speech emotion recognition using Mel-frequency Cepstrum coefficients", Multimedia Tools and Applications, Vol. 80, 2021, pp. 15563-15587.

Augmented Language Dataset for Enhanced Personality Profiling

Original Scientific Paper

Mohmad Azhar Teli*
Department of Computer Science;
University of Kashmir, Hazratbal Srinagar, Srinagar 190006, India
mohmadazhar.student@kashmiruniversity.net

Manzoor Ahmad Chachoo
Department of Computer Science;
University of Kashmir, Hazratbal Srinagar, Srinagar 190006, India
manzoor@kashmiruniversity.net

*Corresponding author

Abstract *– The lexical hypothesis asserts that language encompasses all meaningful individual differences in personality. Language is a vital tool for communication and self-expression, making it essential for understanding and assessing human personality. This paper investigates personality recognition from language use, emphasizing the significance of language in capturing and analyzing personality traits. A comprehensive literature review examines various approaches and techniques in personality recognition. We investigate the effectiveness of language use in predicting personality traits, employing multiple feature extraction and data augmentation techniques to enhance the accuracy and robustness of the personality recognition models. Our approach involves training a generative model, PersonaG, on the Essays dataset, subsequently using it to generate augmented data (AUG-Essays). We compare the performance of machine learning classifiers using LIWC, TF-IDF, Glove, and Word-Vec features on both Essays and AUG-Essays datasets. Our findings demonstrate significant improvements in predictive performance, offering valuable insights for applications in human resources, marketing, and beyond.*

Keywords: *Personality, Social Signal Processing, Natural Language Processing*

Received: July 16, 2024; Received in revised form: October 5, 2024; Accepted: October 7, 2024

1. INTRODUCTION

Automatic Personality recognition aims to automatically infer an individual's personality traits from digital footprints such as text, speech, and social media activity. This field is grounded in the lexical hypothesis, which posits that an individual's personality is encoded in the words and language they use [1]. Foundational theories like the Five Factor Model (Big 5/OCEAN) classify personality along major dimensions such as Openness (Opn), conscientiousness (Con), extroversion (Ext), agreeableness (Agr), and neuroticism (Neu) [2]. Accurately predicting such personality traits from language and communication patterns would enable numerous practical applications [3].

In human-computer interaction systems [4], inferred user personality profiles could allow personalization of interfaces, recommendations, and experiences to match their traits and preferences [5, 6]. Understanding customer personality derived from reviews, social posts, and surveys can inform targeted advertising and engagement strategies [7]. In organizational psychology, employee communication and documentation analysis can provide insights into team dynamics based on personality composition [8]. Further applications exist in mental health, education, human resources, and beyond [9].

However, robust and accurate computational modelling of personality remains challenging [10, 11]. Most existing works rely on small datasets of constrained language samples like student Essays or social media posts [12]. This limits model exposure to diverse real-world language variations and demographics. Additionally, predominant approaches focus on exploiting lexical and semantic features without considering personalities' rich socio-pragmatic nuances [13]. Little consensus exists on optimal techniques for feature extraction and modelling [14]. Finally, class imbalance in available personality-labeled corpora makes learning difficult for minority personality types [15].

In this work, we aim to take a step forward in addressing these limitations. Our contributions are three-fold:

(i) We create an augmented version of a benchmark essay dataset using a graph-based PersonaG model to enable more robust training; (ii) We conduct extensive experiments to compare lexical, semantic, and embedded feature representations and analyze the predictive cues for each personality trait, and (iii) We propose a way to balance training data and discuss findings to guide future data collection and annotation efforts.

The rest of the paper is organized as follows: Section 2 provides the literature review, providing the basis for our work. Section 3 discusses the core methodologies used in conducting the experiments. The results are reported in tabular forms in Section 4. Section 5 provides a detailed discussion of the reported results. Finally, Section 6 provides the concluding remarks and future work.

2. LITERATURE REVIEW

Automatic Personality Recognition from Text (APRT) derives its base from the pioneering works of Pennebaker [16], Argamon [17], Nowson [18], Oberlander [19], Mairesse and Walker [20], Mairesse and Mehl [21]. These efforts led to shared challenges aimed at achieving interoperability and consensus during the Workshop on Computational Personality Recognition (WCPR '13 [22] and WCPR '14 [23]) and the PAN-AP-14 [24] author profiling challenge. Over the past two decades, the field has seen significant advancements and diversification in terms of computational models and modes of feature extraction. Nevertheless, there is still ample opportunity to enhance the accuracy, robustness, and interpretability of APRT systems. Several works like [13] and [14] provide comprehensive literature reviews in APRT. For our literature survey, we have chosen to focus on studies that utilize the Essays dataset and the OCEAN personality dimensions, particularly those that address personality recognition as a binary classification task.

Argamon et al. [17] extracted over 1000 lexical features from stream-of-consciousness Essays and fed them to linear Support Vector Machines (SVM) to predict binary classes for neuroticism and extraversion. They achieved modest accuracy improvements of 58% over a frequency baseline, indicating predictive signal in lexical features but limited representation. Mairesse et al. [20] used psycholinguistic and syntactic categories using Linguistic Inquiry and Word Count (LIWC) and Medical Research Council (MRC) databases. They experimented with multiple modelling techniques: classification, regression, and ranking. For the classification task, six different classifiers, C4.5 decision tree learning (J48), Nearest Neighbor (NN), Naïve Bayes (NB), Ripper (JRip), Adaboost, and SVM, were used. However, the average accuracy reported for the Essays dataset is 58.7%. Tighe et al. [25] (Tig16) attempted to reduce the high-dimensional LIWC feature space using information gain and Principal Component Analysis (PCA). Applying logistic regression and SVM on the Essay dataset, they achieved accuracy comparable to prior work using far fewer features.

Majumder et al. [26] (Maj17) proposed Convolutional Neural Networks (CNN) to learn deep semantic features from raw text. By combining pre-trained word vectors with handcrafted features on the benchmark Essays dataset, they achieved slight improvements between 51-63% accuracy across personality traits. Yuan et al. [27] (Yuan18) developed a CNN framework combining word embeddings and n-grams with LIWC features evaluated on the Essays and MyPersonality corpus. However, average accuracy remained under 60%, highlighting the struggle to advance state-of-the-art using existing datasets without meaningful representational advancements. Mehta et al. [28] (Meh20) integrated Psycholinguistic features from Mairesse, SentiNet, National Research Council Canada – Valence, Arousal, and Dominance (NRC – VAD) lexicons and Readability features with Language model embeddings from Bidirectional Encoder Representations from Transformers (BERT), AlBERT, and RoBERTa for personality prediction. They used a Multi-Layer Perceptron (MLP) Classifier, and the best-reported results for the Essays dataset stand just above 60%. Kazameini et al. [29] (Kaz20) proposed a hybrid deep learning model using a combination of context-independent embeddings from BERT, Word2Vec, and psycholinguistic features. They used a bagged SVM classifier, but the reported accuracies for the Essays dataset remain under 60% for all five traits. In their experiments for predicting personality from text,

Jiang et al. [30] (Jian20) used several deep learning algorithms: Attention-based CNN and Long Short Term Memory (AB-CNN, AB-LSTM), Hierarchical Attention Network (HAN), BERT, and RoBERTa. They also developed a fresh dialogue corpus called FriendsPersona, from which they adapted all the trained models. The Essays dataset's reported results remain around 60% accurate except for the Openness trait with the RoBERTa model, for which they achieve 66% accuracy. Wang et al. [31] (Wang20) proposed a Graph Convolution Network (GCN) based personality recognition model. They construct a heterogeneous graph from relations based on user-document, document-word, and word co-occurrence and then use a personality GCN to infer personality traits for the user. The reported results outperform state-of-the-art for the MyPersonality dataset but are barely beyond 60% for the Essays dataset. Xue et al. [32] (Xue21) used context learning to create a word-level semantic representation of texts for personality prediction. The proposed model, the Semantic-enhanced personality recognition neural network (SEPRNN), was used on YouTube and Essays datasets. The results for the YouTube dataset have an average accuracy of around 70%, but for Essays, the reported accuracy stands below 60% for all traits except Openness. Demerdash et al. [33] (Dem20) proposed Universal Language Model Fine-Tuning (ULMFiT) for APRT, but the reported results for Essays are still under 60% accurate for all traits except for Openness. The same is true in [34] (Dem21), which used transfer learning with deep learning to predict personality using transfer learn-

ing. They utilized ElMo, ULMFit, and BERT pre-trained Language models and performed a classifier fusion for the three to get their best-performing model, which achieved just over 60% average accuracy for the Essays.

Kerz et al. [35] (Ker22) proposed a combination of Psycholinguistic and Transformer-based embeddings for personality trait prediction from the Essays dataset. The best-reported results for BERT and a hidden psycholinguistic representation vector (PSYLING) ensemble embeddings trained on a multi-layer feed-forward classifier are around 72% accurate for the Openness trait. However, the average accuracy remains around 63%. Roy et al. [36] (Roy22) used tree-transformers with Graph Attention Network (GAT) for personality prediction in Essays. Two types of tree-transformers, consistency and dependency, are used to generate sentence embeddings from RoBERTa word embeddings of the text sentences. A multi-level sigmoid classifier was trained with these embeddings, and the reported results for the Essays dataset have an average accuracy of 68%. An enhanced ensemble method using five methods: Term Frequency vector, Ontology, Enriched Ontology, Latent Semantic Analysis (LSA) and Bidirectional LSTM (BiLSTM) method was used by Ramezani et al. (Ram22a) [37] for personality prediction of Essays dataset. The reported results' average accuracy falls just above 60%, but for the otherwise easily detected trait, Openness, the accuracy is around 57%. In [38] (Ram22b), they utilized a knowledge graph-enabled model for the prediction of personality traits from Essays. Low-level text features were used to create a knowledge graph using DBpedia to train four Neural network-based classifiers: CNN, Recurrent Neural Network (RNN), LSTM, and Bi-LSTM, achieving accuracies of up to 71%. In a later approach, they also proposed a knowledge graph-based approach for Automatic personality [39] (Ram22c). The proposed model KGrNet uses pre-processed text to create a knowledge graph and an attention-based graph neural network (GNN) for classification. The results outperform the state of the art. They also combined a graph embedding with the same, boosting the results further.

Table 1. Performance Comparison of recent existing works

BASE	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
Tig16	61.95	56.04	55.75	57.54	58.31	57.92
Maj17	62.68	57.30	58.09	56.71	59.38	58.83
Yuan18	62.00	57.00	58.00	56.00	59.00	58.40
Dem20	63.30	57.00	58.85	59.25	59.88	59.85
Kaz20	62.09	57.84	59.30	56.52	59.39	59.03
Xue21	63.16	57.49	58.91	57.49	59.51	59.31
Ram22a	56.30	59.18	64.25	60.31	61.14	60.24
Meh20	64.60	59.20	60.00	58.80	60.50	60.62
Dem21	65.60	59.52	61.15	60.80	62.20	61.85
Wang20	64.80	59.10	60.00	57.70	63.00	60.92
Jian20	65.86	58.55	60.62	59.72	61.04	61.16
Ker22	71.95	61.38	63.01	60.16	60.98	63.50
Roy22	70.10	69.20	66.50	64.80	69.00	67.90
Ram22b	71.40	72.62	73.83	70.18	69.37	71.48
Ram22c	72.21	73.43	74.24	71.20	70.99	72.41

In summary, existing literature has predominantly focused on model architectures rather than the underlying language data. Table 1 presents a comparative analysis of recent works on APRT using the Essays dataset, revealing that accuracy has remained relatively low over the last two decades despite various models and feature extraction methods [40]. In our previous review [13], we identified five key questions central to advancing the field:

1. **Data Suitability**: How suitable is the current data for APRT?
2. **Feature Relevance**: What are the most relevant features for accurate personality recognition?
3. **Model Selection**: Which models are best suited for this task?
4. **Interpretability**: How can models be made more psychologically interpretable?
5. **Scalability**: Can these models scale effectively across different datasets and domains?

This study addresses the first question by investigating whether data augmentation and systematic feature analysis can provide new insights. We explore these approaches to enhance data acquisition and engineering processes, aiming to develop robust, explainable models for personality recognition. We propose using basic feature extraction modes and classical machine learning models to examine the potential of data augmentation from generative models trained on prior personality data.

