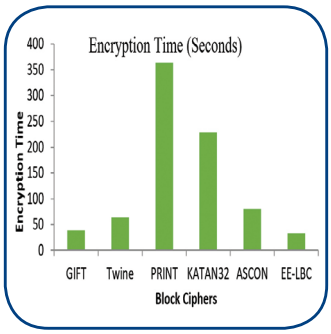
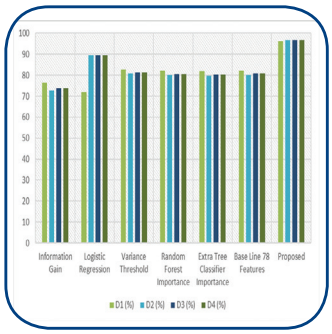
Volume 16, Number 7, 2025 ISSN 1847-6996

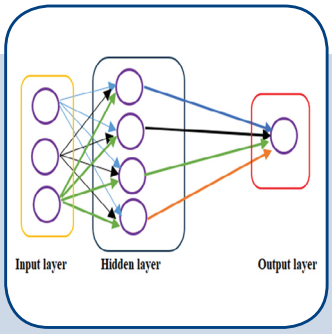
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY OSIJEK

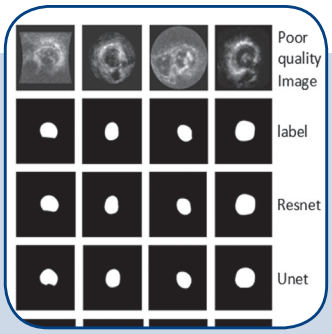
IJECES International Journal of Electrical and Computer Engineering Systems

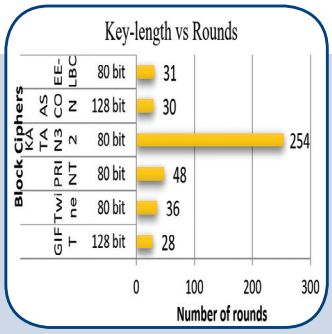
International Journal of Electrical and Computer Engineering Systems











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 7, 2025 | Pages 497 - 564

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

Optimized Triple-Slot Patch Antenna with Electromagnetic Band Gap Structures for Enhanced Performance

Original Scientific Paper

Gowri Chaduvula*
Department of Electronics and Communication Engineering,
Jawaharlal Nehru Technological University, Kakinada,
Andhra Pradesh, India
gowrich.ece@gmail.com

Leela Kumari B
JNTU Kakinada,
Faculty of Engineering, Department of Electronics and Communication
Kakinada, Andhra Pradesh, India
leela8821@yahoo.com

*Corresponding author

Abstract *– This paper presents an antenna integrated with an Electromagnetic Bandgap (EBG) structure to enhance its radiation performance compared to a conventional antenna. MEBG (Mushroom EBG) and EEBG (Edge via EBG) structures are analyzed, integrating MEBG with the triple-slot patch antenna, which demonstrates superior performance. Using the same conventional dimensions, the proposed antenna achieves a gain of 6.15 dB, a directivity of 7.51 dB, and a return loss of 37 dB at 5.2 GHz, providing a 1.92 dB gain improvement over the conventional design. This design is simulated using the HFSS software. The measurement results are validated with simulation results. The fabricated, compact antenna can be used for IoT applications at 5.2 GHz.*

Keywords: *patch antenna, EBG, triple-slot, MEBG, EEBG, gain, IoT*

Received: January 1, 2025; Received in revised form: April 5, 2025; Accepted: April 9, 2025

1. INTRODUCTION

Today, with the evolution of wireless communication technologies, billions of IoT devices are connected to the internet, exchanging information wirelessly [1]. The key requirement for these systems is the integration of compact, high-performance antennas to ensure reliable data transfer [2]. Antennas are the primary components used to transmit or receive information in wireless communication, making their design a vital factor in optimizing IoT connectivity. Microstrip patch antennas are preferred in wireless applications since they are lightweight, small, easy to fabricate, and adaptable to feeding networks [3].

There is a high demand for IoT applications, hence, research is being put forth for compact, high-gain antenna designs that can serve longer distances with improved bandwidth. However, conventional microstrip patch antennas suffer from low gain and low bandwidth. In the literature, several strategies have been applied to antennas to improve their radiation characteristics, making them appropriate for IoT applications. In [4], an L-slotted patch antenna is designed for IoT applications at 2.4 GHz. Improved patch antenna performance is achieved by incorporating slots in the patch to enable operation at 868 MHz for IoT applications [5]. A pixel antenna array is designed for high-gain IoT applications [6]. The U-slot microstrip antenna is designed for IoT applications [7]. DGS, combined with slots in patch antenna, is used for IoT applications [8]. However, designing compact antennas for effective integration with IoT devices is challenging.

Designing antennas on high permittivity substrates results in compact sizes, but it introduces challenges related to the generation of surface waves [9]. Patch antenna performance can be significantly affected by surface waves travelling along the interface between metal and dielectric boundaries. This dispersion causes distortion in radiation patterns and leads to multipath interference, resulting in issues like deep nulls, increased back lobe radiation, reduced gain, and overall decreased performance [10].

A powerful and widely adopted technique for mitigating surface waves is the use of Electromagnetic Band Gap (EBG) structures. EBG structures are defined as "artificial periodic objects that prevent the propagation of electromagnetic waves in a specified range of frequencies for all incident angles and polarization states" [11]. Due to their unique bandgap features, these are considered a special type of metamaterial. Additionally, EBGs exhibit high-impedance surface properties and artificial magnetic conductor (AMC) behavior, making them valuable in antenna engineering and microwave circuits. EBGs are classified based on their geometry into 1D, 2D, and 3D types. Mushroom EBGs (MEBGs) are the most popular choice in two-dimensional structures for effectively reducing surface waves. As discussed in the literature, novel EBG structures are integrated with antenna configurations to enhance performance. Improved isolation and lower radar cross-sections are achieved with frequency-selective surfaces in MIMO antennas [12]. TVDS-EBG is implemented for bandwidth enhancement of the UWB monopole antenna [13]. Isolation improvement in dual-band meander lines has been achieved using split EBG in multiple antenna systems [14].

In particular, several EBG structures are employed to increase the gain of the patch antenna. Mushroom-like EBG structures are integrated with patch antennas to enhance performance at 28 GHz [15]. Improved patch antenna performance for C-band applications is achieved by incorporating mushroom EBGs, which successfully suppress surface waves at 6 GHz [16]. A polarization-dependent metamaterial surface made of EBG structures provides high-gain, low-RCS patch antenna at 3.25 GHz [17]. Reducing the propagation modes of surface waves, thereby enhancing the gain of a patch antenna is achieved by integrating an I-shaped EBG structures with antenna [18]. Gain enhancement for a patch antenna fed by a coaxial probe is achieved through the utilization of L-slotted EBG structures at 5.8 GHz [19]. Performance improvement in terms of gain and radiation pattern of patch antennas is accomplished by using steps like EBG at 5.8 GHz [20]. Fractal-shaped EBG is utilized for gain enhancement of patch antennas and improves the directionality by successfully suppressing the surface waves [21].

From the above-mentioned literature, it is noted that improving the antenna performance requires a larger number of EBG cells, leading to increased design complexity. However, for specific IoT applications, antenna design requires high directivity without increasing the size operated to a particular band. The proposed antenna incorporates a triple-slot design integrated with MEBGs, resulting in a substantial gain improvement. While this design is simple, the triple-slot configuration is combined with a minimal number of MEBGs with improved performance. The results, validated through both simulation and fabrication, demonstrate that this approach provides an optimized balance between performance enhancement and design simplicity, making it suited for IoT-enabled wireless applications.

2. ANTENNA DESIGN WITH EBG STRUCTURES

This section improves the gain by incorporating EBG cells around the patch antenna at the resonant frequency of 5.2 GHz.

2.1. REFERENCE ANTENNA DESIGN

A basic microstrip antenna with a rectangular patch is designed to operate at 5.2 GHz. The radiating patch is mounted on an FR-4 substrate with a height of 1.6 mm. A microstrip feed line is contacted directly to the patch. Basic antenna dimensions are calculated using equations and tabulated in Table 1, and the layout is displayed in Fig. 1. This antenna serves as a reference to compare the performance with the proposed configurations in the subsequent section [22].

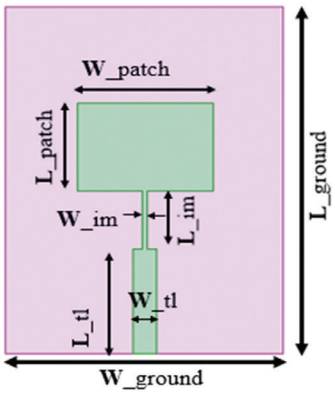


Fig. 1. Design of conventional microstrip patch antenna

Table 1. Specifications of conventional antenna

Parameters	Dimensions (mm*mm)
Substrate	35.9 * 58.9
Ground	35.9 * 58.9
Patch	12.56 * 17.56
$L_{im} * W_{im}$	7.29 * 0.723
$L_{tl} * W_{tl}$	14.59*3.059

Another modified antenna incorporates slots in a conventional patch antenna to improve performance. This modified antenna has three rectangular slots that are each 6 by 2 mm^2 in size. A triple-slot patch antenna layout is depicted in Fig. 2.

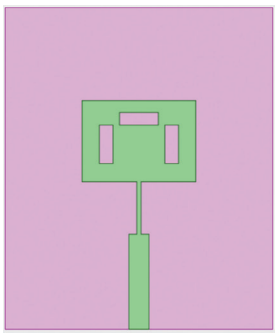


Fig. 2. Triple-slot patch antenna design

2.2. DESIGN AND ANALYSIS OF EBG UNIT CELL

This study includes designing and analyzing both Mushroom EBG (MEBG) and Edge via EBG (EEBG) structures. Initially, EBG unit cell characteristics are thoroughly explained, and later, the design and analysis of both EBG cells are explored in this section.

Basic EBG structures are typically arranged periodically and include four main parts: a metallic patch, a ground plane, a substrate, and a vertical connecting rod extending through the substrate. These periodic EBG structures act as high-impedance surfaces, effectively preventing surface waves from entering inside a specific bandgap. Analysis of a larger array is quite challenging; a simple way to find the characteristics is by applying the periodic boundary condition (PBC) to a unit cell. When the periodicity of the EBG structures is shorter than λ (wavelength), they are referred to as lumped elements (LC) [23]. It functions as a parallel resonance LC filter. An LC filter can exhibit the characteristics of the EBG structure. The presence of vias is crucial in the formation of the well-known two-dimensional mushroom EBG (MEBG). In MEBG, a central via connects the patch to the ground plane. This enables the current to flow, producing inductance (L) and capacitance (C) between the metal planes and the dielectric. The periodic arrangement of MEBG, along with the equivalent circuit diagram, is illustrated in Fig. 3.

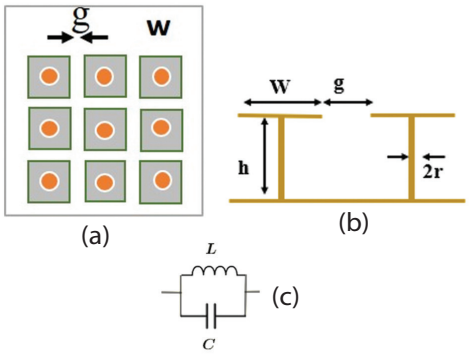


Fig. 3. MEBG unit cells (**a**) top view (**b**) side view (**c**) equivalent circuit

The L and C values determine the frequency band gap, resonant frequency, and surface impedance. These parameters are calculated using the following formulae [24].

$$L = \mu_0 \, h \tag{1}$$

$$C = \frac{w\epsilon_0(1+\epsilon_r)}{h} \cosh^{-1}\frac{(2w+g)}{g} \tag{2}$$

$$f_0 = {}^{1}/_{2\pi(\sqrt{LC})} \tag{3}$$

$$Z = \frac{j\omega L}{1-\omega^2 LC} \tag{4}$$

Based on the parameters of EBG, which include patch width (w), gap between cells (g), and substrate thickness (h), operating frequency and forbidden bandgap are determined. Equation (4) shows that at the resonant frequency, the EBG creates a high-impedance state, which blocks surface waves [25].

In the case of EEBG, the via is moving from the center to the border of the patch. The arrangement of EEBG is shown in Fig. 4. Via routes from the center to the edge, it extends the electrical path to carry out the high impedance transformation [26].

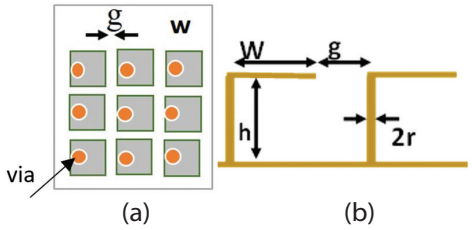


Fig. 4. EEBG unit cells (**a**) front view (**b**) side view

Both the MEBG and the EEBG are designed to achieve an operating frequency of 5.2 GHz, utilizing HFSS software for implementation. In both scenarios, the unit cell of the metallic patch features a square configuration. The metallic patch for both MEBG and EEBG is mounted on an FR-4 substrate with a height of 1.6 mm. This simulation is conducted using periodic boundary conditions (PBCs), which effectively replicate an infinitely periodic structure on all four sides of the cell, as shown in Fig. 5 for MEBG and EEBG. To establish the periodic boundary conditions in HFSS, apply master-slave settings to all four sides of the unit cell. A perfectly matched layer (PML) composed of anisotropic material serves as a boundary at the top of the model volume to prevent reflections and ensure the effective absorption of outgoing electromagnetic waves. The observation plane is positioned at a height nearly ten times greater than the substrate height to reduce the effects of higher-order modes in the results of the EBG unit cell [27]. Using a floquet port in HFSS, plane waves are incident from the top of the EBG cell.

The optimized MEBG and EEBG designs operate at 5.2 GHz. MEBG has a patch width of 8.7 mm, a via radius of 0.2 mm, and a 0.3 mm gap, while EEBG features a 5.4 mm patch width, a 0.3 mm via radius, and a 0.6 mm gap.

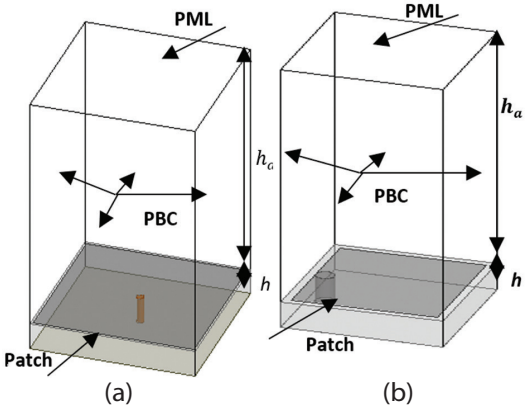


Fig. 5. Simulation setup (**a**) MEBG unit cell (**b**) EEBG unit cell

The forbidden band characteristics and operating frequency for these structures are determined by either the reflection phase or the dispersion diagram. A dispersion diagram interprets the relation between wave frequencies and wave numbers. In the case of the reflection phase, the phase of surface impedance varies concerning frequency from 180° to -180°. AMC features appear in the range of 90° to -90°, where surface currents shift phase to support antenna currents, creating a stopband. Fig. 6 illustrates the reflection phase of both MEBG and EEBG.

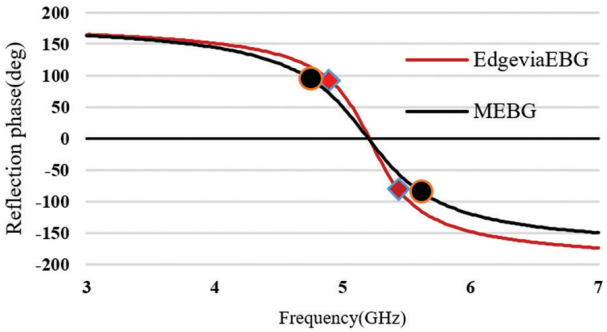


Fig. 6. Reflection phase vs frequency for MEBG and EEBG

From Fig. 6, both the EBGs having a 0° reflection are at 5.2 GHz. The band gap of MEBG is exhibited at 4.78-5.65 GHz. Whereas EEBG exhibits a bandgap around 4.91-5.47 GHz. Compared to MEBG, EEBG exhibits a narrow bandwidth; it shows nearly a 6% decrement in the band gap.

2.3. INTEGRATION OF EBG WITH ANTENNA

In this section, different proposed configurations of antennas are integrated with EBG unit cells while maintaining the same dimensions. For all the configurations the substrate dimensions are 40.8*50.5*1.6 mm^3. These configurations include a conventional antenna with EEBG, a conventional antenna with MEBG, a triple-slot antenna with EEBG, and a triple-slot antenna with MEBG, as shown in Fig. 7.

The first configuration is designed by incorporating EEBG cells around a conventional antenna. The patch dimensions are not changed, and feeding is also the same as that given by the edge feed technique. The gap between EBG cells is kept at 3 mm, and 19 EEBG cells are positioned on the same substrate, as shown in Fig. 7(a).

The second configuration consists of the conventional patch, surrounded by MEBG unit cells, which forms a new design known as the conventional patch antenna with MEBG. The gap between EBG cells is kept at 1 mm, and 12 MEBG cells are positioned on the same substrate. The layout of this new antenna is shown in Fig. 7(b).

The third proposed configuration consists of the conventional patch antenna with EEBG substituted by incorporating a triple slot in the patch. As seen in Fig. 7(c), this transformation produces a unique antenna known as the triple-slot patch antenna with EEBG. Another configuration is that the conventional antenna is replaced by a triple-slot patch antenna and surrounded with MEBG unit cells. This became a new antenna triple-slot patch antenna with MEBG, as shown in Fig. 7(d).

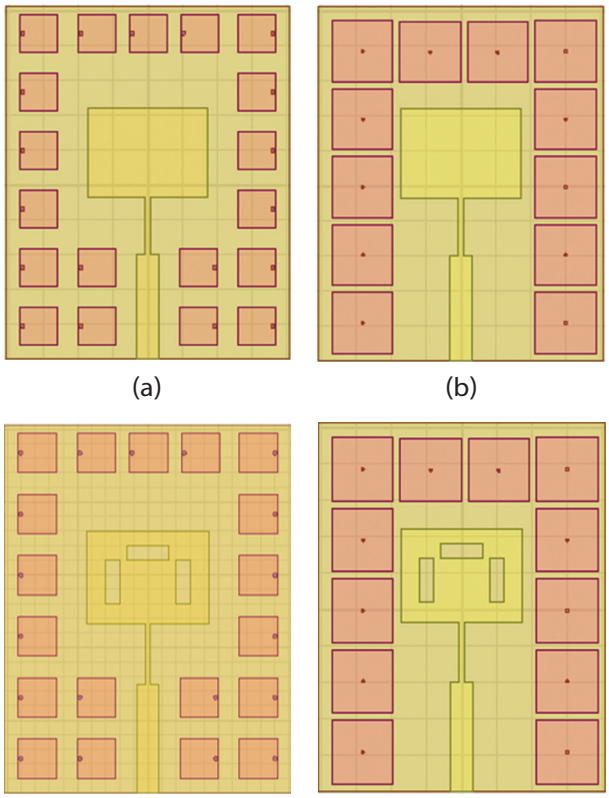


Fig. 7. Design of proposed configurations
(**a**) Layout of conventional patch antenna with EEBG
(**b**) Layout of conventional patch antenna with MEBG
(**c**) Layout of triple slot patch antenna with EEBG
(**d**) Layout of triple slot patch antenna with MEBG

3. RESULTS AND DISCUSSION

This section discusses the simulation results of all design configurations, including the best-performing prototype, which is validated through fabrication results.

3.1. SIMULATION RESULTS

The design of six different configurations of antennas is done using HFSS software. The simulated findings are analyzed in this section. The configurations include a conventional antenna, a triple slot antenna, a conventional antenna with EEBG, a conventional antenna with MEBG, a triple-slot antenna with EEBG, and a triple-slot antenna with MEBG. The simulation findings for each configuration address basic antenna performance parameters, including gain, peak directivity, reflection coefficient (S11) or return loss, and voltage standing wave ratio (VSWR). The quantitative parameters of the different antenna configurations are tabulated in Table 2.

Simulated return loss curve changes with the frequency for each of the six designs are illustrated in Fig. 8, where the conventional antenna and its EEBG and MEBG are radiated at a 5.2 GHz resonant frequency. Whereas a triple-slot antenna and its EEBG and MEBG slightly move the operating frequency to 5 GHz because slots create an additional current path, which increases the electrical length of a patch; hence, the resonant frequency decreases. The simulated S11 parameter for each configuration is as follows: -17.8 dB (conventional), -25.7 dB (triple slot), -29.3 dB (conventional with EEBG), -20.8 dB (conventional with MEBG), -31.3 dB (triple-slot with EEBG), and -37.2 dB (triple-slot with MEBG).

An excellent impedance match between the patch and feedline is achieved when the triple-slot antenna with MEBG exhibits a steep decrease in the reflection coefficient when compared to the other configurations. There is an 87% incremental return loss when comparing a conventional antenna to a triple-slot antenna with MEBG. According to the VSWR values obtained from Fig. 9, MEBG performs better than EEBG in attaining impedance matching, which is a crucial component of antenna design.

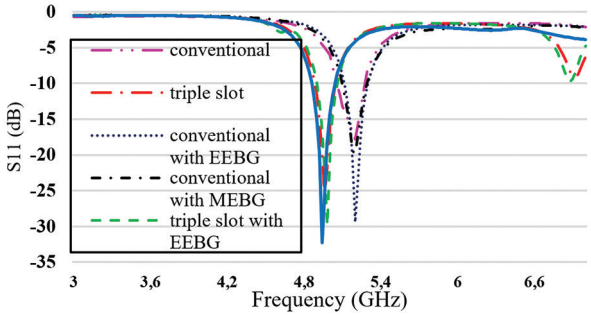


Fig. 8. Return loss vs frequency for all antenna configurations

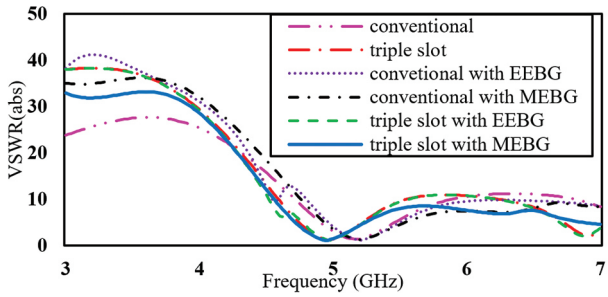


Fig. 9. VSWR vs frequency for all antenna configurations

The radiation pattern, which describes the transmission/reception power from an antenna that varies in different directions, gives important information on how the antenna is directing the power, polarization, and gain properties. The E plane (φ=0°) and H plane (φ=90°) of the radiation pattern are simulated for conventional and triple-slot antennae as shown in Fig. 10(a), conventional with EEBG and MEBG, as shown in Fig. 10(b). E & H plane of triple-slot antenna and it's with EEBG and MEBG, as depicted in Fig. 10(c) and 10(d).

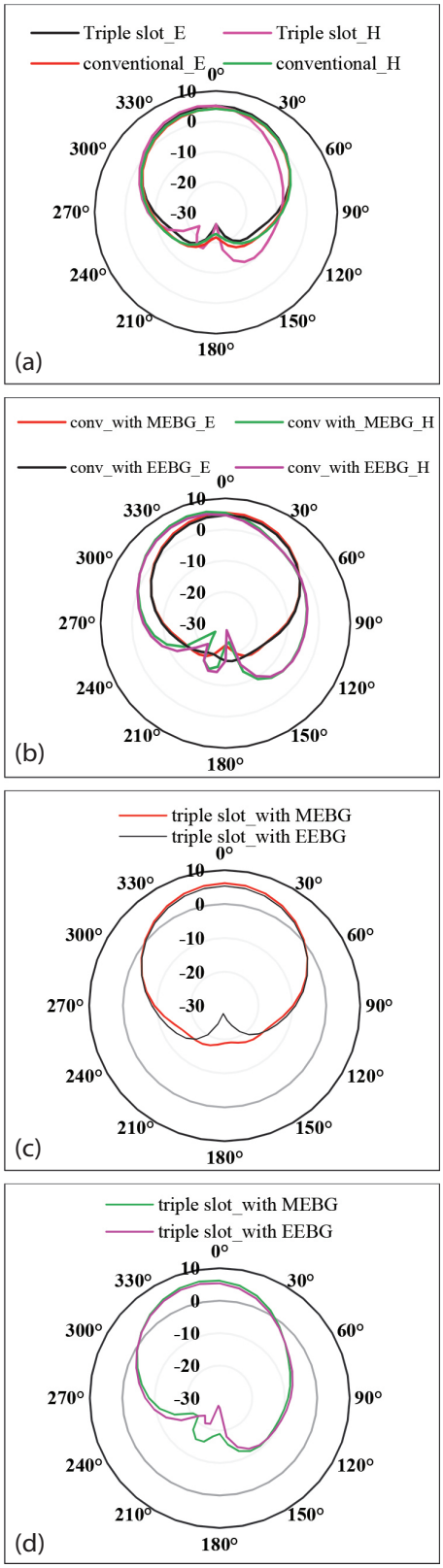
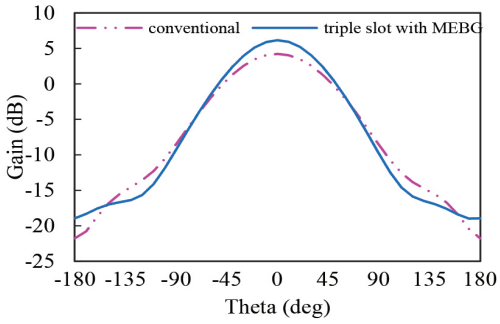
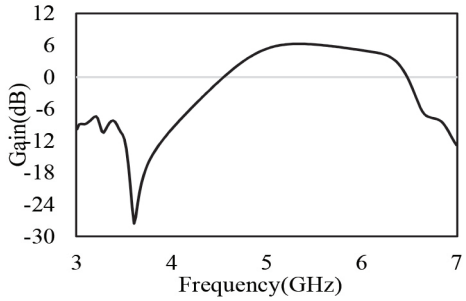


Fig. 10. Radiation plots (**a**)E, H plane for conventional and triple-slot antenna (**b**) E, H radiation planes for a conventional antenna with EEBG & MEBG (**c**) E plane for triple-slot antenna with EEBG & MEBG (**d**) H plane for triple-slot antenna with EEBG & MEBG

Both the planes of the conventional and triple-slot antennas are similar. The addition of slots in the conventional antenna leads to improved gain and reduced back lobe radiation. When comparing EEBG with MEBG, MEBG further reduces back lobe radiation and enhances gain. Fig. 10 illustrates the achievement of a unidirectional pattern for the triple- slot antenna with MEBG, showcasing improved radiation performance compared to EEBG. There is a substantial improvement in gain quantity for a triple-slot antenna with an MEBG compared with a conventional antenna as depicted in Fig. 11(a). When the MEBG cells are included in a triple slot antenna, broadside gain is raised from 4.22 dB to 6.15 dB. The proposed antenna's gain varies with frequency, as illustrated in Fig. 11(b). The three-dimensional radiation pattern at 5.2 GHz is shown in Fig. 12, demonstrating that the realized gain reaches 6.43 dBi.



(a)



(b)

Fig. 11. Gain plot (**a**) gain varies with spatial coordinates for conventional and proposed antenna (**b**) gain varies with frequency for the proposed antenna

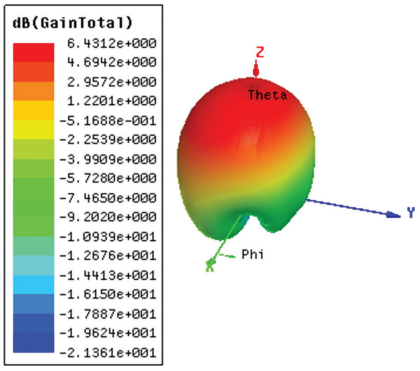


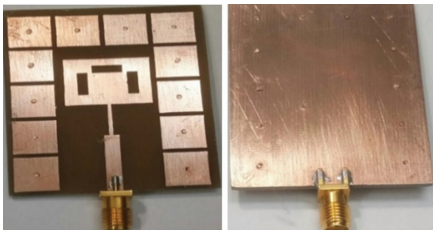
Fig. 12. The 3-D gain plot of the proposed triple-slot antenna with MEBG

Table 2. Antenna parameters of all configurations

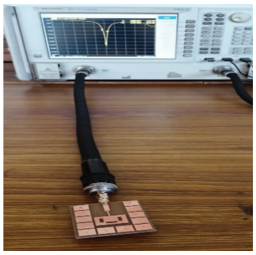
Antenna configurations	S11 (dB)	VSWR (abs)	Gain (dB)	Frequency (GHz)	Peak Directivity (dB)
Conventional	-17.8	1.29	4.22	5.2	5.3
Triple-slot	-25.7	1.12	5.11	4.96	6.64
Conventional with EEBG	-29.3	1.07	4.63	5.2	6.26
Conventional with MEBG	-20.8	1.21	5.38	5.2	6.73
Triple-slot with EEBG	-31.3	1.07	5.31	4.97	6.96
Triple-slot with MEBG	-37.2	1.03	6.15	4.94	7.51

3.2. FABRICATION RESULTS

The best given simulated antenna is a triple-slot antenna with MEBG fabricated with the specifications (ε_r = 4.4, h = 1.6 mm, tan δ = 0.02). It features a 50-ohm SMA connector attached at the end of the feed line. Measurements are conducted using an Agilent N5247A network analyzer, which supports a maximum frequency of up to 18 GHz. Fig.13 illustrates the front and back views of the fabricated antenna along with the measurement setup for return loss analysis. The main challenge in fabricating MEBG structures is achieving precise etching, especially in maintaining gap width and via placements, which were accomplished using UV photoresist etching. Dipping of copper wires into the vias increased complexity, resulting in fabrication tolerances and measurement errors. Impedance mismatches from SMA connectors were addressed through calibrated soldering and VNA testing. These steps ensured that the fabricated prototype closely matched simulations, validating the design. The measured and simulated return loss curves for the triple-slot patch antenna with MEBG are presented in Fig.14.



(a) (b)



(c)

Fig.13. Proposed constructed antenna (**a**) top view (**b**) rear view (**c**) measurement setup of fabricated antenna with VNA

It appears that the operational frequency of the manufactured antenna is merely shifted to 5 GHz from 5.2 GHz. The fabricated antenna produces a return loss of 44.82 dB at an operating frequency of 5 GHz. From conventional to fabricated antenna, the return loss is increased from 17.8 dB to 44.2 dB.

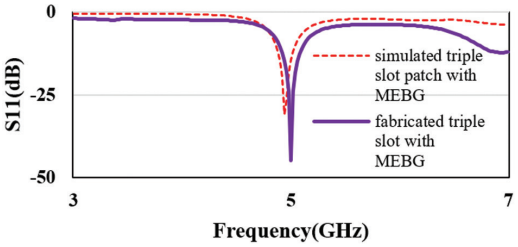
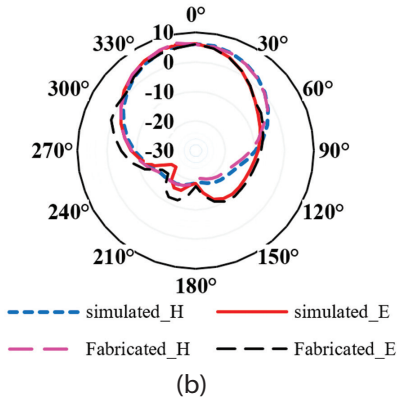


Fig.14. Plot of simulated vs fabricated results of S11 w.r.t frequency

Fig. 15 displays the E-plane and H-plane radiation patterns for the fabricated antenna measured in the anechoic chamber. Both the simulated and developed antennas produce and match the far-field radiation pattern of the E and H planes acceptably, and the gain of the proposed antenna is 6.15 dB. A minor lobe is slightly increased compared with simulation results due to measurement errors. Introducing a conventional slot surrounded by MEBG raises the gain quantity by 1.93 dB compared to the conventional antenna, as mentioned in the measured gain plot at 5.2 GHz, and efficiency is about 70%, as shown in Fig. 19(c). The suggested antenna is compact and has better performance. Table 3 compares the gain augmentation attained by the suggested antenna with other antennas available in the existing research, accounting for the quantity of EBG cells used.

(a)



(b)

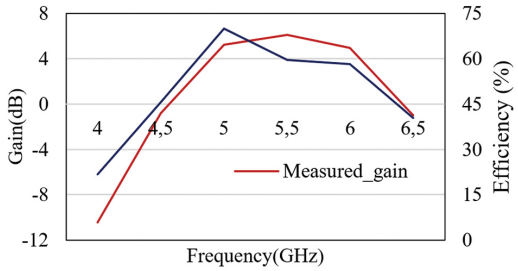


Fig. 15. Proposed fabricated antenna (**a**) setup of radiated power measuring with an anechoic chamber (**b**) radiation pattern of E- and H-plane (**c**) measured gain and radiation efficiency plot varies with frequency

Table 3. Comparisons between the suggested antenna and prior research

References	Overall size (mm*mm* mm)	Frequency (GHz)	S11 (dB)	Enhanced Gain (dB)	No. of EBG cells	Apllications
[15]	7.5*6.1*0.13	28	-47.7	1.6	20	5G
[16]	40*40*2.2	6	-25	1.0	12	C band
[17]	101.4*313.8* 2.2	3.25	-46.1	2.5	200	RCS reduction
[18]	70*70*1.6	5.2	--	2	28	WLAN
[19]	41.3*41.3*1.5	5.8	-12.5	1.9	40	ISM
[20]	58*58*3.81	5.8	-42	2.3	120	RFID
[28]	50*50*1.27	5.2	-28	1.1	72	WLAN
[29]	23*18*0.35	24	-23	2	24	IoT
[30]	68*73*3	5.2	-19.25	2.6	25	IoT
Proposed Work	40.8*50.5*1.6	5.2	-37.2	1.93	12	IoT

Table 3 shows the comparison of previous antenna designs incorporating EBG structures for IoT, 5G, and ISM applications. The previous studies indicate that using a larger number of EBG structures can enhance gain but often leads to increased design complexity. The proposed design uses fewer MEBG structures to achieve optimal gain. This suggested work balances the optimum performance and simple design that can be suitable for practical applications.

4. CONCLUSION

This paper discusses the limitations of traditional substrates, which are overcome by integrating the EBG structures to enhance the radiation performance of antennas. The proposed antenna is built on an FR-4 substrate with a thickness of 1.6 mm to mitigate the effect of surface wave propagation. Both MEBG and EEBG structures are examined to prevent the surface wave propagation around operating frequency 5.2 GHz. According to the examination, MEBG outperforms EEBG in suppressing surface wave propagation over a wider frequency range of 4.78–5.65 GHz with improved performance, whereas EEBG has a narrower bandgap of 4.97–5.41 GHz. The proposed antenna, which consists

of triple slots with only 12 MEBG unit cells, significantly improves the antenna parameters compared with conventional antenna while maintaining the compact size. The gain is improved substantially to 1.93 dB, and the return loss is greatly increased to 19.4 dB over the conventional antenna. The simulation results have been demonstrated and are consistent with fabrication results. The simple proposed design shows a high directivity of 7.51 dB, which can be suited for IoT-enabled applications such as home automation and industrial wireless monitoring.

Conflict of interest:

On behalf of all authors, the corresponding author states that there is no conflict of interest.