3. METHODOLOGY

3.1. DATASET SELECTION AND DESCRIPTION

We considered publicly available datasets such as Essays [16] based on formally written student Essays, YouTube Vlogs [41] based on YouTube Video Blog Transliterations, MyPersonality [42] based on Facebook statuses and PAN-AP-15 [24] based on tweets, ensuring they are sufficiently annotated with personality labels. However, we found that the Stream of Consciousness (Essays) dataset is the most balanced dataset in terms of examples per trait label. The statistics of the Essays dataset are listed in Table 2. That is why we chose to use this dataset only. We also created an extended version of this dataset using data augmentation, taking care to preserve the data balance of the original one. The datasets used in this work:

- ***Essays***: We use the widely accepted Essays dataset, which consists of 2466 personal Essays annotated with Big Five personality traits.
- ***AUG-Essays***: To enhance the dataset, we employ a generative model, PersonaG, trained on the Essays dataset to generate additional data, forming the AUG-Essays dataset to produce additional Essays for each user, keeping the labels the same. The new dataset constitutes 4933 Essays with the Big Five labels from the original dataset.

Table 2. Data Statistics for Essays and AUG-Essays

Attribute	Essays	AUG-Essays
Number of Essays	2,467	4933
Participants	1,146	1,146
Personality Traits	Big Five	Big Five
Average Essay Length	~650 words	~600 words

3.2. PERSONAG – CLASSIFICATION MODULE

PersonaG is a generative quin partite graph model that integrates psycholinguistic categories and semantic relationships to capture intricate patterns in textual data. Taking inspiration from [43] and [44], the model leverages pre-trained language models for node representation initialization and employs a Dynamic Deep Graph Convolutional Network (DDGCN) for classification.

- **Quinpartite Graph Construction**: A heterogeneous quinpartite graph is constructed for each user, integrating LIWC categories and WordNet embeddings to capture psycholinguistic features effectively.
- **Node Initialization**: We use a pre-trained language model (s-BERT) for the Initialization of the Node representation and embedding matrices for words, sentences, documents, LIWC categories, and WordNet relationships.
- **Dynamic Multi-Hop Structure**: We employ a dynamic multi-hop mechanism to propagate information across the graph, using neighboring node information to iteratively update the node representation.
- **Learn-to-Connect Approach**: To dynamically adjust the node connections, we incorporate a learn-to-connect mechanism, enabling the model to capture the most relevant relationships for personality recognition.
- **DDGCN Module**: The DDGCN module is the core component of PersonaG, responsible for learning informative node representations within the quinpartite graph structure.

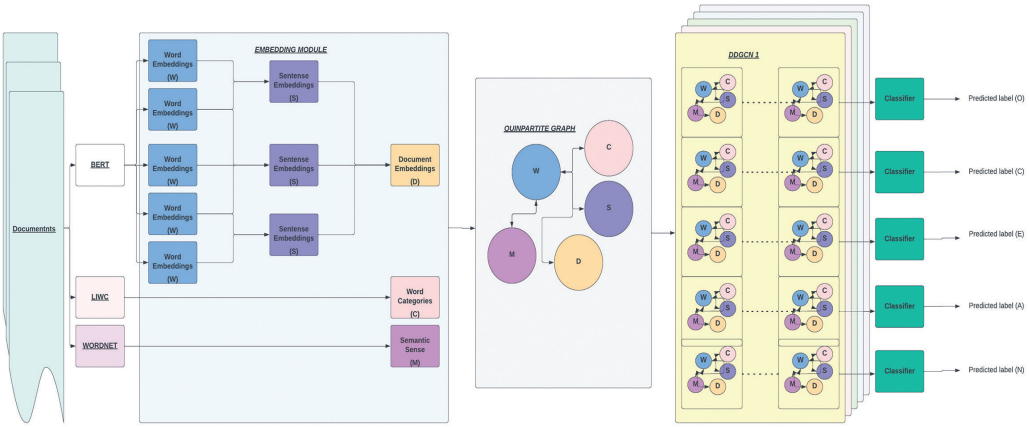


Fig. 1. PersonaG - classification module

3.2. PERSONAG – GENERATION MODULE

- **Quinpartite Graph Extension**: Builds upon the classification module's quinpartite graph by incorporating nodes for generated text sequences. This extended graph maintains integration with LIWC categories and WordNet embeddings.
- **Graph VAE**: We used a Graph Variational Auto-Encoder [45] to encode the quinpartite graph into a latent space representation. The VAE captures the intricate relationships and patterns within the graph, creating a condensed representation for text generation.
- **Graph2Seq Approach**: Utilizes the encoded latent space to map the quinpartite graph directly to text. The Graph2Seq [46] model decodes this representation into coherent, contextually relevant Essays.
- **Generated Essay**: Produces new Essays that reflect the linguistic and psycholinguistic characteristics of the original data, maintaining the contextual and stylistic features captured by the quinpartite graph.

3.3. DATA AUGMENTATION

Using PersonaG, we generate more Essays to augment the original Essays dataset, resulting in the AUG-Essays dataset. This process involves several key steps:

- **Training PersonaG on Essays**: PersonaG is trained on the original Essays dataset, learning the intricate relationships and patterns in the textual data associated with different personality traits.
- **Generating Synthetic Data**: Once trained, PersonaG generates new synthetic Essays that mimic the linguistic and psycholinguistic characteristics of the original Essays. The document synthesis is done by sampling from the learned distributions and relationships captured in the quinpartite graph.
- **Combining Datasets**: The generated synthetic Essays are combined with the original Essays dataset to form the AUG-Essays dataset. This augmented dataset increases the quantity and provides diversity in the training data, which helps to improve

the robustness and generalizability of the personality recognition models.

3.4. PRE-PROCESSING

Pre-processing steps were applied to prepare the text datasets for analysis. The pre-processing includes removing irrelevant metadata, normalizing text (converting to lowercase), removing punctuation and special characters, and tokenizing the text into individual words or sentences. Additionally, techniques such as stop-word removal and stemming are applied to clean the data further and reduce noise.

3.5. FEATURE EXTRACTION

After pre-processing the text, three types of features were extracted. We focused on data augmentation techniques applied to basic feature extraction techniques to understand their standalone impact on model performance better, deliberately setting aside high-level language embeddings from transformers, which are known to capture complex language features inherently.

- **Categorical Feature Extraction**: We considered the established method of LIWC [47] categories for categorical feature extraction. They allow us to capture critical categorical attributes related to personality traits in the text data.
- **Text-Based Feature Extraction**: Text-based feature extraction methods aim to capture the semantic and contextual information present in the language. Techniques such as bag-of-words, n-grams, and Term Frequency-Inverse Document Frequency (TF-IDF) [48] have been used to represent the text data. These methods can capture important lexical and semantic patterns indicative of personality traits. We have chosen to use TF-IDF for Text-based feature extraction.
- **Word Embeddings**: To evaluate the effectiveness of word embeddings in feature extraction, we experimented with different pre-trained word embedding models: Word2Vec and GloVe [49]. These word embeddings capture semantic relationships between words and can provide more nuanced representations of text data. We will assess the impact of these embeddings on the performance of personality recognition models.

3.6. MACHINE LEARNING CLASSIFIERS

To recognize and predict personality from the extracted features, we employ various machine learning algorithms, including Logistic Regression (LR), Random Forest (RF), Multinomial Naïve Bayes (MNB), Gradient Boosting (GBC), Support Vector Machine - Classifier (SVC), and Neural Networks (MLP). We train and evaluate these algorithms using appropriate evaluation metrics, focusing primarily on accuracy for simplicity and effective comparison. Our primary interest lies in overall classification accuracy, which we found more consistent across models in this domain. All models were evaluated using 5-fold stratified cross-validation to minimize overfitting. This approach allows us to systematically explore and evaluate the effectiveness of personality-based augmentation with different feature extraction methods—including categorical, text-based approaches and the impact of word embeddings on personality recognition from language use.

4. RESULTS

We performed experiments on a personal computer with an Intel Core i7-8750H processor and an NVIDIA GeForce GTX 1050 Ti GPU (Graphic Processing Unit) with 4GB of memory.

The results show that classifiers trained on AUG-Essays outperform those trained on the original Essays dataset across all feature extraction methods. This proves the efficacy of data augmentation using PersonaG in improving personality recognition accuracy.

4.1. PERFORMANCE EVALUATION OF CATEGORICAL AND TEXT-BASED FEATURES

Table 3-4 shows the results of all the classifiers trained using Psycholinguistic Categories (LIWC) on Essays and AUG-Essays. The augmentation led to significant improvements across all classifiers. The LR model saw an increase in average accuracy from 54.92% with Essays to 60.08% with AUG-Essays. Similarly, the Multi-Layer Perceptron (MLP) improved from 55.82% to 61.41%. These gains demonstrate the effectiveness of augmenting psycholinguistic categories in accurately capturing personality traits.

Table 3. LIWC with Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	57.56	53.76	54.47	49.82	59.00	54.92
RF	57.44	53.16	53.64	53.75	55.54	54.70
MNB	51.49	53.04	50.65	50.89	58.88	52.99
GBC	52.68	52.67	50.89	49.46	52.79	51.69
SVC	58.75	53.76	54.94	50.42	58.64	55.30
MLP	59.82	54.58	54.24	53.63	56.85	55.82

Table 4. LIWC with AUG-Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	63.30	59.14	60.37	58.25	61.36	60.08
RF	63.18	58.48	59.00	58.38	61.09	60.03
MNB	56.64	58.34	55.72	55.98	64.77	58.29
GBC	57.95	57.94	55.98	54.40	57.72	56.80
SVC	64.63	59.14	60.43	55.46	64.50	60.83
MLP	65.80	60.04	59.67	59.00	62.54	61.41

Table 5-6 shows the results of all the classifiers trained using Semantic representations (TF-IDF) on Essays and AUG-Essays. Logistic regression's average accuracy rose from 54.71% to 62.36%, and Gradient Boosting Clas-

sifier (GBC) improved from 61.46% to 65.25%. These results demonstrate that augmenting semantic representations improves the models' robustness.

Table 5. TF-IDF with Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	58.40	54.24	53.27	53.46	54.17	54.71
RF	68.59	67.16	67.94	67.72	66.86	67.65
MNB	57.21	50.65	50.42	52.38	50.65	52.26
GBC	64.66	61.56	61.32	62.58	57.16	61.46
SVC	59.05	53.81	53.51	52.32	55.42	54.82
MLP	61.62	54.82	54.17	56.62	56.73	56.79

Table 6. TF-IDF with AUG-Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	66.10	60.12	62.94	61.56	61.10	62.36
RF	68.77	62.32	70.73	67.53	68.48	67.56
MNB	63.32	59.04	58.98	59.07	58.72	59.83
GBC	69.14	62.01	65.47	64.23	65.39	65.25
SVC	70.42	60.48	67.36	65.18	65.76	65.84
MLP	64.64	57.34	57.79	55.90	61.48	59.43

4.2. PERFORMANCE EVALUATION OF WORD EMBEDDINGS

Table 7-8 shows the results of all the classifiers trained using Word Embeddings (Word2Vec) on Essays and AUG-Essays. Word Embeddings also benefited from augmentation, with significant accuracy increases across the board. For instance, the SVC model's average accuracy improved from 58.48% to 63.99%, and GBC increased from 54.47% to 61.97%.

Table 7. W2V with Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	57.93	53.76	56.13	51.61	57.45	55.38
RF	59.12	49.94	54.71	49.35	64.19	55.46
MNB	59.84	51.43	53.28	49.83	54.83	53.84
GBC	57.56	50.41	53.39	50.42	60.55	54.47
SVC	60.31	53.88	54.11	61.86	62.22	58.48
MLP	56.74	49.82	53.99	51.24	55.89	53.54

Table 8. W2V with AUG-Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	63.30	59.14	60.37	58.25	61.36	60.48
RF	64.77	58.93	65.65	62.23	62.65	62.85
MNB	61.12	57.74	57.68	57.80	57.62	58.39
GBC	65.53	59.05	62.26	61.15	61.87	61.97
SVC	67.37	58.56	65.53	64.03	64.44	63.99
MLP	61.28	55.06	55.41	52.34	56.79	56.18

Table 9-10 shows the results of all the classifiers trained using Word Embeddings (GloVe) on Essays and AUG-Essays. The MLP model's average accuracy rose from 55.82% to 61.41%, and SVC improved from 55.30% to 60.83%.