5. REFERENCES

[1] L. Marco, P. Francesco, S. Domenico, "Internet of Things: A General Overview between Architectures Protocols and Applications", Information, Vol. 12, No. 2, 2021, p. 87.

[2] D. Arnaoutoglou, T. Empliouk, T. Kaifas, M. Chryssomallis, G. Kyriacou, "A review of multifunctional antenna designs for the Internet of Things", Electronics, Vol. 13, No. 6, 2024, p. 3200.

[3] Z. Wang, B. Yuan, X. Zhang, L. Guo, "An Axial-Ratio Beam-Width Enhancement of Patch-Slot Antenna Based on EBG", Microwave and Optical Technology Letters, Vol. 59, 2017, pp. 493-497.

[4] M. Zambak, S. Al-Bawri, M. Jusoh, A. Rambe, V. K. Hamza, A. Almuhlafi, M. Himdi, "A compact 2.4 GHz L-shaped microstrip patch antenna for ISM-band Internet of Things (IoT) applications", Electronics, Vol. 12, No. 9, 2023, p. 2149.

[5] L. Anchidin, A. Lavric, M. Marian, A. Petrariu, V. Popa, "The design and development of a microstrip antenna for Internet of Things applications", Sensors, Vol. 23, No. 3, 2023, p. 1062.

[6] Y. Madany, H. Elkamchouchi, H. A. Elmonieum, "High-Gain Pixel Patch Antenna Array for Miniature Wireless Communications and IoT Applications", Progress In Electromagnetics Research C, Vol. 131, 2023, p. 209-225.

[7] A. Ayomikun, M. Mokayef, "Miniature microstrip antenna for IoT application", Materials Today: Proceedings, Vol. 29, 2020.

[8] S. M. Refaat, A. Abdalaziz, E. K. I. Hamad, "Tri-Band Slot-Loaded Microstrip Antenna for Internet of Things Application", Advanced Electromagnetics, Vol. 10, No. 1, pp. 21-28.

[9] Y. I. Ashyap et al. "An Overview of Electromagnetic Band-Gap Integrated Wearable Antennas", IEEE Access, Vol. 8, 2020, pp. 7641-7658.

[10] D. Sievenpiper, L. Zhang, R. F. J. Broas, N. G. Alexopoulos, E. Yablonovitch, "High-impedance electromagnetic surfaces with a forbidden frequency band", IEEE Transactions on Microwave Theory and Techniques, Vol. 47, No. 11, 1999, pp. 2059-2074.

[11] Y. Rahmat-Samii, F. Yang, "Microstrip Antennas Integrated with Electromagnetic Band-Gap (EBG) Structures: A Low Mutual Coupling Design for Array Applications", IEEE Transactions on Antennas and Propagation, Vol. 51, 2003 pp. 2936-2946.

[12] Y. Li, K. Zhang, L. Yang, L. Du, "Gain enhancement and wideband RCS reduction of a microstrip antenna using triple-band planar electromagnetic band-gap structure", Progress In Electromagnetics Research Letters, Vol. 65, 2017, pp. 103-108.

[13] P. P. Bhavarthe, S. S. Rathod, K. T. V. Reddy, "A Compact Dual Band Gap Electromagnetic Band Gap Structure", IEEE Transactions on Antennas and Propagation, Vol. 67, No. 1, 2019, pp. 596-600.

[14] X. Tan, W. Wang, Y. Liu, A. Kishk, "Enhancing Isolation in Dual-Band Meander-Line Multiple Antenna by Employing Split EBG Structure", IEEE Transactions on Antennas and Propagation, Vol. 67, 2019, pp. 2769-2774.

[15] A. Alsudani, H. Marhoon, "Design and Enhancement of Microstrip Patch Antenna Utilizing Mushroom Like-EBG for 5G Communications", Journal of Communications, Vol. 18, 2023, pp. 156-163.

[16] M. Abdulhameed, M. M. Isa, M. Saari, Z. Zakaria, M. Mohsen, M. Attiah, "Mushroom-Like EBG to Improve Patch Antenna Performance for C-Band Satellite Application", International Journal of Electrical and Computer Engineering, Vol. 8, 2018, pp. 3875-3881.

[17] Z. J. Han, W. Song, X. Q. Sheng, "Gain Enhancement and RCS Reduction for Patch Antenna by Using Polarization-Dependent EBG Surface", IEEE Antennas and Wireless Propagation Letters, Vol. 16, 2017, pp. 1631-1634.

[18] P. Ketkuntod, T. Hongnara, W. Thaiwirot, P. Akkaraekthalin, "Gain enhancement of microstrip patch antenna using I-shaped Mushroom-like EBG structure for WLAN application", Proceedings of the International Symposium on Antennas and Propagation, Phuket, Thailand, 30 October - 2 November 2017, pp. 1-2.

[19] S. Venkata, R. Kumari, "Gain and isolation enhancement of patch antenna using L-slotted mushroom electromagnetic bandgap", International Journal of RF and Microwave Computer-Aided Engineering, Vol. 30, 2020.

[20] N. Melouki, A. Hocini, T. Denidni, "Performance enhancement of a compact patch antenna using an optimized EBG structure", Chinese Journal of Physics, Vol. 69, 2020.

[21] N. Rao, D. Kumar, "Gain enhancement of microstrip patch antenna using Sierpinski fractal-shaped EBG," International Journal of Microwave and Wireless Technologies, Vol. 7, 2015, pp. 1-5.

[22] G. Chaduvula, B. Kumari, "Implementation of EBG Structure to Reduce Surface Wave Excitations for IoT Range Applications", Proceedings of the 2nd International Conference on Artificial Intelligence, Computational Electronics and Communication System, Manipal, India, 16-17 February 2023, p. 012038.

[23] P. Bhavarthe, S. S. Rathod, K. Reddy, "A Compact Two Via Slot-Type Electromagnetic Bandgap Structure", IEEE Microwave and Wireless Components Letters, Vol. 27, No. 5, 2017, pp. 446-448.

[24] E. Wang, Q. Liu, "GPS patch antenna loaded with fractal EBG structure using an organic magnetic substrate", Progress In Electromagnetics Research Letters, Vol. 58, 2016, pp. 23-28.

[25] K. Peter, Z. Raida, Z. Lukes, "Design and optimization of periodic structures for simultaneous EBG and AMC operation", Proceedings of the 15th Conference on Microwave Techniques COMITE, Brno, Czech Republic, 19-21 April 2010, pp. 195-198.

[26] E. R. Iglesias, L. I. Sanchez, J. L. V. Roy, E. G, "Size Reduction of Mushroom-Type EBG Surfaces by Using Edge-Located Vias", IEEE Microwave and Wireless Components Letters, Vol. 17, 2007, pp. 670-672.

[27] R. Remski, "Analysis of Photonic Bandgap Surfaces Using Ansoft HFSS", Microwave Journal, Vol. 43, 2000.

[28] N. Jaglan, S. Dev Gupta, "Surface waves minimisation in microstrip patch antenna using EBG substrate", Proceedings of the International Conference on Signal Processing and Communication, Noida, India, 16-18 March 2015, pp. 116-121.

[29] W. May, I. Sfar, L. Osman, J. M. Ribero, "A Textile EBG-Based Antenna for Future 5G-IoT Millimeter-Wave Applications", Electronics, Vol. 10, No. 2, 2021, p. 154.

[30] A. Ahmad, F. Faisal, S. Ullah, D. Choi, "Design and SAR Analysis of a Dual Band Wearable Antenna for WLAN Applications", Applied Sciences, Vol. 12, No. 18, 2022, p. 9218.

Lightweight Block Cipher for Security in Resource-Constrained Network

Original Scientific Paper

Aruna Gupta*
CSE, Sathyabama Institute of Science and Technology,
Chennai, India
agupta7.2018@gmail.com

T. Sasikala
CSE, Sathyabama Institute of Science and Technology,
Chennai, India
dean.computing@sathyabama.ac.in

*Corresponding author

Abstract *– As the proliferation of resource-constrained devices continues in various application domains, the need for energy-efficient cryptographic algorithms becomes paramount for ensuring their security. Lightweight block ciphers play a crucial role in securing communication and data integrity in resource-poor environments. This paper presents the design, simulation, and evaluation of a novel symmetric Energy Efficient Lightweight Block Cipher (EE-LBC), tailored for such environments, which employs a balanced combination of substitution-permutation network (SPN) structure with larger diffusion and substitution box activation properties to achieve high security with minimal energy consumption and implementation cost. Through rigorous cryptanalysis and performance evaluations, EE-LBC demonstrates superior throughput and efficiency compared to prevailing lightweight block ciphers, making it an ideal choice for securing resource-constrained network.*

Keywords*: block cipher, IoT, lightweight cryptography, MANET, resource-constrained network*

Received: March 15, 2025; Received in revised form: April 14, 2025; Accepted: April 19, 2025

1. INTRODUCTION

With the rapid expansion of the Internet of Things (IoT) and the increasing integration of connected devices into various domains, ensuring data security in resource-constrained environments has become a paramount concern. Wireless networks having limited resources for storage, processing, communication and power are all categorized as Resource-Constrained Network (RCN) such as Mobile Ad-hoc NETwork (MANET), Vehicular Ad-hoc NETwork (VANET), Flying Ad-hoc NETwork (FANET) and Internet of Things (IoT). Nodes in such network must always retain minimum energy level so as to maintain the network connectivity [1]. On the other hand, security is the biggest challenge in these networks due to wireless communication media and lack of in-built security. Attacks done by the malicious node in the network can destructively affect the integrity, confidentiality, and secrecy of nodes in the network [2].

To guard the resource-constrained networks from active attacks, various proposals are available which are either proactive or reactive in nature. After the intruder disturbs the network, sense the attack and then try to recover from it. This is called a reactive method of protection. On the other hand, in proactive method, necessary care is taken to confirm that the intruder will not be able to damage the system or authentic user. Intrusion Detection Systems (IDS) are built on the reactive approach of defense whereas cryptography is preferred in the proactive method. One of the disadvantages of IDS is that it cannot detect the source of the attack and it just locks the whole network. This parallelizes it completely [3]. Secondly, IDS continuously monitor node behavior and network traffic; so, it keeps engaging the resources [4].

Proactive technique of cryptography can be split into two categories: cryptographic algorithms designed for resource-rich networks are not suitable for RCN due to their high resource requirements and other one is lightweight cryptographic primitives that uses limited resources without compromising the security level achieved [5, 6]. In this paper, we propose a novel en-

ergy-efficient lightweight block cipher, EE-LBC, specifically designed to address the security and energy constraints for application in the resource-poor networks.

The block cipher proposed in this paper is energy efficient as well secure and possesses following properties:

- Symmetric key cryptography-based block cipher EE-LBC uses Substitution-Permutation Network (SPN) structure as its base which comprises of substitution and permutation layer.
- High diffusion and S-box activation properties of EE-LBC makes it strongly resistance against differential and linear cryptanalysis.
- After taking the 80-bit key through key generation algorithm, the cipher assured to be resistant against key schedule attacks due to the shuffling of bits in the key by XORing round counter with the middle portion of the round key.
- It makes use of a single 4-bit non-linear S-box that operates on 16-bit data at a time. This leads to the simplicity in design while minimizing the computational complexity and implementation cost but improving throughput.
- Reduced number of rounds and smaller key size of this cipher results in desired performance in terms of energy efficiency.
- It is suitable for the constrained devices and networks for variety of applications.

Further portion of the paper is systematized by elaborating on distinguishable facts about various lightweight cryptographic protocols in Section II. Overview with the architecture and details of proposed block cipher are presented in Section III. Cryptanalysis of the proposed protocol is covered in Section IV whereas results of the comparison of EE-LBC with other block cipher and the discussion through analysis of the same is conversed in Section V. Paper is summarized in the Section VI followed by the references used.

2. RELATED WORK

For the past decade, various researchers have been engaged in discussing about the resource-constrained wireless networks and the security facets in its context. Precisely, the network layer attacks on routing protocols are a subtle issue in this and any other network [7, 8]. Traditional cryptographic algorithms are not suitable in this case due to the high resource requirements. Lightweight Cryptography is the method of encryption that leads to small-sized and computationally lower complexity as well as low power consumption. Thus, it extends the battery life of resource-limited devices while maintaining strong security levels. [9]. Lightweight cryptography consists of cryptographic protocols customized for implementation in constrained setup. Its standardization process is still in progress [10].

A contest held by NIST (Mar 2019-2021), to identify secure and efficient cryptographic algorithms suitable for constrained environments such as IoT devices and RFID systems, was graced by 57 innovative submissions out of which 56 could pass through the first round and 32 could reach to the final round of the contest. On the critical evaluation of those protocols, 10 were declared as finalists [11].

Prior research in lightweight cryptography has led to the development of various algorithms optimized for resource-constrained devices. Stream ciphers, block ciphers, and hash functions have been extensively studied and tailored for lightweight applications. One of the symmetric key cryptographic techniques for providing the confidentiality and integrity to the sensitive information is block cipher. Block cipher uses combination of confusion and diffusion properties of cryptography that makes the reverse of encryption process to extract the original text harder in block cipher [12]. Existing lightweight block ciphers, such as PRESENT [13], SIMON, SPECK [14] and LEA [15], have demonstrated promising results in terms of area efficiency and computational performance [16, 17]. However, these ciphers may still consume significant energy when implemented on power-constrained devices. In this section, prominent lightweight cryptographic block ciphers in this framework are studied and analyzed for security.

Two most popular examples of protocols standardized by NIST are AES [18] and DES [19]. Former is SPN based algorithm whereas later follows Feistel Network (FN) structure. AES has the implementation requirement of around 2400 GEs where DES needs around 2310 GEs. Such larger area pre-requisite makes both protocols unsuitable for RCN.

TWINE presented in [20] takes 64-bit input and has two variants with 80-bit and 128-bit key. It can operate with lesser memory also but has requirement of around 2000 GEs which is still higher for constrained environment. Ultra-light block cipher RECTANGLE in [21] works with least rounds i.e., 25 by applying little alterations to the SPN structure which makes it useful for large variety of applications in constrained environment. Ill-advisedly, it is suspectable to related-key and side-channel attacks.

Symmetric key block cipher E3LCM proposed in [22] uses Multi-sequence Linear Feedback Shift Register (MLFSR) in substitution layer for reducing design area. Though it has smallest key of 64-bit, since the output of MLFSR is deterministic, it is not secure.

PRINT [23] is another cipher that makes use of an 80-bit key to perform 48 iterations with least GE requirement. It performs 3-bit operations which is infeasible due to odd number of bits. On the other hand, PRINCE in [24] is one of most efficient lightweight algorithms that uses 128-bit key for 12 rounds. It has low energy consumption and smaller hardware requirement. But it is not used widely due to its fixed larger key size.

Major focus of TEA in [25] is to achieve higher speed while minimizing the memory requirements. It is designed to work for commodity hardware with the application of 128-bit key used across 32 rounds. But its key scheduling part is too simple to get forged by brute force attack. SKINNY [26] has three key variants 64/128/192-bit key to perform 32 to 40 rounds on the data blocks of varying sizes. Limitation of this cipher is that it is prone to birthday attack.

GIFT [27] was finalist in NIST contest since it offers lighter S-box and occupies lesser physical space. It uses 128-bit key for 64-bit data block with 40 rounds. But it's not safe against differential cryptanalysis. SLIM [28] is used for Internet of Health Things and is based on Feistel structure. It works on 32-bit block with 80-bit key in 32 rounds. Its security is suitable only for RFID-based systems. It is immune against linear and differential cryptanalysis.

The implementation of the KATAN32 algorithm [29] on the ESP32 microcontroller demonstrates low computational power requirements, making it suitable for resource-constrained environments. While KATAN32 is lightweight, it may not offer the same level of security robustness as more modern algorithms. The implementation is tailored to the ESP32 microcontroller, which may limit its generalizability to other hardware platforms without additional modifications.

Lastly, ASCON [30], selected as the NIST lightweight cryptography standard, requires significantly fewer resources compared to AES-128, making it well-suited for edge devices. It has been found resistant against side-channel attacks, a critical consideration for IoT devices. Its limitation is that performance metrics such as resource utilization, operating frequency, and power consumption can vary across different FPGA platforms, necessitating platform-specific optimizations.

As seen from the above study, SLIM, KATAN32 and ASCON have platform-specific requirements. AES, DES, and Twine ciphers exhibit significant area overhead, making them resource-intensive for hardware implementations. E3LCM, PRINT, and TEA ciphers exhibit inherent structural vulnerabilities, impacting their cryptographic robustness. A block cipher is expected to provide balance among implementation cost, performance in terms of encryption time and security.

3. EE-LBC ALGORITHM

The algorithm EE-LBC is predicated on Substitution-Permutation Network (SPN) structure that shudders the data through a combination of substitution layer and permutation layer and puts it together for the further round. It spins the data through the 31 rounds of permutations with the assistance of separate roundkey for every round. The encryption method of proposed cryptographic protocol accepts two inputs, one data block of size 64-bit and a key stream that is 80-bit long and generates ciphertext block as an output. During each of the 31 rounds, XOR operation is performed between the data block and the corresponding roundkey. In each round, the non-linear Substitution Layer that consists of 4-bit S-box is applied 16 times (64-bit data block/4-bit S-box=16) parallelly. Decryption process of this cipher is the reverse of steps applied during encryption. The diagram showing functioning of proposed algorithm EE-LBC is shown in Fig. 1.

The resistance of symmetric-key based block ciphers broadly depends on the cryptographic potency of the Substitution Layer. To scale back the design area, the Substitution Box structure is slightly altered and constructed using 4-bit single S-box rather than eight S-boxes. The rifeness of this structure is that it occupies less space with finest speed and energy consumption. The research work in this paper focuses solely on a software-based implementation and analysis, which aims to demonstrate the fundamental design, correctness, and performance of the proposed cipher on constrained devices. Implementation of EE-LBC is focused on embedding a lightweight cryptographic protocol in routing protocol of MANET-enabled IoT. Detailed stages in the form of bit operations performed during encryption and decryption processes of the proposed block cipher are delineated below.

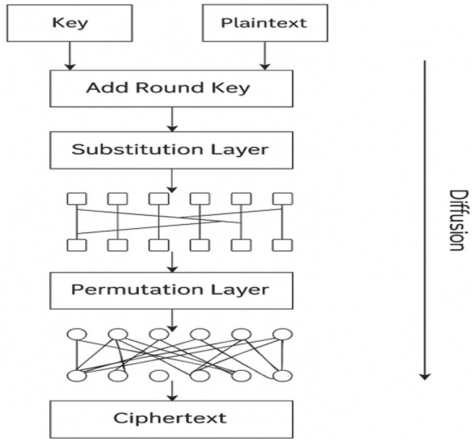


Fig. 1. Block Diagram for EE-LBC

A. Encryption

Key Scheduler

EE-LBC incorporates a lightweight key scheduling algorithm that minimizes computational overhead and energy consumption during key expansion. The key scheduling algorithm efficiently generates round keys from the master key, ensuring robust key mixing without sacrificing performance.

One key K_i is generated for each round i where 1<= i <= 31. At first, the original 80-bit key (K_{79}, K_{78}……K_0) is split into two keys, namely Key16 which is formed with lower 16 bits of the key and Key64 which is made up of upper 64 bits of the key.

$$\text{Key16} = (K_{15}, K_{14}\ldots\ldots K_0)$$
$$\text{Key64} = (K_{79}, K_{78}\ldots\ldots K_{16})$$

For round 1 the key K_1 is simply the Key64. Remaining round keys are generated using following steps:

- Shift the key 61 bits to the left: $K_i = K_i << 61$
- Key64 goes through the substitution layer
- The bits at the mid of the two keys are XORed with the counter of the current round

$$(K_{19}, K_{18}, K_{17}, K_{16}, K_{15}) = (K_{19}, K_{18}, K_{17}, K_{16}, K_{15}) \oplus i$$

For each round, the upper 64-bit part of the generated key is used as the round key.

After generating subkeys for all rounds, the encryption algorithm iterates through following three phases 31 times:

Add Round Key

The round function combines the operations of the substitution and permutation layers with the round key to produce the output ciphertext block. It consists of multiple iterations of the substitution and permutation operations followed by key mixing.

The 64-bit data block is XORed with the 64-bit round key Ki generated by the key scheduler as follows:

$$(B_{63}, B_{62}\ldots\ldots B_0) = (B_{63}, B_{62}\ldots\ldots B_0) \oplus (K_{63}, K_{62}\ldots\ldots K_0)$$

Substitution Layer

The simple design of S-box in EE-LBC focuses on achieving a balance between cryptographic strength and reduced computational complexity. This also maintains sufficient non-linearity and resistance against differential and linear cryptanalysis. In EE-LBC, single 4-bit S-box is used. The above data block after addition of round key is passed through the substitution layer. The S-box is defined with the hexadecimal values as follows:

S[] = {0xC, 0x5, 0x6, 0xB, 0x7, 0x0, 0xA, 0xD, 0x1, 0xE, 0xF, 0x8, 0x4, 0x9, 0x3, 0x2}

Each word (4-bit) in the data block is replaced by the corresponding value in the S-box at the same position.

Permutation Layer

The permutation layer shuffles the output of the substitution layer to achieve diffusion. It ensures that each bit of the input affects multiple bits of the output, thereby spreading the influence of each input bit across the entire block. This layer is implemented using efficient permutation techniques to minimize energy consumption.

For performing permutation at each round, following table (Table 1) is referred:

Table 1. Permutation Table

Bit Position 1															
0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63
NewBit Position														Bit Position 64	

Reading the table row-wise, each cell number represents, input bit position as marked on the table. So, ith bit is moved to the position indicated by the value inside the cell.

During each round of the encryption process, the permutation layer performs a series of bit-level operations designed to enhance diffusion by disrupting bit positions systematically. The steps are as follows:

1. Bit Position Identification: Determine the position of the bit to be processed by calculating its distance from the least significant bit (LSB). This helps in isolating its exact contribution to the data block.

2. Bit Alignment: Right shift the data block to align the target bit with the LSB position. This normalization simplifies subsequent operations.

3. Bit Isolation: Apply a bitwise AND operation with 1 to mask and extract the target bit. This ensures only the bit of interest is manipulated.

4. New Position Calculation: Compute the target location for the bit based on a permutation rule or key-driven logic. This step ensures that the permutation is non-linear and key-dependent.

5. Bit Placement: Shift the isolated bit to its new position and use a bitwise OR operation to merge it into the permuted data block. This ensures that each bit contributes uniquely to the final encrypted output.

B. Decryption

Key scheduler module, adding round key and substitution layer for the decryption process is same as that of encryption process. Last phase is replaced by inverse permutation layer. During each round of the decryption process, the inverse permutation phase systematically reconstructs the original bit positions to reverse the scrambling introduced during encryption. The following steps are executed:

1. Target Bit Identification: Determine the original position of the bit that should occupy the i-th position in the final output. This step ensures accurate reversal of the permutation logic.

2. Result Alignment: Left shift the intermediate result by 1 bit to create space for inserting the next bit in its correct sequence.

3. Bit Alignment in Cipher Block: Shift the data

block such that the required bit is brought to the least significant bit (LSB) position, simplifying its extraction.

4. Bit Extraction: Apply a bitwise AND operation with 1 to mask and isolate the required bit from the shifted data block.

5. Bit Integration: Insert the extracted bit into the result using a bitwise OR operation. This reconstructs the inverse permutation step-by-step, ultimately restoring the original data structure.

4. CRYPTANALYSIS OF EE-LBC

The attacks that alter the data or affects the memory typically catches hold of parameters of block cipher for performing the attack and they do not exploit the inner structure of the block cipher. In further part of this section, crucial security measure is discussed in context to the proposed technique. Cryptanalysis is employed to assess how resistant the algorithm is to various types of attacks. Two most commanding tools for cryptanalyst for finding linear or differential path through various rounds of the block cipher are linear and differential cryptanalysis [31]. But if the search space of the cipher is sufficiently large then finding the optimal path becomes challenging task for the cryptanalyst.

Differential Cryptanalysis:

During this technique, attacker selects the plaintext as input to the algorithm and then access the corresponding ciphertext. To perform this cryptanalysis, a pair of plaintexts is taken which is related with each other by a constant difference. It exploits how difference in plaintext pairs propagate through the cipher's structure [32].

The difference Δc between the generated pair of ciphertexts for the given pair of plaintexts can be calculated as

$$\Delta c = S\,(P \oplus \Delta p) \oplus S\,(P).$$

where S is the substitution function and P is the permutation function, Δp indicates the difference between the given pair of plaintexts.

Consider following pair of inputs with just one-bit difference:

P1: 4172756e61204b47

P2: 4172756e61204b48

It generates following pair of ciphertexts with countable difference:

C1: fff2f964d2012604

C2: 5a3c5405043f0d45

The 4-bit S-box used in EE-LBC has a maximum differential probability (MDP) of 2^{-2}. Since the cipher has 16 S-boxes per round and 31 rounds, the best differential characteristic over many rounds involves selecting active S-boxes carefully.

- Over 5 rounds: at least 10 active S-boxes.

- Over 25 rounds: ≥ 62 active S-boxes.

If 62 S-boxes are active and each contributes max 2^{-2}, total probability ≤ $(2^{-2})^{62} = 2^{-124}$. So the cipher resists differential attacks beyond about 25 rounds with time complexity of around 2^{63} encryptions.

Linear Cryptanalysis:

Linear estimate showing relation between plaintext and ciphertext is computed in linear cryptanalysis to analyze the block cipher [33]. We can calculate the linear trails through various rounds of the algorithm. Linear trail is calculated by producing the linear estimates of the S-box.

As per theorem in [34] for linear estimates, like in the differential case, let's say 62 active S-boxes:

Total linear trail bias: $(2^{-1})^{62} = 2^{-62}$.

Required plaintexts ≈ 1 / $(\text{bias})^2 = 2^{124}$

This is again infeasible with cipher's 64-bit block size. Best known linear attack reaches up to 26 rounds with time complexity ~2^{65} and data complexity ~2^{63}.

Also, diffusion property of the bit permutation used in EE-LBC as explained in section III is strong enough to defend against linear cryptanalysis.

Key Schedule Attack:

By shuffling the bits of the key with the help of non-linear operations, the block cipher EE-LBC can resist key schedule attacks like round key attack where the intruder tries to identify the relationship between various sets of subkeys. This is done in the third step of key scheduler module by using round counter that is XORed with the middle portion of the round key. While generating each of the round key, use of non-linear function is made as described in the section III.

Although the current implementation of EE-LBC is software-based and not evaluated for side-channel resistance, potential countermeasures such as constant-time operations and masking techniques can be considered in future hardware implementations to enhance resilience against side-channel attacks.

To summarize the methodology of EE-LBC for attack resistance, while generating round keys, shuffling bits in the key using non-linear function by XORing round counter with the middle portion of the round key, makes it possible to protect the cipher from key schedule attacks. Due to its proper S-box activation properties, it has strong resistance against differential cryptanalysis. Efficient bit permutation leads to high diffusion which directly strengthens its security against linear cryptanalysis.

5. RESULTS AND DISCUSSION

The core goal of the proposed protocol of designing simplistic solution for security of resource-constrained network is consummated by optimizing the implementation for the performance. The key size of EE-LBC that is 80-bit provides more than acceptable security for the

applications requiring basic security, typically the one that uses tag-based deployments like applications in health-care sector, smart agriculture based on IoT etc.

Experimental Set-up:

The proposed protocol is implemented and executed using the simulator Omnet++ 5.7 [35, 36] to test its functionality. This simulator offers open-source framework based on C++ library. It supports for the communication of mobile and wireless network nodes [37]. The base network intended for the experiment is MANET which is wireless ad-hoc network. Simulation environment set-up indicating parameter values for the same are as follows:

Number of nodes	50
Number of connection links	varying
Broadcast delay	0.01 ms
Datarate	1 kbps
Project reference	queueinglib
Simulation runtime GUI	Qtenv
Simulation run mode	Fast
Data block size	64-bit
Message frequency	0s
Routing Protocol	AODV

As detailed in section II, several block ciphers have been examined, among which few with smaller key size and smaller data block processing per operation, have been simulated and further analyzed in this research for comparative purposes. These block ciphers along with EE-LBC as detailed in Table 2, were simulated using Omnet++. Comparative results of these ciphers are presented below. The comparison presented in Table 2 illustrates the security parameters of EE-LBC in relation to other block ciphers, including number of rounds, key size, throughput and immunity against cryptographic attacks etc.

Table 2. Security Analysis of Block Ciphers

Block Cipher	Key Length	Data Block Size	Rounds	No. of S-boxes	Throughput (Kbps)	Cryptanalysis
GIFT	128 bit	64 bit	28	08	20.40	Not safe against related-key attack
Twine	80 bit	64 bit	36	08	12.48	Not immune against related-key differential attack
PRINT	80 bit	48 bit	48	08	2.2	Resistant only to related key attack
KATAN32	80 bit	32 bit	254	-	3.5	Not safe against linear, differential and related-key attacks
ASCON	128 bit	64 bit	30	05	10	Resilient to differential and linear attacks
EE-LBC	80 bit	64 bit	31	01	23.7	Resistant against key scheduling attack, linear and differential cryptanalysis

Comparative Analysis:

Data Block Size and Key Size:

The 64-bit data block size is typical for lightweight ciphers, balancing between security and efficiency. The 80-bit key size provides a reasonable level of security for most practical applications but may be susceptible to brute-force attacks in the long term. GIFT and ASCON have a longer key compared to other ciphers that increases complexity during key scheduling. Relation of key-length with number of rounds for above block ciphers is presented in Fig. 2.

Throughput:

Throughput measures how quickly data can be processed. Higher throughput is generally favorable, especially in scenarios where real-time processing is critical. As compared to others, the cipher EE-LBC has higher throughput 23.7 Kbps due to its lesser block size, S-box and number of rounds, whereas PRINT has lowest of 2.2 Kbps. Comparison of throughput for all above ciphers is shown in Fig. 3 below.

Number of S-boxes:

The number and design of S-boxes contribute significantly to the security and cryptographic strength of the cipher. More S-boxes can enhance security against certain types of attacks but might increase computational complexity. Where GIFT, Twine and PRINT use 8 S-boxes, EE-LBC lowers the complexity by applying single 4-bit S-box 16 times.

Cryptanalysis Resistance:

This parameter assesses how resilient the cipher is against known attacks, such as differential and linear cryptanalysis, which are common in the evaluation of block ciphers. GIFT is strong against linear cryptanalysis due to its robust S-boxes and balanced diffusion properties. Twine and ASCON are resilient against differential and linear cryptanalysis through their strong S-boxes, round structure, and key schedule design. PRINT and KATAN32 are not immune against these cryptanalyses. Whereas EE-LBC is resistant to key scheduling attack, differential and linear cryptanalysis through its non-linear layers, permutation layer, complex key schedule and careful round function design as conferred in Section IV.

Encryption Time:

The simulation encompassed various message ranges, recording the encryption process time for the above ciphers depicted in Fig. 4. Encryption time is inversely proportional to the throughput. For comparison, we consider the simulation run for 100 messages, each of

size 1KB (8000 bits). By referring the throughput values from Table 2, encryption time for the above ciphers is computed using following formula:

Encryption Time = (Block Size (bits) x Number of Messages) / Throughput (bps)

Graph portraying the same in Fig. 4 clearly shows that EE-LBC requires the least encryption time of 33.76s and PRINT is ahead of the other ciphers due to the least throughput.

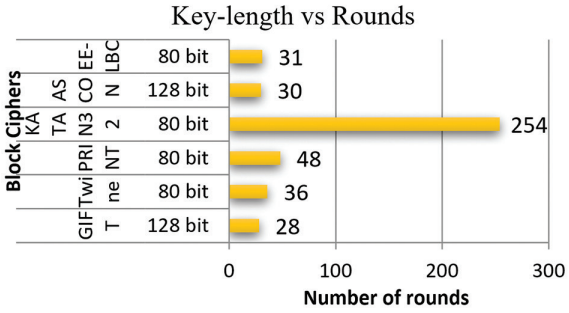


Fig. 2. Number of Rounds for ciphers

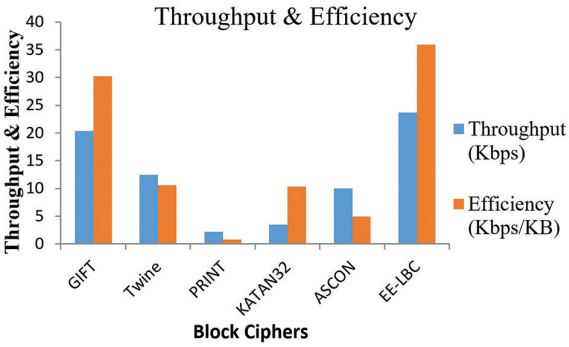


Fig. 3. Throughput and Efficiency of ciphers

Efficiency:

Another essential metric efficiency provides performance while minimizing resource requirements. It is dominated by the lengthier algorithms. It can be determined as follows:

Efficiency = Throughput (Kbps) / Code_Size (KB)

Graph in Fig. 3 for efficiency of the block ciphers is led by EE-LBC with the peak efficiency 35.91 Kbps/KB due to its smaller code size and trailed by PRINT with its extensive code.

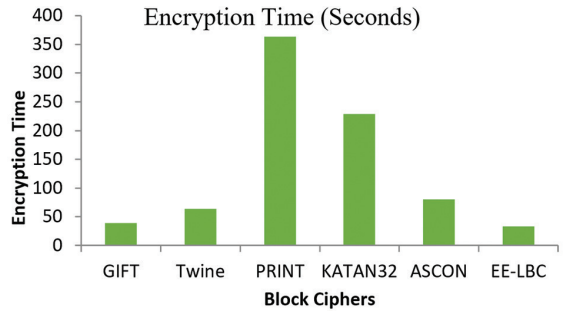


Fig. 4. Time for Encryption Process

6. CONCLUSION

The study focuses on securing resource-constrained wireless networks, which are vulnerable to attacks like man-in-the-middle that threaten control message integrity. The key challenge is developing energy-efficient security solutions, as conventional cryptographic algorithms are too resource-intensive. Lightweight cryptography offers a viable alternative with lower power and computational demands.

This paper introduces design and simulation of EE-LBC, a symmetric lightweight cryptographic block cipher structured on SPN), operating on a 64-bit data block with an 80-bit key, swirling through 31 rounds. The algorithm prioritizes simplicity in design by dropping S-box count to one and reduction in implementation cost while ensuring a satisfactory level of security making it an ideal choice for securing IoT devices and other energy-constrained systems. The algorithm exhibits resilience against key schedule attack, algebraic attack as well as linear and differential cryptanalysis. However, there remains a marginal vulnerability to biclique attacks, contingent upon the attacker conducting exhaustive computations involving approximately 2^{80} encryption attempts to determine the correct key.

Performance evaluation clarifies significant rise in throughput i.e., 23.7 Kbps whereas there is reduction of encryption time, for 100 messages each of size 1KB, of 13.92% compared to GIFT with least encryption time among others. Future work may involve further optimizations and extensions to EE-LBC, as well as exploration of its applicability to emerging IoT scenarios and use cases.

7. REFERENCES:

[1] Q. V. Khanh, L. A. Ngoc, "An energy-efficient routing protocol for MANET in Internet of Things environment", International Journal of Online and Biomedical Engineering, Vol. 17, No. 7, 2021, pp. 35-45.