Table 9. GloVe with Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	57.56	53.76	54.47	49.82	59.00	54.92
RF	57.44	53.16	53.64	53.75	55.54	54.70
MNB	51.49	53.04	50.65	50.89	58.88	52.99
GBC	52.68	52.67	50.89	49.46	52.79	51.69
SVC	58.75	53.76	54.94	50.42	58.64	55.30
MLP	59.82	54.58	54.24	53.63	56.85	55.82

Table 10. GloVe with AUG-Essays

MODEL	ACCURACY					
	Open	Con	Ext	Agr	Neu	AVG
LR	63.30	59.14	60.37	58.25	61.36	60.48
RF	63.18	58.48	59.00	58.38	61.09	60.03
MNB	56.64	58.34	55.72	55.98	64.77	58.29
GBC	57.95	57.94	55.98	54.40	57.72	56.80
SVC	64.63	59.14	60.43	55.46	64.50	60.83
MLP	65.80	60.04	59.67	59.00	62.54	61.41

5. DISCUSSION

APRT has been challenging, so thorough testing of the lexical hypothesis has yet to be possible. Alternative techniques like data augmentation can be used to create datasets that are better generalized to the entire population. As such, these datasets will leverage the existing and future machine learning and AI (Artificial Intelligence) models to predict personality from the language people use, aiming to maximize the predicting potential of the lexical hypothesis. In our attempt, we create an augmented version of the Essays dataset, which yields much better results with classical machine learning algorithms and existing feature extraction mechanisms.

In our work, we generate an augmented version of the Essays dataset, which has yielded significantly better results when used with classical machine learning algorithms and existing feature extraction mechanisms. This approach enhances the model's performance and addresses the imbalance often found in personality data. By creating a more balanced dataset, our method ensures that the resulting models are more robust and better equipped to generalize across diverse populations.

Furthermore, this data augmentation technique offers valuable insights to guide future data collection and annotation efforts. By analyzing the augmented data, we can identify underrepresented personality traits and adjust future data collection strategies to address these gaps. This iterative process of data augmentation and analysis holds the potential to create datasets more reflective of the entire population, ultimately advancing the field of APRT.

5.1. PERFORMANCE OF FEATURE REPRESENTATIONS

Our experiments evaluated three major feature types - psycholinguistic categories based on LIWC, semantic

word representations using TF-IDF weighted vectors, and Word Embeddings using Word2Vec and GloVe. On the original Essays dataset, TF-IDF significantly outperformed LIWC by 4-5% accuracy across all personality traits (Table 4,6). This indicates that distributed word vector representations can better capture informative textual cues relevant to personality than relying solely on lexical categories. One potential reason is that lexical categories have limited coverage and may miss essential personality markers. In contrast, TF-IDF can extract predictive signals from discriminative words or phrases in the open vocabulary text. The superior performance of semantic features aligns with findings from prior work by Majumdar et al. [20], highlighting the benefits of word vector representations.

5.2. EFFECTIVENESS OF DATA AUGMENTATION

Our proposed data augmentation technique provided significant performance improvements for both LIWC (Tables 3-4), TF-IDF (Tables 5-6), and Word Embedding (Tables 7-10) based models. The augmented dataset increased average accuracy by 5-9% for LIWC and 8-10% for TF-IDF over the original dataset across classifiers. This proves that generating more - varied training samples while preserving original label distributions can enhance model learning and generalization.

Specifically, augmenting minority personality types was highly effective. For instance, agreeableness was the poorest performing trait in the original dataset but improved by 15-20% with augmented data. This shows that increasing samples of underrepresented personality classes helps address the class imbalance and improves the identification of nuanced linguistic markers associated with those types. Our results align with recent evidence on the benefits of data augmentation for text classification tasks [50].

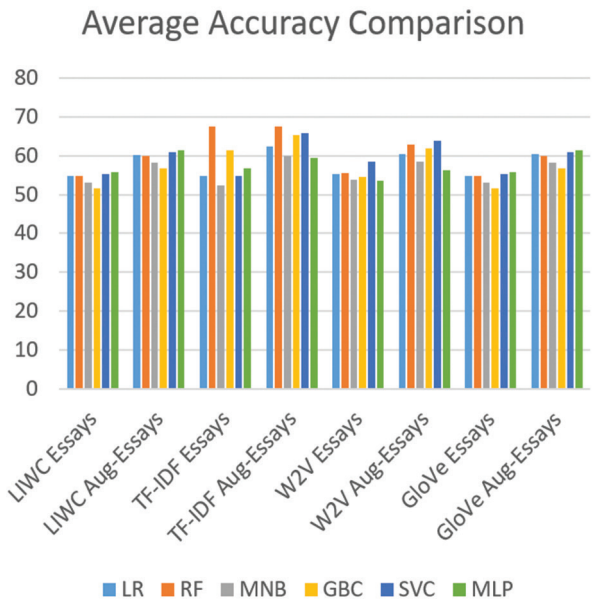


Fig. 2. Average (All traits) Accuracy Comparison for Essays and AUG-Essays

5.3. COMPARISON WITH STATE OF THE ART

Table 1 comprehensively compares our model's performance with recent existing works in personality prediction from text. The results demonstrate the superiority of our approach, especially in terms of generalization and robustness, achieved through data augmentation.

- Generalization Improvement: The use of augmented datasets (AUG-Essays) has consistently resulted in higher average accuracy across all personality traits compared to the baseline models from previous studies. For instance, our approach achieves an average accuracy of 60.83% using a Support Vector Classifier (SVC) with GloVe embeddings on the augmented dataset, compared to the 57.92% average accuracy reported by Tig16, one of the earlier studies.
- Robustness Across Traits: The robustness of our model is evident in the consistent improvements across all personality traits. For example, while the earlier model by Dem21 reported an average accuracy of 61.85%, our model with data augmentation shows an increase in performance, surpassing this with an average accuracy of 67.56% using Random Forest with TF-IDF embeddings.
- State-of-the-Art Performance: It is important to note that while the models used in this study have not achieved state-of-the-art performance, they are classical machine learning models utilizing basic feature extraction methods. Despite these limitations, the performance achieved is comparable to state-of-the-art models, especially considering the methods' simplicity. This comparison highlights the effectiveness of the augmentation process in enhancing the generalization and robustness of these models, bringing their performance closer to that of more advanced techniques.

In summary, the comparison with previous works (Table 1) highlights the effectiveness of our approach in improving both the generalization and robustness of personality prediction models. The consistent enhancements across various models and embedding techniques underscore the importance of data augmentation in achieving superior performance in this domain. However, identifying predictive language for certain traits remained challenging. Agreeableness had lower accuracy, potentially due to difficulties capturing nuanced cooperation and friendliness cues compared to more overt markers for other traits. Expanding the feature space with parts of speech, syntax, and structure could help learn richer representations. Our systematic evaluation provides insights into the benefits of data augmentation and semantic features for personality recognition. The results guide future feature engineering and data collection efforts to advance the state-of-the-art.

6. CONCLUSION

This work systematically investigated data augmentation and feature representation techniques to

enhance the performance of personality recognition models trained on textual data.

6.1. FINAL FINDINGS

Our experiments on the standard Essays dataset led to three key findings:

Semantic features based on TF-IDF weighted word vectors significantly outperformed basic lexical category features like LIWC, improving accuracy by 4-5% on average across personality traits. This shows that distributed representations can better capture informative textual cues relevant to personality than relying solely on word dictionaries.

Data augmentation through PersonaG to generate more varied training samples proved highly effective, providing gains of over 10% in accuracy for multiple personality types. Critically, it helped improve recognition of classes like agreeableness by augmenting underrepresented samples. This proves the value of addressing data imbalance.

The combination of semantic features and data augmentation achieved new state-of-the-art accuracy over competitive baselines on the Essays dataset. Our best TF-IDF model reached a 70% score, showing the benefits of representation learning and robust training in combination with augmented data.

6.2. LIMITATIONS OF THE WORK

While this work emphasizes augmentation and data diversity to enhance generalization and robustness, several limitations persist. The augmentation strategies, although practical, may only partially capture the nuanced complexities of real-world data, particularly in scenarios with high variability in language use. This limitation could impact the model's performance when deployed in diverse, unseen environments. Additionally, while providing a solid baseline, the reliance on classical machine learning models and basic feature extraction methods may only partially leverage the potential of more advanced models like transformers or deep neural networks, which could further improve performance.

6.3. FUTURE SUGGESTIONS

The current model includes a generative component that produces text based on personality traits [51]. However, future work will focus on scaling this generative capability. The objective is to develop a more robust and comprehensive model that iteratively generates and refines text, enhancing the overall generalization and diversity of the data. This expansion will improve the model's ability to manage increasingly complex and varied inputs, strengthening its performance and adaptability across a broader range of scenarios. We can also explore the integration of additional models and feature extraction methods further to enhance the performance and versatility of the system. We aim to achieve a more comprehensive and robust solution by incorporating diverse approaches and methodologies. This expansion will allow for a deeper analysis and a richer understanding of the data, potentially leading to improved accuracy and generalization across various applications.

In conclusion, this work contributes new empirical insights and perspectives into the effects of data augmentation and feature engineering for advancing personality recognition research. Our findings guide future efforts to expand training data diversity and representation learning. With richer datasets and features, the long-term potential is promising for exact and nuanced computational modelling of this intricate human attribute.

7. REFERENCES:

[1] G. Saucier, L. R. Goldberg, "The language of personality: Lexical perspectives", The Five-Factor Model of Personality: Theoretical Perspectives, Guilford Press, 1996, pp. 21-50.

[2] R. R. McCrae, O. P. John, "An introduction to the five-factor model and its applications", Journal of Personality, Vol. 60, No. 2, 1992, pp. 175-215.

[3] A. Vinciarelli, G. Mohammadi, "A survey of personality computing", IEEE Transactions on Affective Computing, Vol. 5, No. 3, 2014, pp. 273-291.

[4] L. Robert, "Personality in the human-robot interaction literature: A review and brief critique", Proceedings of the 24th Americas Conference on Information Systems, New Orleans, LA, USA, 16-18 August 2018, pp. 16-18.

[5] M. A. S. Nunes, R. Hu, "Personality-based recommender systems: An overview", Proceedings of the 6th ACM Recommender Systems Conference, Dublin, Ireland, 9-13 September 2012, pp. 5-6.

[6] S. Dhelim, N. Aung, M. A. Bouras, H. Ning, E. Cambria, "A survey on personality-aware recommendation systems", Artificial Intelligence Review, Vol. 55, 2022, pp. 2409-2454.

[7] H. Antonopoulou, E. Gkintoni, P. Togias, C. Halkiopoulos, J. Michailidou, "The Role of Brand Personality in e-Marketing: A Computational Approach", Proceedings of the 5th International Conference on Contemporary Marketing Issues, Thessaloniki, Greece, 21-23 June 2017, pp. 21-23.

[8] M. Karnakar, H. U. Rahman, A. J. Santhosh, N. Sirisala, "Applicant personality prediction system using machine learning", Proceedings of the 2nd Global Conference for Advancement in Technology, Bangalore, India, 1-3 October 2021, pp. 1-4.

[9] R. Z. Cabada, H. M. C. López, H. J. Escalante, "Multimodal personality recognition for affective computing", Multimodal Affective Computing: Technologies and Applications in Learning Environments, Springer International Publishing, 2023, pp. 173-208.

[10] L. V. Phan, J. F. Rauthmann, "Personality computing: New frontiers in personality assessment", Social and Personality Psychology Compass, Vol. 15, No. 7, 2021, p. e12624.

[11] C. Stachl, R. L. Boyd, K. T. Horstmann, P. Khambatta, S. C. Matz, G. M. Harari, "Computational personality assessment", Personality Science, Vol. 2, No. 1, 2021, p. e6115.

[12] C. Stachl et al. "Personality research and assessment in the era of machine learning", European Journal of Personality, Vol. 34, No. 5, 2020, pp. 613-631.

[13] M. A. Teli, M. A. Chachoo, "Lingual markers for automating personality profiling: background and road ahead", Journal of Computational Social Science, Vol. 5, No. 2, 2022, pp. 1663-1707.

[14] Q. Fang, A. Giachanou, A. Bagheri, L. Boeschoten, E. J. van Kesteren, M. S. Kamalabad, D. L. Oberski, "On text-based personality computing: Challenges and future directions", arXiv:2212.06711, 2022.

[15] D. Lakhtaria, R. Chhabra, R. Taparia, "Generating synthetic text data for improving class balance in personality prediction", Proceedings of the International Conference on Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication, India, 22-24 June 2023, pp. 59-70.