[2] M. A. Khan, I. M. Qureshi, F. Khanzada, "A hybrid communication scheme for efficient and low-cost deployment of future flying ad-hoc network (FANET)", Drones, Vol. 3, No. 1, 2019, p. 16.

[3] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study", IEEE Access, Vol. 8, 2020, pp. 106576-106584.

[4] J. Jabez, B. Muthukumar, "Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach", Procedia Computer Science, Vol. 48, 2015, pp. 338-346.

[5] A. Biswas et al. “LRBC: a lightweight block cipher design for resource constrained IoT devices”, Journal of Ambient Intelligence and Humanized Computing, Vol. 14, 2023, pp. 5773-5787.

[6] Y. Zhong, J. Gu, “Lightweight block ciphers for resource-constrained environments: A comprehensive survey,” Future Generation Computer Systems, Vol. 157, 2024, pp. 288-302.

[7] B. V. dos Santos, A. Vergütz, R. T. Macedo, M. Nogueira, “A dynamic method to protect user privacy against traffic-based attacks on smart home”, Ad Hoc Networks, Vol. 149, No. C, 2023.

[8] S. Ganesh, R. Amutha, “Efficient and secure routing protocol for wireless sensor networks through SNR based dynamic clustering mechanisms”, Journal of Communications and Networks, Vol. 15, No. 4, 2013, pp. 422-429.

[9] F. Dridi, S. El Assad, W. El Hadj Youssef, M. Machhout, R. Lozi, “Design, implementation, and analysis of a block cipher based on a secure chaotic generator”, Applied Sciences, Vol. 12, No. 19, 2022, p. 9952.

[10] O. Toshihiko, “Lightweight cryptography applicable to various IoT devices”, NEC Technical Journal, Vol. 12, No. 1, 2017, pp. 67-71.

[11] NIST Contest, “Lightweight Cryptography: Finalists and Standardization”, https://csrc.nist.gov/projects/lightweight-cryptography (accessed: 2025)

[12] M. B. İlter, “Differential and Linear Cryptanalysis of Lightweight Block Ciphers with MILP Approach”, Graduate School of Applied Mathematics, Middle East Technical University, Ankara, Turkey, 2023, Ph.D. thesis.

[13] A. Bogdanov et al. “PRESENT: An Ultra-Lightweight Block Cipher”, Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10-13 September 2007.

[14] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers”, IACR Cryptology ePrint Archive, Vol. 2013, 2013, p. 404.

[15] D. Hong et al. "LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors", Proceedings of the 14th International Workshop on Information Security Applications, Jeju Island, Korea, 19-21 August 2013, pp. 3-27.

[16] V. A. Thakor, M. A. Razzaque, M. R. A. Khandaker, "Lightweight Cryptography for IoT: A State-of-the-Art", IEEE Internet of Things Journal, Vol. 7, No. 10, 2020, pp. 9370-9383.

[17] V. A. Thakor, M. A. Razzaque and M. R. A. Khandaker, “Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities”, IEEE Access, Vol. 9, 2021, pp. 28177-28193.

[18] N. Pub, “Advanced encryption standard (AES)”, Federal Information Processing Standards, Vol. 197, No. 441, 2001, p. 0311.

[19] R. Anusha, V. V. D. Shastrimath, “LCBC-XTEA: High throughput lightweight cryptographic block cipher model for low-cost RFID systems”, Proceedings of the 8th Computer Science On-line Conference: Cybernetics and Automation Control Theory Methods in Intelligent Algorithms, Vol. 3, 2019.

[20] T. Suzaki, K. Minematsu, S. Morioka, E. Kobayashi, “Twine: A lightweight, versatile block cipher”, Proceedings of the 19th International Conference on Selected Areas in Cryptography, Windsor, Canada, 15-16 August 2012, pp. 1-5.

[21] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, I. Verbauwhede, “RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms”, Science China Information Sciences, Vol. 58, No. 12, 2015, pp. 1-15.

[22] Periasamy et al. “An Enhanced Energy Efficient Lightweight Cryptography Method for various IoT devices”, ICT Express, Vol. 7, 2021.

[23] L. Knudsen, G. Leander, A. Poschmann, M. Robshaw, “Printcipher: A block cipher for IC-printing”, Proceedings of the 12th International Workshop, Santa Barbara, CA, USA, 17-20 August 2010, pp. 16-32.

[24] J. Borgho et al. “PRINCE—A low-latency block cipher for pervasive computing applications”, Proceedings of the 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, 2-6 December 2012, pp. 208-225.

[25] D. Williams, "The tiny encryption algorithm (TEA)", Network Security, Vol. 26, 2008, pp. 1-14.

[26] C. Beierle et al. "Sim.: The skinny family of block ciphers and its low-latency variant mantis", Proceedings of the 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, 14-18 August 2016, pp. 123-153.

[27] S. Banik et al. "Gift: A Small Present: Towards Reaching the Limit of Lightweight Encryption", Proceedings of the 19th International Conference on Cryptographic Hardware and Embedded Systems, Taipei, Taiwan, 25-28 September 2017.

[28] B. Aboushosha et al. "SLIM: A Lightweight Block Cipher for Internet of Health Things", IEEE Access, Vol. 8, 2021, pp. 203747-203757.

[29] A. Ukpebor, J. Addy, K. Ali, A. A. Humos, "Secure End-to-End Communications with Lightweight Cryptographic Algorithm", IEEE Internet of Things Journal, Vol. 10, No. 2, 2023, pp. 1023-1034.

[30] J. Kaur, A. C. Canto, M. M. Kermani, R. Azarderakhsh, "A Comprehensive Survey on the Implementations, Attacks, and Countermeasures of the Current NIST Lightweight Cryptography Standard", IEEE Access, Vol. 11, 2023, pp. 12345-12367.

[31] J. S. Teh, L. J. Tham, N. Jamil, W.-S. Yap, "New differential cryptanalysis results for the lightweight block cipher BORON", Journal of Information Security and Applications, Vol. 66, 103129, 2022.

[32] J. Lu, "A Methodology for Differential-Linear Cryptanalysis and Its Applications", Designs, Codes and Cryptography, Vol. 77, No. 1, pp. 11-48, Oct. 2015.

[33] S. Sallam, B. D. Beheshti, "A survey on lightweight cryptographic algorithms", Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea, 28-31 October 2018, pp. 1784-1789.

[34] A. Bogdanov, P. S. Vejre, "Linear Cryptanalysis of DES with Asymmetries", in ASIACRYPT 2017, Vol. 10624, Eds. Cham: Springer, 2017, pp. 187-216.

[35] OmNET++ 5.7, https://omnetpp.org/download/ (accessed: 2025)

[36] A. Varga and OpenSim Ltd. "OMNeT++ User Guide Version 6.0", https://omnetpp.org/doc/omnetpp5/UserGuide.pdf (accessed: 2025)

[37] S. Manzoor, M. Manzoor, H. Manzoor, D. E. Adan, M. A. Kayani, "Which Simulator to Choose for Next Generation Wireless Network Simulations? NS-3 or OMNeT++", Engineering Proceedings, Vol. 46, No. 1, 2023, p. 36.

Optimized t-Test Feature Selection for Real-Time Detection of Low and High-Rate DDoS Attacks

Original Scientific Paper

Raghupathi Manthena
Research Scholar, Department of Computer Science and Engineering,
Jawaharlal Nehru Technological University Hyderabad, Telangana, India
mraghu30@gmail.com

Radhakrishna Vangipuram*
Department of Information Technology,
VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India
radhakrishna_v@vnrvjiet.in

*Corresponding author

Abstract *– Distributed Denial of Service (DDoS) attacks stand out as a serious threat, capable of disrupting online services and businesses. The main aim of Distributed Denial of Service (DDoS) attacks is to make system services unavailable to the legitimate users. To detect these attacks, intrusion detection systems (IDS) continually monitor the network traffic. During this process, the IDS system generates high false positive rates while distinguishing low-rate DDoS (LRDDoS) and high-rate DDoS (HRDDoS) attack traffic from legitimate traffic. The idea behind feature selection is that picking the right network features is a key part of interpreting the difference between normal traffic and LRDDoS or HRDDoS attack traffic. This means the IDS performance will automatically get better. In this paper, we propose a scalable feature selection method that utilizes the statistical t-test to identify an optimal feature subset from original feature set at a low computational cost. We strongly hypothesize that the proposed feature selection method yields an optimal feature subset and the machine learning classifiers trained on this feature set can effectively distinguish benign, LRDDoS, and HRDDoS network traffic. We evaluated the proposed method on the publicly available benchmark datasets CICIDS2017, CICIDS2018, and CICDDoS2019, utilizing twelve supervised machine learning classifiers. Among the twelve classifiers, the Extra Tree Classifier (EXT) demonstrated superior performance, achieving an average accuracy of 96.50%, precision of 96.58%, and an F-Score of 96.50% across the four sample test datasets (D1, D2, D3, and D4). The proposed method showed consistent and superior performance in distinguishing the LRDDoS, HRDDoS, and benign traffic to the state-of-the-art existing works over the four test datasets.*

Keywords*: t-Test, Feature selection, DDoS traffic, LRDDoS, HRDDoS, CICDDoS2019, Balanced accuracy*

Received: December 5, 2024; Received in revised form: February 26, 2025; Accepted: April 3, 2025

1. INTRODUCTION

As our dependence on technology continues to rise, the importance of cybersecurity has surged to unprecedented heights. The Internet of Things (IoT), cloud computing, mobile devices, and the widespread use of digital communication have all contributed to an exponential increase in the attack surface for cyber-attacks [1]. Among all the cyberattacks, Distributed Denial of Service (DDoS) attacks stand out as a serious threat, capable of disrupting online services and businesses.

Attackers carry out these network attacks by overwhelming the networks or server's resources (CPU, memory, bandwidth, etc.) with massive traffic because of which even a legitimate user will not able to access services of network or server. According to the NetScout threat report H1 2024, DDoS attacks rose 12.8% compared to H2 2023 NetScout threat report. The longest attack lasted for over an hour, resulting in a count of 825,217 DDoS attacks. Furthermore, in less than 5 minutes, the number of DDoS attacks increased to 4,137,582. In this paper, we attempt to categorize modern DDoS attacks into two groups based on rapid disruption of network or server services. The first group of DDoS attacks are high-rate DDoS (HRDDoS) attacks, which make the services unavailable within short period of time. The second category of attacks consists of low-rate DDoS (LRDDoS) attacks. In this kind of network attack, attackers ex-

ploit potential logic errors or vulnerabilities in the service to send the malicious requests. These malicious requests slowly make the server unavailable for legitimate users. Some research works focus on model building instead of feature selection tasks, as mentioned in [2]. These types of models result in high false positive rates.

Research works [3-5] developed an IDS system to detect DDoS attacks applying feature selection and machine learning techniques. These research studies evaluated learning machines on the CICDS2017 dataset. However, the limitation is that the CICDDoS2017 dataset does not represent a wide range of DDoS attacks. Also, the dataset only included a limited number of DDoS samples, failing to cover full spectrum of DDoS classes. Though studies [6-8] propose feature subsets to identify DDoS attacks they are not better representative features for LRDDoS and HRDDoS attacks.

Usually, most research studies directly apply conventional feature selection methods to select the optimal feature subset. These methods are divided into three categories: filter-based, wrapper-based, and embedded methods. Each of these methods have their advantages and limitations. The filter-based methods calculate feature importance using statistical properties without employing machine learning algorithms. The information gain (IG), chi-square, and correlation coefficient (CrC) methods are fast and computationally efficient, making them suitable for high-dimensional data. These methods may ignore interactions among features when feature relationship is complex. Additionally, when working with continuous data, there may be a bias towards more diverse features, which could result in the inclusion of irrelevant features in the subset.

Wrapper methods generate different feature subsets using random selection or heuristic selection. The machine learning algorithm will select the optimal feature subset based on each subset performance. While these methods enhance the performance of a model by using an optimal feature subset, they can be computationally costly when dealing with large datasets, may also limit the model generalizability.

Embedded methods use algorithms like Lasso (L1 regularization), Ridge (L2 regularization), and decision trees to build feature selection right into the model training process. These methods often result in improved generalization and performance of the model. Specific algorithms tailor these methods, potentially limiting their broader applicability and making them less interpretable than filter methods. Thus, feature subsets obtained by conventional feature selection techniques for binary classification of DDoS attacks fail to discriminate LRDDoS and HRDDoS attacks. Therefore, it is essential to identify a more appropriate and optimal feature subset for detection of LRDDoS and HRDDoS attacks.

The problem of accurately distinguishing between legitimate user traffic, LRDDoS and HRDDoS network traffic is the present challenge w.r.t DDoS attacks. Incorrectly identifying attack traffic can lead to system overload or failure, while misclassifying legitimate traffic can cause service disruptions and financial losses. This problem can be addressed using machine learning with t-test feature selection, which helps to identify the most important features to discriminate between benign and attack traffic. By focusing on the relevant features, the system can improve its accuracy in detecting attacks while minimizing errors, ensuring both security and continuous service for legitimate users.

The motivation for this study coins from the gap in the present literature which does not address detection of LRDDoS and HRDDoS variants using machine learning techniques integrated with a lightweight feature selection method. Also, the immense volume of modern network traffic necessitates the immediate need for identification of key features in minimal time and at the same time to obtain high detection accuracy. LRDDoS attacks can gradually merge with legitimate traffic, while HRDDoS attacks result in abrupt and massive data spikes. Thus, for effective detection in both scenarios, it is essential to focus on selecting the most relevant features that can distinguish between low-rate DDoS attack traffic and legitimate user traffic.

At the outset, to address the limitations of conventional feature selection methods, in this research we propose a lightweight feature selection method to obtain an optimal feature subset that detects LRDDoS and HRDDoS attacks by employing the Extra Tree classifier (EXT).

The proposed method is a lightweight solution for selecting significant features with less computation time. Following are highlights of the present work.

- In this research, we propose a light-weight feature selection method that leverages the inferential statistical t-Test to identify optimal feature set to discriminate LRDDoS and HRDDoS attacks.
- Based on the proposed feature selection method, we strongly suggest 58 network traffic features for differentiating low-rate and high-rate DDoS traffic from benign traffic.
- To evaluate the proposed method, we utilized reliable and publicly available benchmark dataset CICDDoS2019. We tested our method on four different testing datasets (D1, D2, D3, and D4) obtained from testing day traffic of CICDDoS2019 dataset to achieve generalizability.
- We evaluated performance of twelve machine learning classifiers on the feature subset identified by our feature selection method. Among all classifiers, the Extra Tree classifier (EXT) performed the best, with an average accuracy of 96.50%, precision of 96.58%, and F-score of 96.50% across four test datasets (D1, D2, D3, and D4).
- On the CICIDS2017 and CICIDS2018 datasets, an accuracy 99.81% and 99.99% is achieved by proposed method.

2. RELATED STUDY

In this section, we provide a brief discussion of the popular feature selection methods used to select a subset of features and implement the IDS system. A number of IDS datasets are available publicly, and it is proved that the CICDDoS2019 dataset is the most reliable and contains updated DDoS attack vector instance [6]. This led us to concentrate on the CICDDoS2019 IDS dataset, which has been the subject of evaluation in recent studies.

In cybersecurity, especially when it comes to detecting Distributed Denial-of-Service (DDoS) attacks, the performance of machine learning models largely depends on the quality of the input data. For each class of the CICDDoS2019 dataset DDoS attacks, Sharafaldin et al. [6] suggested a significant feature subset with 24 different features using a weighted standard mean of random forest feature importance. The performance of ID3, Random Forest, and logistic regression machine learning classifiers is evaluated on these 24 features. Overall, ID3 classifier outperformed the other two, achieving a high detection rate of 65% and an accuracy of 78% on more than 7 crore instances. The approach they used did not address the detection of LRDDoS and HRDDoS attacks. They did not discuss the generalizability of their model, which could lead to variations in the rate of DDoS attacks in different network data.

To address the high dimensionality problem, S. Li et al. [7] suggested Truncated Lanczos-Tensor SVD to reduce the dimensionality of large-scale datasets. However, they did not address the adaptability, and practical evaluation of this method. Hajimaghsoodi and Jalili [8] suggested a novel method, i.e., a 3-phase RAD model, to detect DDoS attacks using a statistical approach. The number of features used to evaluate their model remains undisclosed and did not address the low-rate and high-rate DDoS attacks detection. To detect, identify, categorize, and classify IoT DDoS attacks, Jia et al. [9] developed the edge-centric protection system FlowGuard. However, the system has certain drawbacks, including its dependence on artificial datasets, which could potentially impact its real-world applicability, and its use of high-performance edge servers, which could limit its scalability in resource-constrained environments. Maheswari et.al [10], developed an optimized weighted voting-based ensemble model for detection of DDoS attacks in SDN environment. and selected 20 features using statistical analysis. However, in this study they did not discuss the computational cost and detection of high- and low- rate DDoS attacks. The most recent work by S. Mahdavifar and A. A. Ghorbani [11], suggested 22 significant features using the mutual information gain and developed CapsRule method to detect the reflection-based DDoS attacks but they did not study for low rate and high-rate DDoS network attacks. Enock Q.E. et al [12] proposed a feature selection method by integrating mutual information gain, correlation, and random forest feature importance for DDoS attacks detection using RCHT method.

G.C. Amaizu et al. [13] developed the DDoS detection system for 5G and B5G networks, which uses an effective feature extraction technique in conjunction with a composite multilayer perceptron to detect and categorize DDoS attacks. The multilayer perceptron models and feature extraction in real-time applications may increase the computational burden. Cil et al. [14] suggested a deep learning (DL) model that combines feature extraction and classification. Using DNN algorithm, they achieved improved DDoS attack detection and classification. However, for multiclass classification, the model accuracy was lower. In order to construct an effective intrusion detection system (IDS) for detecting and categorizing DDoS attacks, A.A. Najar et al. [15] suggested a feature subset which contains 43 features. The creative feature selection, efficient preprocessing, and comprehensive dataset analysis are important contributions of this study. Although it provides a high detection accuracy and quick detection times, drawbacks include (i) inability to handle unbalanced data, (ii) computational complexity, (iii) dependence on the dataset quality, and (iv) difficulties of extrapolating to other attack types. Wei et.al [16] suggested, a deep learning-based hybrid AE-MLP method to classify the DDoS attacks. They used an autoencoder to denoise the DDoS attacks and then classified them using MLP. Ferrag et.al [17], developed a deep learning-based CNN method to address the classification problems of DDoS attacks in the smart agriculture sector. A. Alashhab et al. [18] suggested an IDS framework utilizing online machine learning (OML) to detect DDoS attacks within software-defined networks (SDN). They identified 22 features using their custom dataset. However, these works did not address the generalizability of the model, low-rate and high-rate DDoS attacks detection. To address the generalizability issue in IDS system, O. Barut et.al [19] developed R1D1T model to classify the DDoS attacks using raw packet data. The R1D1T model converts the data into 1-D image and applies self-attention-based neural networks to perform classification. Despite addressing these issues, the generalizability and computational cost of this model pose serious limitations.

Li et al. [20], developed a mathematical model for detecting and mitigating low-rate DDoS attacks in cloud computing environments, specifically targeting container-based DDoS attacks. However, this work focused on LRDDoS attacks and neglected HRDDoS attacks. They designed their mathematical model based on the number of requests per unit time in the test bed network. Makhduma F. Saiyed et al. [21], designed a lightweight method FLUID, to differentiate DDoS attacks from legitimate traffic. The development of this approach relied on the theories of Kullback-Leibler (KL) divergence and greedy bin-packing information. In this approach, they have achieved an average 90% accuracy on the CICDS2017, CICDDoS2019, and ToN-IoT datasets based on a single threshold value. However, threshold value-based methods may not be suitable to discriminate the LRDDoS and HRDDoS attack traffic from the legitimate traffic.

Raghupathi et al. [22], suggested a feature selection method using independent sample t-test. This method was used to identify significant features for the detection of DDoS attacks and focused solely on binary classification. M. F. Saiyed and I. Al. Anbagi [23], suggested GADAD model to select key features using GAStats method to detect low-rate and high-rate DDoS attacks. However, this model computational time is high.

Thus, the majority of research studies focused on detecting DDoS attacks but did not address for LRDDoS and HRDDoS traffic detection. Few studies [20], [21] and [23] focused on detecting either LRDDoS or HRDDoS attacks. The research literature shows that there has been limited research w.r.t detection of LRDDoS and HRDDoS attacks. Thus, in this paper we mainly focused on addressing three key areas: We aim to (1) identify the significant feature subset with less computational cost and at 95% confidence interval; (2) discriminate between LRDDoS and HRDDoS attacks with high accuracy and precision rates, and (3) develop a generalizable model. The current study addresses low-rate and high-rate DDoS attack detection.

3. METHODOLOGY

A key statistical method for determining whether there is a significant difference between the means of two groups is the statistical t-test. It plays a crucial role in the hypothesis testing, which evaluates whether observed differences are genuine or merely due to random chance. Various fields widely employ this straightforward yet powerful technique to derive meaningful insights from data comparisons. Researchers frequently use the t-test to test hypotheses and make inferences about population parameters based on sample data. Using the power of the t-test for statistics, we propose a method to obtain a feature subset that can unearth the difference between DDoS traffic and normal traffic.

The proposed feature selection method utilizes the training dataset (D_{MxN}), where M represents the number of instances and N denotes the number of features, as outlined in the algorithm. The preprocessing steps, from step 1 to step 6, involve eliminating static features, those with standard deviation zero, and highly correlated features. The algorithm also eliminates duplicate instances; if they represent less than 0.05% of the total instances of the class label and contain NaN or infinite values. The proposed feature selection method is performed from step 7 to step 14. The feature selection method starts by dividing the preprocessed dataset into three groups: 1. BENIGN, 2. LOW, and 3. HIGH. In step 8, algorithm selects the feature in sequential order from the BENIGN group and the LOW group. Next, we calculated the mean and variance of the current feature using Eq. 1 and Eq. 2 respectively. In Eq.1 and Eq.2 $\bar{x}_{BENIGN}$ and $\bar{x}_{DDoS\ Attack}$ are the mean value of current feature considered from two groups; σ^2_{BENIGN} and $\sigma^2_{DDoS\ Attack}$ are variances of the current feature.

$$Mean\ (\bar{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

$$Variance\ (\sigma^2) = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{2}$$

$$T_{testscore} = \frac{\bar{x}_{BENIGN} - \bar{x}_{DDoS\ Attack}}{\sqrt{\left(\frac{\sigma^2_{BENIGN}}{n_{BENIGN}} - \frac{\sigma^2_{DDoS\ Attack}}{n_{DDoS\ Attack}}\right)}} \tag{3}$$

Then, T_{test_score} is computed using Eq.3 wherein variables n_{BENIGN} and $n_{DDoS\ Attack}$ represent the number of samples in respective groups. Here, the $DDoS\ Attack$ can belong to LOW or HIGH group. In the next step, the degree of freedom (DOF) is computed using Eq. 4, and $T_{critical}$ value is obtained from $t_{distribution}$ table with a 95% confidence interval w.r.t DOF.

$$DOF = \frac{\left(\frac{\sigma^2_{BENIGN}}{n_{BENIGN}} + \frac{\sigma^2_{DDoS\ Attack}}{n_{DDoS\ Attack}}\right)^2}{\frac{(\sigma^2_{BENIGN})^2}{n_{BENIGN} - 1} + \frac{(\sigma^2_{DDoS\ Attack})^2}{n_{DDoS\ Attack} - 1}} \tag{4}$$

In the subsequent steps, features that satisfy the given constraint are identified and considered as significant, if T_{test_score} of the feature is greater than $T_{critical}$ value. From step 8 to step 10, we proceed until we reach (N-1)th feature. The same process is repeated by considering HIGH and BENIGN groups to identify significant features for detection of HRDDoS attacks. After identifying significant feature subsets separately for LRDDoS and HRDDoS attacks these feature subsets are merged to obtain the final feature subset. The T_{test_score} value is determined using the statistical evaluation method described in the Algorithm, where the ability of each feature to discriminate between Benign, Low and High classes is analyzed based on the independent t-Test value.

Algorithm: Proposed Feature Selection for Detection of Low-rate and High-rate DDoS Network Traffic

Input: Train dataset D_{MxN}

M → Number of instances

N → Number of input features (1 to N-1 are the iput features and Nth feature is the label)

Output: Optimal feature subset F'

Start

Step 1: Remove socket features. {Unnamed 0, 'Flow Id', 'Source IP' 'Source Port' , 'Destination IP' , 'Detination Port', 'Protocol' , 'Timestamp' and 'SimilarHTTP'}

Step 2: Removal of duplicate instances

D^1 ← D updated dataset after removing duplicate rows

Step 3: Removal of *NaN* or *Inf* values contained rows

$D^2 \leftarrow D^1$ updated dataset after removing NaN and Infinity value rows

Step 4: Removal of highly correlated features

{fwd header length (2), subflow fwd packets, subflow fwd bytes, subflow bwd packets, and suflow bwd bytes}

Step 5: Removal of standard deviation is zero fetures

{fwd avg packets/bulk, fwd avg bulk rate, bwd avg bytes/bulk, bwd psh flags, fwd urg flags, bwd urg flags, fwd avg bytes/bulk, bwd avg packets/bulk, and bwd avg bulk rate, fin flag count, psh flag count and ece flag count}

Step 6: Handling the negative values in the dataset

Step 7: Split the dataset into 3 groups based on class labels: BENIGN, 2.LOW and 3. HIGH

Step 8: Calculate mean and variance of current feature from BENIGN group and LOW/HIGH group

Step 9: Calculate T_{test_score} value of current feature using mean and variance

Step 10: Calculate degree of freedom for current feature

Step 11: Obtain $T_{critical}$ value with respect to degree of freedom at 95% confidence interval

Step 12: If T_{test_score} value greater than $T_{critical\ value}$, Consider the feature is significant

Step 13: Repeat from step 8 to step 12, until the end of feature set and groups.

Step 14: Return optimal feature subset F'

Stop

4. DATASET

To evaluate the performance of the proposed method, we utilized the CICIDS2017 [24], CICIDS2018 [25], and CICDDoS2019 widely used benchmark IDS datasets. Among these, the CICDDoS2019 dataset satisfies all the eleven properties listed by Gharib et al. [26], making it a reliable IDS dataset. It offers diverse DDoS attack scenarios, including normal traffic, enabling researchers to evaluate the effectiveness of their detection algorithms. Its comprehensive feature set aids in the development of sophisticated methods for feature selection and model training. The training dataset has more than 50 million instances split into 13 class labels. Of these, one label represents benign (legitimate user) traffic, and remaining 12 labels represent various types of DDoS attack traffic. The testing dataset includes more than 20 million instances, categorized into 8 class labels, where one label denotes benign traffic and the other 7 represents different attacks. For our experimentation, we utilized a sample of 399,998 instances for training day dataset and 112,611 instances for testing day dataset to evaluate the proposed method.

Table 1. Class distribution of the CICDDoS2019 training dataset utilized for evaluation

SN	Class Label	Number of Instances
1	BENIGN	56425
2	WebDDoS	439
3	UDP_Lag	31194
4	NetBIOS	31194
5	LDAP	31194
6	MSSQL	31194
7	DNS	31194
8	SYN	31194
9	UDP	31194
10	TFTP	31194
11	NTP	31194
12	SNMP	31194
13	SSDP	31194
Total		**399998**

Table 2. Class distribution of the CICDDoS2019 testing dataset utilized for evaluation

SN	Class Label	Number of Instances
1	BENIGN	56306
2	UDP_Lag	1873
3	NetBIOS	9072
4	LDAP	9072
5	MSSQL	9072
6	PortMap	9072
7	SYN	9072
8	UDP	9072
Total		**112611**

Tables 1 and 2 depict the class distribution of the training and testing datasets respectively. To ensure generalizability, robustness, reduce overfitting; we have sampled four different testing datasets (D1, D2, D3, and D4) from 20 million testing day network traffic instances, each having 112611 network flow samples. Additionally, we derived a validation dataset (V_D) of 112612 samples from 50 million sequential samples of the CICDDoS2019 dataset. Thus, validation and testing datasets contain unique instances and are of same size. For experimental analysis, Classes labels are encoded by encoding benign instances as BENIGN, low-rate attack traffic as LOW, and high-rate attack traffic as HIGH, using Andrew Visualization Plot.

To ensure the generalizability of the proposed method, we have also considered benign, low-rate, and high-rate DDoS instances from publicly available datasets (CICIDS2017 and CICIDS2018). The CICIDS2017 and CICIDS2018 sample datasets information is depicted in Table 3.

Table 3. Class distribution of CICIDS2017 and CICIDS2018 datasets used for evaluation

SN	Dataset	Class Label	Number of Instances		
			Training	Validation	Testing
1	CICIDS 2017	BENIGN	30065	9956	9979
		LOW	12974	4349	4265
		HIGH	59913	20013	20074
2	CICIDS 2018	BENIGN	119981	39956	40063
		LOW	1067	328	335
		HIGH	118952	39716	39602

5. EXPERIMENTATION AND RESULTS

In this section, we discuss the experimentation and evaluation results. All the experiments were executed on a Dell PC, which has an i7 Intel processor with 2.2Ghz speed and 16 MB RAM. To simulate the proposed work, the Jupiter notebook IDE and python scripts were utilized.

Fig. 1 illustrates the architecture of proposed system for the detection of LRDDoS and HRDDoS attacks. The training dataset, which initially had 87 features along with the target feature was input to the feature selection algorithm. The preprocessing stage reduced the number of feature dimensions from 87 to 61. During the feature selection phase, the application of a statistical t-Test resulted in 58 significant features in just 0.49 seconds. The 58 features resulted from feature selection algorithm are included in the appendix. For evaluation of the proposed method, 58 features resulted from t-Test are retained w.r.t training, validation and testing datasets and twelve machine learning classifiers were used to measure their classification and prediction performance. The classifiers included AdaBoost, K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Naive Bayes (NB), Quadratic Discriminant Analysis (QDA), Random Forest (RF), and Ridge. The performance evaluation metrics considered are Accuracy (Acc), Precision (Prec), Sensitivity (Sns), Specificity (Spe), F-score and Balanced Accuracy (BA) given by Eq.5 to Eq.10 respectively.

$$Acc = \frac{TP + TN}{TP + TN + FN + FP} \tag{5}$$

$$Pre = \frac{TP}{TP + FP} \tag{6}$$

$$Sns\ (or)\ Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Spe = \frac{TN}{TN + FP} \tag{8}$$

$$F - Score = 2 * \frac{Pre * Sns}{Pre + Sns} \tag{9}$$

$$Balanced\ Accuracy = \frac{Sns\ + Spe}{2} \tag{10}$$

Before training the classifiers, we normalized the training dataset using a min-max scalar, which scaled all the data points within the range of 0 and 1. The normalized dataset with 399998 instances, which included 58 features is input to twelve machine learning classifiers for training. These twelve classifiers were then validated using 112612 instances. Table 4 provides a detailed performance analysis of twelve machine learning models on the validation dataset (V_D). Subsequently, the performance of machine learning models is evaluated on four distinct testing day subset datasets. The performance results of twelve models on four test datasets (D1, D2, D3, and D4) is depicted using Table 5, Table 6, Table 7 and Table 8 respectively.

Table 4. Validation metrics of twelve machine learning models using the CICDDOS2019 validation dataset

SN	Model	Acc. (%)	Prec. (%)	F-Score (%)
1	ADB	78.56	78.93	78.72
2	DT	82.67	82.64	82.65
3	EXT	80.92	80.98	80.94
4	KNN	83.77	83.74	83.74
5	LDA	77.44	78.98	77.43
6	LR	80.79	81.53	80.99
7	MLP	80.25	80.45	80.32
8	NB	85.23	87.50	84.44
9	QDA	27.01	9.10	13.61
10	RF	81.11	81.16	81.13
11	Ridge	76.88	78.62	76.99
12	XGB	82.52	82.50	82.50

Table 5. Performance metrics of twelve classifiers using the CICDDOS2019 testing day dataset D1

SN	Model	Acc. (%)	Prec. (%)	F-Score (%)
1	ADB	55.18	50.43	48.87
2	DT	82.17	86.86	83.99
3	EXT	96.12	96.24	96.15
4	KNN	88.12	92.41	89.32
5	LDA	88.10	90.99	88.97
6	LR	81.22	90.72	83.73
7	MLP	82.73	91.22	85.12
8	NB	90.58	89.43	89.55
9	QDA	44.57	73.90	31.11
10	RF	90.07	90.75	90.30
11	Ridge	53.56	71.62	49.54
12	XGB	95.30	95.58	95.41

Table 6. Performance metrics of twelve classifiers using the CICDDOS2019 testing day dataset D2

SN	Model	Acc. (%)	Prec. (%)	F-Score (%)
1	ADB	54.19	50.52	49.30
2	DT	91.60	92.70	91.98
3	EXT	96.56	96.63	96.56
4	KNN	87.09	92.15	88.46
5	LDA	88.32	91.09	89.14
6	LR	79.20	90.35	81.97
7	MLP	80.96	90.86	83.62
8	NB	92.06	91.36	91.47
9	QDA	43.67	73.26	30.39
10	RF	89.65	89.74	89.62
11	Ridge	53.58	60.47	49.56
12	XGB	95.31	95.63	95.43

Table 7. Performance metrics of twelve classifiers using the CICDDOS2019 testing day dataset D3

SN	Model	Acc. (%)	Prec. (%)	F-Score (%)
1	ADB	54.18	48.73	48.63
2	DT	85.23	89.39	86.68
3	EXT	96.67	96.73	96.67
4	KNN	87.35	92.19	88.69
5	LDA	88.06	91.04	88.96
6	LR	79.20	90.41	82.00
7	MLP	81.16	90.90	83.81
8	NB	92.04	91.23	91.35
9	QDA	43.98	73.43	30.60
10	RF	88.29	89.61	88.75
11	Ridge	53.51	60.54	49.55
12	XGB	95.49	95.73	95.59

Table 8. Performance metrics of twelve classifiers using the CICDDOS2019 testing day dataset D4

SN	Model	Acc. (%)	Prec. (%)	F-Score (%)
1	ADB	55.16	48.98	49.23
2	DT	85.12	89.36	86.60
3	EXT	96.66	96.72	96.65
4	KNN	87.46	92.22	88.76
5	LDA	88.12	91.08	89.01
6	LR	79.12	90.39	81.94
7	MLP	81.16	90.91	83.82
8	NB	92.06	91.27	91.39
9	QDA	43.94	73.41	30.58
10	RF	89.10	89.99	89.41
11	Ridge	53.49	66.25	49.54
12	XGB	95.30	95.60	95.41

From these results, we observed that ADB, QDA and Ridge performance is lower, and the test accuracy ranges between 43% and 55% for these classifiers. With the exception of the EXT model, the accuracy of the remaining models ranged from 80% to 95%. Overall, the EXT model showed superior performance compared to the remaining eleven machine learning models. The EXT tree model outperformed eleven models with an average accuracy of 98.31%, precision of 99.97%, and F-score of 98.29%. Using our method, the model is able to detect benign traffic and HRDDoS attack traffic with an average balanced accuracy of 98%, while LRDDoS attacks are detected with an average balanced accuracy of 91.20% which is very significant. The hyperparameter settings used for EXT classifier are n_estimaters = 10, criterion='gini', max_depth=none, min_samples_split = 2, min_samples_leaf = 1, max_features = 'auto', n_jobs =1, random_state = none.

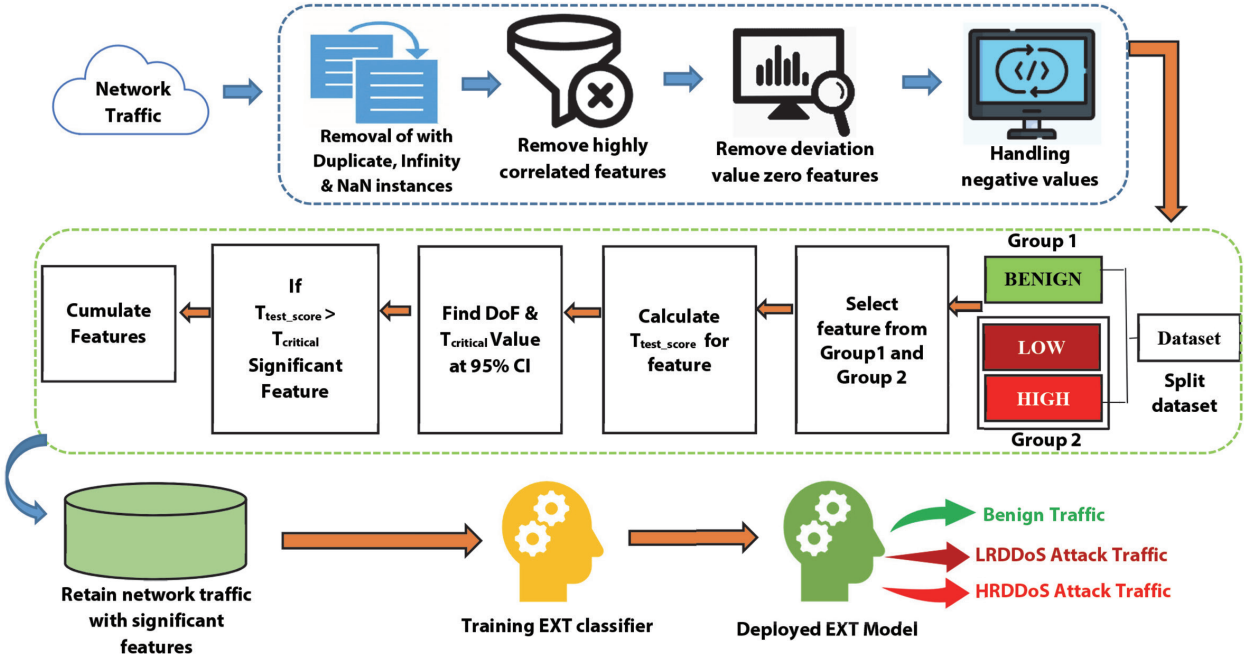


Fig. 1. Architecture for detection of low-rate and high-rate DDoS attacks

Furthermore, to analyze the model in-depth we utilized receiver operating characteristics (ROC) curves as an evaluation metric. Fig. 2 depicts ROC curves obtained when the model is evaluated on the four test datasets (D1, D2, D3 and D4).

Table 9 displays the detailed performance results of the EXT model against BENIGN, LRDDoS, and HRDDoS attacks. In this study, we also present the balanced accuracy metric, which reveals the performance of individual classes w.r.t three-class classification.

COMPARISON WITH EXISTING FEATURE SELECTION METHODS:

We have compared our feature selection method to widely applied methods such as filtering (information gain and variance threshold), embedding (logistic regression), and wrapping (Random Forest Importance and Extra Tree Classifier Importance). Table 10 depicts the comparison of the proposed method to some of the widely used feature selection methods. The Information Gain method identified 32 features and achieved an average accuracy of 74% across four test datasets using the EXT classifier. In contrast, the logistic regression method identified 27 features with an average accuracy of 80%. Similarly, the random forest importance method also reached an average accuracy of 80% with 18 features, while the Extra Tree classifier importance achieved the same accuracy using 17 features. However, our proposed feature selection method outperformed all five methods, achieving an average accuracy of 96.50% using 58 significant features. In Table 10, NOF denotes number of features. Fig. 3 shows comparison of how well the EXT model worked on four testing day datasets using conventional feature selection methods vs. proposed method.

Table 9. Performance metrics obtained for validation and four testing day datasets of CICDDOS2019 for the EXT model using the proposed feature selection

Dataset	Class Label	Sns. (or) Recall (%)	Spe. (%)	Pre. (%)	Acc. (%)	F-Score (%)	BA (%)
VD	BENIGN	99.99	99.99	99.99	99.99	99.99	99.99
	LOW	60.24	87.07	58.03	80.93	59.12	73.65
	HIGH	63.06	87.53	65.16	80.92	64.09	75.29
D1	BENIGN	96.60	99.94	99.94	98.27	98.25	98.27
	LOW	84.24	97.97	81.69	96.63	82.94	91.10
	HIGH	98.39	96.62	95.16	97.34	96.75	97.51
D2	BENIGN	96.67	99.96	99.96	98.31	98.28	98.31
	LOW	84.24	98.40	85.03	97.03	84.63	91.32
	HIGH	99.40	96.69	95.30	97.78	97.31	98.05
D3	BENIGN	96.75	99.99	99.99	98.37	98.34	98.37
	LOW	84.13	98.51	85.87	97.11	84.99	91.32
	HIGH	99.61	96.69	95.31	97.87	97.41	98.15
D4	BENIGN	96.65	99.99	99.99	98.32	98.29	98.32
	LOW	84.17	98.53	86.08	97.14	85.11	91.35
	HIGH	99.69	96.63	95.23	97.86	97.41	98.16

Table 10. Balanced accuracy of proposed feature selection vs. conventional methods using EXT model

SN	Feature Selection Method	NOF	D1 (%)	D2 (%)	D3 (%)	D4 (%)	Model	Time (Sec.) for feature selection
1	Information Gain	32	76.23	72.57	73.71	73.66	EXT	94.76
2	Logistic Regression	27	71.73	89.53	89.58	89.57	EXT	4.53
3	Variance Threshold	18	82.74	80.71	81.27	81.32	EXT	0.55
4	Random Forest Importance	21	82.07	79.97	80.54	80.56	EXT	174.80
5	Extra Tree Classifier Importance	17	81.96	79.85	80.36	80.37	EXT	16.78
6	Base Line 78 Features	78	82.18	80.06	80.78	80.71	EXT	-
7	Proposed	58	96.12	96.56	96.67	96.66	EXT	0.49

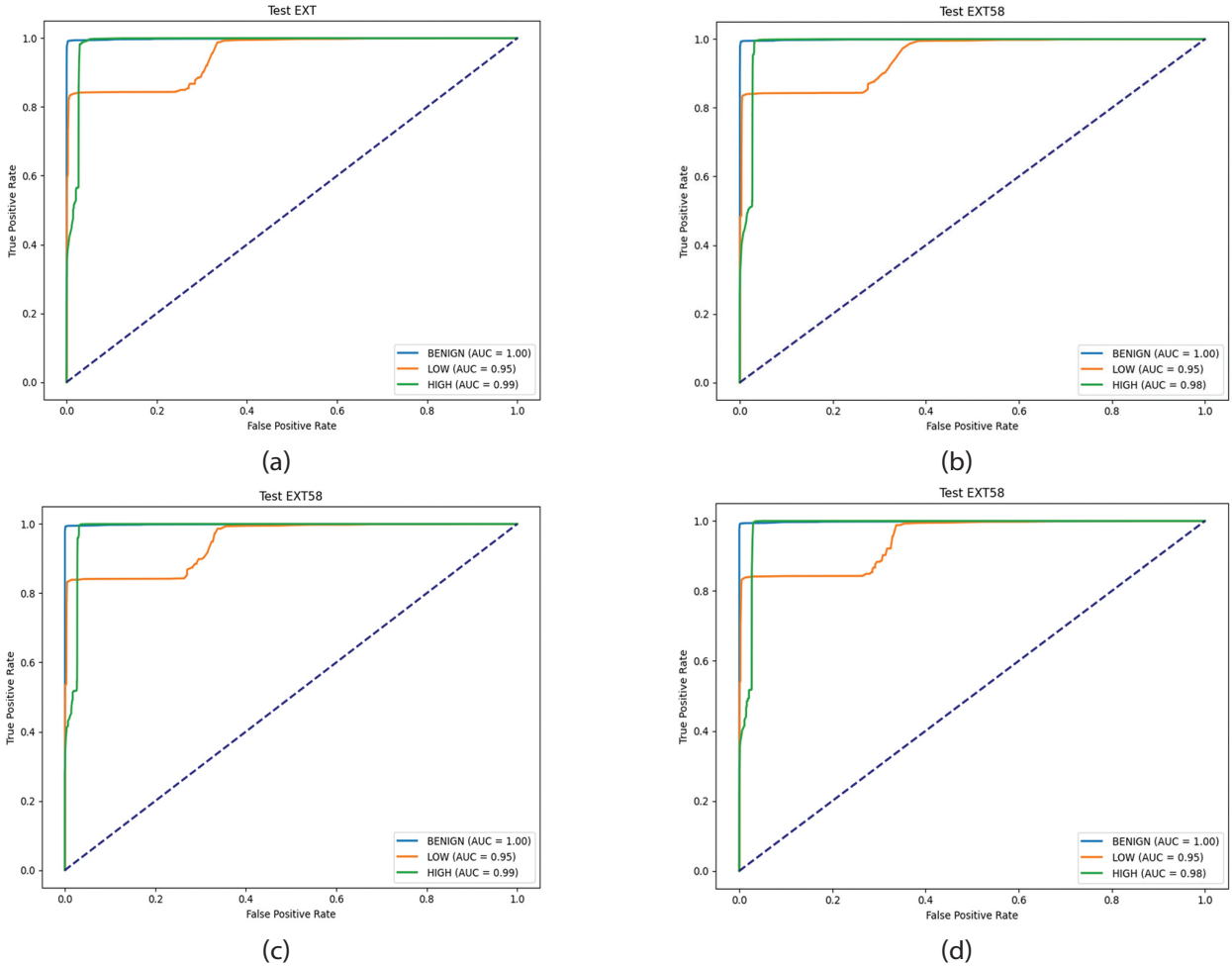


Fig. 2. (**a**) ROC curve of EXT model for test dataset D1 (**b**) ROC curve of EXT model for test dataset D2 (**c**) ROC curve of EXT model for test dataset D3 (**d**) ROC curve of EXT model for test dataset D4

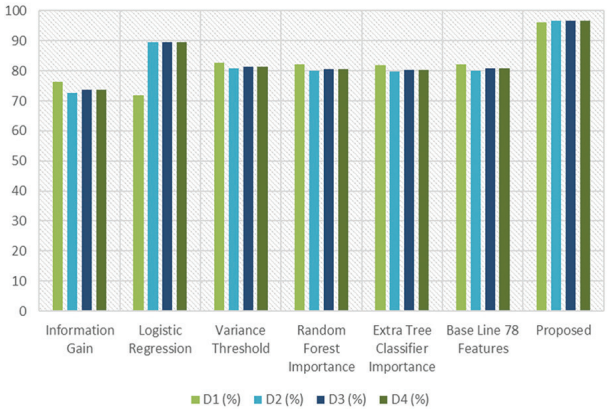


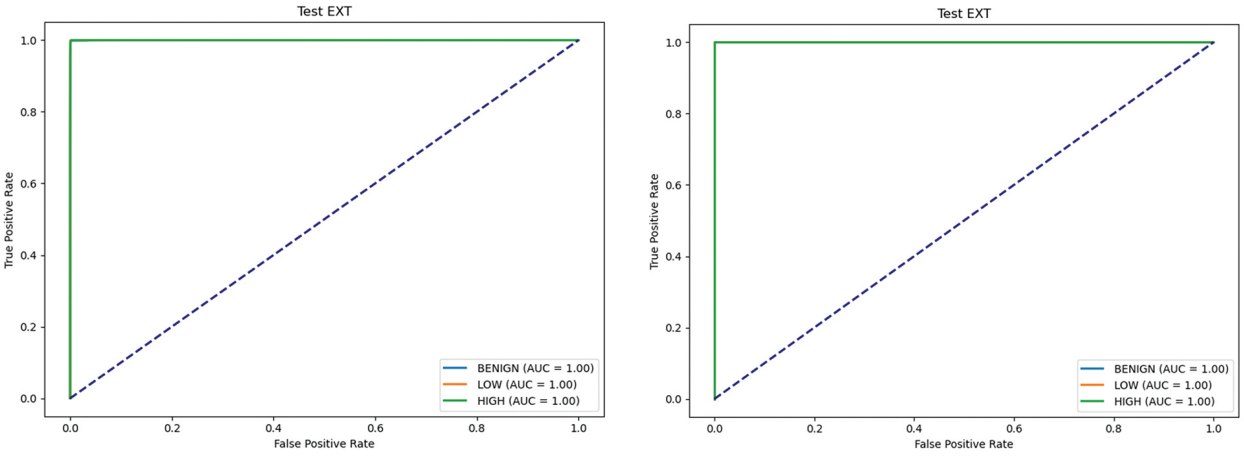
Fig. 3. Comparison of proposed feature selection vs. Conventional methods using EXT model

STATE-OF-THE-ART COMPARISON WITH EXISTING WORKS:

Table 11 compares the proposed method with state-of-the-art existing systems over the four test datasets (D1, D2, D3, and D4). The existing system [27] showed the balanced accuracy (97.16%) is higher than the proposed method over the D4 test dataset. However, this method demonstrated inconsistent performance over the four test datasets. When compared to [27], the proposed system showed consistent performance with 96.50% balanced accuracy on average. Though existing system's feature dimensions are lower compared to the proposed method, the low-rate and high-rate DDoS attack detection performance of the proposed method is superior to the existing methods and systems.

EVALUATION ON CICIDS2017 AND CICIDS2018:

The 58 features obtained using the proposed feature selection method are projected on CICIDS2017 and CICIDS2018 datasets and normalized using min-max scalar. The normalized training dataset is then used to train the EXT classifier and is validated using validation dataset. Then the proposed method is evaluated w.r.t testing dataset using the trained and validated EXT model. The performance of EXT model, over the CICIDS2017 and CICIDS2018 datasets are depicted in Table 12. The experiment results proved that the proposed method showed better balanced accuracies (an average of 99.81% and 99.99%) over the two datasets.



(a) (b)

Fig 4. (**a**) ROC curve of EXT model for CICIDS2017, (**b**) ROC curve of EXT model for CICIDS2018.

Table 11. Comparison of proposed method to existing research studies on DDoS attack detection w.r.t balanced accuracy metric metric over the CICDDoS2019 dataset

SN	Author & Year	FC	Model	Balanced accuracy of testing datasets			
				D1 (%)	D2 (%)	D3 (%)	D4 (%)
1	[6] & 2019	24	EXT	84.93	83.53	83.88	83.98
2	[9] & 2020	10	EXT	64.13	59.52	62.73	62.81
3	[13] & 2021	10	EXT	72.81	86.96	86.50	90.91
4	[14] & 2021	68	EXT	93.15	78.80	78.54	93.32
5	[27] & 2022	40	EXT	96.66	91.40	92.42	97.16
6	[10] & 2022	20	EXT	82.87	82.84	82.50	83.12
7	[28] & 2023	6	EXT	77.27	76.85	76.05	76.12
8	[18] & 2024	14	EXT	64.40	61.59	61.74	61.74
9	[11] & 2024	22	EXT	73.69	71.42	72.38	72.36
10	[15] & 2024	43	EXT	80.25	75.85	77.61	78.07
11	[29] & 2024	10	EXT	73.55	74.72	73.35	73.25
12	Base line	78	EXT	82.19	80.06	80.79	80.72
13	Proposed	58	EXT	96.12	96.56	96.67	96.66

Table 12. Performance metrics of EXT model using CICIDS2017 and CICIDS2018 datasets

Dataset	Class Label	Sns. or Recall (%)	Spe. (%)	Pre. (%)	Acc. (%)	F-Score (%)	BA (%)
CICIDS2017	BENIGN	99.53	99.94	99.85	99.85	99.69	99.74
	LOW	99.88	99.97	99.83	99.96	99.85	99.92
	HIGH	99.93	99.69	99.78	99.83	99.86	99.81
CICIDS2018	BENIGN	100	99.99	99.99	99.99	99.99	99.99
	LOW	100	100	100	100	100	100
	HIGH	99.99	100	100	99.99	99.99	99.99

The ROC curves of EXT model over the CICIDS2017 and CICIDS2018 datasets is depicted in Fig. 4

Results proved that our proposed method also performed better on the other two popular datasets (CICIDS2017 and CICIDS2018). The EXT model performance on the CICIDS2017 dataset in terms of accuracy, precision, recall, and f-score was 99.81%, while on the CICIDS2018 dataset, it was 99.99%. This indicates that our proposed model achieved generalizability.

Here is a summary of the results and key observations:

- Authors & Years: The studies range from 2019 to 2024, with each study offering performance scores for four different datasets (D1, D2, D3, and D4).
- Performance (FC): The number of features selected (FC) varies across studies, from as low as 6 to as high as 78.
- Performance Scores (D1 to D4): The accuracy scores (D1, D2, D3, and D4) vary across various IDS studies, with values generally falling within a range from 59.52% to 97.16%. The highest performance scores tend to appear in more recent studies (2022-2024).
- Baseline: The baseline performance, with 77 features selected, shows moderate accuracy (ranging from 80.06% to 82.19%).
- Proposed Model: The proposed model, with 58 features, achieves very high performance, with accuracy scores of 96.12%, 96.56%, 96.67%, and 96.66% w.r.t D1, D2, D3, and D4 datasets respectively, outperforming the baseline and other state-of-the-art studies.

Key Observations

i. The proposed model demonstrates significant improvement over previous research studies on DDoS attack detection, with higher accuracy across all datasets.

ii. The proposed feature selection method reduced 33.33% of the feature space.

iii. The computational cost of the EXT model using 78 features (baseline features) is 25.5 seconds, whereas using 58 features obtained by proposed method it is just 14.99 sec.

iv. The baseline and earlier studies (2019-2020) generally show lower performance, indicating that newer models, including the proposed one, offer enhanced results.

v. Studies with fewer features (e.g., [6] in 2019 and [9] in 2020) typically show lower accuracy, while more recent studies (especially from 2022 and 2023) exhibit better accuracy, possibly due to improved methodologies or optimizations in feature selection and model performance.

Thus, this summary highlights the significant advantage of the proposed method in comparison to previous work, both in terms of accuracy and feature selection.

6. CONCLUSION

Distributed Denial of Service (DDoS) attacks can severely impact IT services by rendering systems inaccessible to legitimate users. Despite the challenge involved in detection of DDoS attacks, a much more critical challenge is to differentiate LRDDoS traffic from legitimate traffic. In this paper, we propose a feature selection method that leverages the statistical t-Test to improve the IDS ability to predict LRDDoS and HRDDoS attack traffic more accurately and precisely.

The features obtained using the proposed feature selection method aids the machine learning model to detect LRDDoS and HRDDoS attacks at a 95% confidence level. We evaluated the proposed method on CICIDS2017, CICIDS2018, and CICDDoS2019 datasets. To generalize the learning model for intrusion detection, we evaluated the performance of the trained model using four distinct testing datasets obtained using CICDDoS2019 dataset which contains network traffic flows unseen during training and validation phase. For evaluation, we have considered twelve machine learning classifiers. Among all learning models, the Extra Tree (EXT) model has performed better. When these four testing day datasets are used for experimental study, the EXT model has achieved an average accuracy of 96.50%, a precision of 96.58%, and an F-Score of 96.50%. Overall, the EXT model showed an average accuracy of 99.81% and 99.99% on CICIDS2017 and CICIDS2018 datasets respectively. These results indicate that feature set obtained using the proposed feature selection with extra tree learning machine addressed generalizability.

It is also observed that the computational time for finding the feature subset is much lower compared to the conventional methods and that the proposed method shows comparatively better performance in discriminating low-rate DDoS attack, high-rate DDoS attack and benign network traffic.

In this paper, the research work is limited to finding an optimal feature subset based on feature selection using t-Test and integrating t-Test feature selection with

EXT classifier for machine learning. Future research work could focus on improving the accuracy of LRDDoS attacks detection using new feature extraction methods.

7. REFERENCES:

[1] S. Rajagopal, P. P. Kundapur, Hareesha K. S., "Towards Effective Network Intrusion Detection: From Concept to Creation on Azure Cloud", IEEE Access, Vol. 9, 2021, pp. 19723-19742.

[2] U. S. Chanu, K. J. Singh, Y. J. Chanu, "A dynamic feature selection technique to detect DDoS attack", Journal of Information Security and Applications, Vol. 74, 2023, p. 103445.

[3] N. Singh, A. Kaur, "Feature selection for artificial neural network based intrusion detection system", International Journal For Technological Research In Engineering, Vol. 2, No. 11, 2015, pp. 2681-2683.

[4] S. Das, S. Saha, A. T. Priyoti, E. K. Roy, F. T. Sheldon, A. Haque, "Network Intrusion Detection and Comparative Analysis Using Ensemble Machine Learning and Feature Selection", IEEE Transactions on Network and Service Management, Vol. 19, No. 4, 2023, pp. 4821-4833.

[5] Kurniabudi, D. Stiawan, Darmawijoyos, M. Y. Bin Idris, A. M. Bamhdi, R. Budiarto, "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection", IEEE Access, Vol. 8, 2020, pp. 132911-132921.

[6] I. Sharafaldin, A. H. Lashkari, S. Hakak, A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy", Proceedings of the International Carnahan conference on Security Technology, Chennai, India, 1-3 October 2019, pp. 1-8.

[7] S. Li, J. Xu, P. Liu, X. Li, P. Wang, X. Jin, "Truncated Lanczos-TSVD: An Effective Dimensionality Reduction Algorithm for Detecting DDoS Attacks in Large-Scale Networks", IEEE Transactions on Network Science and Engineering, Vol. 11, No. 5, 2024, pp. 4689-4703.

[8] M. Hajimaghsoodi, R. Jalili, "RAD: A Statistical Mechanism Based on Behavioral Analysis for DDoS Attack Countermeasure", IEEE Transactions on Information Forensics and Security, Vol. 17, 2022, pp. 2732-2745.

[9] Y. Jia, F. Zhong, A. Alrawais, B. Gong, X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism Against IoT DDoS Attacks", IEEE Internet of Things Journal, Vol. 7, No. 10, 2022, pp. 9552-9562.

[10] A. Maheshwari, B. Mehraj, M. S. Khan, M. S. Idrisi, "An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment", Microprocessors and Microsystems, Vol. 89, 2022, p. 104412.

[11] S. Mahdavifar, A. A. Ghorbani, "CapsRule: Explainable Deep Learning for Classifying Network Attacks", IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, No. 9, 2024, pp. 12434-12448.

[12] E. Q. Effah, E. O. Osei, M. Dorgbefu Jnr, A. Tetteh, "Hybrid Approach to Classification of DDoS Attacks on a Computer Network Infrastructure", Asian Journal of Research in Computer Science, Vol. 17, No. 4, 2024, pp. 19-43.

[13] G. C. Amaizu, C.I. Nwakanma, S. Bhardwaj, J. M. Lee, D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks", Computer Networks, Vol. 188. 2021, p. 107871.

[14] A. E. Cil, K. Yildiz, A. Buldu, "Detection of DDoS attacks with feed-forward based deep neural network model", Expert Systems with Applications Vol. 169, 2021, p. 114520.

[15] A. A. Najar, S. M. Naik, "A Robust DDoS Intrusion Detection System Using Convolutional Neural Network", Computers and Electrical Engineering, Vol. 117, 2024, pp. 1-19.

[16] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, S. Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification", IEEE Access, Vol. 9, 2021, pp. 146810-146821.

[17] M. A. Ferrag, L. Shu, H. Djallel, K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0", Electronics, Vol. 10, No. 11, 2021, p. 1257.

[18] A. A. Alashhab, M. S. Zahid, B. Isyaku, A. A. Elnour, W. Nagmeldin, A. Abdelmaboud, T. A. A. Abdullah, U. D. Maiwada, "Enhancing DDoS Attack Detection and Mitigation in SDN Using an Ensemble Online Machine Learning Model", IEEE Access, Vol. 12, 2024, pp. 51630-51649.

[19] O. Barut, Y. Luo, P. Li, T. Zhang, "R1DIT: Privacy-Preserving Malware Traffic Classification with Attention-Based Neural Networks", IEEE Transactions on Network and Service Management, Vol. 20, No. 2, 2023, pp. 2071-2085.

[20] Z. Li, H. Jin, D. Zou, B. Yuan, "Exploring New Opportunities to Defeat LRDDoS Attack in Container-Based Cloud Environment", IEEE Transactions on Parallel and Distributed Systems, Vol. 31, No. 3, 2020, pp. 695-706.

[21] M. F. Saiyed, I. Al-Anbagi, "Flow and unified information-based DDoS attack detection system for multi-topology IoT networks", Internet of Things, Vol. 24, 2023, p. 100976.

[22] R. Manthena, R. Vangipuram, "Integrating Machine Learning and T-tests to Optimize Distributed Denial of Service Attacks Detection", International Journal of Intelligent and Engineering Systems, Vol. 17, No. 6, 2024, pp. 1023-1043.

[23] M. F. Saiyed, I. Al-Anbagi, "A Genetic Algorithm-and t-Test-Based System for DDoS Attack Detection in IoT Networks", IEEE Access, Vol. 12, 2024, pp. 25623-25641.

[24] I. Sharafaldin, A. Gharib, A. H. Lashkari, A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset", Software Networking, Vol. 1, 2018, pp. 177-200.

[25] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", Proceedings of the 4th International Conference on Information Systems Security and Privacy, Portugal, 22-24 January 2018, pp. 108-116.

[26] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, "An evaluation framework for intrusion detection dataset", Proceedings of the International Conference on Information Science and Security, Pattaya, Thailand, 19-22 December 2026, pp. 1-6.

[27] D. Akgun, S. Hizal, U. Cavusoglu, "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity", Computers and Security, Vol. 118, 2022, p. 102748.

[28] R. K. Batchu, H. Seetha, "On improving the performance of DDoS attack detection system", Microprocessors and Microsystems, Vol. 93, 2022, p. 104571.

[29] D. M. Sharif, H. Beitollahi, "Detection of application-layer DDoS attacks using machine learning and genetic algorithms", Computers and Security, Vol. 135, 2023, p. 103511.

Abbreviations:

IG	Information Gain
LR	Logistic Regression
RFFI	Random Forest Feature Importance
EXT	Extra Tree Classifier
VT	Variance Threshold
FC	Feature Count
SN	Serial Number

Appendix:
List of 58 features selected using proposed feature selection

SN	Feature Name	SN	Feature Name	SN	Feature Name	SN	Feature Name
1	Total Fwd Packets	16	Flow IAT Max	31	Packet Length Std	46	Active Mean
2	Total Backward Packets	17	Fwd IAT Mean	32	Packet Length Variance	47	Active Std
3	Total Length of Fwd Packets	18	Fwd IAT Max	33	SYN Flag Count	48	Active Max
4	Fwd Packet Length Max	19	Bwd IAT Total	34	RST Flag Count	49	Active Min
5	Fwd Packet Length Min	20	Bwd IAT Mean	35	ACK Flag Count	50	Idle Mean
6	Fwd Packet Length Mean	21	Bwd IAT Std	36	URG Flag Count	51	Idle Std
7	Fwd Packet Length Std	22	Bwd IAT Max	37	CWE Flag Count	52	Idle Min
8	Bwd Packet Length Max	23	Bwd IAT Min	38	Down/Up Ratio	53	Inbound
9	Bwd Packet Length Min	24	Fwd PSH Flags	39	Average Packet Size	54	Flow Duration
10	Bwd Packet Length Mean	25	Bwd Header Length	40	Avg Fwd Segment Size	55	Fwd IAT Total
11	Bwd Packet Length Std	26	Fwd Packets/s	41	Avg Bwd Segment Size	56	Fwd IAT Std
12	Flow Bytes/s	27	Bwd Packets/s	42	Init_Win_bytes_forward	57	Fwd Header Length
13	Flow Packets/s	28	Min Packet Length	43	Init_Win_bytes_backward	58	Idle Max
14	Flow IAT Mean	29	Max Packet Length	44	act_data_pkt_fwd		
15	Flow IAT Std	30	Packet Length Mean	45	min_seg_size_forward		

Echocardiographic Left Ventricular Segmentation Using Double-layer Constraints on Spatial Prior Information

Original Scientific Paper

Jin Wang
[1]College of Computing,
Informatics and Mathematics, Shah Alam, Malaysia
[2]Department of Electrical Engineering,
Taiyuan Institute of Technology, Taiyuan, China
wangjin@studyedus.cn

Sharifah Aliman
Universiti Teknologi MARA,
College of Computing, Informatics and Mathematics
Shah Alam, Malaysia
sharifahali@uitm.edu.my

*Corresponding author

Shafaf Ibrahim*
Universiti Teknologi MARA,
College of Computing, Informatics and Mathematics
Shah Alam, Malaysia
email: shafaf2429@uitm.edu.my

Yanli Tan
Universiti Teknologi MARA,
College of Computing, Informatics and Mathematics
Shah Alam, Malaysia
email: tanyanli@studyedus.cn

Abstract *– Real-time segmentation of echocardiograms is of great practical significance for doctors' clinical diagnosis. This paper addresses the existing echocardiogram segmentation models' pursuit of high segmentation accuracy in insufficient training data, which leads to high model complexity and low learning efficiency. This paper fully exploits the spatial prior characteristics of the image itself. It proposes an echocardiographic left ventricular segmentation algorithm that utilizes double-layer constraints of prior information on spatial anatomical structures. The algorithm is based on the following two principles. Firstly, the segmentation model is initialized using a self-supervised sorting model based on the spatial anatomy to fully learn the orderly image features of the left ventricular spatial anatomy and achieve same-domain transfer of images, allowing the segmentation network to learn segmentation information more effectively; Secondly, the segmentation network is subjected to mask shape constraints, and the output space is limited by imposing anatomical shape priors to expand the global training goals of the CNN model. Finally, the algorithm proposed in this paper was verified using three classic segmentation models. The experimental results showed that on the public echocardiography dataset CETUS (Challenge on Endocardial Three-dimensional Ultrasound Segmentation), compared with the classic Resnet, Unet, and VGG segmentation models, the double-layer constrained segmentation model that introduces prior features has increased the segmentation accuracy (Dice index) by 5.6%, 4.9%, and 4.8%, respectively. The MIOU (Mean Intersection over Union) index increased by 7%, 5.5%, and 6.8%, respectively, demonstrating robustness to slice misalignment.*

Keywords: *echocardiographic segmentation, deep learning, spatial prior, CETUS*

Received: January 20, 2025; Received in revised form: May 1, 2025; Accepted: May 2, 2025

1. INTRODUCTION

Due to the portability, cost-effectiveness, non-radiation, and real-time nature of echocardiography, accurate segmentation of the left ventricle from ultrasound images can help doctors with less clinical experience to analyze cardiac images conveniently and accurately to serve actual clinical diagnosis [1]. However, due to the ultrasonic imaging mechanism, echocardiography has characteristics such as considerable dynamic noise, low image contrast, and loss of edges [2]. This makes achieving fully automated real-time segmentation of the left ventricle in echocardiography a well-known challenge. In recent years, the most advanced deep learning technology has been used for cardiac image segmentation to automatically measure size and functional assessment of the left ventricle, effectively improving the diagnostic efficiency of echocardiography

[3, 4]. However, it also faces some limitations. For example, deep learning networks depend on the learning capabilities and results of a large amount of annotated data and powerful storage computing units; there are currently very few publicly available datasets, and the scale is difficult to meet research needs.

To solve this problem, some researchers have proposed deep network fusion algorithms to improve segmentation accuracy and convergence speed, especially when training datasets are limited. Literature [5-8] combines deep learning networks with deformable models, and features extracted by trained deep neural networks are used instead of handcrafted features to improve accuracy and robustness. Literature [9] proposed a method combining convolutional neural networks and ASM (Active Shape Model) to achieve automatic segmentation of the left ventricle of echocardiograms. It uses the Nakagami distribution to integrate the shape prior of the image to provide preprocessing classification. The results show that the segmentation accuracy and convergence are improved at the same time. Literature [10-11] uses generative adversarial networks to make segmentation masks, and image frame structures correspond one-to-one, increasing the number of training samples and improving segmentation accuracy. Literature [12] fuses two convolutional neural networks, YOLOv7 and U-Net, to automatically segment echocardiographic images. Some researchers have effectively utilized unlabeled data and proposed semi-supervised and unsupervised deep learning methods to improve the segmentation performance of the model by combining multiple strategies [13-16].

At this stage, deep network fusion algorithms have performed well in left ventricular segmentation tasks on ultrasound cardiac images. Algorithms that introduce prior information about intensity, shape, time, topology, and atlas show obvious advantages in improving the accuracy and efficiency of segmentation. However, most deep learning networks are based on feature classification of pixel sets, ignore the structural characteristics and related prior knowledge of echocardiograms, and lack the learning of global features related to segmentation target structures, resulting in limited feature learning capabilities of the model. Some researchers realize the importance of prior knowledge, such as image anatomy and imaging information, and try to utilize prior features better to optimize deep learning models. Literature [17] incorporates the perceptual similarity information between the generated and original frames into the segmentation model as prior knowledge. It uses unlabeled data for semi-supervised learning to improve segmentation performance. Literature [18] introduces a prior information encoding module, and the results show that the accuracy of this method is close to the segmentation result of the current optimal nnU-Net, with the convergence speed increased by 145%. Literature [19] proposes a Unet network model (MCCT-Unet) based on a multi-channel cross-fusion transformer. By effectively combining deep information with shallow information in the encoding stage, the segmentation performance of the network is improved. Literature [20] constructs a multi-fusion residual attention U-Net (MURAU-Net) automatic segmentation model by strengthening the connection of spatial features. Literature [21] introduced spatial and temporal prior features and achieved excellent segmentation results through deep network fusion. These research results demonstrate the effectiveness of introducing prior left ventricular anatomical structure features in improving the deep network fusion algorithm's segmentation accuracy and convergence speed.

This paper proposes a segmentation algorithm for the left ventricle in echocardiography using double-layer constraints on the prior information of spatial anatomy. The algorithm uses the orderliness of the anatomical position of the left ventricle to construct a self-supervised sorting model to initialize the segmentation network. The shape prior is incorporated into the mask part of the segmentation network to constrain the output space, reduce the extraction depth of the feature layer of the segmentation network, and improve the segmentation performance. Experimental results show that with relatively limited training data, the model has achieved significant improvements in both the Dice and MIOU indicators of segmentation accuracy, fully verifying its excellent performance and practicality. Specifically, the model brings the following benefits: (1) By utilizing the strong correlation between different positions of the image simultaneously, efficient model pre-training is achieved based on same-domain transfer, effectively solving the problem of insufficient generalization ability in different domain transfer learning. (2) By analyzing the imaging characteristics of echocardiography and the anatomical structure of the left ventricle and taking advantage of the natural order of the short-axis section of the left ventricle in spatial position, a self-supervised sorting model is constructed, aiming to explore a reasonable model initialization method to improve the performance and efficiency of the model. (3) The short-axis section of the left ventricle presents a fixed position relationship from top to bottom at any time in the cardiac cycle. This spatial anatomical prior knowledge is not only applicable to echocardiography. Still, it can also be extended to image segmentation of other modalities, providing a new way to solve the training requirements of medical image segmentation problems.