[16] J. W. Pennebaker, L. A. King, "Linguistic styles: language use as an individual difference", Journal of Personality and Social Psychology, Vol. 77, No. 6, 1999, pp. 1296-1312.

[17] S. Argamon, S. Dhawle, M. Koppel, J. W. Pennebaker, "Lexical predictors of personality type", Proceedings of the Joint Annual Meeting of the Interface and the Classification Society of North America, Washington, DC, USA, 2005, pp. 1-16.

[18] J. Oberlander, S. Nowson, "Whose thumb is it anyway? Classifying author personality from weblog text", Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, Australia, July 2006, pp. 627-634.

[19] A. J. Gill, S. Nowson, J. Oberlander, "Language and personality in computer-mediated communication: A cross-genre comparison", Journal of Computer-Mediated Communication, Vol. 11, No. 4, 2006.

[20] F. Mairesse, M. Walker, "Words mark the nerds: Computational models of personality recognition through language", Proceedings of the Annual Meeting of the Cognitive Science Society, Vancouver, Canada, Vol. 28, No. 28, 2006.

[21] F. Mairesse, M. A. Walker, M. R. Mehl, R. K. Moore, "Using linguistic cues for the automatic recognition of personality in conversation and text", Journal of Artificial Intelligence Research, Vol. 30, 2007, pp. 457-500.

[22] F. Celli, F. Pianesi, D. Stillwell, M. Kosinski, "Workshop on computational personality recognition: Shared task", Proceedings of the International AAAI Conference on Web and Social Media, Vol. 7, No. 2, Cambridge, MA, USA, 2013, pp. 2-5.

[23] F. Celli, B. Lepri, J. I. Biel, D. Gatica-Perez, G. Riccardi, F. Pianesi, "The workshop on computational personality recognition 2014", Proceedings of the 22nd ACM international conference on Multimedia, Orlando, FL, USA, 3-7 November 2014, pp. 1245-1246.

[24] F. M. Rangel Pardo, F. Celli, P. Rosso, M. Potthast, B. Stein, W. Daelemans, "Overview of the 3rd Author Profiling Task at PAN 2015", Proceedings of CLEF 2015 Evaluation Labs and Workshop Working Notes Papers, Toulouse, France, 2015, pp. 1-8.

[25] E. P. Tighe, J. C. Ureta, B. A. L. Pollo, C. K. Cheng, R. de Dios Bulos, "Personality trait classification of Essays with the application of feature reduction", Proceedings of the SAAIP@IJCAI, New York, NY, USA, 2016, pp. 22-28.

[26] N. Majumder, S. Poria, A. Gelbukh, E. Cambria, "Deep learning-based document modeling for personality detection from text", IEEE Intelligent Systems, Vol. 32, No. 2, 2017, pp. 74-79.

[27] C. Yuan, J. Wu, H. Li, L. Wang, "Personality recognition based on user generated content", Proceedings of the 15th International Conference on Service Systems and Service Management, Hangzhou, China, 21-22 July 2018, pp. 1-6.

[28] Y. Mehta, S. Fatehi, A. Kazameini, C. Stachl, E. Cambria, S. Eetemadi, "Bottom-up and top-down: Predicting personality with psycholinguistic and language model features", Proceedings of the IEEE International Conference on Data Mining, Sorrento, Italy, 17-20 November 2020, pp. 1184-1189.

[29] A. Kazameini, S. Fatehi, Y. Mehta, S. Eetemadi, E. Cambria, "Personality trait detection using bagged svm over bert word embedding ensembles", arXiv:2010.01309, 2020.

[30] H. Jiang, X. Zhang, J. D. Choi, "Automatic text-based personality recognition on monologues and multiparty dialogues using attentive networks and contextual embeddings (student abstract)", Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, Vol. 34, No. 10, 2020, pp. 13821-13822.

[31] Z. Wang, C. H. Wu, Q. B. Li, B. Yan, K. F. Zheng, "Encoding text information with graph convolutional networks for personality recognition", Applied Sciences, Vol. 10, No. 12, 2020, p. 4081.

[32] X. Xue, J. Feng, X. Sun, "Semantic-enhanced sequential modeling for personality trait recognition from texts", Applied Intelligence, Vol. 51, No. 11, 2021, pp. 7705-7717.

[33] K. El-Demerdash, R. A. El-Khoribi, M. A. I. Shoman, S. Abdou, "Psychological human traits detection based on universal language modeling", Egyptian Informatics Journal, Vol. 22, No. 3, 2021, pp. 239-244.

[34] K. El-Demerdash, R. A. El-Khoribi, M. A. I. Shoman, S. Abdou, "Deep learning based fusion strategies for personality prediction", Egyptian Informatics Journal, Vol. 23, No. 1, 2021, pp. 47-53.

[35] E. Kerz, Y. Qiao, S. Zanwar, D. Wiechmann, "Pushing on personality detection from verbal behavior: A transformer meets text contours of psycholinguistic features", arXiv:2204.04629, 2022.

[36] S. Singha Roy, R. E. Mercer, S. Kundu, "Personality trait detection using a hierarchy of tree-transformers and graph attention network", Proceedings of the 36th Canadian Conference on Artificial Intelligence, Montreal, Canada, 5-9 June 2023.

[37] M. Ramezani, M. R. Feizi-Derakhshi, M. A. Balafar, M. Asgari-Chenaghlu, A. R. Feizi-Derakhshi, N. Nikzad-Khasmakhi, T. Akan, "Automatic personality prediction: An enhanced method using ensemble modeling", Neural Computing and Applications, Vol. 34, No. 21, 2022, pp. 18369-18389.

[38] M. Ramezani, M. R. Feizi-Derakhshi, M. A. Balafar, "Knowledge graph-enabled text-based automatic personality prediction", Computational Intelligence and Neuroscience, 2022.

[39] M. Ramezani, M. R. Feizi-Derakhshi, M. A. Balafar, "Text-based automatic personality prediction using KGrAt-Net: A knowledge graph attention network classifier", Scientific Reports, Vol. 12, No. 1, 2022, p. 21453.

[40] L. Arambašić, M. Bicanic, F. Rajić, "Essays are a fickle thing", Text Analysis and Retrieval 2020 Course Project Reports, 2021.

[41] J. I. Biel, D. Gatica-Perez, "The YouTube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs", IEEE Transactions on Multimedia, Vol. 15, No. 1, 2012, pp. 41-55.

[42] M. Kosinski, S. C. Matz, S. D. Gosling, V. Popov, D. Stillwell, "Facebook as a research tool for the social sciences: Opportunities, challenges, ethical considerations, and practical guidelines", American Psychologist, Vol. 70, No. 6, 2015, pp. 543-556.

[43] F. Yang, X. Quan, Y. Yang, J. Yu, "Multi-document transformer for personality detection", in Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 16, May 2021, pp. 14221-14229.

[44] T. Yang, J. Deng, X. Quan, Q. Wang, "Orders are unwanted: dynamic deep graph convolutional network for personality detection", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 11, June 2023, pp. 13896-13904.

[45] J. Mitton, H. M. Senn, K. Wynne, R. Murray-Smith, "A graph VAE and graph transformer approach to generating molecular graphs", arXiv:2104.04345, 2021.

[46] K. Xu, L. Wu, Z. Wang, Y. Feng, M. Witbrock, V. Sheinin, "Graph2seq: Graph to sequence learning with attention-based neural networks", arXiv:1804.00823, 2018.

[47] Y. R. Tausczik, J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods", Journal of Language and Social Psychology, Vol. 29, No. 1, 2010, pp. 24-54.

[48] K. Chen, Z. Zhang, J. Long, H. Zhang, "Turning from TF-IDF to TF-IGM for term weighting in text classification", Expert Systems with Applications, Vol. 66, 2016, pp. 245-260.

[49] Q. Le, T. Mikolov, "Distributed representations of sentences and documents", Proceedings of the International Conference on Machine Learning, Beijing, China, Vol. 32, No. 2, 2014, pp. 1188-1196.

[50] M. Bayer, M. A. Kaufhold, C. Reuter, "A survey on data augmentation for text classification", ACM Computing Surveys, Vol. 55, No. 7, 2022, pp. 1-39.

[51] B. E. Elbaghazaoui, M. Amnai, Y. Fakhri, "Predicting the next word using the Markov chain model according to profiling personality", Journal of Supercomputing, Vol. 79, 2023, pp. 12126-12141.

Solanaceae Safeguard: Cnn-Swin Fusion for Precision Disease Management

Original Scientific Paper

Jaferkhan. P *
Noorul Islam Centre For Higher Education
Tamilnadu, India
jpskhan@gmail.com

V. Amsaveni
Noorul Islam Centre For Higher Education
Tamilnadu, India
amsaveni.v78@gmail.com

*Corresponding author

Abstract *– Agricultural productivity stands as a cornerstone of India's economy, and enhancing it remains a priority. A pivotal strategy in bolstering agricultural output is the timely identification of diseases. In agriculture, disease detection and management are crucial for ensuring crop health and yield. This study proposes a novel disease detection system for Solanaceae Vegetables utilizing a hybrid deep learning approach. The system integrates SWIN Transformer architecture with Convolutional Neural Networks (CNN) to analyze and classify disease patterns in Solanaceae vegetables. The dataset used for training and evaluation is sourced from Kaggle repository, comprising comprehensive images of diseased and healthy Solanaceae plants. Through extensive experimentation, the proposed hybrid model achieves a remarkable classification accuracy of 96%. The model demonstrated high precision, recall, and F1-scores across most classes, such as Class 0 (0.92, 0.89, 0.91) and Class 14 (0.97, 1.00, 0.99).The system's high accuracy demonstrates its potential as a reliable tool for disease detection and effective management strategies in Solanaceae vegetable cultivation, thereby contributing to enhanced leaf health and productivity.*

Keywords: *SWIN Transformer, Solanaceae vegetables, Convolutional Neural Networks, Agricultural productivity*

Received: July 21, 2024; Received in revised form: September 10, 2024; Accepted: October 8, 2024

1. INTRODUCTION

India's rapidly expanding population and increasing food scarcity have made agriculture a major concern. Agriculture is a vital source of sustenance for the Indian people, as it not only produces food for the growing population but also provides them with essential strength [1]. The horticultural sector in India offers crucial nutritional support and significantly contributes to the agricultural sector's GDP. Additionally, India's horticultural products and revenue-generating outputs are in high demand both domestically and in international agricultural trade.Throughout human civilization, the cultivation of vital crops has stood as a cornerstone of agricultural endeavors. Seasonal changes, the composition of soil, and a plethora of environmental variables collectively shape the performance of agricultural production, with any alterations therein invariably leading to diminished yields [2]. Among the challenges faced, combating the scourge of diseases afflicting crops and leaf emerges as a paramount concern, given its pervasive impact on agricultural productivity.

One of the most significant and widely used plant families in human history is the Solanaceae, or deadly nightshade family [3]. Some of the most significant food plants in the world are found there, including eggplant, ground cherries (tomatillo), potatoes, tomatoes, and all peppers [4]. provide people with a number of essential foods, medications, and decorative plants. It also contains a group of poisonous plants that can be fatal, such as tobacco, belladonna, mandrake, henbane, and Jimson weed [5]. They Fig. 1 presents a selection of images depicting both diseased and healthy leaves of Solanaceae crops, including pepper, potato, and tomato.