2. METHOD

The overall framework of the echocardiographic left ventricular segmentation algorithm using double-layer constraints of prior information on spatial anatomy is shown in Fig. 1. It mainly consists of two parts: a self-supervised ranking model and a shape-constrained image segmentation model. The self-supervised rank-

ing model aims to learn the anatomical prior features of the image. It uses pre-training of the self-supervised ranking model to initialize the segmentation network. It encourages the model to learn more about the segmentation task by obtaining anatomical position features when training the segmentation task. This information helps improve the prediction accuracy and convergence speed of segmentation models. The shape-constrained image segmentation model incorporates the shape prior of the label structure into the deep learning network. In this way, by constraining the training process of the neural network, the network is guided to make more anatomically meaningful predictions. This study uses the prior spatial structure features and integrates high-dimensional and low-dimensional features into the deep learning segmentation network simultaneously. It is expected to achieve better segmentation results, reduce the segmentation network learning model's complexity, and reduce the scale of the training dataset.

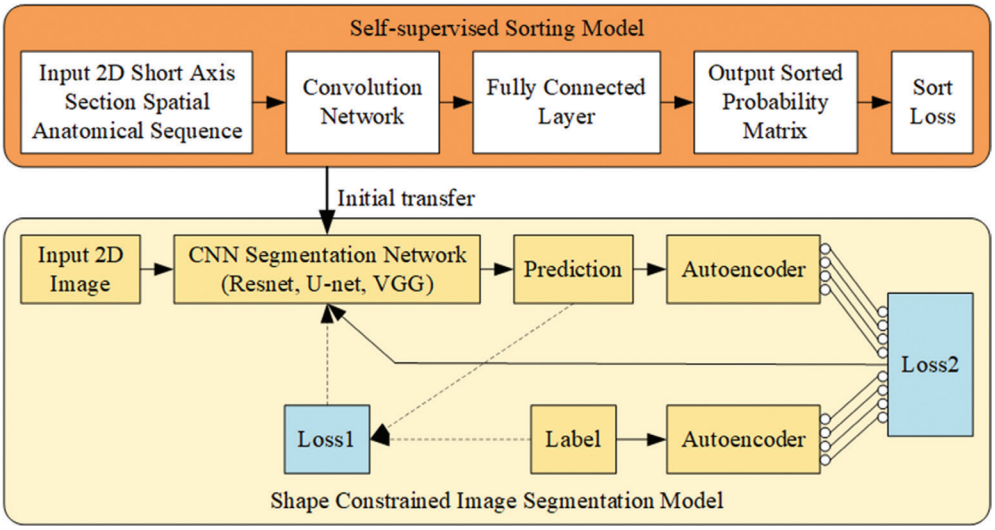


Fig. 1. Overall algorithm framework

2.1. IMAGE DESCRIPTION

The dataset for the pre-training model and the shape-constrained segmentation model are both from the public CETUS dataset. CETUS comprises 45 3D echocardiography sequences, which are evenly distributed in three different subgroups: healthy subjects, patients with muscle damage, and patients with dilated cardiomyopathy, and has been widely verified for its superior algorithm [22]. First, the 3D volume data is sliced along the short axis to obtain 2D slices. Second, 2D slices of the left ventricle from the apex to the base were obtained manually, and other parts were removed.

Finally, the acquired left ventricular short-axis slice sequence is normalized to ensure that all data have the exact spatial resolution along the short axis, and the left ventricular short-axis slice data are sampled according to the sorting scale, 10,658 2D short-axis slice images were obtained, of which 8,276 were used for training and 2,382 for testing.

The self-supervised sorting pre-training model uses slice sequences as input data. The input sequence is randomly shuffled to prevent the problem of the absolute position of the left ventricular 2D slice feature map. The shape constraint segmentation model uses a single 2D slice as input for subsequent segmentation tasks.

2.2. SELF-SUPERVISED RANKING MODEL BASED ON SPATIAL PRIOR

The puzzle problem trains a deep learning network to identify the components of the target [23]. This paper analyzes echocardiography's imaging characteristics and the left ventricle's short-axis structure based on this concept. Its spatial anatomy resembles a cone, with these slices appearing in a fixed order from top to bottom in spatial position. This paper takes advantage of the natural spatial ordering of short-axis slices of the left ventricle to build a self-supervised sorting pre-training model, initializes the parameters of the segmentation network, and effectively integrates spatial anatomy prior knowledge into the Segmentation network for deep learning. As shown in Fig. 2, the self-supervised sorting pre-training model includes four parts: input, backbone feature extraction, output, and loss module. The input is an echocardiographic left ventricular short-axis slice aerial anatomical structure sequence, a set of short-axis slices from the apex to the base. The number of slices N defines the sorting scale ($20>N>2$), based on the selected N Slices, with different sorting for different inputs. The backbone feature extraction module is a general convolutional network structure. In this paper, three classical structures, VGG [24], Unet [25], and Resnet [26], are chosen for the performance comparison of self-supervised sorting models. The output is an $N*N$-dimensional probability matrix, and the loss module is mainly used to evaluate the accuracy of N-slice sorting.

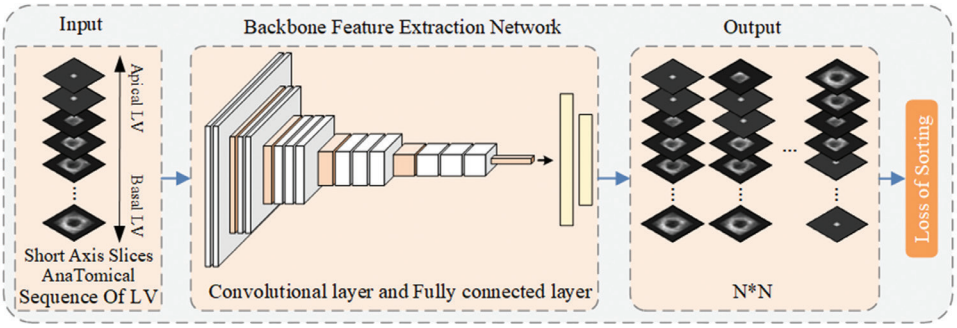


Fig. 2. Self-supervised ranking pre-training model based on spatial prior

The loss module is mainly used to evaluate the sorting accuracy of echocardiographic left ventricular short-axis section image sequences. The optimization goal of the spatial anatomical structure self-supervised sorting task is the multi-level Softmax loss function. The specific loss function is as follows:

$$\text{Loss} = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N} y_{ji} p_{ji} \tag{1}$$

Among them, the formula of p_{ji} is as follows:

$$p_{ji} = \frac{e^{Z_{ji}}}{\sum_{c}^{N} Z_{jc}} \tag{2}$$

N represents the number of categories, y_{ji} p_{ji} indicating that the j-th image belongs to the i-th category, y_{ji}=1 indicating that the j-th image belongs to the i-th category, y_{ji}=0indicating that the j-th image does not belong to the i-th category, p_{ji} indicating that the input j-th image belongs to the i-th category Probability, Z represents the input of the softmax activation function.

2.3. SHAPE-CONSTRAINED IMAGE SEGMENTATION MODEL

The shape-constrained segmentation model implements the shape constraint function by adding convolutional autoencoding, applying anatomical shape priors to the predicted images of the segmentation model to constrain the output space, expanding the global training goals of the CNN segmentation model, and using two loss functions to adjust the feedback of the segmentation network. This approach improves sub-pixel segmentation accuracy by training an upsampling layer with high-resolution ground truth maps.

Fig. 3 shows the structure of the shape-constrained segmentation model. The convolutional autoencoding constraint model AE (autoencoder) [27] is integrated into the basic segmentation network based on deep learning and predicts image class label shape constraints. It fully uses the anatomical low-dimensional features of 2D echocardiographic left ventricular images to improve model segmentation accuracy.

The basic segmentation network uses a cross-entropy loss function to run predictions at the single-pixel level, since the backpropagation gradient is only parameterized by the individual probability divergence term at the pixel level, it provides little global context. It cannot guarantee the consistency of the overall anatomical shape. Class label prediction obtains the parameters and underlying structure of the lower-dimensional segmentation by performing AE-based nonlinear low-dimensional projections of the predicted image and the true label [28]. This paper builds a segmentation network with a double-constraint loss function to obtain more global information and local features, thereby improving the performance of the segmentation model.

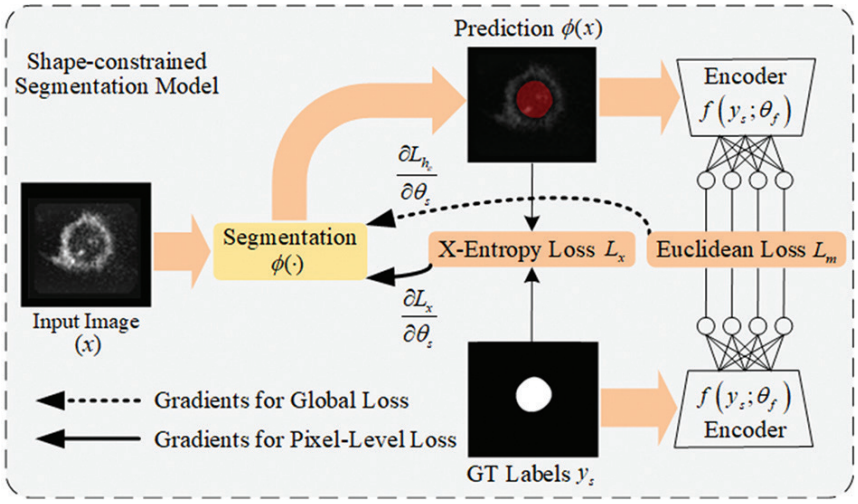


Fig. 3. Shape-constrained segmentation network model

The shape Constrained Segmentation Network via Cross-Entropy Loss of Basic CNN Segmentation Network L_x (ϕ(x; θ_s), y_s)and the linear combination of the shape loss L_m from AE to train the objective function, as shown in Equation 3. ω is the weight of the convolution filter of the segmentation network. The third term corresponds to weight decay, which limits the number of free parameters in the model to avoid over-fitting, weight decay to restrict the number of free parameters in the model to avoid overfitting. θ_s represents all trainable parameters of the segmentation model, θ_f represents all trainable parameters of the AE model, which are updated during training. The coupling parameters λ_1 and λ_2 determine the weight of the shape loss and the weight decay terms used in the training. In this equation, the second term L_m ensures that the generated segmentations are in a similar low-dimensional space as the ground-truth labels.

$$L_m = \left\| f\left(\phi(x);\theta_f\right) - f\left(y;\theta_f\right)\right\|_2^2$$
$$\min_{\theta_s}\left(L_x\left(\phi\left(x;\theta_s\right),y_s\right) + \lambda_1 \cdot L_{h_e} + \frac{\lambda_2}{2}\|\omega\|_2^2 \right) \quad (3)$$

2.4. PERFORMANCE EVALUATION

To measure the accuracy of echocardiographic left ventricular segmentation, this paper used three different metrics, namely Dice, two-dimensional HD (Hausdorff Distance), and MIOU to evaluate the segmentation accuracy [29][30][31]. Let U={u_1, u_2,…u_m} be the prediction area, and R={r_1, r_2,…r_m } be the reference area.

Dice is a measure of the similarity between two sets. It evaluates the similarity between the network prediction structure and the human annotation result. The segmentation task classifies the pixels in the image. Set similarity evaluates the similarity between two contours and generally requires the index to be greater than 0.7 for the segmentation effect to be considered relatively good.

$$\text{Dice}=\frac{2|U\cap R|}{|U|+|R|} \quad (4)$$

HD is the maximum distance from one set to the nearest point in another set. This distance is directional, meaning that $h_{(U, R)}$ is not equal to $h(_{R, U})$. H takes the larger of the two distances. A smaller value indicates a higher degree of similarity for parameters sensitive to differences in location information. The calculation formula is as follows:

$$h(\text{R,U}) = \max_{u\in U}\left\{\min_{r\in R}\|\text{u-r}\|\right\} \quad (5)$$

$$h(\text{U,R}) = \max_{r\in R}\left\{\min_{u\in U}\|\text{r-u}\|\right\} \quad (6)$$

MIOU is the average intersection and union ratio, including the heart and background areas. IOU is used to test the overlapping area of each category, calculated as the intersection area of a specific category divided by the union area of a particular category. The MIOU is calculated as the sum of the Io of all categories divided by the total number of categories.

$$\text{MIOU}=\frac{1}{2}\times\left(\frac{n_{ff}}{t_f+n_{bf}}+\frac{n_{bb}}{t_b+n_{fb}}\right) \quad (7)$$

Among them, n_{ff} represents the number of correctly classified foreground pixels, t_f represents the total number of pixels belonging to the foreground, n_{bf} represents the number of incorrectly classified background pixels, n_{bb} represents the number of correctly classified background pixels, t_b represents the total number of pixels belonging to the background, and n_{fb} represents the number of misclassified foreground pixels.

3. RESULTS AND DISCUSSION

During model training, the classic CNN network structures, including VGG [24], Unet [25], and Resnet [26], were selected for medical image segmentation tasks. The echocardiographic left ventricle segmentation algorithm using double-layer constraints of spatial prior information was verified to be effective. The model was implemented using the PyTorch deep learning framework, with Kaggle selected as the running platform.

3.1. RESULTS AND ANALYSIS OF SELF-SUPERVISED SORTING PRE-TRAINING MODEL

First, the feasibility of the self-supervised ranking model based on spatial priors for different deep-learning networks was verified under various input image sequence sizes. During the self-supervised model training process, the hyperparameters Epoch=20, batch size=16, and learning rate=1e-5 were set, with all parameter size limits chosen based on tracking and error to provide higher accuracy. VGG [24], Unet [25], and Resnet [26] were used as the basic network structures for deep learning, and the performance of the self-supervised model was evaluated through average ranking accuracy. Table 1 shows the ranking accuracy of the self-supervised ranking model on the test set using different basic network structures and input image sequence sizes. Experimental results indicate that for sorting tasks with a sorting vector of less than 10, the sorting model can achieve an accuracy higher than 50%, which validates the results to a certain extent. This paper illustrates the rationale behind constructing a feasible sorting task for a self-supervised sorting model based on spatial anatomical priors. It demonstrates that developing a self-supervised sorting model has significant potential for application in pre-training left ventricular segmentation tasks.

Table 1. Accuracy of different deep learning network structures

Method	Accuracy at different sorting scales								
	2	3	4	5	6	7	8	9	10
Unet	0.80	0.77	0.72	0.71	0.70	0.68	0.62	0.55	0.53
VGG	0.85	0.80	0.76	0.74	0.73	0.64	0.62	0.61	0.58
ResNet	0.88	0.85	0.83	0.80	0.73	0.70	0.64	0.63	0.61

Next, the effect of self-supervised ranking based on spatial anatomy priors on the pre-trained segmentation model is verified. The Models used for comparison and verification include: Resnet, Resnet_S; Unet, Unet_S; VGG, VGG_S, where S represents self-supervised sorting based on spatial anatomy prior, which is used for the pre-training of segmentation models.

Given the performance analysis results of the pre-training model, the input scale of the ranking model in the experiment utilized 8 2D image slices, and the ranking output was an 8×8 probability matrix. After the training of the ranking task is completed, the model that performed best on the test set is selected, and the segmentation network is initialized. The ranking model and segmentation network use the same base network to facilitate simple and effective model parameter migration. The initial learning rate (Lr) is set to 0.01, the minimum learning rate is 1e-5, the optimizer used is AE, the batch size is 8, and the limits of all parameter sizes are selected based on tracking and error to improve accuracy. The model is evaluated through the Loss_epoch curve of the training set, and the experimental results are shown in Fig. 4.

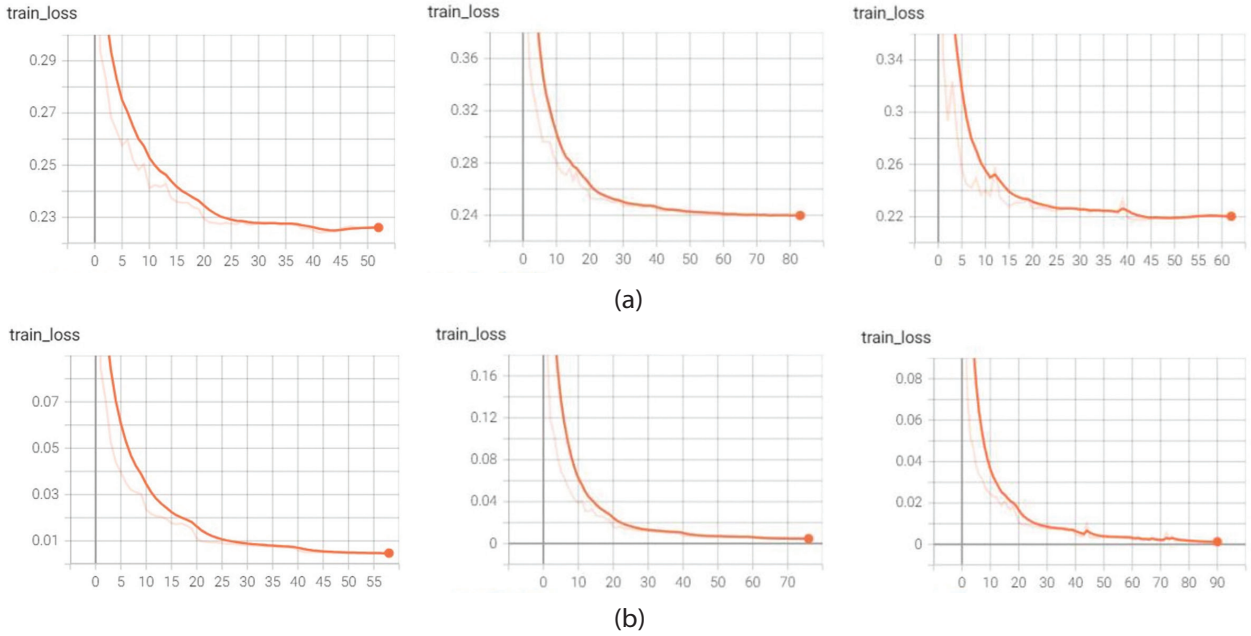


Fig. 4. Loss_epoch curves at different model training stages. (**a**) Resnet(loss-epoch), Unet(loss-epoch) and VGG(loss-epoch), (**b**) Resnet_S(loss-epoch), Unet_S(loss-epoch) and VGG_S(loss-epoch)

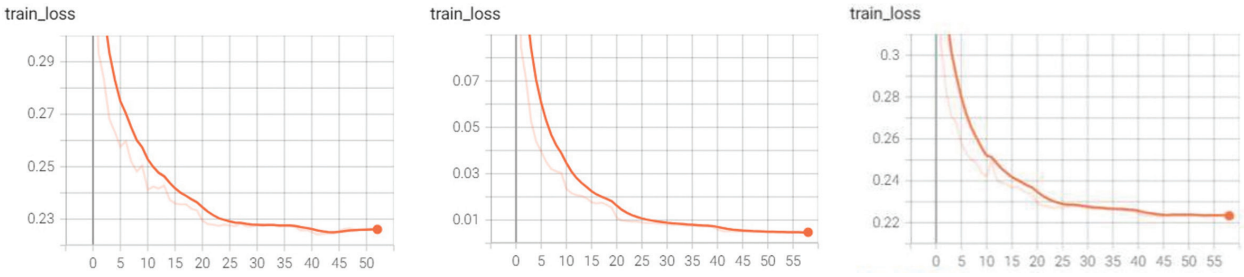
Fig. 4 shows the Loss-Epoch curves of different base network model training stages. By comparison, it is found that the segmentation models Resnet_S, Unet_S, and VGG_S, which are based on spatial anatomy prior self-supervised sorting pre-training, exhibit faster convergence capabilities. The reason is that during pre-training, the model learns effective prior information naturally ordered in the spatial dimensions of left ventricular ultrasound images. When the segmentation task training is completed, it will learn more task-related information and perform segmentation based on the learned prior features, which helps improve segmentation accuracy and speeds up training convergence.

3.2. RESULTS AND ANALYSIS OF THE DOUBLE-LAYER CONSTRAINT SEGMENTATION NETWORK MODEL

First, the impact of adding anatomical constraints on the convergence performance of the segmentation network was verified during the model training phase. Resnet was selected as the representative backbone network for evaluation in the experiment, and the performance of the segmentation network before and after the introduction of anatomical constraints was comparatively analyzed. Here, S represents the use of pre-training, and L represents the introduction of anatomical constraints. The Model parameter settings included a batch size of 10 and a learning rate of 2e-4, which was reduced to 1e-5 in the later stage of training. The loss function used was the cross-entropy loss function. Fig. 5 shows the Loss_epoch curve of the model training stage represented by the Resnet base network. It was found that pre-training effectively improves the convergence speed of the model. After the anatomical constraints are introduced, since two losses limit the model, it increases the learning difficulty of the model, making the convergence speed slower and consistent with the convergence of the model without pre-training.

Next, a comprehensive performance evaluation of the double-constraint left ventricular segmentation model based on self-supervised sorting pre-training proposed in this paper was conducted from two aspects: segmentation accuracy and model segmentation effect. The model's accuracy is evaluated through three indicators: Dice, HD, and MIOU. The experiment

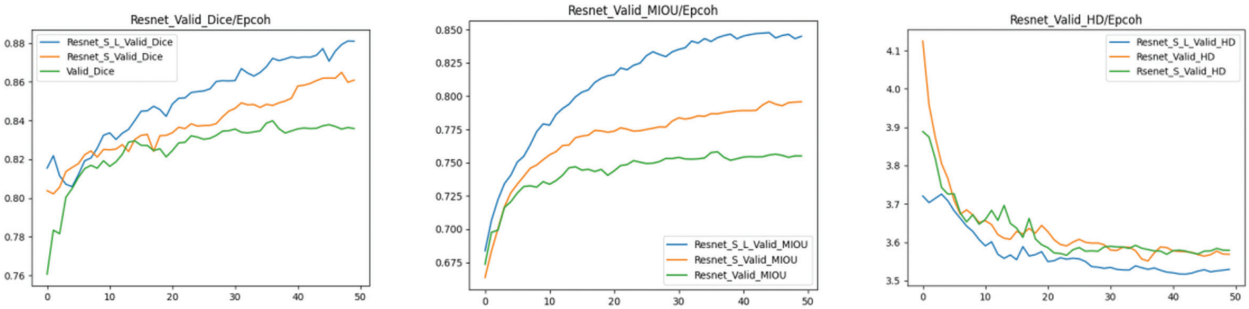
was implemented using the PyTorch deep learning framework, and the running platform used was Kaggle. The test set comprises 2382 2D slices from 36-45 patients in the CETUS dataset. 9 segmentation models, including Resnet, Resnet_S, Resnet_S_L, Unet, Unet_S, Unet_S_L, VGG, VGG_S, and VGG_S_L were tested. The Segmentation effects were intuitively assessed through qualitative visual comparisons of different image qualities, segmentation masks produced by different models, and corresponding ground truth values.



(a) (b) (c)

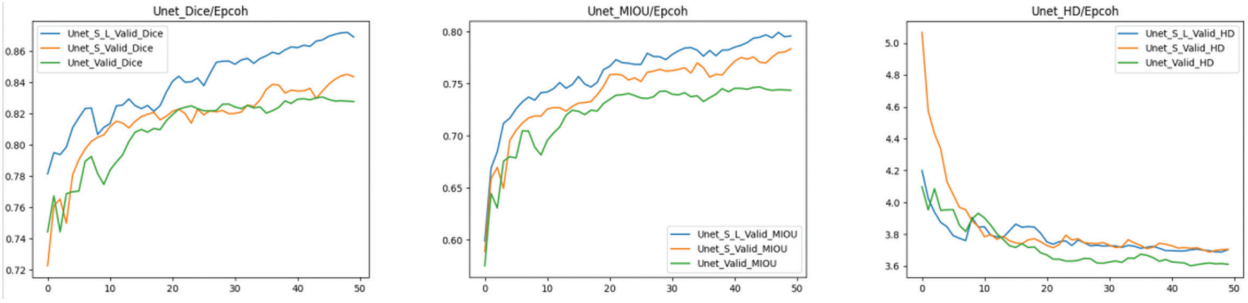
Fig. 5. Loss-epoch curves of Resnet, Resnet_S, and Resnet_S_L models. (**a**) Resnet(Loss-epoch), (**b**) Resnet_S(Loss-epoch), (**c**) Resnet_S_L(Loss-epoch)

Three pre-training models, Resnet_S, Unet_S, and VGG_S, refer to the performance analysis results of the pre-training models, using eight image slices as input, with a batch size of 16, a learning rate of 1e-5, and an epoch count of 10. The loss function used is multi-level Softmax. After completing the self-supervised sorting task, the model with the best performance on the test set is selected as the pre-training model for the segmentation task. The parameters for the three basic network models of Resnet, Unet, and VGG are set with a batch size of 10 and a learning rate of 2e-4, which is reduced to 1e-5 in the later stage of training, with an epoch count of 50. The loss function used is the cross-entropy loss function. Based on anatomical prior pre-training, the shape-constrained segmentation models Resnet_S_L, Unet_S_L, and VGG_S_L proposed in this paper incorporate a linear combination of shape-constrained Loss1 and Loss2. The objective function is trained using a linear combination of the cross-entropy loss of the basic CNN segmentation network and the AE shape loss. The experimental results are shown in Fig. 6, Fig. 7, Fig. 8, and Table 2.



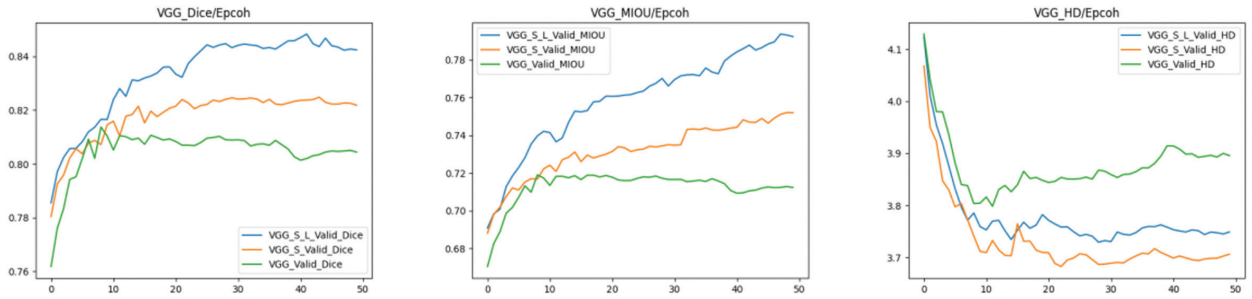
(a) (b) (c)

Fig. 6. The performance of Resnet, Resnet_S, and Resnet_S_L models using (**a**) Dice, (**b**) MIOU, and (**c**) HD methods



(a) (b) (c)

Fig. 7. The performance of Unet, Unet_S, and Unet_S_L models using (**a**) Dice, (**b**) MIOU, and (**c**) HD methods



(a) (b) (c)

Fig. 8. The performance of VGG, VGG_S, and VGG_S_L models using (**a**) Dice, (**b**) MIOU, and (**c**) HD methods

Fig. 6, Fig. 7, and Fig. 8 visually show the changes in test indicators of the model. By comparison, it is found that the Resnet_S, Unet_S, and VGG_S models based on self-supervised pre-training have significantly higher accuracy than the randomly initialized Resnet, Unet, and VGG basic segmentation models. In the sorting task, the model learns many basic features, such as the spatial anatomical structure of the image. This effective prior information is suitable for migrating the weights of the segmentation model, allowing it to learn based on the acquired prior features once the segmentation model training is completed. More information related to segmentation tasks can enhance model accuracy and speed up training convergence. Resnet_S_L, Unet_S_L, and VGG_S_L, which incorporate shape constraints, have further improved segmentation accuracy compared to Resnet_S, Unet_S, and VGG_S. The segmentation network learns low-dimensional position and shape information using sticky note shape constraints, significantly improving segmentation accuracy.

Table 2. Accuracy of different segmentation models

Accuracy	Resnet	Resnet_S	Resnet_S_L	Unet	Unet_S	Unet_S_L	VGG	VGG_S	VGG_S_L
Dice	0.827	0.843	0.874	0.813	0.827	0.853	0.805	0.821	0.844
MIOU	0.746	0.773	0.816	0.725	0.747	0.765	0.714	0.734	0.772
HD	3.617	3.605	3.565	3.796	3.774	3.759	3.871	3.761	3.727

Table 2 shows the segmentation results of the model on the test set. The Dice index for Resnet_S is improved by 1.8% compared to Resnet, Unet_S is improved by 1.7% compared to Unet, and VGG_S is improved by 1.9% compared with VGG; the MIOU index for Resnet_S is improved by 3.6% compared with Resnet, and Unet_S is 3% higher than Unet, and VGG_S is 2.8% higher than VGG; The HD parameters have not changed significantly. These results fully demonstrate that the self-supervised sorting tasks can be used to pre-train deep learning-based segmentation tasks. A double-layer constrained segmentation model using spatial prior information, the Dice index for Resnet_S_L further improved by 3.7% compared to Resnet_S, Unet_S_L further improved by 3.9% based on Unet_S, and VGG_S_L improved by 2.8% based on VGG_S; the MIOU index for Resnet_S_L further increased by 5.6% compared to Resnet_S, Unet_S_L further increased by 2.4% based on Unet_S, and VGG_S_L further increased by 5.2% based on VGG_S. The model constructed in this paper shows excellent segmentation performance, with Dice and MIOU indicators as high as 0.874 and 0.816, respectively, but it still has significant potential for performance improvement. Future research will focus on optimizing the segmentation network, incorporating spatial prior information, introducing cutting-edge attention mechanisms, integrating multi-scale feature fusion technology, and deep mining contextual information. These advancements are expected to improve the Dice and MIOU indicators, ensuring the model provides more accurate and reliable segmentation results across various image segmentation application scenarios.

Fig.9 shows the segmentation experimental results of each model under different image qualities. Comparing the segmentation masks produced by various models against the corresponding ground truth values allows for an intuitive evaluation of each model's segmentation performance. The results show that the Resnet_S, VGG_S, and Unet_S models based on self-supervised sorting pre-training perform better than the randomly initialized Resnet, VGG, and Unet models in the echocardiography left ventricle segmentation task. The pre-trained models showed satisfactory segmentation results in the face of segmentation challenges such as artifacts, speckle noise, and blurred anatomical boundaries. When evaluating the two key indicators of boundary accuracy and region overlap, the segmentation results of these models matched the ground truth values very well, significantly outperforming the basic segmentation network models. However, the Resnet_S_L, VGG_S_L, and Unet_S_L models that further incorporated the double-layer prior information constraints of the shape mask did not show significant performance improvements.

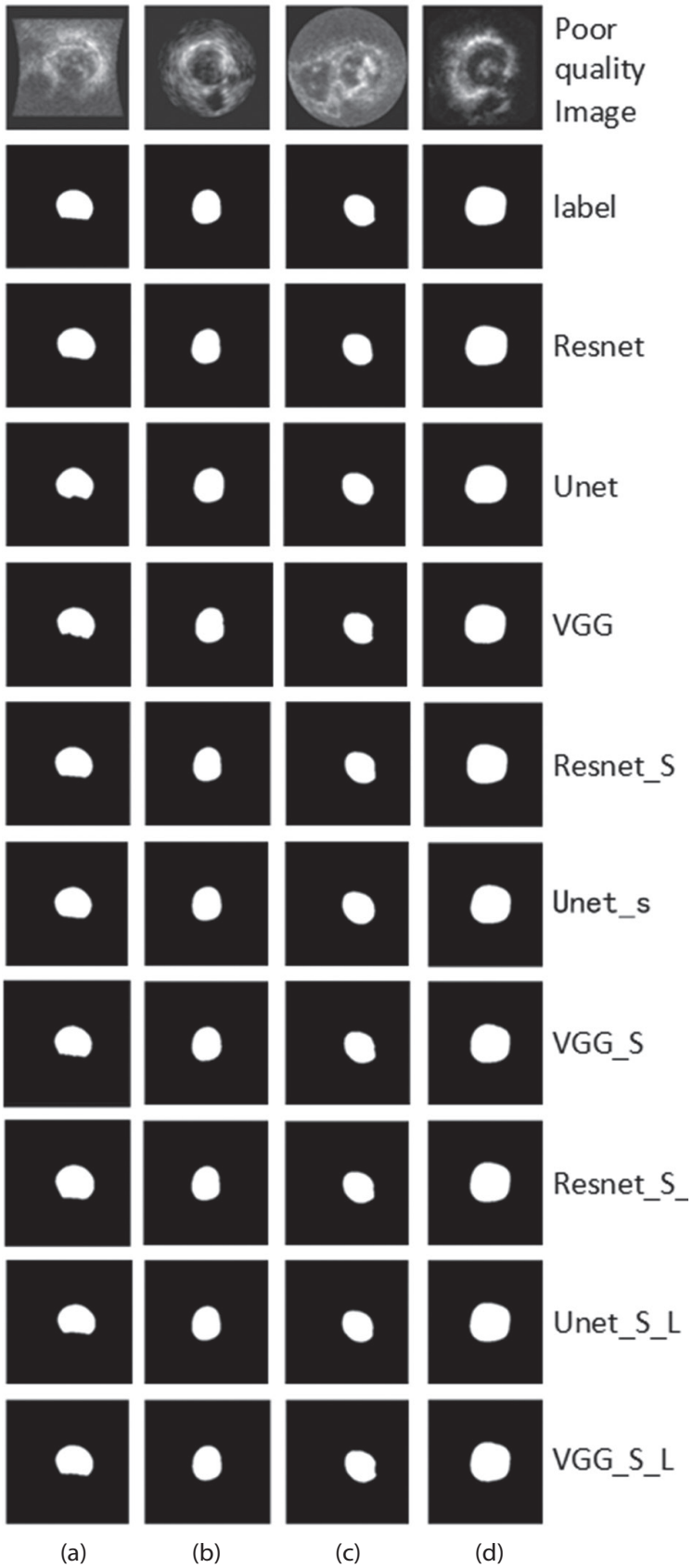


Fig. 9. Echocardiographic left ventricular segmentation renderings in different scenarios

The segmentation model based on self-supervised sorting pre-training uses a sequence of adjacent slices as input. During the pre-training process of the model, the basic structural features of the left ventricular image are learned. This prior information has a high reuse potential in the segmentation task, which helps to improve the segmentation accuracy and accelerate the convergence of the model. It is worth noting that fur-

ther integrating shape-constrained AE on top of these pre-trained models did not show significant performance improvements. The reason is that, on the one hand, the spatial anatomical prior of the image in the pre-training stage has already implied some shape information, thus weakening the gain effect brought by the additional shape constraint.

On the other hand, the AE model is mainly trained based on the left ventricle segmentation mask to capture the anatomical variation of the left ventricle accurately. Considering that the shape variation of the heart's left ventricle is relatively limited in the public dataset used in this study, the shape constraint of the second layer improves the performance. This improvement is more reflected in the subtle optimization of the existing performance. It fails to achieve the significant improvement brought by pre-training.

4. CONCLUSION

To solve the problem that the fully supervised segmentation algorithm for the left ventricle in echocardiography needs to deepen the network learning depth to improve segmentation accuracy due to insufficient training data, this paper constructs a segmentation model that integrates image prior information to achieve an effective combination of low-dimensional and high-dimensional features, as well as global and local features. On the one hand, through self-supervised sorting pre-training of the left ventricle from apex to base based on the spatial anatomical structure, the weights of the segmentation model are initialized, so that the model can fully obtain more local information related to the segmentation when performing the segmentation task, thereby improving the segmentation accuracy and accelerating the convergence speed. On the other hand, the segmentation network model is implemented to predict the shape constraints of image class labels, and the anatomical low-dimensional features of the left ventricle image of the two-dimensional echocardiogram are used to capture more global information and further improve the segmentation accuracy of the model. The model can extract basic features from related images, which are common to similar image analysis tasks, and can improve the performance of subsequent tasks.

Future research will continue to explore the sorting relationships implied by other spatial anatomical prior knowledge and try to model these relationships into a self-supervised sorting framework, study the specific impact of different sorting modes on the performance of the segmentation model, and find better sorting strategies. In addition, by optimizing the data input method, the sorting model can learn richer knowledge, significantly shorten the model training time, and improve the sorting model's learning effect and generalization ability, providing more solid technical support for applications in medical image processing.

5. ACKNOWLEDGMENT

The study was supported by the Ministry of Higher Education Malaysia (MoHE) and Universiti Teknologi MARA through the Fundamental Research Grant Scheme (FRGS/1/2022/ICT02/UITM/02/2).

6. REFERENCES

[1] R. M. Lang et al. "Recommendations for Cardiac Chamber Quantification by Echocardiography in Adults: An Update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging", European Heart Journal - Cardiovascular Imaging, Vol. 16, No. 3, 2015, pp. 233-271.

[2] J. Zhang et al. "Fully Automated Echocardiogram Interpretation in Clinical Practice", Circulation, Vol. 138, No. 16, 2018, pp. 1623-1635.

[3] M. Balasubramani, C.-W. Sung, M.-Y. Hsieh, E. P.-C. Huang, J.-S. Shieh, M. F. Abbod, "Automated Left Ventricle Segmentation in Echocardiography Using YOLO: A Deep Learning Approach for Enhanced Cardiac Function Assessment", Electronics, Vol. 13, No. 13, 2024, p. 2587.

[4] S. Ferraz, M. Coimbra, J. Pedrosa, "Deep Learning for Segmentation of the Left Ventricle in Echocardiography", Proceedings of the IEEE 7th Portuguese Meeting on Bioengineering, Porto, Portugal, 22-23 June 2023, pp. 159-162.

[5] G. Veni, M. Moradi, H. Bulu, G. Narayan, T. Syeda-Mahmood, "Echocardiography segmentation based on a shape-guided deformable model driven by a fully convolutional network prior", Proceedings of the IEEE 15th International Symposium on Biomedical Imaging, Washington, DC, USA, 4-7 April 2018, pp. 898-902.

[6] G. Carneiro, J. C. Nascimento, "Combining Multiple Dynamic Models and Deep Learning Architectures for Tracking the Left Ventricle Endocardium in Ultrasound Data", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 11, 2013, pp. 2592-2607.

[7] S. Dong, G. Luo, G. Sun, K. Wang, H. Zhang, "A left ventricular segmentation method on 3D echocardiography using deep learning and snake", Proceedings of the Computing in Cardiology Con-

ference, Vancouver, BC, Canada, 11-14 September 2016, pp. 473-476.

[8] S. Dong, G. Luo, K. Wang, S. Cao, Q. Li, H. Zhang, "A Combined Fully Convolutional Networks and Deformable Model for Automatic Left Ventricle Segmentation Based on 3D Echocardiography", BioMed Research International, Vol. 2018, No. 1, 2018, p. 5682365.

[9] Y. Ali, S. Beheshti, F. Janabi-Sharifi, "Echocardiogram segmentation using active shape model and mean squared eigenvalue error", Biomedical Signal Processing and Control, Vol. 69, 2021, p. 102807.

[10] V. Zyuzin, J. Komleva, S. Porshnev, "Generation of echocardiographic 2D images of the heart using cGAN", Journal of Physics: Conference Series, Vol. 1727, No. 1, 2021, p. 012013.

[11] A. Gilbert, M. Marciniak, C. Rodero, P. Lamata, E. Samset, K. Mcleod, "Generating Synthetic Labeled Data From Existing Anatomical Models: An Example With Echocardiography Segmentation", IEEE Transactions on Medical Imaging, Vol. 40, No. 10, 2021, pp. 2783-2794.

[12] M. J. Mortada, S. Tomassini, H. Anbar, M. Morettini, L. Burattini, A. Sbrollini, "Segmentation of Anatomical Structures of the Left Heart from Echocardiographic Images Using Deep Learning", Diagnostics, Vol. 13, No. 10, 2023.

[13] J. Liang et al. "Echocardiographic segmentation based on semi-supervised deep learning with attention mechanism", Multimedia Tools and Applications, Vol. 83, No. 12, 2024, pp. 36953-3697.

[14] S. Zhuang, H. Zhang, W. Ding, Z. Zhuang, J. Zhang, Z. Gao, "Semi-supervised domain adaptation incorporating three-way decision for multi-view echocardiographic sequence segmentation", Applied Soft Computing, Vol. 155, 2024, p. 11144.

[15] Y. Wan et al. "A Semi-supervised Four-Chamber Echocardiographic Video Segmentation Algorithm Based on Multilevel Edge Perception and Calibration Fusion", Ultrasound in Medicine & Biology, Vol. 50, No. 9, 2024, pp. 1308-1317.

[16] G. F. Cacao, D. Du, N. Nair, "Unsupervised Image Segmentation on 2D Echocardiogram", Algorithms, Vol. 17, No. 11, 2024, p. 515.

[17] M. H. Jafari et al. "Semi-Supervised Learning For Cardiac Left Ventricle Segmentation Using Conditional Deep Generative Models as Prior", Proceedings of the IEEE 16th International Symposium on Biomedical Imaging, Venice, Italy, 8-11 April 2019, pp. 649-652.

[18] D. Cao, J. Dang, Y. Zhong, "Real-Time Segmentation of Echocardiograms with Geometric Information Assistance", Journal of Computer-Aided Design & Computer Graphics, Vol. 34, No. 8, 2022, pp. 1252-1259.

[19] C. Liu, S. Dong, F. Xiong, L. Wang, B. Li, H. Wang, "Echocardiographic mitral valve segmentation model", Journal of King Saud University - Computer and Information Sciences, Vol. 36, No. 9, 2024, p. 10221.

[20] K. Wang, H. Hachiya, H. Wu, "A Multi-Fusion Residual Attention U-Net Using Temporal Information for Segmentation of Left Ventricular Structures in 2D Echocardiographic Videos", International Journal of Imaging Systems and Technology, Vol. 34, No. 4, 2024, p. e23141.

[21] Z. Feng, J. A. Sivak, A. K. Krishnamurthy, "Two-Stream Attention Spatio-Temporal Network For Classification Of Echocardiography Videos", Proceedings of the IEEE 18th International Symposium on Biomedical Imaging, Nice, France, 13-16 April 2021, pp. 1461-1465.

[22] S. Leclerc et al. "Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography", IEEE Transactions on Medical Imaging, Vol. 38, No. 9, 2019, pp. 2198-2210.

[23] M. Noroozi, P. Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11-14 October 2016, pp. 69-84.

[24] K. Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv:1409.1556, 2015.

[25] V. Zyuzin et al. "Identification of the left ventricle endocardial border on two-dimensional ultrasound images using the convolutional neural network Unet", Proceedings of the Ural Symposium on Biomedical Engineering, Radioelectronics and

Information Technology, Yekaterinburg, Russia, 7-8 May 2018, pp. 76-78.

[26] A. Amer, X. Ye, M. Zolgharni, F. Janan, "ResDUnet: Residual Dilated UNet for Left Ventricle Segmentation from Echocardiographic Images", Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society, Montreal, QC, Canada, 20-24 July 2020, pp. 2019-2022.

[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", The Journal of Machine Learning Research, Vol. 11, 2010, pp. 3371-3408.

[28] A. Sharma, O. Grau, M. Fritz, "VConv-DAE: Deep Volumetric Shape Learning Without Object Labels", Proceedings of Computer Vision - ECCV 2016 Workshops, Amsterdam, The Netherlands, 8-10 October 2016, pp. 236-250.

[29] Y. Yu, C. Wang, Q. Fu, R. Kou, W. Wu, T. Liu, "Survey of Evaluation Metrics and Methods for Semantic Segmentation", Computer Engineering and Applications, Vol. 59, No. 6, 2023, p. 57.

[30] C. Wang, Z. Zhao, Q. Ren, Y. Xu, Y. Yu, "Dense U-net Based on Patch-Based Learning for Retinal Vessel Segmentation", Entropy, Vol. 21, No. 2, 2019, p. 168.

[31] X. Li, Y. Wang, W. Yan, R. J. Van der Geest, Z. Li, Q. Tao, "A Multi-Scope Convolutional Neural Network for Automatic Left Ventricle Segmentation from Magnetic Resonance Images: Deep-Learning at Multiple Scopes", Proceedings of the 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, Beijing, China, 13-15 October 2018, pp. 1-5.

Computational intelligence in chromosomal primitive extraction and speaker recognition

Original Scientific Paper

Mohamed Hedi Rahmouni
University of Tunis El Manar
Faculty of Sciences of Tunis. Research Laboratory: LAPER.
Street Bechir Salem Belkhairya - Tunisia
medhedi.rahmouni@fst.utm.tn

Mohamed Salah Salhi
University of Tunis El Manar
National Engineering School of Tunis,
Research Laboratory of Signal Image and Information Technology LR-SITI
Street Bechir Salem Belkhairya -Tunisia
medsalah.salhi@enit.utm.tn

Mounir Bouzguenda
King Faisal University, Department of Electrical Engineering, College of Engineering,
Al Ahsa, 31982, Saudi Arabia
mbuzganda@kfu.edu.sa

Hatem Allagui
University of Tunis El Manar
Faculty of Sciences of Tunis. Research Laboratory: LAPER.
Street Bechir Salem Belkhairya -Tunisia
hatem.allagui@fst.utm.tn

Ezzeddine Touti*
Center for Scientific Research and Entrepreneurship,
Northern Border University,
Arar 73213, Saudi Arabia
esseddine.touti@nbu.edu.sa

*Corresponding author

Abstract *– This research presents an innovative approach leveraging computational intelligence for chromosomal primitive extraction and speaker recognition. The study emphasizes real-time digital signal processing (DSP) embedded systems integrating chromosomal-inspired techniques to enhance auditory feature extraction and speaker identification accuracy. By applying Gamma chromosomal factors, Mel-Frequency Cepstral Coefficients (MFCC) are refined through convolution, emulating human cochlear functionality. This integration aligns well with the perceptual auditory mechanisms and computational intelligence paradigms. The proposed methodology incorporates feature extraction techniques like Linear Predictive Coding (LPC), Linear Predictive Cepstral Coefficients (LPCC), and MFCC, followed by robust classifiers such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Self-Organizing Maps (RSOM). Experimental results demonstrate superior performance of RSOM, achieving a recognition rate of up to 99.7% with Gamma-enhanced MFCCs, compared to 98.6% for SVM and 91% for SOM. The RSOM model effectively identifies speakers across diverse conditions, albeit with slightly increased response times due to its dynamic recurrence loop. This work addresses challenges like environmental noise and variability in speech styles by introducing the Gamma chromosomal factor, a logarithmic nonlinear enhancement model. The experimental setup, executed on DSP boards using Python, highlights the advantages of computationally intelligent systems in real-world applications such as biometric authentication and decision-making systems. These findings underscore the potential of chromosomal-inspired computational techniques to advance speaker recognition technology, offering high accuracy and reliability in adverse conditions. Future research will focus on optimizing architectural and software frameworks to improve response times and further integrate this approach into constrained real-time systems.*

Keywords: *Computational intelligence; embedded systems; chromosomal primitive extraction; speaker recognition*

Received: January 17, 2025; Received in revised form: April 26, 2025; Accepted: April 28, 2025

1. INTRODUCTION

Speech is a natural and variable process that serves as the primary means of human communication, conveying information through acoustic signals. Speaker recognition has evolved from statistical models like HMM and GMM-UBM to deep learning approaches such as CNNs, RNNs, and wav2vec 2.0. Computational intelligence techniques, including Recurrent Self-Organizing Maps (RSOM) and Genetic Algorithms (GA), have shown promise in speech processing. However, their integration remains

underexplored, particularly for embedded systems. This study considers the foundation of speaker identification, a subset of artificial intelligence (AI), that involves distinguishing individuals based on their speech [1]. It aims to enhance recognition accuracy and efficiency while benchmarking against existing methods. An evolutionary recurrent neural system processes these acoustic vectors through unsupervised learning, associating each vector with a speaker's identity stored in a database. During the control phase, it compares new inputs to its stored data and makes an identification decision [2]. This process is implemented in embedded systems, such as digital signal processors (DSPs), under real-time constraints. Speech and speaker recognition, alongside facial recognition, are critical fields in Industry 4.0, IoT, blockchain, and cloud computing. These technologies are vital for security and decision-making applications [3].

Approaches such as Cochlear coefficients and its derivatives are widely employed, with the selection guided by the specific demands and limitations of the system. This paper introduces, as second contribution, a novel approach to extracting speech signal features and examines the most accurate classification algorithms for speaker identification. The aim is to enhance robotic safety, voice control, and decision-making, targeting zero-error performance, even in challenging environments.

This work is structured as follows: Section 1 contextualizes speaker recognition, tracing key models to the adopted computational intelligence approach. Section 2 s speech feature extraction methods and classifiers, highlighting accuracy, decision speed, and comparative analysis. Finally, sections 3 and 4 focus, respectively, on experimental results and discussion.

2. APPLIED METHODS

2.1. MAIN TECHNIQUES FOR EXTRACTING PRIMITIVES

ASR transcribes speech into text, while speaker identification determines identity using vocal features. Despite differing goals, they share techniques and can be integrated for more robust, personalized systems. [4]. Fig. 1 illustrates the global architecture of the voice recognition system.

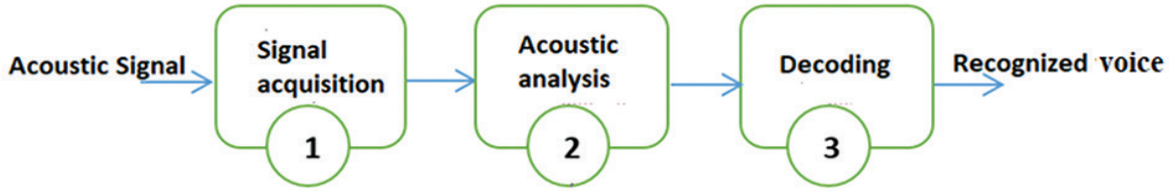


Fig. 1. Global architecture of a voice recognition system

Speaker recognition captures speech, digitizes it via DSP, extracts features (e.g., LPC, MFCC), and matches them to a database. It identifies speakers by balancing feature detail with reduced dimensionality. The primary techniques for extracting speech primitives are as follows:

a. Linear predictive coding (LPC)

Since the 1960s, LPC models speech as a filter with poles, representing the vocal tract, using filter coefficients to describe its transfer function [5].

Linear prediction (LP) is a key tool in speech analysis, modeling the signal $s(n)$ at time n based on p previous samples. The weighted sum of these samples produces a prediction error, $e(n)$, as shown in equation 1.

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \quad (1)$$

Linear Prediction (LPC) computes coefficients ak to minimize the error $e(n)$, commonly using autocorrelation or covariance, with autocorrelation preferred for its efficiency and stability. The LPC technique's block diagram is shown in Fig. 2.

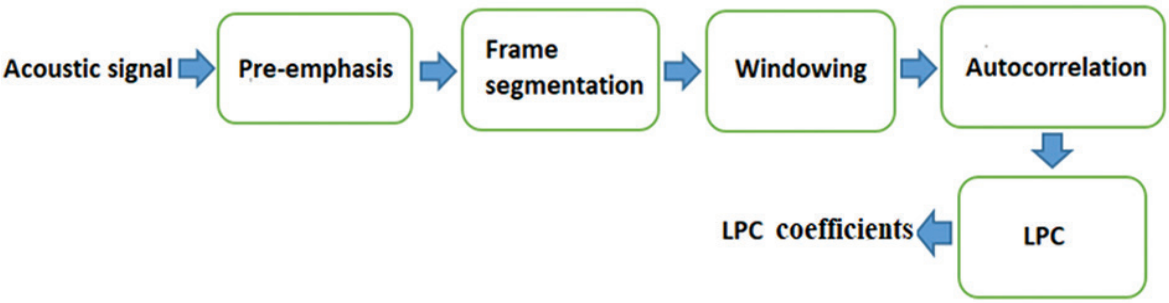


Fig. 2. Schematic representation of the LPC method

Consider $x(t)$ as the speech signal; the temporal autocorrelation function is expressed as:

$$c(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)x(t-\tau)dt \quad (2)$$

This represents the average over time of the signal multiplied by its own version shifted by a time delay τ. For a digital signal xk, sampled with a period Te, the discrete autocorrelation function is calculated using the equation:

$$C_n = \frac{1}{M}\sum_{k=i}^{i+M-1} x_k\, x_{k-n} \quad (3)$$

Here, M represents the number of points considered in computing the average, where the total duration is $T=M.Te$. The Levinson-Durbin algorithm [6] is used to determine signal coefficients by applying it to the filter signal for linear prediction. It calculates the linear prediction coefficients that minimize the mean squared error, as defined by:

$$E = \frac{1}{N}\sum_{n=0}^{N} e(n) \;\; where\; e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \quad (4)$$

The autocorrelation method computes LPC coefficients from windowed frames, precisely modeling the vocal tract's spectral envelope.

b. Linear Predictive Cepstral Coefficients (LPCC)

LPCC smooths the speech signal's spectral envelope while extracting speaker characteristics. Based on LPC analysis, it derives coefficients from the prediction process. Fig. 3 shows the LPCC extraction block diagram.

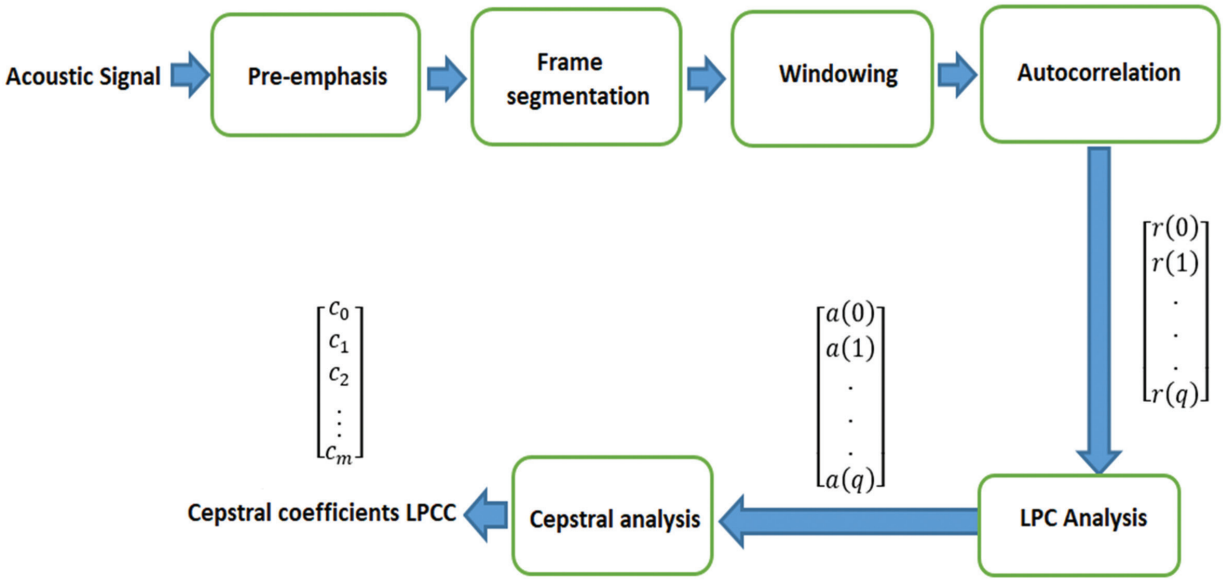


Fig. 3. Schematic illustrating the process of LPCC feature extraction.

The parameters are determined using the following equation:

$$\begin{aligned} c_n &= \sum_{k=1}^{n-1}(1-\frac{k}{n})a_k c_{n-k} + a_n \\ c_1 &= a_1 \quad 1 < n \le p \end{aligned} \quad (5)$$

C_n : the n^{th} coefficient of cepstrum
A_n : the n^{th} linear predictor coefficient LPC

c. Mel Frequency Cepstral Coefficient (MFCC)

In 1980, Davis and Mermelstein introduced Mel-Frequency Cepstral Coefficients (MFCC) analysis [7], a robust parameter extraction method based on the Mel scale. It uses FFT and DCT to derive decorrelated coefficients that closely simulate human auditory perception. The Mel scale, reflecting the human ear's sensitivity, is linear at low frequencies and logarithmic at high frequencies, as defined by the following equation [8]:

$$Mel(f) = 2595 log_{10}\left(1 + \frac{f}{700}\right) \quad (6)$$

MFCC extraction involves pre-emphasis, segmentation with a Hamming window, FFT, and Mel-scaled filter banks. The first 12 coefficients from 20-30 ms overlapping windows are used for analysis.

$$s(n) = x(n) * w(n) \qquad 0 \le n \le N-1 \quad (7)$$

The Fast Fourier Transform (FFT) is an efficient algorithm for calculating the Discrete Fourier Transform (DFT) of a discrete signal $x(n)$.

$$X(f) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j2\pi nf} \quad (8)$$

Proceed to the discretization of the frequency on N points among [-F_e/2 ,F_e/2] by putting:

$$f = \frac{k}{N} \;\; avec\; k = 0,1 \ldots \ldots N-1$$

We write in this case:

$$X\left(\frac{k}{N}\right) = X(k) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j2\pi \frac{k}{N} n} \quad (9)$$

The discrete Fourier transform (DFT) for N frequency points of a discrete signal is expressed as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}} \quad (10)$$

Where, $X(k)$ is the DFT output.

N is the sample count per frame, enabling time-to-frequency conversion. The Mel scale (1937) models auditory spectra using triangular filters, crucial for cepstral coefficient calculation. [9 -10]. Fig. 4 shows the general shape of the Mel-scale filter bank.

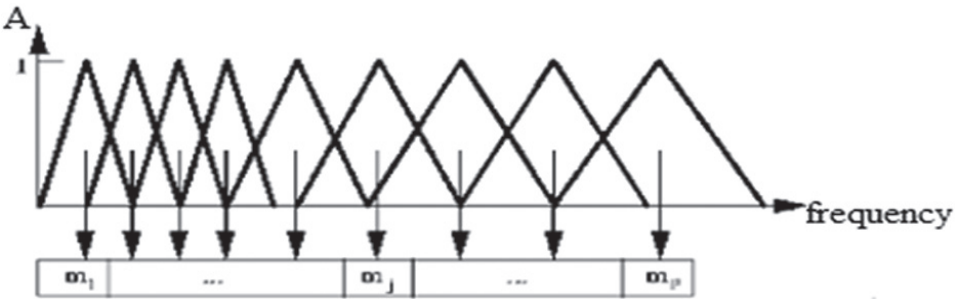


Fig. 4. Mel Scale Filter Bank

To ensure a smooth, stable spectrum, the energy logarithm (amplitude spectrum logarithm) is computed as follows:

$$s(m) = 20log_{10}\left(\sum_{k=0}^{N-1}|X(k)|\,H(k)\right) \quad (11)$$

Here, m represents the number of Mel scale filters, ranging from 20 to 40.

$X(k)$ denotes the FFT of the frame, while $H(k)$ refers to the transfer function of the Mel filter.

The Discrete Cosine Transform (DCT) is applied to filter coefficients, enhancing discriminative power and noise robustness for speech recognition. The coefficients $c(n)$ are calculated using the following equation [11]:

$$c(n) = \sum_{m=0}^{N-1} \mathrm{s(m)}\cos\left[\frac{\mathrm{n\pi(m-0.5)}}{\mathrm{M}}\right] \quad 0 \le \mathrm{n} \le \mathrm{M} \quad (12)$$

In this context $c(n)$ represents the MFCC coefficients. s(m) denotes the logarithmic spectrum. N indicates the number of samples within each frame. M refers to the number of filter banks.

MFCC dynamic features are captured by delta and acceleration coefficients, reflecting temporal changes, with typical speech systems sampling at 16 kHz. and extracts these features [12].

$$x_k = \begin{cases} c_k \\ \Delta c_k \\ \Delta\Delta c_k \end{cases}$$

Where, c_k : is the MFCC vector of the kth frame

$\Delta ck = c_{k+2} - c_{k-2}$: first derivative of the MFCCs calculated from the MFCC vectors of the kth+2 frame and kth-2 frame;

$\Delta\Delta c_k = \Delta c_{k+1} - \Delta c_{k-1}$: second derivative of MFCC.

d. Comparative study between primitives' extraction techniques

A comparative study of MFCC, PCA, and ARMA for voice feature extraction assesses their effectiveness, efficiency, and suitability in speech processing. Additional methods like spectral subtraction, LPC, Wiener filtering, and independent component analysis ICA aid in noise separation. Key comparison factors include computational complexity, noise robustness, and speech quality [13].

Each method has its strengths depending on the application. For speech recognition, MFCC and LPCC are commonly used. PCA is mainly for dimensionality reduction. LPC and ARMA are more relevant for speech synthesis and modeling. [14 -15].

2.2. CLASSIFICATION MODELS

Classification identifies speakers by matching features with a database using classifiers like HMM, SVM, k-means, PCA, and ANN [16].

a. Artificial Neural Networks ANN

An artificial neural network (ANN), inspired by the human brain, processes and produces information. Multi-layer perceptron (MLP) networks have three layers: input, hidden (for non-linear processing), and output (for results). See Fig. 5 [17 - 18].

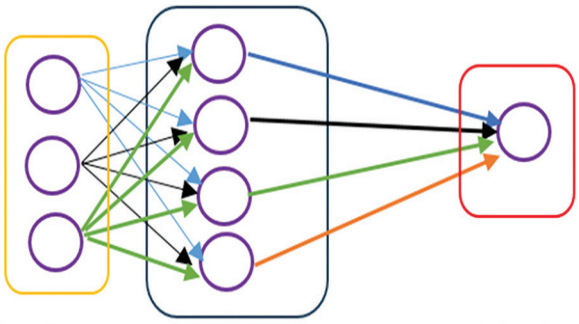


Fig. 5. Representative of ANN structure

ANN is widely used in speech and speaker recognition under artificial intelligence applications (Deep learning and Q-learning). The self-organizing map (SOM), as a static tool, achieves 85-90% speaker recognition, while the recurrent dynamic neural map (RSOM) improves this to 97-100% under optimal conditions. See Fig. 6.

The layer consists of neurons functioning as interconnected, fundamental processing units, operating through the following sequence of steps:

Unsupervised Learning: The neurons are trained by processing MFCC vectors that represent the speech signals of known individuals.

Neuron Count Estimation: The total number of neu-

rons, N_n, in the RSOM map is calculated using the formula $N_n = 2.5 \times C$, where C is the number of individuals (or vectors) involved in training. For example, recognizing 40 speakers typically requires around 100 neurons.

Neuron Specialization: After multiple training iterations, each neuron becomes fine-tuned to a specific input vector. In our case, the stop criterion is set at 100 iterations.

Testing and Identification: Once trained, the RSOM map can process any speech samples, analyze them, and visualize potential identification outcomes.

Weight Vector Representation: The weight vector associated with a specific neuron, indexed as iii, is described using the expression provided below [19]:

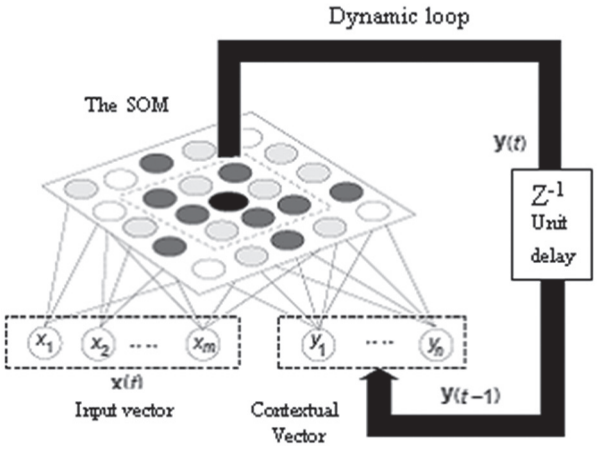


Fig. 6. Representation of Recurrent Self-Organizing Map RSOM

$$\boldsymbol{v}_{pij} = \{w_{i1}; w_{i2}; w_{i3}; \ldots; w_{ij}\} \tag{13}$$

Each signal vector is sent to all neurons of the RSOM map. The neuron whose weight vector has the smallest Euclidean distance to the input vector is activated, determining whether the input corresponds to a known or unknown individual. The Euclidean distance between the input $x(t)$ and the weight w_i is calculated as follows:

$$\begin{aligned} E_i &= \|x(t) - w_i\| \\ E_v &= \min E_i \;\; ; i \in N \end{aligned} \tag{14}$$

The winning neuron is the one that minimizes this Euclidean distance.

b. Support Vector Machine (SVM)

In the early 1990s, Vladimir Vapnik introduced the support vector machine (SVM), which projects data into a higher-dimensional space to find the best hyperplane for classification or regression. SVM solves the discrimination problem by constructing a function f that maps an input vector x to an output vector y [20].

$$y = f(x) = wx + b \tag{15}$$

The linear discriminant function is derived as a linear combination of the input vector $x=(x_1, x_2., x_N)$ and the weight vector w : $f(x)=wx+b$, $b \in R$ a scalar referred to as the bias.

If $f(x)>0$, it is decided that x is of class 1, otherwise, if $f(x)<0$, we decide x of class -1.

For classifying speech primitives using SVM, the following criteria are taken into account:

$$\begin{aligned} &\text{class of (x)} = \text{sign f(x)} \\ &= \text{sign (wx + b)} = \begin{cases} -1, \text{if f(x)} < 0 \\ 0, \text{if f(x)} = 0 \\ 1, \text{if f(x)} > 0 \end{cases} \end{aligned} \tag{16}$$

The margin of a hyperplane is defined as the shortest distance between the hyperplane and the closest data points. Let dis (x, w, b) denote the distance between a point x located on the plane H1 and the hyperplane defined by $f(x) = 0$. The margin M can be expressed as:

$$M = min\ \{dis\ (w \cdot x + b\ )\} \tag{17}$$

This distance is calculated as: $(f(x))/\|w\| = 1/\|w\|$, resulting in the distance between the two planes H1 and H2 being $2/\|w\|$.

The vectors w and b define the separating hyperplane, also known as the optimal hyperplane. Optimizing this hyperplane involves minimizing the squared norm $\|w\|^2$, leading to the objective: min($1/2\ \|w\|^2$).

This problem is typically solved using the Lagrange multipliers method. The classification function is represented as: $class(x) = sign(w \cdot x + b)$. The indicator function can also be reformulated based on the following expression [21].

$$\begin{aligned} w &= \textstyle\sum_{i=1}^{l} \alpha_i y_i x_i \\ \text{So, } class\ (x) &= sign\ \left(\textstyle\sum_{i=1}^{l} (\alpha_i y_i x_i . x) + b\right) \end{aligned} \tag{18}$$

In practical classification scenarios, data frequently necessitates separation via a nonlinear decision boundary. This is accomplished by applying a kernel-based transformation $K(x, y)$, which optimizes the input data and is represented in the following form:

$$f(x) = \left(\textstyle\sum_{i=1}^{l} \alpha_i y_i K(x_i . x) + b\right) \tag{19}$$

Among the kernels used are:

$$\begin{cases} \text{linear:} \quad K(x,y) = x.y \\ \text{polinomial:} \quad K(x,y) = [(x,y) + 1]^d \\ \text{radial basis function RBF:} \quad K(x,y) = exp\{-\Psi(|x.y|^2\} \end{cases}$$

c. Comparative study of main classifier models

A comparative study of classifiers like HMM, RSOM, CSOM, SVM, and DNN assesses their effectiveness in speech recognition. HMMs achieve around 90% accuracy, while RSOMs capture temporal speech dynamics and CSOMs handle classification. CNNs excel in visual data analysis, and DNNs surpass 95% recognition rates.

SVMs, effective with non-linear boundaries, achieve 85-95% accuracy. X-vector and i-vector methods, combined with DNNs, exceed 98%, while Deep Speaker Embeddings (DES) can achieve over 99%, though environmental conditions can reduce performance to 92%. Each method's choice depends on application requirements and data quality.

Various performance parameters are used to evaluate the suggested model. The RSOM in its evolutionary form (hybridized with GA) demonstrates a strong balance between recognition accuracy and computational efficiency in speaker recognition tasks. It achieves high precision (93-99%), recall (89-97%), and F1-score (92-96%), making it competitive with deep learning models while maintaining a lower computational cost. Compared to traditional models like HMM (Precision: 75-82%, Recall: 72-80%, F1-score: 73-81%) and SVM (Precision: 78-85%, Recall: 75-83%, F1-score: 76-84%), RSOM outperforms in handling dynamic speech variations. However, modern deep learning approaches such as CNN (Precision: 88-94%, Recall: 86-93%, F1-score: 87-93%), LSTM (Precision: 90-96%, Recall: 89-95%, F1-score: 89-95%), and wav2vec 2.0 (Precision: 93-97%, Recall: 92-96%, F1-score: 92-96%) achieve higher recognition accuracy but at the expense of increased computational complexity. In terms of time efficiency, RSOM outperforms deep learning models, with processing times comparable to HMM and SVM, making it a viable choice for embedded systems and real-time speaker recognition applications.

The Table 1 below highlights some parameter scores supported by TIMIT database.

Table 1. Compared performances over existing models

Model	Recognition Accuracy (%)	Computational Cost (ms)	Dataset Used
HMM	81	100	TIMIT
SVM	87	200	TIMIT
CNN	92	350	TIMIT
DNN	92.5	500	TIMIT
i-vector	89	275	TIMIT
Deep RSOM Embeddings	96	850	TIMIT

2.3. ADOPTED METHOD

Speaker recognition on embedded systems uses machine learning tailored to resources, speed, and accuracy. Lightweight models like SVM suit low-resource systems, while complex algorithms work on high-resource systems. This work employs Python-programmed DSP cards for a comparative study of SVM and RSOM, with Fig. 7 illustrating the SVM algorithm.

This approach introduces Gamma Chromosomal Factors, a novel technique convolved with MFCC primitives to enhance speech feature extraction, especially in adverse environments. The resulting convolutional output is then fed into an evolutionary RSOM model embedded on a DSP. A comparative analysis with an embedded SVM model, known for its lightweight nature, underscores the advantages of our approach.