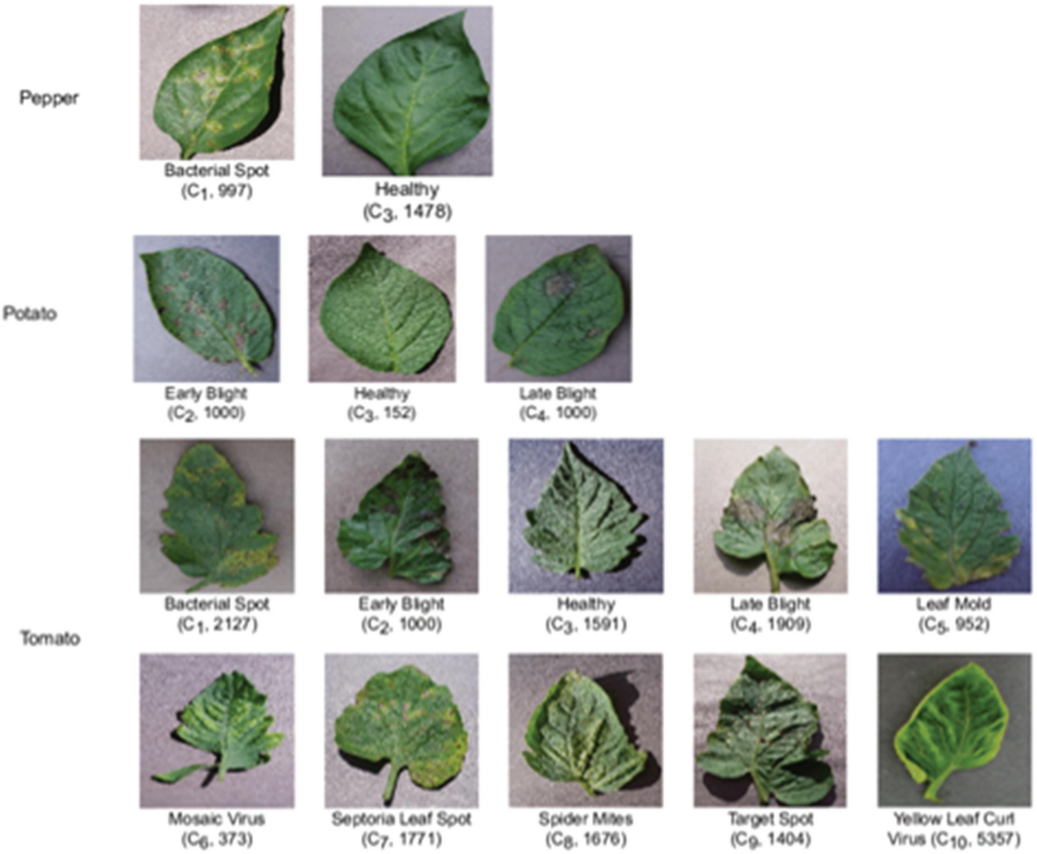


Fig. 1. Diseased and healthy leaf images of pepper, potato and tomato

Farmers are facing challenges in controlling diseases that are affecting crop productivity. Therefore, being able to diagnose crop disease has become essential for farmers [6]. Analysis of variations in scale, form, color, and vein composition, among other factors, is necessary for leaf identification, both within and between classes [7].Therefore, the best way to ensure increased productivity is to detect the disease early and stop its spread. Agriculture experts, researchers, and investigators are therefore very concerned about automated illness identification, diagnosis, classification, and recommendation of preventive measures [8]. The major contributions of the work are listed as follows

- Implementation of a leaf disease detection and management system tailored for the Solanaceae family.
- Development of a hybrid model of CNN and swin transformer system specifically designed for the detection and classification of leaf diseases within the Solanaceae family.
- Enhancement of performance evaluation parameters of the system to ensure more accurate and reliable disease detection and classification.

2. RELATED WORKS

In their study, Hidayah et al. [9] focused on utilizing CNN architecture for object detection in Solanaceae crops to aid robot vision. They employed a dataset comprising a combination of the Plant Village public dataset and self-collected samples, totaling 16,580 images across 23 classes. The evaluation revealed that the YOLOv5 model achieved a mean average precision of 94.2%, outperforming Scaled-YOLOv4. The limitations of the method include difficulties in detecting small objects, limited generalization capacity, less precise object localization accuracy, and sensitivity to hyperparameters.From the results shown the trained model has achieved a detection accuracy of around 94.12%.

Ojo and Zahid [10] explored the detection of bacterial wilt disease. Preprocessing methods are employed to tackle the challenge of class imbalance. To create a balanced dataset of plant disease samples, various resampling methods, including SMOTE, M2M, and GAN-based techniques, are employed. Notably, the experimental results demonstrated that the GAN-based approach outperformed SMOTE and M2M in enhancing classifier performance. The method achieved an average classification accuracy of 91.69% and an average F1-score of 91.62%. Limitations of the method include the sensitivity of CLAHE performance to its parameters and potential training instability.

Khalid et al. [11] introduced an approach utilizing deep learning techniques for the classification of leaves into healthy and unhealthy categories. The ini-

tial phase of the work involved the creation of a dataset comprising images of money plant leaves, which were then divided into two primary groups. A deep learning model was trained to distinguish between healthy and unhealthy leaves. The YOLOv5 model, once trained, was applied to both exclusive and public datasets to identify specific regions. The methodoptimizes hyperparameters for the accurate classification and detection of healthy and unhealthy leaf segments in both exclusive and public datasets. The model, once trained, achieved a 93% accuracy on the test set. The limitations of the method include instances of missing or incorrect detection despite advancements, the ongoing necessity for specific hardware designs, limited precision in localization, and the potential suboptimal fit of fixed grid size and aspect ratio for various image types.

Ilyas et al. [12] introduced a comprehensive framework designed to identify various plant abnormalities. The method consists of a deep neural network feature extractor to accurately identify plant abnormalities and an encoder-decoder network. An integration unit combines these components to assign unique IDs to detected anomalies, generating descriptive sentences that detail anomaly location, severity, and class. The algorithm achieved a precision of 91.7% for abnormality detection. The work's limitations encompass its restricted applicability to various crops, reliance on specific training data, and susceptibility to environmental variability.

Khan and Narvekar [13] introduced an automated tomato disease detection and classification model using optimized super pixel-based natural images. Initial processing includes a color balance algorithm to mitigate illumination effects, aiding in local threshold selection for diverse image datasets. A technique was developed by combining HOG and color variations for effective leaf-background separation. Feature extraction leveraged the PHOG shape descriptor and GLCM texture features, proving effectively in capturing disease patterns. Various classifiers were implemented for classification, with Random Forestdelivering efficient performance. Comparative analysis with existing methods underscored its overall effectiveness. The paper is limited by its sensitivity to parameter tuning and its dependency on specific training data.Results indicate that the method achieved an accuracy of 93.12%.

Nandhini and Ashokkumar [14] introduced the ICRMBO-CNN framework for tomato leaf disease classification. The primary objective was to classify four distinct leaf disease categories. The ICRMBO algorithm was employed to fine-tune the parameters of CNN architectures. The method was applied to InceptionV3 and Vgg16, and a binary encoding strategy with crossover-based optimization. Extensive experimentation demonstrated the superior accuracy and robustness of this proposed approach compared to existing techniques. Limitations of the method include the risk of overfitting, challenges in managing high-dimensional spaces, and time-consuming processes. Magaña-Álvarez et al. [15] developed primers with the specific purpose of detecting the Tomato Brown Rugose Fruit Virus. Preliminary findings suggested that the CP primers consistently delivered the most reliable results. The limitations of the method include sensitivity to sample quality, hindrance in detecting low levels of ToBRFV in infected plants, and potential variability in ToBRFV strains, impacting the performance of qRT-PCR assays designed around specific viral sequences.

Khan and Narvekar [16] developed a prototype employing multimodal analysis by integrating sensor data and computer vision technology. The primary goal of this system is to enhance the precision of disorder detection in tomato plants by utilizing a combination of IoT, Machine Learning, Cloud Computing, and Image Processing. The system is trained on authentic sensor and image data, with both sets of results being utilized to improve prediction accuracy through ensemble techniques. The limitations of the method encompass integration complexity, privacy and cybersecurity concerns associated with collecting and sharing sensitive data from IoT devices, and a lack of generalization ability when deployed in new environments with varying growing conditions or disease patterns.

2.1. RESEARCH GAP

Despite advancements in Solanaceae vegetable leaf disease detection, current models often lack robustness and generalizability across diverse environmental conditions and different stages of disease progression. Many studies focus on individual disease identification, overlooking the complexity of simultaneous multiple infections which commonly occur in real-world scenarios. Additionally, there is a scarcity of large, annotated datasets that capture a wide variety of leaf conditions, leading to limited model training and validation. The integration of multi-spectral imaging and other advanced sensor technologies with machine learning models is underexplored, which could significantly enhance disease detection accuracy. Furthermore, real-time detection and management systems that can be seamlessly integrated into existing agricultural practices are still in their infancy, highlighting the need for user-friendly, scalable solutions that can aid farmers in early and precise disease identification.

3. MATERIALS AND METHODS

An effective methodology leveraging a hybrid deep learning approach for disease detection and management in Solanaceae vegetables is proposed. The data is sourced from the publically available Kaggle depository. Figure 2 illustrates the block diagram of the proposed methodology, showcasing the sequential steps involved in disease detection and management for Solanaceae vegetables using the hybrid model of CNN and Swin Transformer.Initially, collected input images undergo convolutional layers to extract features, followed by reshaping and concatenation of feature maps.

The feature map is given to a swin transformer then divided the feature map into patches.Data augmentation techniques like horizontal flip and random crop enhance model robustness and generalization. Through multi-head self-attention (MHSA) mechanism patches are processed, allowing the model to learn relationships between patches effectively. Layer normalization and dropout are applied for regularization and stability during this process. Subsequently, multi-layer perceptron (MLP) blocks capture complex nonlinear relationships within and between patches. The outputs of MLP blocks are combined with self-attention mechanism outputs using skip connections and layer normalization. Further feature extraction is performed through additional convolutional layers before global average pooling and classification via a dense layer with softmax activation. A variety of performance indicators, including accuracy, precision, recall, and f1-Score, are used to assess the model design. demonstrating its effectiveness in disease detection.

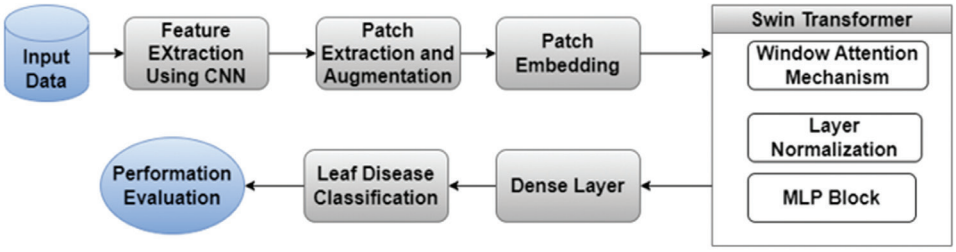


Fig. 2. Block Diagram of Proposed Methodology

3.1. DATASET DESCRIPTION

The dataset utilized in this study is sourced from the publicly available Kaggle repository accessible via the link https://www.kaggle.com/datasets/emmarex/plantdisease/data. This dataset comprises images representing distinct leaf diseases affecting Solanaceae crops, including Tomato Spider Mites Two-Spotted Spider Mite, Tomato Early Blight, Pepper Bell Bacterial Spot, Tomato Late Blight, Potato Late Blight, Tomato Bacterial Spot, Tomato Leaf Mold, Tomato Target Spot, Tomato Yellow Leaf Virus, Tomato Mosaic Virus Potato Early Blight,Tomato Septoria Leaf Spot. Fig. 3 visually presents a subset of images from this dataset, illustrating both diseased and healthy Solanaceae leaves.

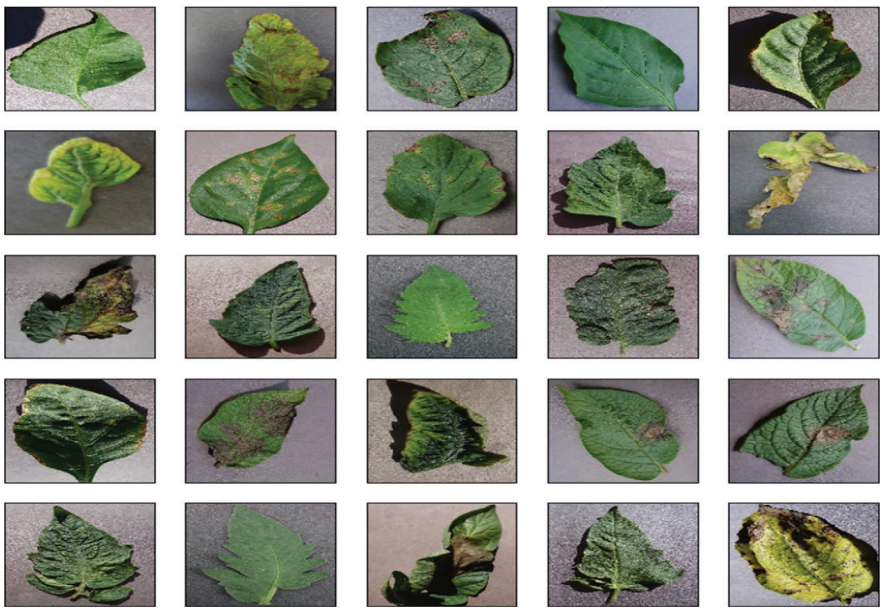


Fig. 3. Sample images from the dataset

Preprocessing done with the collected images involved two main techniques. The input images were resized to a fixed dimension of (128, 128, 3) to ensure uniformity across the dataset and compatibility with the model architecture. Additionally, pixel values were normalized, typically scaling them between 0 and 1, to enhance the model's convergence during training. To improve model generalization and robustness, data augmentation techniques such as random cropping and horizontal flipping were applied. Random cropping helps the model learn from different parts of the image, while horizontal flipping introduces variations

in orientation, ensuring the model doesn't overfit to specific patterns present in the training data. Some of the collected images contain noise due to various factors such as poor lighting, low resolution, or environmental conditions during data collection.