The experimental setup includes:

- a PC running the Code Composer Studio CCS software environment for programming a DSP board.
- a Texas Instruments TMS 320 DSP.
- a USB cable for downloading the program describing the model to be implemented to the DSP.
- a screen interface for viewing results and curves

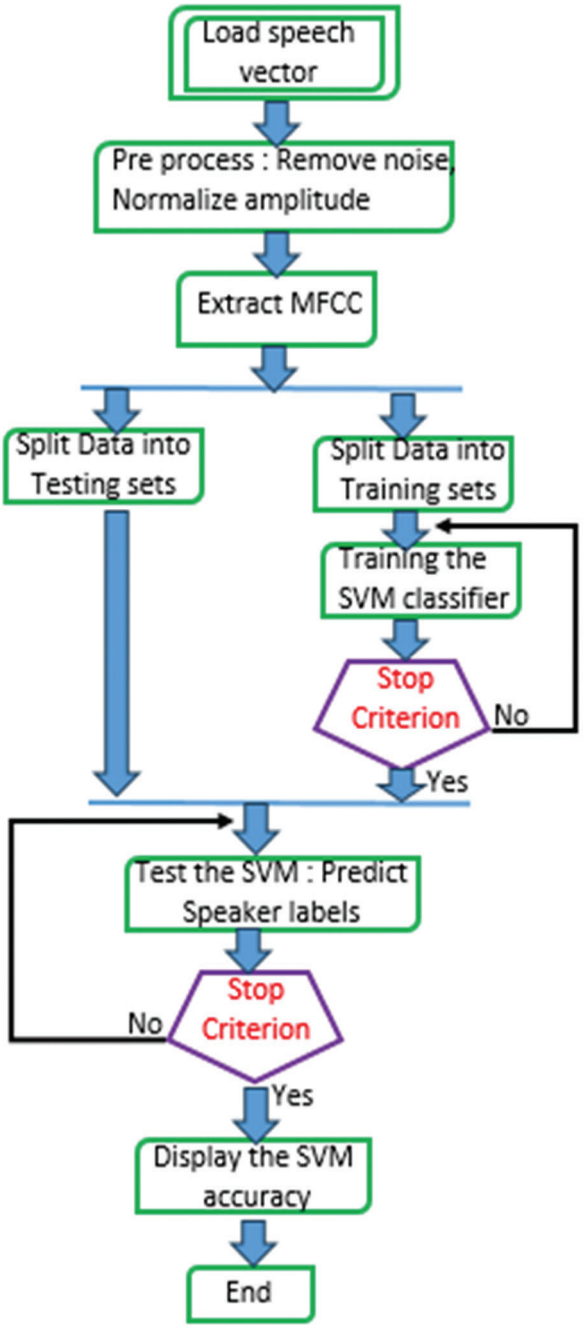


Fig. 7. Algorithm of experimented SVM Function

The algorithm assumes pre-processed, labeled speech signals split into training and testing sets. MFCC extraction functions are pre-implemented.

Compared to RSOM, it offers limited execution time and accuracy. Fig. 8 illustrates the optimized RSOM, where the BMU (Best Matching Unit) acts as a small intelligent processor, identifying the speaker.

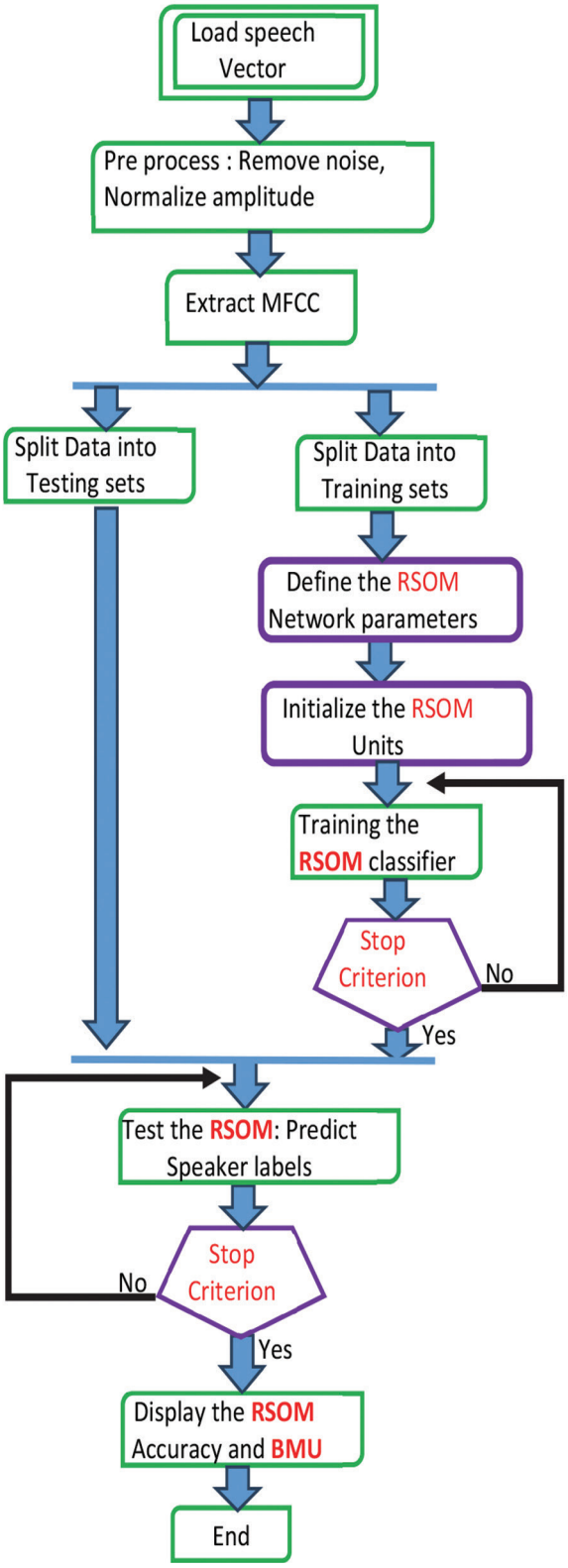


Fig. 8. Algorithm of adopted RSOM Function

3. RESULTS

MFCC coefficients in speaker recognition can be affected by environmental conditions, causing errors. Our approach uses a chromosomal factor, Gamma, to refine and enhance these coefficients. Tests on diverse speakers show Gamma ranges from 0.1 to 1.0. The results of this study are mentioned in Table 2 below.

Table 2. Chromosomal Gamma scores over conditions

Gamma	Day	Night
Men	0.9	1.0
Women	0.6	0.78
Children	0.4	0.55

Our contributed chromosomal factor Gamma is calculated using a logarithmic, non-linear model developed through experiments and validations.

$$Gamma\ (\gamma) = \alpha * log(\beta + \lambda) \quad (19)$$

Alpha (α) represents a membership factor that characterizes the state of an individual speaker. Its value ranges between 0 and 1.

Beta (β) serves as an indicator of geographic conditions and atmospheric pressure, varying within the interval [0, 10].

Lambda (λ) denotes a coefficient associated with neighborhood noise. This coefficient is typically negligible, ensuring that $(\beta+\lambda) \leq 10$.

Gamma chromosome, derived from deoxyribonucleic acid (DNA) composition, influences human biological traits. Its variation with day-night cycles can impact pronunciation.

Speaker recognition results on a DSP using Python depend on speech quality, feature extraction, classification algorithms, and system parameters. Accuracy ranges from 70% to 99%, and performance is assessed through metrics like precision, recall, and F1-score. Experimentation results for the sentence "I am Happy" spoken by five public people, using SOM, SVM, and RSOM without MFCC filtering, are presented in Table 3. Results may slightly vary with databases like TIMIT or VoxCeleb, but the performance gap between models remains consistent.

Table 3. Comparison of recognition rates across models using MFCC, excluding chromosomal Gamma

Models/ speakers	SOM rates in %	SVM rates in %	RSOM rates in %
Person 1	81.5	86.9	92
Person 2	83.2	88	94.5
Person 3	90.1	89.9	97.4
Person 4	87.6	95	99.2
Person 5	89	96.8	98

These results are illustreted in Fig. 9 below.

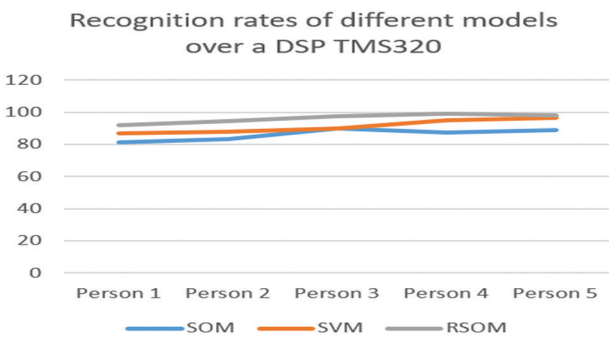


Fig. 9. Representation of recognition rates for different models over DSP

Nevertheless, RSOM exhibits a slightly longer response time of 30 ms, attributed to its dynamic recurrence loop, in contrast to 22 ms for SOM and 17 ms for SVM. The results highlight a trade-off: RSOM offers higher precision, while SVM is faster. RSOM remains the preferred choice, as real-time efficiency can be optimized through DSP hardware enhancements and software acceleration. RSOM's diverse neuron weights enhance adaptability to Deep Learning and Q-learning, while its recurrence loop adds dynamism, further improving results. When these models are tested using chromosomal Gamma MFCC primitives, an improvement in the results is observed, as shown in Table 4 below:

Table 4. Comparison of recognition rates across models using chromosomal Gamma MFCC primitives

Models/ speakers	SOM rates in %	SVM rates in %	RSOM rates in %
Person 1	82.1	87.4	92.7
Person 2	83.6	90	95
Person 3	91	91.5	98
Person 4	88.3	97,2	99.7
Person 5	89.5	98.6	98.5

Fig. 10 below illustrates the response of each model on their respective embedded systems, showing recognition rates when chromosomal Gamma is applied to MFCC primitives.

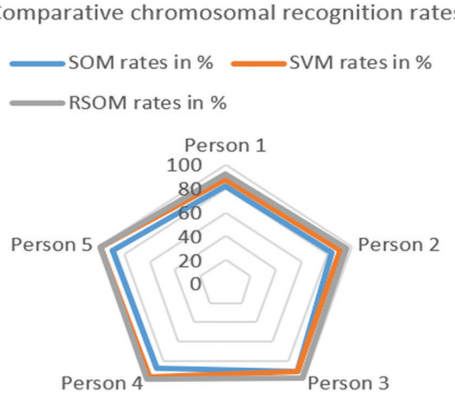


Fig.10. Visualization of recognition rates with chromosomal Gamma across various models on DSP

4. DISCUSSION

Speaker recognition is challenging due to factors like speech style, background noise, and microphone variations. Therefore, thorough evaluation under diverse conditions is crucial for reliability. As shown in Fig. 9, the RSOM model outperforms the SVM and SOM with a maximum recognition rate of 99.2%, compared to 96.8% for SVM and 90.1% for SOM, all without the chromosomal factor Gamma applied to the MFCC primitives.

By applying the convolutional method of MFCC primitives combined with the chromosomal factor Gamma to the spoken sentence during the testing phase of the various models, the following outcomes were observed:

- RSOM achieved the highest speaker recognition rate of approximately 99.7% in 34 ms.
- SVM reached a peak recognition rate of around 98.6% in 20 ms.
- SOM attained a maximum recognition rate of about 91% in 25 ms.

The embedded DSP model, utilizing computational sensors with the RSOM classifier, demonstrates superior speaker recognition performance compared to other models. However, it requires more processing time to generate its response. Consequently, architectural and software optimizations are suggested to enhance its suitability for a real-time, constrained system.

5. CONCLUSION

This research demonstrates the significant advancements made in the field of speaker recognition by leveraging computational intelligence and chromosomal-inspired techniques. The study developed an embedded real-time system using a combination of Mel-Frequency Cepstral Coefficients (MFCC) and Gamma chromosomal factors for feature extraction, along with advanced classifiers such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Self-Organizing Maps (RSOM). The results underscore the potential of these methods to significantly enhance recognition accuracy, even in challenging environmental conditions. Speaker recognition is inherently complex due to various challenges, including variability in speech styles, environmental noise, and device inconsistencies. The introduction of the Gamma chromosomal factor addresses these issues by providing a non-linear enhancement to MFCC, inspired by biological auditory processes. This factor adapts to variations in environmental conditions and speaker characteristics, enabling robust feature extraction and improving recognition rates. Experimental results demonstrate the superiority of RSOM in achieving a maximum recognition rate of 99.7% with Gamma-enhanced MFCCs, compared to 98.6% for SVM and 91% for SOM. However, RSOM's slightly increased response time due to its dynamic recurrence loop highlights a trade-off between accuracy and computational efficiency. The

study also emphasizes the versatility and adaptability of computational intelligence techniques. The integration of MFCC primitives with the Gamma factor not only improves recognition performance but also aligns with human auditory perception, bridging the gap between biological inspiration and technological application. The real-time implementation on DSP boards demonstrates the feasibility of deploying these advanced techniques in embedded systems, making them suitable for various practical applications, including security, robotics, and voice-controlled systems.

Data availability statements

The data used to support the findings of this research are available from the corresponding author upon request.

Declaration of interest

The authors confirm that there are no conflicts of interest associated with this Paper.

Acknowledgements

The authors extend their appreciation to Northern Border University, Saudi Arabia, for supporting this research work through project number "NBU-CRP-2025-2448".

6. REFERENCES:

[1] T. Voegtlin, "Recursive Principal Components Analysis", Inria Campus Scientifique, Nancy, France, 2002.

[2] Q.-B. Hong, C.-H. Wu, H.-M. Wang, "Speaker-Specific Articulatory Feature Extraction Based on Knowledge Distillation for Speaker Recognition", Journal APSIPA Transactions on Signal and Information Processing, Vol. 12, No. 2, 2023.

[3] Z. Wang et al. "A hybrid model of sentimental entity recognition on mobile social media", EURASIP Journal on Wireless Communications and Networking, 2016, p. 253.

[4] M. S. Salhi, El M. Barhoumi, Z. Lachiri, "Effectiveness of RSOM Neural Model in Detecting Industrial Anomalies", Diagnostyka, Vol. 23, No. 1, 2023.

[5] M. S. Salhi, N. Khalfaoui, H. Amiri, "Evolutionary Strategy of Chromosomal RSOM Model on Chip for Phonemes Recognition", International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.

[6] Z. Chen, P. Li, R. Xiao, T. Li, W. Wang, "A Multiscale Feature Extraction Method for Text-independent Speaker Recognition", Journal of Electronics & Information Technology, Vol. 43, No. 11, 2021.

[7] H. Liang, X. Sun, Y. Sun, Y. Gao, "Text feature extraction based on deep learning: a review", EURASIP Journal on Wireless Communications and Networking, 2017, p. 211.

[8] C. Hema, F. P. G. Marquez, "Emotional speech Recognition using CNN and Deep learning techniques", Applied Acoustics, Vol. 211, 2023, p. 109492.

[9] V. S. Dharun M. E, "Intelligent system speech recognition", Manonmaniam Sundaranar University, Tamil Nadu, India, 2012, PhD thesis.

[10] B. Medhi, P. H. Talukdar, "Different acoustic feature parameters ZCR, STE, LPC and MFCC analysis of Assamese vowel phonemes", Proceedings of the ICFM International Conference on Frontiers in Mathematics, Assam, India, 26-28 March 2015.

[11] S. Chen, Z. Luo, H. Gan, G. Mesnil, X. He, L. Deng, Y. Bengio, "An entropy fusion method for feature extraction of EEG", Neural Computing and Applications, Vol. 29, 2018, pp. 857-863.

[12] K. Bharti, P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering", Expert Systems with Applications, Vol. 42, No. 6, 2015, pp. 3105-3114.

[13] Y. Shen, X. He, J. Gao, "Learning semantic representations using convolutional neural networks for web search", WWW '14 Companion: Proceedings of the 23rd International Conference on World Wide Web, Seul, Korea, 7-11 April 2014, pp. 373-374.

[14] A. Severyn, A. Moschitti, "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks", Proceedings of the 38th International ACM SIGIR conference on research and development in Information Retrieval, Santiago, Chile, 9-13 August 2015, pp. 373-382.

[15] H. A. Elharati, M. Alshaari, V. Z. Këpuska, "Arabic Speech Recognition System Based on MFCC and HMMs", Journal of Computer and Communications, Vol. 8 No. 3, 2020.

[16] B. M. Tarabya, As. Khateb, S. Andria, "Processing Printed Words in Literary Arabic and Spoken Arabic: An fNIRS Study", Open Journal of Modern Linguistics, Vol. 11 No. 3, 2021.

[17] P. J. Worth, "Word Embeddings and Semantic Spaces in Natural Language Processing", International Journal of Intelligence Science, Vol. 13 No. 1, 2023.

[18] M. Koizumi, M. Maeda, Y. Saito, M. Kojima, "Correlations between Syntactic Development and Verbal Memory in the Spoken Language of Children with Autism Spectrum Disorders and Down Syndrome: Comparison with Typically Developing Children", Psychology, Vol. 11 No. 8, 2020.

[19] M. S. Salhi, S. Kashoob, Z. Lachiri, "Progress in Smart Industrial Control based on Deep SCADA Applied to Renewable Energy System", Turkish Online Journal of Qualitative Inquiry, Vol. 12, No. 8, 2021, pp. 871-882.

[20] F. A. Zadeh, A. R. Salehi, A. H. Mohammed, "An Analysis of New Feature Extraction Methods Based on Machine Learning Methods for Classification Radiological Images", Computational Intelligence and Neuroscience, 2022.

[21] Z. Yang, S. Serikawa, "Optimizing Speech Emotion Recognition with Hilbert Curve and convolutional neural network", Cognitive Robotics, Vol. 4, 2024, pp. 30-41.

A Scalable Distributed Approach for Exploration Global Frequent Patterns

Original Scientific Paper

Houda Essalmi*
Laboratory of Engineering Sciences, Polydisciplinary Faculty of Taza,
University of Sidi Mohamed Ben Abdellah
Fez, Morocco
houda.essalmi@usmba.ac.ma

Anass El Affar
Laboratory of Engineering Sciences, Polydisciplinary Faculty of Taza,
University of Sidi Mohamed Ben Abdellah
Fez, Morocco
anass.elaffar@usmba.ac.ma

*Corresponding author

Abstract *– Finding patterns in transactional databases regularly is an essential part of data mining since it makes it simpler to identify significant connections and reoccurring patterns in datasets. Scalable, high-performance computing solutions that employ parallel computing systems to optimize resource efficiency and data analysis as data volumes continue to grow are necessary for efficiently processing large databases. To solve these issues, this paper presents Exploration Global Frequent Patterns (EGFP), a new parallel algorithm designed to generate global frequent patterns in different distributed datasets. By facilitating the distribution of workloads and data partitioning, the approach reduces communication costs and ensures efficient parallel execution. Our approach uses two prefix-tree structures to generate a significantly compacted and structured representation of frequent patterns. The first structure local-tree serves to store local support values to effectively collect and arrange transaction data. Global prefix counts are then aggregated and ranked to improve frequency-based analysis and provide a more organized and useful representation of frequent patterns. To find the globally prevalent patterns, a Master site develops a second structure global-tree for each prefix based on this arranged data. Experimental results on large-scale benchmark datasets show that EGFP outperforms other existing methods including CD and PFP-tree in terms of execution time and scalability, while incurring considerably less communication cost.*

Keywords: *Data mining, Parallel Processing, Frequent Patterns tree, Communication costs*

Received: March 2, 2025; Received in revised form: April 25, 2025; Accepted: April 28, 2025

1. INTRODUCTION

The great progress in technology and research in recent years has greatly affected the increasing data volume. Datasets including many complex attributes usually grow exponentially. Distributed data mining is the method of evaluating large datasets maintained across several linked sources or servers, therefore supporting decision-making and revealing hidden information inside the distributed database that calls for specific knowledge. Essential in Data Mining are classification, association rule mining, sequential pattern detection, and other activities [1]. In a transaction database, the interactions among data values are complicated and many of those relationships are effectively implicit. In the discipline of data mining, association rule mining [2] is a rather popular method, it aims to find relationships among itemsets contained in transaction databases or other data sources [3]. Effective counting of all frequent patterns depends on Apriori methods, which produce appropriate rule sets. To find regular patterns inside a transactional database, Apriori algorithms [4] require two main phases candidate generation and pruning, i.e., the elimination of uncommon itemsets is used in an iterative approach in the process. Initially, it finds individual frequent items with values above a minimum support threshold; then by combining them with other frequent itemsets, it generates more combinations. The candidates are further evaluated using the set support threshold. This process continues till no more frequent itemsets can be generated.

The sequential Apriori technique is essentially an essential component of both parallel and distributed algorithms. Association rule mining and optimization depend much on parallel and distributed techniques, thereby improving load distribution and accelerating computation execution. Candidate Distribution (CD) [5] among these approaches assigns the produced candidates to several sites to reduce computational repetition. By using transaction allocation, the Distributed Mining Algorithm (DMA) [6] improves distributed data management. Fast Distributed Mining (FDM) [7] maintains result accuracy while reducing communication costs across nodes, improving efficiency. Optimal Distributed Association Mining (ODAM) [8] is interested in mitigating load imbalance and improving association rule computation's efficiency. The Distributed Decision Miner (DDM) [9] addresses distributed data analysis to boost the decision-making process in vast settings. Distributed Decision (DD) [5], which strategically distributes tasks based on resource availability, and Intelligent Data Distribution (IDD) [10], which flexibly impacts data distribution to maximize processing performance, are alternative techniques. Employing a hash-based approach, hash-based Parallel Association Rule Exploration (HPA) [11] increases the effectiveness of parallel association rule exploration. Integration of CD and DD approaches by Candidate Distribution (CaD) [5] helps to effectively manage candidates and reduce computing costs. Skew Handling (SH) [12] solves data distribution differences to distribute the load. For enhanced performance in a distributed environment, hybrid distribution (HD) [10] combines several distribution techniques.

Through tree-based approaches, such as FP-Growth (Frequent Pattern Growth), Apriori-based methods presently facilitate the analysis of frequent patterns. opposed to the Apriori approach, which needs both the generation and assessment of many candidates, tree-based [13-16] solutions develop this process by grouping the data in a simple and organized hierarchical structure.

Although the FP-Tree (Frequent Pattern Tree) [16] reduces searches for patterns and eliminates excessive candidate generation, therefore decreasing searches for patterns and avoiding unnecessary candidate generation, it also presents many drawbacks when used with very large databases. Building the FP-Tree requires maintaining all transactions in memory as a hierarchical structure retained in a record. This structure can grow excessively large and surpass RAM limits for large databases; therefore, it's ineffective for operation. Building an FP-Tree demands multiple processes, including organizing frequently occurring elements and incorporating transactions into the tree. Regarding time and resources, this approach could be very costly, particularly if the database is large and contains several different components. Mostly operating in memory, the FP-Growth approach, which utilizes the features of the FP-Tree, makes implementation difficult in distributed systems. Different versions, such as Parallel FP-Growth [17] and Load Balancing FP-tree (LFP-tree) [18], have been developed to advance scalability, even though they typically involve complex changes.

LFP-Tree intelligently distributes FP subtrees and transactions among compute sites to maximize load balancing, avoiding bottlenecks and minimizing the processing time. Developed for distributed systems such as Hadoop and Spark, PFP-Tree partitions data into subsets handled separately before the final results are combined. Applied to large databases, both the LFP-Tree and PFP-Tree approaches have weaknesses. LFP-Tree maximizes load balancing; however, the issue can find it challenging to distribute subtrees dynamically in the presence of wildly different transactions, thus generating residual problems with balance and overloading some sites. Moreover, subtree operations and coordination could contribute to higher computing costs. Although PFP-Tree is suitable for distributed environments, it causes major communication expenses between sites during the aggregation of final results, therefore influencing general performance.

In most cases, parallel systems in distributed environments improve scalability and performance; yet, they also have some limitations. The communication overhead between sites represents a main issue that may become a limiting factor in the case of frequent essential data exchanges. Moreover, the control of synchronizing across operations may ultimately result in significant latencies, especially when some activities need close coordination. Load imbalance poses a major problem since certain sites are unused while others show too much demand, therefore limiting the general performance of the system. Especially when the number of data needed to be evaluated is significant, these algorithms often require many database scans, therefore optimizing processing time and resource requirements.

Designed to solve the above-mentioned problems and efficiently uncover global frequent patterns inside distributed datasets, this paper presents a new parallel approach called Exploration Global Frequent Patterns (EGFP). Our approach is based on two tree structures, local and global, including the prefix data of the global database.

Unlike most parallel approaches (peer to peer), we first construct the Master-Slave paradigm by distributing the workload among several Slave sites, improving execution time and system scalability. This architecture constitutes a conscious decision in distributed systems when efficient management of resources and centralized control are required. The slave sites principally aim at building the first local tree structure based on prefix items depending on the defined transaction sequence in the local database. Our method simultaneously develops an ancestor table for each prefix to rearrange the locale-tree structures for all Slave sites in descending order by executing a single scan at each local database. Then, depending on the ancestry information of

the initial localized tree structure, the Master site constructs a global tree for all prefixes, iteratively generating frequent global patterns without requiring Slave site communication.

Our EGFP reduces the communication load across several sites of the system by limiting data processing to a single pass (one scan), a crucial consideration in distributed architectures where an excessive number of exchanges can negatively impact performance. Compared to techniques requiring multiple reads and synchronizations, our method facilitates all Slave sites operating their processing independently, hence reducing the need for synchronizing and temporarily storing. We examine our EGFP algorithm's performance against PFP-Tree and CD on actual increasing data quantities. To develop and investigate a tree structure, PFP-Tree and CD both need several data readings. For real-time analytics and large-scale systems, our method maximizes processing by reducing the amount of data accessed to an alone pass.

The structure of the work is as follows: Section 2 contains a review of the existing works. Section 3 presents the notation and problem definition. In Section 4 our proposed method of algorithm is explained. the results with discussion are given in Section 5. Then the paper is concluded this work in Section 6.

2. RELATED WORK

Several research works have been proposed to enhance the efficiency, scalability, and flexibility of pattern mining algorithms in large-scale data settings. In this section, we review seminal contributions in the field of distributed frequent pattern mining.

Fernandez-Basso et al. [19] presented an original method to improve the extraction of prevalent patterns and association rules inside a distributed system by using Apache Spark. The study aims to address in the framework of large datasets the drawbacks of conventional algorithms such as Apriori and FP-Growth. The authors search for several optimizations, including using Spark's distributed design to provide efficient in-memory parallel computing, hence reducing scans and improving productivity. This work simplifies the consumption of resources and execution time while improving the scalability and applicability of knowledge extraction for large datasets.

The authors in [20] provided a novel method for sequential pattern extraction in databases using a tree structure (SP-Tree structure). This approach applies an optimal tree for better sequence structure, hence reducing data redundancy and database scans. The method advances performance on speed and memory use by applying an effective structure; consequently, pattern extraction becomes more suitable for large datasets. By establishing upgraded efficiency and scalability of sequential pattern analysis, this work progresses data mining methods.

Van and Josef [21] introduced a new approach for extracting frequent itemsets inside a distributed and parallel architecture. The FPO-Tree (Frequent Pattern Ordered Tree) structure that the authors provided improves memory compression and organization of transactions, hence reducing database scans. Moreover, they developed the DP3 (Distributed Parallel Preprocessing Pattern Mining) method, indicated to efficiently apply distributed architectures and parallelize the frequent pattern extraction. This approach reduces data transfers between nodes and maximizes workload distribution, thus boosting the scalability and efficiency of Frequent Itemset Mining (FIM). By reducing computational costs and improving analytical performance, the work presents a successful method for handling large data sets.

In [22], the researchers suggested a selective and adaptive approach for extracting frequent patterns in distributed transactional databases. In contrast to conventional methods that generate all frequent patterns, this particular approach operates on demand by extracting only the relevant patterns depending on user requests. In a distributed system, this greatly reduces computing costs and improves resource economy. This concept depends on the quick identification of patterns at the instant the analyst needs them. They offer DDSampling, a new pattern sampling technique. This program chooses a pattern at random from a distributed transactional database such that the selection probability corresponds to its degree of interest.

The study in [23] investigated the issues and possible solutions for parallelizing frequent itemset mining algorithms in large data settings. Their discussion is centered on enhancing the efficiency of conventional mining methods by their execution in parallel computing architectures. In particular, they suggested an approach that splits large datasets and shares computational tasks across several processing units. This method reduces redundancy in candidate generation and enhances overall efficiency by load balancing and better data access patterns. The implementation, evaluated on cloud computing setups, showed that parallelization substantially speeds up mining with result accuracy intact. These observations highlight the effectiveness of parallel mining techniques in managing the growing volume and velocity of data that are characteristic of contemporary big data systems.

In [24], the authors suggested a distributed association rule mining method named Mine-first Association Rule Mining that solves data decentralization and privacy issues in distributed networks. Their method allows each node to mine local frequent patterns without revealing raw data, thereby ensuring data confidentiality and conserving communication overhead. The incorporation of local patterns into global rules is facilitated by an effective aggregation mechanism that considers support and confidence measures, thereby guaranteeing the integrity and validity of the rules discovered.

The distributed framework is extremely effective when used with decentralized data, providing a scalable and privacy-conscious solution that is especially applicable to multi-source environments, including cloud-based analytics and federated platforms.

In another work authors [25] designed a parallel frequent itemset mining algorithm named STB_Apriori, which is specifically tailored for big data environments using the Apache Spark framework. The algorithm overcomes the drawbacks of conventional Apriori algorithms, especially the computational burdens resulting from repetitive candidate generation and processing of large datasets. The suggested approach employs a BitSet-based compression technique to preserve transactional data in compressed Boolean matrices to accelerate bitwise calculation and lower memory usage. Further, the system cleverly takes advantage of Spark's distributed computing framework for parallel processing of the mining task across several nodes. Experimental evaluations show STB_Apriori to be considerably better than state-of-the-art algorithms about execution time and scalability, positioning it in a favorable position for mining frequent patterns in large-scale distributed data.

Rochd and Hafidi [26] suggested DSSS (Distributed Single Scan on Spark), a distributed version of SSFIM (Single Scan Frequent Itemset Mining) [27], to efficiently mine frequent itemsets in big data environments with Apache Spark. DSSS conducts a single pass over the data by broadcasting a compressed dataset to nodes via RDDs and broadcast variables, unlike conventional multi-scan approaches. It includes early elimination of infrequent items and pruning of unpromising candidates to enhance memory and communication overhead. The experimental results confirm its ability for scalability and efficiency, showing its suitability for cloud and streaming environments.

3. NOTATION AND PROBLEM DEFINITION

Let $I= \{x_1, x_2, x_3,..., x_n\}$ be a collection of n distinct elements. A pattern or a collection of elements constitutes a subset of this set, defined by $X \subseteq I$. A database DB is essentially a collection of transactions, where each transaction T is a subset of I and is uniquely identified by its transaction identifier (TID). The number of database transactions presenting pattern X expressed as sup(X), signifies its level of support value. The equation of support is calculated as follows [2]:

$$sup(X) = (count(X))/N \quad (1)$$

where $count(X)$ represents the occurrence of X in the database and N defines the entire number of transactions. A pattern is considered frequent if its level of support surpasses the minimal support threshold, marked by the symbol ξ. Frequent pattern mining essentially is interested in identifying each pattern that frequently appears in a transaction database while maintaining the specified support threshold ξ.

A distributed system consists of n sites, each labeled, S_1, S_2, S_3,..., S_n.Under this system, the database DB is horizontally divided into n parts, shown as DB_1, DB_2, DB_3,..., DB_n, where each part is assigned to a particular site for data processing (for $i = 1,..., n$). To measure the frequently occurring pattern X, we indicate $Sup_i(X)$ as its local support count, and $Sup(X)$ as its global support count in the whole distributed system. Globally frequent a pattern X satisfying the minimal support criteria ξ, determined using the formula [4]:

$$Sup(X) \geq \xi \times |DB| \quad (2)$$

In a distributed database setting, this requirement is essential since it ensures that the pattern shows consistently over several partitions.

4. PROPOSED METHOD

This section presents our proposed method of algorithm, centered on the Master/Slave paradigm in a distributed computing environment, which can efficiently enable parallel and distributed frequent pattern mining. Additionally provided is an extensive description of the mining process and its purposes.

4.1. MASTER-SLAVE PARADIGM

Communication in distributed frequent pattern mining is typically decentralized. At each phase, all of S_i shares its locally calculated support values with every other site, which produces a rapid increase in data exchanges [28]. Network performance may be impacted as a result of the higher levels of communication imposed by growing datasets and site interactions. An alternative is the Master/Slave paradigm, which, as described in [28], offers a centralized communication mechanism. The dataset is divided into clusters by a Master site, which then assigns each cluster to a slave site. The Master site coordinates the entire process and collects information from Slave sites. This operational method is highly beneficial for distributed computing systems because it minimizes time and space complexity and maximizes resource efficiency, according to previous research by Vasoya and Koli [29].

A distributed and parallel method is examined in this work, in which the database is split horizontally over multiple Slave sites, each of which handles an equal share of transactions on its site. The Master site reduces the overhead of inter-site communication by distributing and merging processed data, functioning as the coordinator. Each Slave site obtains unique prefixes and their associated support numbers from the dataset in a single database scan to generate a Local-Tree Structure (LTS). By collecting the values of local supports, the Master site ranks prefixes in descending order of global frequency. The Ancestor Table (AT), which is generated from this data, serves as an essential component for the Global-Tree Structure (GTS), which is built at the Master site. The FP-Tree technique processes a single structured tree recursively, while the Mining Global Frequent Patterns (EGFP) approach

splits the task into multiple smaller GTSs. By investigating patterns at different hierarchical levels without requiring an excessive amount of inter-node communication, this method increases computational efficiency. The following paragraph provides a detailed description of the EGFP algorithm and its computational elements.

4.2. CONSTRUCTION OF LOCAL-TREE STRUCTURE

At this stage, the Local-Tree Structure (LTS), a data structure designed to optimize processing and storage efficiency with a single database investigation, gets formed. The primary LTS structure consists of a root node, called null, and multiple child nodes, each of which represents a different prefix subtree. Every node contains crucial information, including the prefix name, which identifies the associated element, the support count, which records the frequency of occurrence of a path segment in local transactions, and a node-link, which connects related nodes, using a structure similar to the FP-tree [16].

A sequential three-step process is used to construct the LTS. The transactions from the local database are initially added to the tree in a predefined transaction sorting order to ensure consistency. They locate and store local frequency counters. Then, the local counters from several Slave sites are combined into a Global Counter Table, where their values are added up and then arranged in decreasing order of frequency. Finally, the LTS paths are reorganized according to these sorted global prefix counters, refining the tree's structure for optimal efficiency. The following example illustrates how an LTS is constructed from a local database.

Consider the database DB illustrated in Fig. 1(a), where transaction records are distributed across two Slave sites: DB_1 and DB_2, as depicted in Fig. 1(b). The minimum support threshold is set at ξ=2. The initial phase involves constructing a Local-Tree Structure (LTS) at each Slave site by organizing transaction prefixes in lexicographic order. To facilitate this process, an empty Counter Table (CT) is first created, listing all database prefixes alongside their respective frequency counts. The LTS is then built using the FP-tree methodology, inserting transactions based on the CT structure. Unlike the FP-tree's vertically structured Header Table, the CT provides a horizontal representation, mapping elements, and their counts to their first occurrences in the LTS. Each Slave site independently constructs its local LTS by processing transactions from its database segment. Every time a transaction is added, the CT updates the occurrence count for each prefix. The initial structures of LTS_1 and LTS_2, representing Slave sites 1 and 2, are illustrated in Fig. 1(c) and (d). Correspondingly, CT_1 and CT_2 maintain accurate prefix frequency counts within their respective Slave sites. Once the LTSs are built, the contents of CT_1 and CT_2 are transmitted to the Master site, which oversees system coordination and management.

In the second phase, the Master site computes the global support count for each prefix by aggregating the support values from all received CTs. After summing the local counts, the prefixes are sorted in descending order of frequency, prioritizing the most significant patterns. The prefix counter structure is shown in Fig. 1(e), while Fig. 1(f) details the computation process for global support values. Once the support values are consolidated, Fig. 1(g) presents the sorted global prefix counts. This information is then redistributed to all Slave sites, allowing them to reorganize their LTSs accordingly, thereby streamlining subsequent pattern analysis and extraction.

The final step aims to enhance tree efficiency by restructuring LTSs at each Slave site based on the sorted global prefix order, thus avoiding redundant database scans. Only the frequently occurring prefixes contribute to tree reorganization. At the start of this step, each Slave site generates an Ancestor Table (AT), which records prefix frequencies alongside their ancestor relationships. As paths are reorganized, prefix occurrences are updated within the AT, preventing data duplication. For example, in Slave site 1, the original transaction path (A, B, C, E) is reordered as (B, C, E, A). The ancestor relationships are updated as follows: Prefix B has no ancestors, as it serves as the root node. Prefix C has a single ancestor, {B:2}, where 2 represents the support count of B in this path and in the previous path (B, C, E). Prefix E has one ancestor consisting of two prefixes, {B, C:1}. Prefix A has two ancestors: The first ancestor is {C:1}, derived from the (C, A) path, and the second ancestor is {B, C, E:1}. This restructuring allows for more efficient pattern extraction in subsequent analyses.

Fig. 1(h) and (j) illustrate the information repository at Slave sites 1 and 2, detailing each prefix, its support count, and ancestor relationships. Meanwhile, Fig. 1(i) and (k) depict the optimized LTS structures following reorganization at both sites.

After completing the final phase, the local LTSs attain a highly compact structure, preserving all essential details from their respective databases. This restructured format ensures that the ancestry of each prefix node can be accurately retrieved via the Ancestor Table (AT). Once the LTSs are finalized, the collected node data is transmitted to the Master site for global frequent pattern extraction. For instance, if a node X has N ancestors, its representation in the local AT takes the following form: {($X.ancestor_1$: sup ($X.ancestor_1$)),…,($X.ancestor_N$: sup ($X.ancestor_N$)), sup (X)} where each $X.ancestor_1$ is distinct from $ancestor_N$.

When an ancestor appears multiple times across subsequent pattern analysis and extraction.

The final step aims to enhance tree efficiency by restructuring LTSs at each Slave site based on the sorted global prefix order, thus avoiding redundant database different paths, its frequency values must be consolidated to maintain an accurate count. Instead of storing redundant entries, the cumulative support count is

computed as follows: {($X.\ ancestor_1$: sup ($X.\ ancestor_1$) + sup ($X.\ ancestor_N$), sup (X)}. Applying this principle to our example, the computed values for *Slave Site*$_1$ are: B: {Ø,2}, C: {B :2,3}, E: {(B, C :2),2}, A: {(C :1), (B, C, E :1),2}. For *Slave Site*$_2$, the results are: B: {Ø,2}, C: {B :2,2}, E: {(B :1), (B, C :2),3}, A: {(B, C, E :1),2}.

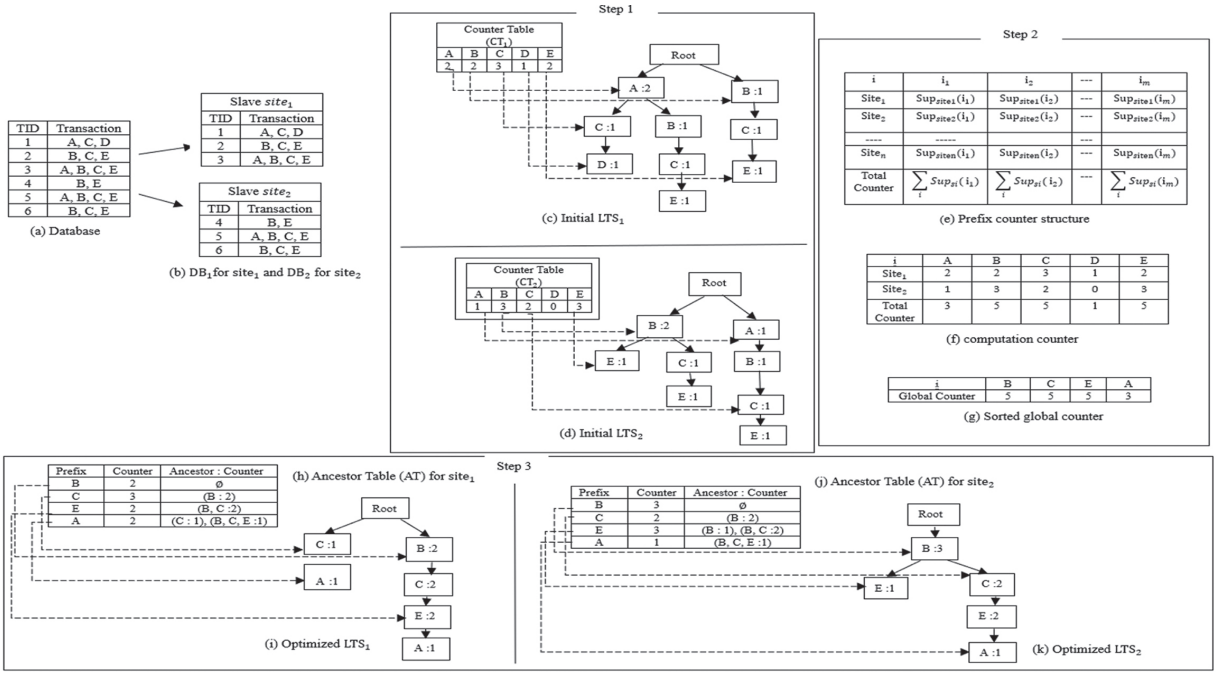


Fig. 1. Construction of Local-Tree Structure

4.3. CONSTRUCTION OF GLOBAL-TREE STRUCTURE

In this section, the pattern exploration process is carried out at the Master site, where we introduce a new frequent pattern extraction technique based on a hierarchical tree structure called the Global-Tree Structure (GTS). Our EGFP algorithm utilizes the GTS through multiple iteration levels K. Instead of relying on a single tree, the approach employs multiple hierarchical GTSs, each designed to organize frequent patterns at different levels of granularity. Each GTS level captures increasingly generalized patterns (e.g., pairs of elements, triplets, etc.) and facilitates targeted pattern exploration. This structured approach reduces the search space while efficiently organizing frequent patterns across various levels of abstraction.

The GTS construction is primarily based on information extracted from the Ancestor Table (AT), which is received from the Slave sites. For each prefix X, all ancestor information gathered from the different Slave sites is combined to build the GTS. The nodes within a GTS contain: all ancestor prefixes, the count of each prefix, node connections, and pointers to a table named Global Counter Table (GCT). The GCT serves as a central repository for aggregated support information from all ancestor prefixes. If the same ancestor elements appear in multiple Slave sites, their support values are accumulated within the corresponding GCT elements. The GTS exploration procedure follows a methodology similar to that of conditional FP-Trees but with a reversed traversal direction. Instead of exploring elements bottom-up, as in an FP-Tree, the GTS traversal moves from top to bottom within the Global Counter Table (GCT). By executing this recursive exploration process, all frequent global itemsets associated with X are efficiently derived.

The GTS construction process is carried out iteratively at each level. At level K=1, a GTS is created for each prefix of size m=1 received from the local Slave sites, as previously described. At level K=2, a GTS is built for each global frequent pattern derived from the first-level patterns of size (m=2). This is achieved by intersecting the paths of frequent global pattern subsets to construct the GTS. At level K≥3, the process continues by identifying the common paths shared among subsets of X of size (m−1) that contain the same (m−2) prefixes at the start of frequent global patterns X. This iterative procedure continues until no further GTSs can be constructed, meaning that all global frequent sets in the distributed database have been identified, and no additional frequent patterns can be generated. If a prefix X has no ancestors, its GTS cannot be constructed. Similarly, if X belongs to a subset of a global frequent pattern XY and X has no ancestors, then the GTS for XY cannot be created, preventing any further global frequent pattern generation.

Fig. 2 illustrates the GTS construction process for each global frequent pattern at different iteration levels K, based on the example provided in Fig. 1(i) and (k). For instance, at level K=1, the GTS extraction process for the global element A proceeds as follows: The global pattern A shares a common ancestor (B, C, E:2) from *Slave Site*$_1$

and *Slave Site*$_2$, as well as a single ancestor (C:1) from *Slave Site*$_1$. Therefore, the GTS is constructed, forming nodes B, C, and E, while accumulating support information for all prefixes in the Global Counter Table (GCT). Since these elements are frequent, a set of frequent item combinations associated with A is generated: {AB:2, AC:3, AE:2}. Next, a GTS is constructed for each derived global frequent pattern (AB, AC, AE) enabling recursive exploration at iteration K=2. At level K=2: The global element AB contains the prefix B, which has no ancestors, so its GTS cannot be built. The global element AC has two prefixes, A and C, meaning an intersection can be made between the paths of GTS$_A$ and GTS$_C$ to construct associated with X are efficiently derived.

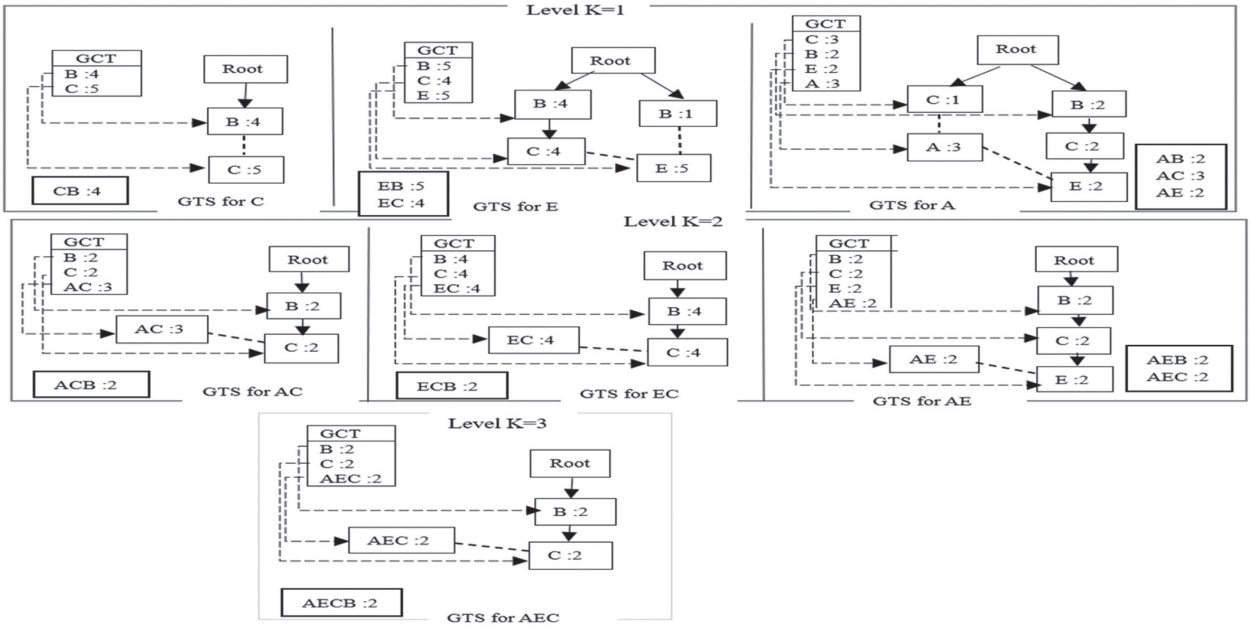


Fig. 2. Construction of Global-Tree Structure

GTS$_{AC}$, leading to the derivation of frequent elements associated with AC: {ACB:2}. The global pattern ACB contains the prefix B, which lacks ancestors, preventing the construction of GTS$_{ACB}$.

This procedure is repeated for the global patterns C and E until all frequent global patterns are extracted. In this example, the process converges in three iterations.

The efficiency of our global frequent pattern exploration procedure lies in its ability to construct a highly compact and optimized GTS, especially in iterations where $k \geq 3$. At this stage, the approach focuses only on shared paths among subsets of a global pattern of size $(m-1)$ that contain the same $(m-2)$ prefixes at the beginning of frequent patterns X. This method significantly enhances the extraction of highly frequent global patterns.

To execute our EGFP (Exploration Global Frequent Patterns) algorithm, we implement a Master/Slave communication model within a fully distributed environment. This intelligent data distribution strategy minimizes communication overhead between sites. Unlike traditional distributed frequent itemset mining algorithms, which require extensive inter-site communication, leading to high network costs, our approach optimizes synchronization while reducing complexity. For comparison, the CD (Count Distribution) algorithm follows the Apriori logic, employing an all-to-all broadcasting approach, which necessitates multiple database scans and explicit candidate generation. While effective, this method can become computationally expensive when handling large-scale data, as it results in increased communication and synchronization overhead.

Applying the above example, the CD algorithm operates as follows: In the first iteration, each Slave site computes the local support of 1-itemsets: *Slave Site*$_1$: {A:2, B:2, C:3, E:2}, *Slave Site*$_2$: {B:3, C:2, E:3}. By exchanging local support counts for each candidate itemset, the two sites can synchronize and calculate the global support: {B:5, C:5, E:5, A:3}. The second iteration computes the local support for 2-itemsets using Apriori-based steps, producing the following outcomes: {AB, AC, AE, BC, BE, CE}. The sites exchange these frequent itemsets with each other. This iterative process continues for iterations 2, 3, and 4, leading to the discovery of: {ABC, ABE, ACE, BCE}, and finally {ABCE}.

The EGFP algorithm leverages two fundamental tree structures: LTS and GTS. Unlike conventional methods, EGFP optimizes database scanning by requiring just a single pass to compute prefix frequencies, significantly enhancing efficiency. In contrast, the parallelized FP-Growth (PFP-Tree) algorithm operates within a fully distributed Peer-to-Peer framework, necessitating two separate scans: the first to compute local frequency counts and the second to reconstruct the local FP-Tree, where items are ordered based on their local frequency. Once local FP-Trees are built, they are progressively merged

into a global FP-Tree. Unlike a Master-Slave model, where coordination is centralized, PFP-Tree requires direct exchanges between sites, increasing communication costs and synchronization complexity especially as FP-Trees grow larger. After the global FP-Tree is formed, it is partitioned into subtrees, and each site executes FP-Growth independently. The resulting frequent patterns are later aggregated to generate the final global set. In contrast, EGFP optimizes communication by reducing it to only two rounds between the Master and Slave sites. Global frequent patterns of size (m=1) are computed in the first round, and Ancestor Table (AT) data is sent to the Master site in the second. To avoid inter-site exchanges during frequent pattern extraction, EGFP uses a highly compressed GTS structure, which reduces the redundancy of frequent elements. However, FP-Growth relies on a single tree to recursively investigate frequent sets without the need for subtree division. EGFP has several benefits, including simplified frequent pattern extraction without extra processing steps and less communication overhead. To maximize overall efficiency and remove reliance on external techniques, the GTS employs an optimized iterative strategy to find frequent global patterns.

4.4. PROCESS OF EGFP ALGORITHM

A detailed explanation of our methodology is provided in Fig. 3. Its purpose is to extract the set of frequently occurring global patterns in a distributed setting. A whole database must first be horizontally divided into local databases that are assigned to various Slave sites by a Master site. Then, using a Counter Table (CT) that determines the elements' support numbers, an initial LTS including local transactions will be constructed for every slave site after calculating each prefix element's support number. To generate the Ancestor Table (AT) and rebuild the paths of each LTS, a global aggregating phase of the local counters is required. The GTS that corresponds to each frequent pattern is subsequently generated, which provides the basis for effectively extracting global frequent patterns.

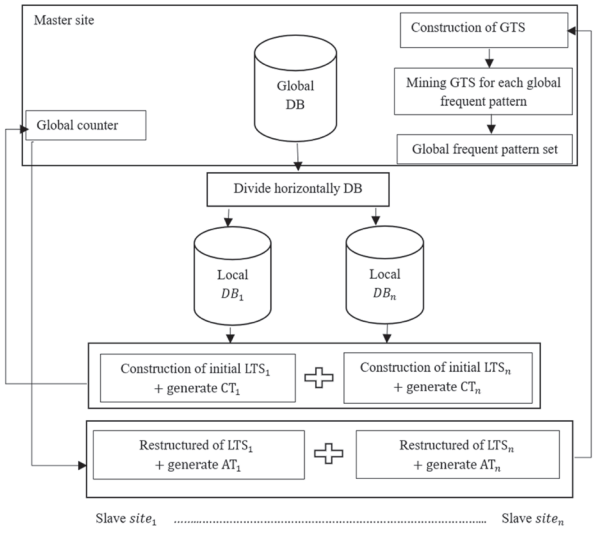


Fig. 3. Process of the Proposed EGFP Algorithm

5. RESULTS AND DISCUSSION

In order to evaluate the performance of our EGFP, we conducted extensive experiments on two kinds of datasets with different characteristics, as shown in Table 1. T10I4D100K and Kosarak are sparse large-scale datasets obtained from FIMI [30]. The T10I4D100K dataset contains a Max TL (Maximum Tree Length) of 29 and an Avg TL (Average Tree Length) of 10.1, showing relatively balanced transaction sizes. Kosarak, on the other hand, has a significantly higher Max TL of 2,498 and an Avg TL of only 8.10, representing a dataset composed of predominantly short transactions with some outliers drastically contributing to tree depth.

We compared EGFP with some previously known algorithms such as PFP-Tree, and CD. The experiments were performed on a system with an Intel® Core™ i7-10875H CPU running at 2.80 GHz, 16 GB of RAM, and operating on Windows 11. To assess scalability and efficiency, the datasets were distributed across 3, 5, and 7 Slave sites. All the programs are implemented in Java using the NetBeans IDE. Communication between sites is facilitated through MPJ Express, a Java-based message-passing library specifically designed for executing parallel applications on multicore processors.

Table 1. Dataset Characteristic

Dataset	Transaction	Items	Max TL (Maximum Tree Length)	Avg TL (Average Tree Length)
T10I4D100K	100000	870	29	10.1
Kosarak	990002	41270	2498	8.10

5.1. ANALYSIS OF RESULTS

The comparative analysis of the PFP-Tree algorithm, CD algorithm, and our proposed EGFP algorithm reveals significant differences in performance, scalability, and efficiency across the T10I4D100K and Kosarak datasets. Fig. 4 compares the execution time of EGFP, PFP-Tree, and CD algorithms for the T10I4D100K dataset, showing how performance scales with the number of Slave sites and minimum support thresholds (MinSupp).

EGFP: Demonstrates superior performance with low execution times across all configurations. For example, with 5 Slave sites and a MinSupp% of 3,5%, EGFP achieves an execution time of 14 seconds, compared to 21 seconds for CD and 19 seconds for PFP-Tree.

CD: Suffers from high communication overhead and redundant computations, leading to significantly higher execution times, especially for lower MinSupp% values.

PFP-Tree: Performs better than CD but is outperformed by EGFP due to the cost of merging FP-Trees and communication overhead.

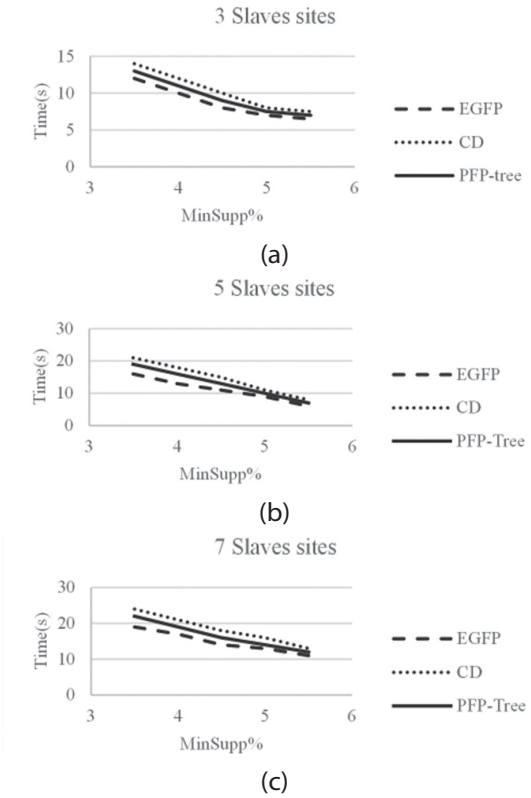


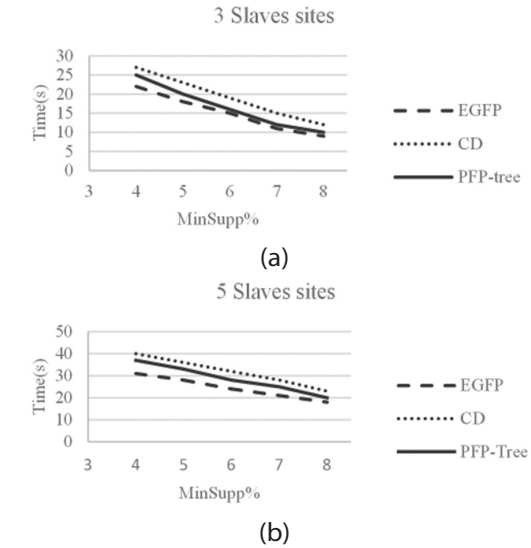
Fig. 4. The running time of T10I4D100K with (**a**) 3 numbers of Slave sites, (**b**) 5 numbers of Slave sites, (**c**) 7 numbers of Slave sites

Fig. 5 compares the execution time of EGFP, PFP-Tree, and CD algorithms for the Kosarak dataset, highlighting the impact of dataset density on scalability.

EGFP: Maintains efficient performance even with the sparse and large Kosarak dataset. For instance, with 7 Slave sites and a MinSupp% of 4%, EGFP achieves an execution time of 40 seconds, compared to 60 seconds for CD and 51 seconds for PFP-Tree.

CD: Struggles with scalability, showing poor performance as the number of Slave sites increases.

PFP-Tree: Performs moderately but is less efficient than EGFP, especially for larger numbers of Slave sites.



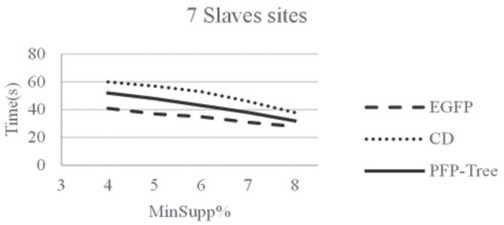


Fig. 5. The running time of Kosarak with (**a**) 3 numbers of Slave sites, (**b**) 5 numbers of Slave sites, (**c**) 7 numbers of Slave sites

The results highlight the strengths and weaknesses of each algorithm in distributed frequent pattern mining:

- EGFP: The use of a Master/Slave architecture and bidirectional communication significantly reduces communication overhead and redundant computations. As a result, EGFP is very scalable and efficient, especially for big datasets Kosarak.
- CD: The broadcast communication approach leads to high communication costs and redundant calculations, making it less efficient for large-scale datasets.
- PFP-Tree: The combination of FP-Trees and communication overhead restricts its scalability and efficiency, even if it outperforms CD.

a) Interpretations

Efficiency of EGFP: EGFP is the most efficient algorithm due to its minimization of redundant computations and concentrated pattern production at the Master site. This is especially evident in its capacity to manage greater numbers of slave sites and capture global frequent patterns using lower minimum support.

Scalability Challenges for CD and PFP-Tree: The communication and computation overheads make CD and PFP-Tree unsuitable for large-scale applications. When the number of slave sites grows, these difficulties become more prominent.

Dataset Impact: The performance gap between EGFP and the other algorithms is more pronounced for the large Kosarak dataset, highlighting EGFP 's ability to handle complex datasets efficiently.

b) Scalability Analysis

The scalability analysis, as shown in Fig. 6, evaluates the performance of EGFP and CD and PFP-Tree as the number of nodes increases.

1. Kosarak Dataset:

- EGFP: Execution time increases gradually with the number of nodes, indicating good scalability. For example, with 5 nodes, EGFP achieves an execution time of 15 seconds, compared to 19 seconds for CD and 18 seconds for FFP-Tree.
- CD and PFP-Tree: Execution time increases significantly with the number of nodes, indicating poor scalability.

2. For T10I4D100K Dataset:

- EGFP: Execution time increases gradually with the number of nodes, maintaining efficient performance.
- CD and PFP-Tree: Execution time increases significantly with the number of nodes, reflecting high communication overhead.

The Key Insights of EGFP Demonstrate excellent scalability, making it suitable for large-scale distributed systems. CD and PFP-Tree Struggle with scalability, particularly for large numbers of nodes and sparse datasets.

Impact of Node Expansion: With an increasing number of Slave nodes, the EGFP algorithm maintained a clear performance edge over PFP-Tree and CD. Even with 7 nodes, EGFP continuously produced faster execution speeds. Although CD and PFP-Tree experienced obvious delays as a result of the growing influence of communication overhead. Efficient communication minimization is a key component of EGFP 's exceptional scalability, enabling it to maintain performance as the system grows.

Performance Trends at Higher Node Counts: As nodes grew, the differences in algorithm efficiency became more evident. At 7 nodes, the increasing communication overload was significantly affecting the performance of PFP-Tree and CD. EGFP provided a distinct advantage, as it benefited from optimized data management and fewer synchronization requests. This capacity allowed it to maintain its efficiency even under high parallelism conditions.

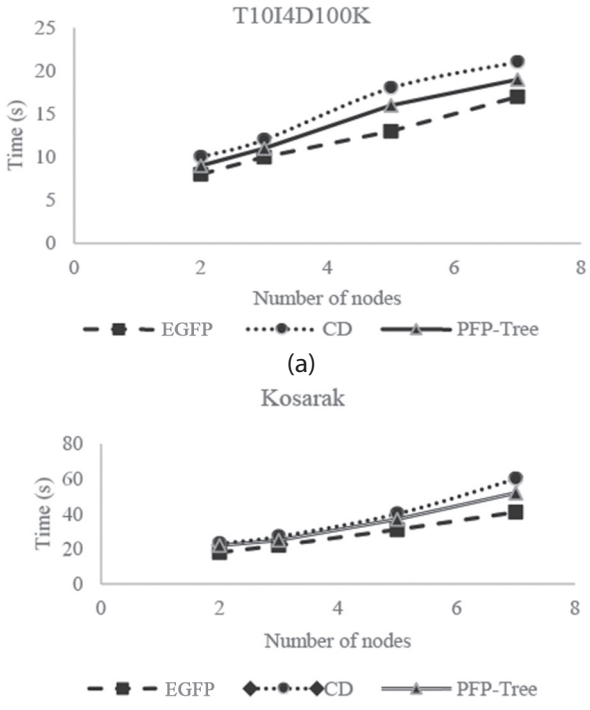


Fig. 6. Scalability of EGFP by various number of nodes for (**a**) T10I4D100K and (**b**) Kosarak with MinSupp = 4%

5.2. DISCUSSION

Our approach employs an iterative process to generate global patterns while leveraging a highly compressed and optimized GTS, ensuring efficient pattern discovery. On the other hand, PFP-Tree relies on conditional sub-tree construction, which reduces redundancy but does not naturally minimize execution overhead. As a result, its computational cost remains significantly higher than that of our algorithm. Additionally, PFP-Tree involves multiple computational steps, such as merging local trees, partitioning the global FP-Tree into subtrees, and executing the FP-Growth algorithm. On the other hand, EGFP removes these complications, which significantly increases execution time and resource efficiency. The CD algorithm exchanges local support values using a straightforward communication paradigm based on all-to-all broadcasting. This strategy generates a lot of candidate sets, which greatly raises the computational and communication overhead even though it is effective in distributing data. A low support threshold causes a rapid increase in the number of generated patterns. Even if parallelization speeds up processing, a significant number of patterns are still extracted overall. EGFP is not a requirement for inter-site communication because it generates all global candidates openly using the GTS. Through the use of the GTS's compression mechanism, this method significantly reduces duplication and communication overhead.

According to the analysis, EGFP performs better on both datasets in terms of efficiency, scalability, and performance than both CD and PFP-Tree. EGFP is the best algorithm for distributed frequent pattern mining because it uses a Master-Slave architecture and bidirectional communication, which significantly reduces the communication cost and unnecessary calculations. With better scalability and faster calculation times than sequential approaches, our algorithm represents a significant advancement in parallel frequent pattern mining. We could focus on investigating fault tolerance strategies to improve EGFP 's resilience in distributed situations and further optimize it for even larger datasets.

6. CONCLUSION

This paper presents a new EGFP algorithm for exploring the discovery of frequent patterns inside a distributed system. Our EGFP remedies the important problems of previous parallel algorithms, including the computational time and communication costs, typically obtained via message exchanges among different sites. The EGFP method compacts and compresses the database using two prefix tree structures. Each Slave site builds its preliminary structure (LTS) to determine the supports of all prefixes, which are subsequently aggregated at the Master site to rearrange the LTS paths for each Slave site by establishing a table including all the ancestors of the local tree nodes. For each element in the local database, the Master site includes ancestral data required for building the Global Tree structure (GTS). The process is car-

ried out iteratively to generate the Global Tree Structure (GTS) for exploring all global patterns. Comparatively to the PFP-Tree and CD algorithms, the performance evaluation of our method on real-world datasets showed its efficiency in speed and scalability.

For distributed systems, real-time data streams, and conditions needing rapid execution, our method, with its optimized approach that limits data processing to a single pass, is extremely fast. In the future work, we plan to concentrate on developing our algorithm with association rule mining to promote efficiency, scalability, and adaptation to complex datasets for the discovery of beneficial patterns and insights.

7. REFERENCES

[1] H. Kargupta, C. Kamath, P. Chan, "Distributed and Parallel Data Mining: Emergence, Growth, and Future Directions", Advances in Distributed and Parallel Knowledge Discovery, AAAI/MIT Press, 2000, pp. 409-416.

[2] R. Agrawal, T. Imieliński, A. Swami, "Mining Association Rules Between Sets of Items in Large Databases", Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington, USA, 25-28 May 1993, pp. 207-216.

[3] P.-N. Tan, M. Steinbach, V. Kumar, "Association Analysis: Basic Concepts and Algorithms", Introduction to Data Mining, Pearson Addison Wesley, 2005, pp. 327-386.

[4] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, Chile, 12-15 September 1994, pp. 487-499.

[5] R. Agrawal, J. C. Shafer, "Parallel Mining of Association Rules", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, 1996, pp. 962-969.

[6] D. W. Cheung, V. T. Ng, A. W. Fu, Y. Fu, "Efficient Mining of Association Rules in Distributed Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, 1996, pp. 911-922.

[7] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, Y. Fu, "A Fast Distributed Algorithm for Mining Association Rules", Proceedings of the 4th International Conference on Parallel and Distributed Information Systems, Miami Beach, FL, USA, 18-20 December 1996, pp. 31-42.

[8] M. Z. Ashrafi, D. Taniar, K. Smith, "ODAM: An Optimized Distributed Association Rule Mining Algorithm", IEEE Distributed Systems Online, Vol. 5, No. 3, 2004, pp. 1-18.

[9] A. Schuster, R. Wolff, "Communication-Efficient Distributed Mining of Association Rules", ACM SIGMOD Record, Vol. 30, No. 2, 2001, pp. 473-484.

[10] E.-H. Han, G. Karypis, V. Kumar, "Scalable Parallel Data Mining for Association Rules", ACM SIGMOD Record, Vol. 26, No. 2, 1997, pp. 277-288.

[11] T. Shintani, M. Kitsuregawa, "Hash Based Parallel Algorithms for Mining Association Rules", Proceedings of the 4th International Conference on Parallel and Distributed Information Systems, Miami Beach, FL, USA, 18-20 December 1996, pp. 19-30.

[12] L. Harada, N. Akaboshi, K. Ogihara, R. Take, "Dynamic Skew Handling in Parallel Mining of Association Rules", Proceedings of the 7th ACM International Conference on Information and Knowledge Management, Bethesda, MD, USA, 3-7 November 1998, pp. 76-85.

[13] S. K. Tanbeer, C. F. Ahmed, B.-S. Jeong, Y.-K. Lee, "Efficient Single-Pass Frequent Pattern Mining Using a Prefix-Tree", Information Sciences, Vol. 179, No. 5, 2009, pp. 559-583.

[14] H. Huang, X. Wu, R. Relue, "Association Analysis with One Scan of Databases", Proceedings of the IEEE International Conference on Data Mining, Maebashi, Japan, 9-12 December 2002, pp. 629-632.

[15] G. Grahne, J. Zhu, "Fast Algorithms for Frequent Itemset Mining Using FP-Trees", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 10, 2005, pp. 1347-1362.

[16] J. Han, J. Pei, Y. Yin, "Mining Frequent Patterns Without Candidate Generation", Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16-18 May 2000, pp. 1-12.

[17] A. Javed, A. Khokhar, "Frequent Pattern Mining on Message Passing Multiprocessor Systems", Distributed and Parallel Databases, Vol. 16, 2004, pp. 321-334.

[18] O. R. Zaïane, M. El-Hajj, P. Lu, "Fast Parallel Association Rule Mining Without Candidacy Generation",

Proceedings of the IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November - 2 December 2001, pp. 665-668.

[19] C. Fernandez-Basso, M. D. Ruiz, M. J. Martin-Bautista, "New Spark Solutions for Distributed Frequent Itemset and Association Rule Mining Algorithms", Cluster Computing, Vol. 27, No. 2, 2023, pp. 1217-1234.

[20] R. A. Rizvee, C. F. Ahmed, M. F. Arefin, C. K. Leung, "A New Tree-Based Approach to Mine Sequential Patterns", Expert Systems with Applications, Vol. 242, 2024, p. 122754.

[21] V. Q. P. Huynh, J. Küng, "FPO Tree and DP3 Algorithm for Distributed Parallel Frequent Itemsets Mining", Expert Systems with Applications, Vol. 140, 2020, p. 112874.

[22] L. Diop, C. T. Diop, A. Giacometti, A. Soulet, "Pattern on Demand in Transactional Distributed Databases", Information Systems, Vol. 104, 2022, p. 101908.

[23] C. Wu, H. Jiang, "Research on Parallelization of Frequent Itemsets Mining Algorithm", Proceedings of the IEEE 6th International Conference on Cloud Computing and Big Data Analysis, Chengdu, China, 23-25 April 2021, pp. 210-215.

[24] B. Mudumba, M. F. Kabir, "Mine-First Association Rule Mining: An Integration of Independent Frequent Patterns in Distributed Environments", Decision Analytics Journal, Vol. 7, 2024, p. 100434.

[25] D. Fan, J. Wang, S. Lv, "Optimization of Frequent Item Set Mining Parallelization Algorithm Based on Spark Platform", Discover Computing, Vol. 27, No. 1, 2024, p. 38.

[26] Y. Rochd, I. Hafidi, "Frequent Itemset Mining in Big Data with Efficient Distributed Single Scan Algorithm Based on Spark", International Journal of Intelligent Engineering and Systems, Vol. 18, No. 2, 2025, p. 101908.

[27] Y. Djenouri, M. Comuzzi, D. Djenouri, "SS FIM: Single Scan for Frequent Itemsets Mining in Transactional Databases", Advances in Knowledge Discovery and Data Mining, Springer, 2017, pp. 644-654.

[28] T. Tassa, "Secure Mining of Association Rules in Horizontally Distributed Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 4, 2014, pp. 970-983.

[29] A. Vasoya, N. Koli, "Mining of Association Rules on Large Database Using Distributed and Parallel Computing", Procedia Computer Science, Vol. 79, 2016, pp. 221-230.

[30] B. Goethals, M. J. Zaki, "Advances in Frequent Itemset Mining Implementations", ACM SIGKDD Explorations Newsletter, Vol. 6, No. 1, 2004, pp. 109-117.

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