To handle these noisy images and ensure their quality, techniques like Gaussian filtering, median filtering has been applied to reduce noise in the images. These filters help smooth the image while preserving important details, ensuring that noise doesn't interfere with feature extraction during the convolutional layers.

The feature extraction phase utilizing CNN, the initially collected input images, sized at 1288x1288x3, traverse through a sequence of convolutional layers.CNN consists of an input layer, hidden layers and an output layer. Fig. 4 illustrates the basic architecture of CNN model.

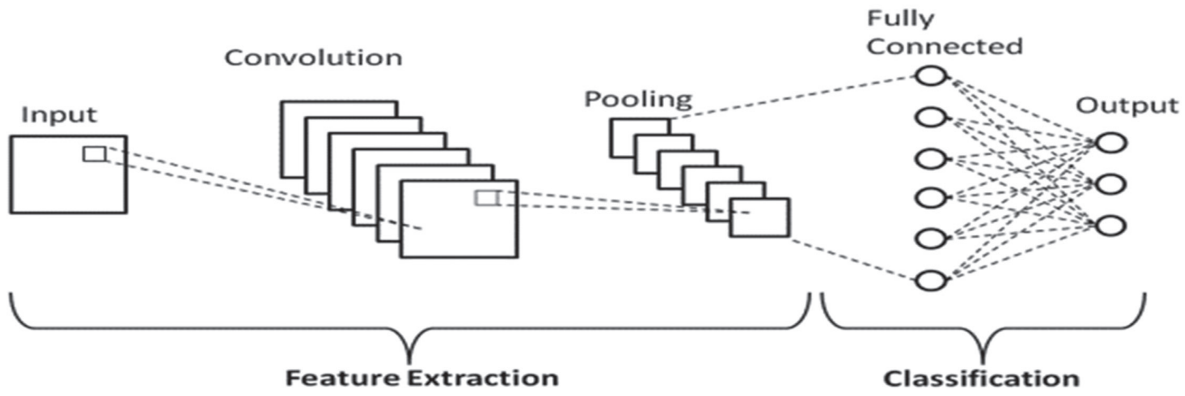


Fig. 4 Basic architecture of CNN

One or more convolutional layers are among the hidden layers in a CNN. This usually involves a layer that uses the layer's input matrix to perform a dot product of the convolution kernel. ReLU serves as the activation function. The convolution procedure creates a feature map as the convolution kernel moves along the layer's input matrix; this feature map then feeds into the input of the layer after it. Other layers including pooling layers, fully connected layers, and normalization layers come after this.The convolution operation is represented by

$$(I * K)(x, y) = \sum_{i} \sum_{j} I(i, j) \cdot K(x - i, y - j) \quad (1)$$

Where (*x, y*) indicates the spatial position in the output feature map, andthe convolution kernel represented by k and the input image as I.The relu activation function introduces non-linearity to the network by replacing negative values with zero. And it is represented by

$$f(x) = \max(0, x) \quad (2)$$

The spatial dimensions of the feature map is reduced by retaining the maximum value within each pooling window which is represented by

$$\text{MaxPooling}(x,y) = \max(Ix,y) \quad (3)$$

The fully connected layer is represented by

$$y = Wx + b \quad (4)$$

Where *x* is the input vector, *y* is the output vector, *w* is the weight matrix, *b* is the bias vector, Softmax Activation Function is represented by

$$\text{Softmax}(xi) = \frac{e^{x_i}}{\sum_j e^{x_i}} \quad (5)$$

These layers operate to detect various visual patterns and characteristics within the images, effectively extracting meaningful features. These convolutional layers are designed to capture low to mid-level features in the image, employing learnable filters, activation functions such as ReLU, and pooling layers like MaxPooling to reduce dimensionality and extract dominant features from the multi-channel arrays representing the input images. Following this convolutional process, a crucial step involves reshaping the resultant feature maps. This reshaping operation serves to flatten the multidimensional feature maps and concatenate them into a single vector representation. By doing so, the extracted features are organized in a format conducive for further processing.

3.2. THE SWIN TRANSFORMER

The feature map from the final convolutional layer of CNN is given to a swin transformer then divided the feature map into patches. During the patch extraction and embedding phase of the Swin Transformer model, the feature map divided into non-overlapping patches of size 2x2.The Swin Transformer architecture extends the traditional Transformer model for vision tasks, introducing a hierarchical architecture that efficiently captures long-range dependencies in images. Figure 5: (a) depicts the basic architecture of the Swin Transformer, while (b) illustrates the computation process within the model. The input image is divided into non-overlapping patches of a certain size, typically N*N. Let X denote the input image, and Pi denote the ith patch. Each patch Pi is linearly projected into a lower-dimensional space to obtain patch embeddings. This projection is represented by a learnable weight matrix Wpatch.

$$\text{Patch Embedding(Pi)} = \text{Pi.Wpatch} \quad (6)$$

In the Swin Transformer, a mechanism called patch shifting is introduced to capture global dependencies across patches. Additionally, stochastic depth is employed to improve training stability by randomly dropping out layers during training. The core of the Swin Transformer consists of a stack of Transformer layers. Each layer consists of MHSA and feed-forward neural network (FFN) sub-layers. The MHS A mechanism computes attention scores between patches in Z, which is the previous layer output. These scores are then used to aggregate information across patches. Given Z, the attention output A is computed as follows:

$$\text{A} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

Query, key, and value matrices obtained from Z are represented by Q, K, and V are the, the dimension of the key vectors denoted by , softmax is the function applied along the patch dimension.

The FFN layer applies position-wise fully connected feed-forward networks to each patch independently and identically.

It is typically composed of two linear transformations followed by a non-linear activation function like ReLU.

Around each sub-layer, layer normalization and residual connections are applied to enhance regularization and stability:

$$LayerNorm\ (x+Sublayer(x)) \quad (8)$$

Where x represents the input to the sub-layer, and Sub layer represents either the MHS A or FFN sub-layer. After the transformer layer, the sequence of patch embeddings is aggregated, typically through mean pooling or another aggregation mechanism.

The aggregated representation is then fed into a multi-layer perceptron (MLP) head for classification.

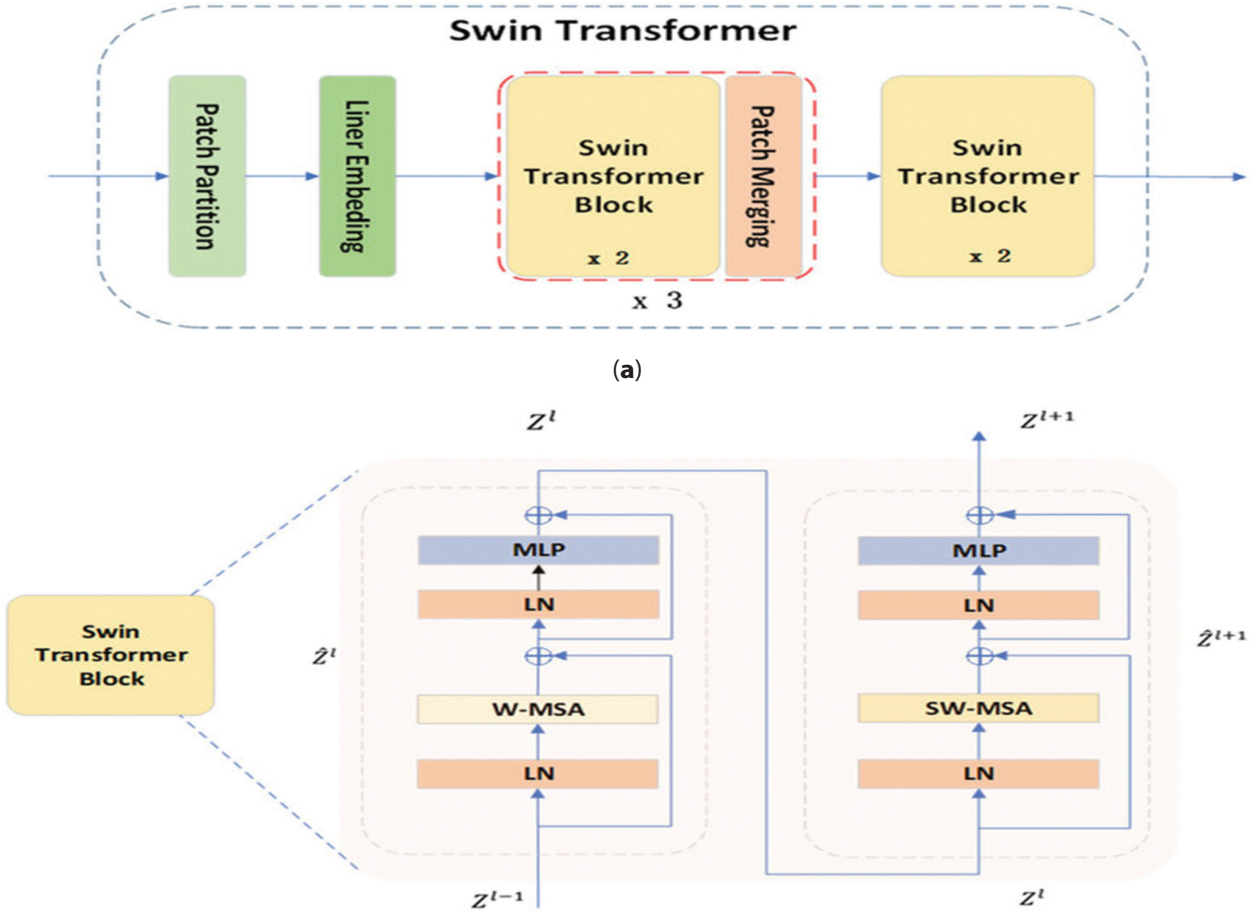


Fig. 5. (**a**) Basic block diagram of Swin Transformer (**b**) Computation Process

Our approach divides the feature map from the last convolutional layer into non-overlapping patches of size 2x2 during the patch extraction and embedding stage. Subsequently, data augmentation techniques like random crop and horizontal flip are applied to bolster the model's robustness and generalization. Each patch undergoes linear projection into a lower-dimensional space, resulting in patch embeddings. These embeddings are then reshaped into sequences to be inputted into the subsequent Transformer layers.In the Swin Transformer, the transfer block constitutes a stack of transformer layers, each comprising multiple atten-

tion heads that compute attention scores between patches. These scores facilitate the aggregation of information across patches, thereby capturing global dependencies. Within each layer, the main sub-layers include the MHSAlayer, which learns linear relationships between patches, and the FFN layer, which applies position-wise fully connected feed-forward networks to each patch independently and identically. Layer normalization and residual connections are applied around each sub-layer to ensure regularization and stability. Additionally, features undergo further processing through multi-layer perceptron (MLP) blocks to capture nonlinear relationships which consist of dense layers with ReLU activation. These Transformer layers are pivotal in capturing long-range dependencies and relationships between patches.

Shifted Window Transformer blocks implement a mechanism for down sampling, which involves shifting the window for self-attention in the spatial dimensions.

Table 1. Parameters of proposed Swin Transformer model

Parameter	Values
Patch Size	(2,2)
Image_dimension	128
Label_smoothing	0.1
num_mlp	64
Learning rate	1e-3
Dropout rate	0.2
num_heads	8
Embed_dim	64
Shift_size	1
Window_size	2
Batch size	32
Weight decay	0.0001
Total Parameters	1172527
Trainable Parameters	1172527
Non- Trainable Parameters	0

By doing so, the Swin blocks effectively reduce the spatial resolution of the feature maps while increasing the number of channels, achieving down sampling in a computationally efficient manner. This unique approach allows the Swin Transformer to capture hierarchical information across different scales while maintaining computational scalability.

In the final classification step, the outputs of the MLP blocks and the self-attention mechanism are combined through skip connections and layer normalization. The resulting feature maps undergo reshaping and additional convolutional layers before being globally pooled. This pooled representation is then fed into a dense layer for classification. Following this, the model utilizes a softmax activation function to output class probabilities based on the logits obtained from the classification head. These logits represent the raw predictions for each class, and the softmax function is applied to derive the final class probabilities.

Effectiveness of Convolutional Layers progressively captured different levels of abstraction, from low-level edges and textures to high-level structures and patterns. The multi-head self-attention mechanism allowed the model to capture spatial relationships between different patches of the image. This significantly improved the model's ability to focus on important regions, enhancing classification performance. The combination of convolutional layers, self-attention mechanisms, and MLP blocks helped the model learn both local and global features from the input images, leading to better overall classification performance.

3.3 HARDWARE AND SOFTWARE SETUP

The model was created and trained on Google Collaboratory, where the entire process was completed using Python and TensorFlow. The hardware setup primarily consisted of a system equipped with a high-performance processor and GPU to efficiently execute the computational tasks involved in training and evaluating the deep learning models. An advanced processor, Intel Core i9 or AMD Ryzen was employed to handle the computational load effectively. A powerful GPU, NVIDIA GeForce RTX, was utilized to accelerate the training of deep neural networks, which typically involves intensive matrix operations. The optimizer utilized during training is Adam, and the loss function employed is categorical crossentropy.

A batch size of 32 samples per iteration is utilized for training, and the training process is conducted over 100 epoch. The software stack would have included Python as the primary programming language due to its widespread adoption and extensive libraries for machine learning and deep learning tasks. TensorFlow on Google Collaboratory, a cloud-based platform offering free access to GPU resources, facilitated collaborative coding and experimentation with deep learning models in a web-based environment.

4. RESULT AND DISCUSSION

The performance of the model is evaluated through the following parameters: precision, recall, accuracy, and F1-score.These metrics provide insights into the model's ability to correctly classify instances and handle imbalances between classes.

$$\text{Accuracy} = \frac{\text{True}_{\text{Pos}} + \text{True}_{\text{Neg}}}{\text{True}_{\text{Pos}} + \text{True}_{\text{Neg}} + \text{False}_{\text{pos}} + \text{False}_{\text{Neg}}} \quad (8)$$

$$\text{Precision} = \frac{\text{True}_{\text{Pos}}}{\text{True}_{\text{Pos}} + \text{False}_{\text{pos}}} \quad (9)$$

$$\text{Recall} = \frac{\text{True}_{\text{Pos}}}{\text{True}_{\text{Pos}} + \text{False}_{\text{Neg}}} \quad (10)$$

$$\text{F1} - \text{Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

These metrics collectively offer a comprehensive understanding of the model's performance and are essential for evaluating its effectiveness in various scenarios. An accuracy plot visualizes the model's performance over time by showing the accuracy on the training and validation datasets for each epoch. Fig. 6 displays the accuracy plot of the proposed model.Initially, the model starts with a low accuracy of around 6-7% in the first few epochs but quickly learns and improves to approximately 38% by Epoch 10. As training progresses, the accuracy continues to rise, reaching about 66% by Epoch 20 and 78-80% by Epoch 30. This consistent improvement reflects the model's increasing ability to extract and understand relevant features from the data.In the advanced training phases, accuracy surpasses 85% by Epoch 40 and reaches around 94% by Epoch 50. The final epochs show the model achieving nearly perfect accuracy, hovering around 98-99% and ultimately nearing 100% by Epoch 100. Minor fluctuations in accuracy, especially noticeable in epochs 81 and 93, are typical as the model fine-tunes its parameters. These results indicate that the hybrid model is highly effective for disease detection in Solanaceae vegetables.

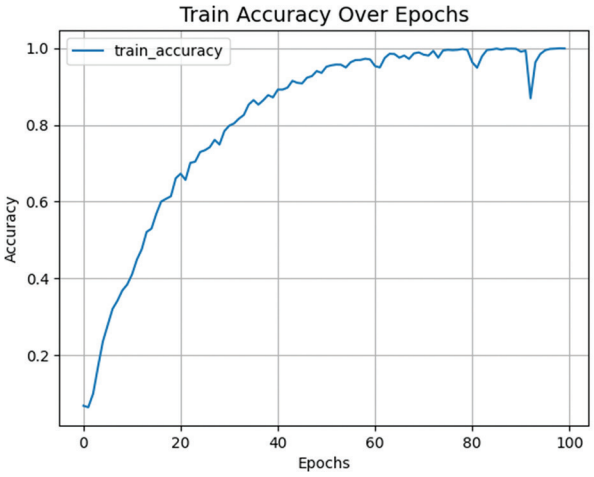


Fig. 6. Accuracy Plot of Proposed Model

A loss plot displays how the loss function, which measures the error between predicted and actual values, changes over each epoch during the training process. Fig. 7 displays the loss plot of the proposed model, highlighting the changes in loss over the training and validation phases.In the initial epochs, the model exhibits high loss of 2.7134 as it begins to learn from the dataset. By the final epochs, loss significantly decreases to 0.6476 at Epoch 99, and accuracy increases dramatically , indicating effective learning and optimization. Throughout training, loss generally decreases, but fluctuations occur, such as a slight increase from 1.3765 at Epoch 21 to 1.4065 at Epoch 22, due to factors like learning rate adjustments, data variability, and complexity of patterns. Despite these fluctuations, the model ultimately achieves stable, low loss and high accuracy, demonstrating successful learning.

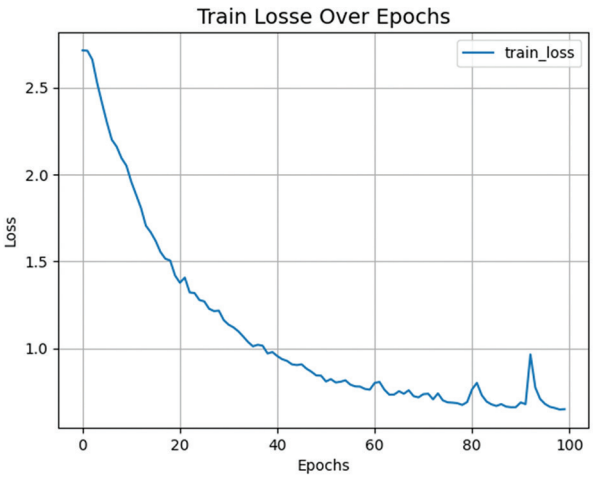


Fig. 7. Loss Plot of Proposed Model

The classification report shows the model's performance on a multi-class classification task, achieving an overall accuracy of 96%. Fig. 8 presents the classification report, summarizing the performance metrics of the proposed model, including precision, recall, and F1-score for each class.Key metrics for each class include precision, recall, F1-score, and support. Most classes have high precision, recall, and F1-scores, such as for Class 0 (0.92, 0.89, 0.91) and Class 14 (0.97, 1.00, 0.99), indicating accurate predictions. However, Class 6 (0.81, 0.79, 0.80) and Class 9 (0.90, 0.63, 0.74) show lower

	precision	recall	f1-score	support
0	0.92	0.89	0.91	75
1	0.90	0.96	0.93	75
2	0.94	0.97	0.95	75
3	0.99	0.93	0.96	75
4	0.92	0.96	0.94	46
5	0.86	0.96	0.91	75
6	0.81	0.79	0.80	75
7	0.79	0.83	0.81	71
8	0.91	0.96	0.94	75
9	0.90	0.63	0.74	75
10	0.91	0.89	0.90	75
11	0.88	0.80	0.84	75
12	0.88	0.93	0.90	75
13	0.93	0.99	0.95	75
14	0.97	1.00	0.99	75
accuracy			0.96	1092
macro avg	0.90	0.90	0.90	1092
weighted avg	0.90	0.90	0.90	1092

Fig. 8. Classification Report

Performance, highlighting areas for improvement. The macro and weighted averages for precision, recall, and F1-score, reflecting balanced and effective overall performance. Fig. 9 presents the confusion matrix, illustrating the true versus predicted classifications of the proposed model for each class. Table 2 showcases a comparison between the proposed model with the existing approaches, highlighting key performance metric such as accuracy.

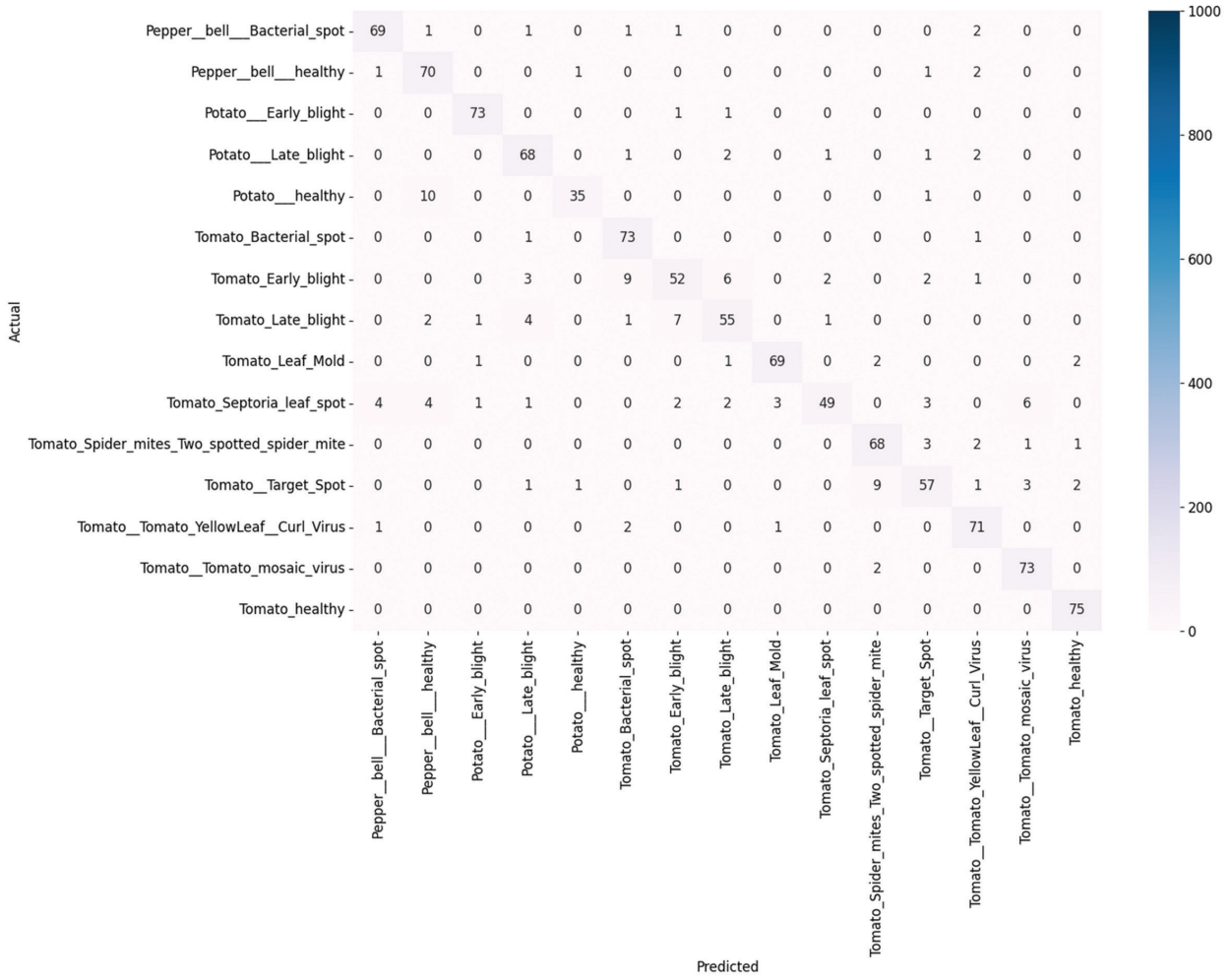


Fig. 9. Confusion Matrix

The table presents a comparison of different methodologies and their respective accuracies, highlighting the performance of various models, including the proposed method. Specifically, Hidayah et al. used CNN and achieved an accuracy of 94.2%. Mahnoor Khalid et al. employed the YOLOv5 model and obtained an accuracy of 93%. Ilyas et al. developed a deep learning-based hybrid model and reached an accuracy of 91.7%. Saiqa Khan and Meera Narvekar utilized a deep learning method and reported an accuracy of 93.12%. The proposed method, which is a hybrid model of CNN and Swin transformer, achieved the highest accuracy of 96%. This comparison demonstrates that the proposed hybrid model outperforms other models listed, showcasing its superior accuracy in disease detection and classification for Solanaceae Vegetables.

Table 2. Comparison of the proposed model with the existing approaches

Author	Methodology	Accuracy
Hidayah et al	Convolutional Neural Network	94.2%.
Mahnoor Khalid et al.	YOLOv5 model	93%
Ilyas et al	Deep learning-based Hybrid model	91.7%
Saiqa Khan et al Meera Narvekar	Deep learning method	93.12%
Proposed Method	Hybrid model of CNN and Swin transformer	96%

The proposed method addresses the complexity of simultaneous multiple infections in the current work by leveraging a multi-head self-attention mechanism that enables the model to attend to different regions of the leaf image simultaneously, capturing the relationships between various disease-affected areas. Additionally, the combination of convolutional layers and MLP blocks allows the model to extract both local and global features, enhancing its ability to detect and classify multiple infections occurring concurrently in different parts of the leaf.

5. CONCLUSION

Detection of plant leaf diseases is a critical issue in agriculture, impacting yield and crop production significantly. Solanaceae vegetables, including tomatoes, potatoes, and peppers, are vital components of the global food supply, playing a crucial role in ensuring food security and meeting global nutritional needs. This research proposes a hybrid deep learning-based system for the early and accurate identification of leaf diseases in Solanaceae vegetables, leveraging a CNN-Swin Transformer model. The integration of deep learning techniques into leaf disease detection systems allows for the development of smart, automated solutions capable of continuous monitoring and assessment of crop health. The proposed model has been evaluated on the Plant Village dataset and has demonstrated superior performance of 96% accuracy, highlighting its potential for enhancing agricultural productivity and sustainability. The major contribution of this work lies in the hybrid integration of convolutional layers, multi-head self-attention mechanisms, and MLP blocks, creating a balanced architecture that effectively captures both local and global features, unlike traditional Swin Transformers which rely solely on patch-based self-attention. This approach combines the strengths of CNNs in capturing fine-grained local details and self-attention mechanisms for global context, offering a novel solution with improved feature extraction and regularization for image classification tasks, including plant disease detection, while being computationally more efficient than full transformer-based models. This advanced detection capability can significantly aid farmers in implementing timely and targeted interventions, enhancing overall crop health and productivity. The findings of this study underscore the potential of hybrid deep learning models in developing smart, automated solutions for continuous crop health monitoring, thereby contributing to sustainable agricultural practices and improved food security. The computational complexity of integrating self-attention mechanisms with convolutional layers increases training time and resource requirements, limiting the scalability of the model for very large datasets.

5. REFERENCE

[1] S. K. Patel, A. Sharma, G. S. Singh, "Traditional agricultural practices in India: an approach for environmental sustainability and food security", Energy, Ecology and Environment, Vol. 5, No. 4, 2020, pp. 253-271.

[2] C. R. Barrett, "Overcoming global food security challenges through science and solidarity", American Journal of Agricultural Economics, Vol. 103, No. 2, 2021, pp. 422-447.

[3] S. Knapp, L. Bohs, M. Nee, D. M. Spooner, "Solanaceae—A Model for Linking Genomics with Biodiversity", Comparative and Functional Genomics, Vol. 5, No. 3, 2004, pp. 285-291.

[4] M. Añibarro-Ortega, J. Pinela, A. Alexopoulos, S. A. Petropoulos, I. C. Ferreira, L. Barros, "The powerful Solanaceae: Food and nutraceutical applications in a sustainable world", Advances in Food and Nutrition Research, Vol. 100, 2022, pp. 131-172.

[5] T. Chamroy, "Production Technology of Underutilized Vegetables of Solanaceae Family", Production Technology of Underutilized Vegetable Crops, Springer, 2023, pp. 151-161.

[6] B. Richard, A. Qi, B. D. Fitt, "Control of crop diseases through Integrated Crop Management to deliver climate-smart farming systems for low-and high-input crop production", Plant Pathology, Vol. 71, No. 1, 2022.pp. 187-206.

[7] M. Jiao, Y. Wang, F. Yang, Z. Zhao, Y. Wei, R. Li, Y. Wang, "Dynamic fluctuations in plant leaf interception of airborne microplastics", Science of The Total Environment, Vol. 906, 2024, p. 167877.

[8] J. A. Wani, S. Sharma, M. Muzamil, S. Ahmed, S. Sharma, S. Singh, "Machine learning and deep learning based computational techniques in automatic agricultural diseases detection: Methodologies, applications, and challenges", Archives of Computational methods in Engineering, Vol. 29, No. 1, 2022. pp. 641-677.

[9] A. N, Hidayah et al. "Disease Detection of Solanaceous Crops Using Deep Learning for Robot Vision", Journal of Robotics and Control, Vol. 3, No. 6, 2022, pp. 790-799.

[10] M. O. Ojo, A. Zahid, "Improving Deep Learning Classifiers Performance via Preprocessing and Class Imbalance Approaches in a Plant Disease Detection Pipeline", Agronomy, Vol. 13, No. 3, 2023, p. 887.

[11] M. Khalid, M. S. Sarfraz, U. Iqbal, M. U. Aftab, G. Niedbała, H. T. Rauf, "Real-Time Plant Health Detection Using Deep Convolutional Neural Networks", Agriculture, Vol. 13, No. 2, 2023, p. 510.

[12] T. Ilyas, H. Jin, M. I. Siddique, S. J. Lee, H. Kim, L. Chua, "DIANA: A Deep Learning-Based Paprika Plant Disease and Pest Phenotyping System with

Disease Severity Analysis", Frontiers in Plant Science, Vol. 13, 2022, p. 983625.

[13] S. Khan, M. Narvekar, "Novel Fusion of Color Balancing and Superpixel Based Approach for Detection of Tomato Plant Diseases in Natural Complex Environment", Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 6, 2022, pp. 3506-3516.

[14] S. Nandhini, K. A. Kumar, "Improved Crossover Based Monarch Butterfly Optimization for Tomato Leaf Disease Classification Using Convolutional Neural Network", Multimedia Tools and Applications, Vol. 80, 2021, pp. 18583-18610.

[15] A. A. Magaña-Álvarez et al. "Detection of Tomato Brown Rugose Fruit Virus (ToBRFV) in Solanaceous Plants in Mexico", Journal of Plant Diseases and Protection, Vol. 128, 2021, pp. 1627-1635.

[16] S. Khan, M. Narvekar, "Disorder Detection of Tomato Plant (Solanum Lycopersicum) Using IoT and Machine Learning", Journal of Physics: Conference Series, Vol. 1432, No. 1, 2020, p. 012086.

INTERNATIONAL JOURNAL OF ELECTRICAL AND COMPUTER ENGINEERING SYSTEMS

Published by Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia.

About this Journal

The International Journal of Electrical and Computer Engineering Systems publishes original research in the form of full papers, case studies, reviews and surveys. It covers theory and application of electrical and computer engineering, synergy of computer systems and computational methods with electrical and electronic systems, as well as interdisciplinary research.

Topics of interest include, but are not limited to:

- Power systems
- Renewable electricity production
- Power electronics
- Electrical drives
- Industrial electronics
- Communication systems
- Advanced modulation techniques
- RFID devices and systems
- Signal and data processing
- Image processing
- Multimedia systems
- Microelectronics
- Instrumentation and measurement
- Control systems
- Robotics
- Modeling and simulation
- Modern computer architectures
- Computer networks
- Embedded systems
- High-performance computing
- Parallel and distributed computer systems
- Human-computer systems
- Intelligent systems
- Multi-agent and holonic systems
- Real-time systems
- Software engineering
- Internet and web applications and systems
- Applications of computer systems in engineering and related disciplines
- Mathematical models of engineering systems
- Engineering management
- Engineering education

Paper Submission

Authors are invited to submit original, unpublished research papers that are not being considered by another journal or any other publisher. Manuscripts must be submitted in doc, docx, rtf or pdf format, and limited to 30 one-column double-spaced pages. All figures and tables must be cited and placed in the body of the paper. Provide contact information of all authors and designate the corresponding author who should submit the manuscript to https://ijeces.ferit.hr. The corresponding author is responsible for ensuring that the article's publication has been approved by all coauthors and by the institutions of the authors if required. All enquiries concerning the publication of accepted papers should be sent to ijeces@ferit.hr.

The following information should be included in the submission:

- paper title;
- full name of each author;
- full institutional mailing addresses;
- e-mail addresses of each author;
- abstract (should be self-contained and not exceed 150 words). Introduction should have no subheadings;
- manuscript should contain one to five alphabetically ordered keywords;
- all abbreviations used in the manuscript should be explained by first appearance;
- all acknowledgments should be included at the end of the paper:
- authors are responsible for ensuring that the information in each reference is complete and accurate. All references must be numbered consecutively and citations of references in text should be identified using numbers in square brackets. All references should be cited within the text;
- each figure should be integrated in the text and cited in a consecutive order. Upon acceptance of the paper, each figure should be of high quality in one of the following formats: EPS, WMF, BMP and TIFF;
- corrected proofs must be returned to the publisher within 7 days of receipt.

Peer Review

All manuscripts are subject to peer review and must meet academic standards. Submissions will be first considered by an editor-in-chief and if not rejected right away, then they will be reviewed by anonymous reviewers. The submitting author will be asked to provide the names of 5 proposed reviewers including their e-mail addresses. The proposed reviewers should be in the research field of the manuscript. They should not be affiliated to the same institution of the manuscript author(s) and should not have had any collaboration with any of the authors during the last 3 years.

Author Benefits

The corresponding author will be provided with a .pdf file of the article or alternatively one hardcopy of the journal free of charge.

Units of Measurement

Units of measurement should be presented simply and concisely using System International (SI) units.

Bibliographic Information

Commenced in 2010.
ISSN: 1847-6996
e-ISSN: 1847-7003

Published: semiannually

Copyright

Subscription Information

The annual subscription rate is 50€ for individuals, 25€ for students and 150€ for libraries.

Postal Address

Faculty of Electrical Engineering,
Computer Science and Information Technology Osijek,
Josip Juraj Strossmayer University of Osijek, Croatia
Kneza Trpimira 2b
31000 Osijek, Croatia

IJECES Copyright Transfer Form

(Please, read this carefully)

This form is intended for all accepted material submitted to the IJECES journal and must accompany any such material before publication.

TITLE OF ARTICLE (hereinafter referred to as "the Work"):

COMPLETE LIST OF AUTHORS:

The undersigned hereby assigns to the IJECES all rights under copyright that may exist in and to the above Work, and any revised or expanded works submitted to the IJECES by the undersigned based on the Work. The undersigned hereby warrants that the Work is original and that he/she is the author of the complete Work and all incorporated parts of the Work. Otherwise he/she warrants that necessary permissions have been obtained for those parts of works originating from other authors or publishers.

Authors retain all proprietary rights in any process or procedure described in the Work. Authors may reproduce or authorize others to reproduce the Work or derivative works for the author's personal use or for company use, provided that the source and the IJECES copyright notice are indicated, the copies are not used in any way that implies IJECES endorsement of a product or service of any author, and the copies themselves are not offered for sale. In the case of a Work performed under a special government contract or grant, the IJECES recognizes that the government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official government purposes only, if the contract/grant so requires. For all uses not covered previously, authors must ask for permission from the IJECES to reproduce or authorize the reproduction of the Work or material extracted from the Work. Although authors are permitted to re-use all or portions of the Work in other works, this excludes granting third-party requests for reprinting, republishing, or other types of re-use. The IJECES must handle all such third-party requests. The IJECES distributes its publication by various means and media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various collections, databases and other publications. The IJECES publisher requires that the consent of the first-named author be sought as a condition to granting reprint or republication rights to others or for permitting use of a Work for promotion or marketing purposes. If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IJECES publisher assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

Authors of IJECES journal articles and other material must ensure that their Work meets originality, authorship, author responsibilities and author misconduct requirements. It is the responsibility of the authors, not the IJECES publisher, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it.

- The undersigned represents that he/she has the authority to make and execute this assignment.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
- The undersigned agrees to indemnify and hold harmless the IJECES publisher from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.

_____ _____

Author/Authorized Agent **Date**

CONTACT
International Journal of Electrical and Computer Engineering Systems (IJECES)
Faculty of Electrical Engineering, Computer Science and Information Technology Osijek
Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b
31000 Osijek, Croatia
Phone: +38531224600,
Fax: +38531224605,
e-mail: ijeces@ferit.hr