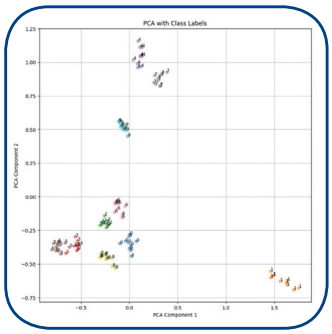
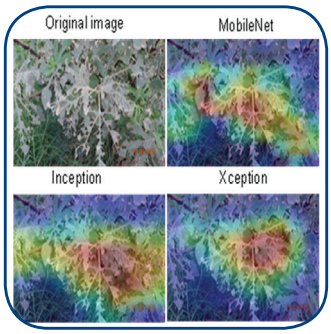
Volume 16, Number 8, 2025 ISSN 1847-6996

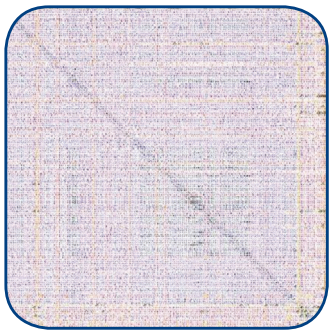
FERIT
FACULTY OF ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY **OSIJEK**

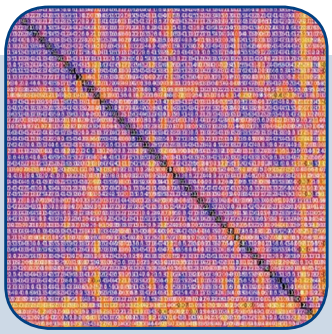
IJECES International Journal of Electrical and Computer Engineering Systems

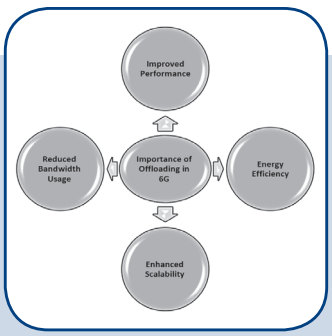
International Journal of Electrical and Computer Engineering Systems

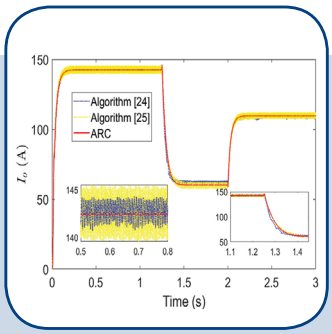












ISSN: 1847-6996

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 8, 2025 | Pages 565 - 640

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

Optimizing Computation Offloading in 6G Multi-Access Edge Computing Using Deep Reinforcement Learning

Original Scientific Paper

Mamoon M. Saeed
Department of Communications and Electronics Engineering,
University of Modern Sciences (UMS), Sana'a, Yemen
mamoon530@gmail.com

Rashid A. Saeed*
College of Business and Commerce,
Lusail University, Lusail, Qatar
rabdelhaleem@lu.edu.qa

Hashim Elshafie
Department of Computer Engineering,
College of Computer Science,
King Khalid University,
Main Campus, Al Farah,
Abha 61421,
Kingdom of Saudi Arabia, KSA
helshafie@kku.edu.sa

*Corresponding author

Ala Eldin Awouda
Mechanical Engineering Department,
College of Engineering,
Bisha University, Bisha, KSA
aadam@ub.edu.sa

School of Electronics of Engineering,
Faculty of Engineering,
Sudan University of Science and Technology,
Khartoum, Sudan

Zeinab E. Ahmed
Department of Computer Engineering,
University of Gezira,
Wad-Madani, Sudan
Zeinab.e.ahmed@gmail.com

Mayada A. Ahmed
School of Electronics of Engineering,
Faculty of Engineering,
Sudan University of Science and Technology,
Khartoum, Sudan
mayadanott13@gmail.com

Rania A Mokhtar
School of Electronics of Engineering,
Faculty of Engineering,
Sudan University of Science and Technology,
Khartoum, Sudan
ragiliter@gmail.com

Abstract *– One of the most important technologies for future mobile networks is multi-access edge computing (MEC). Computational duties can be redirected to edge servers rather than distant cloud servers by placing edge computing facilities at the edge of the wireless access network. This will meet the needs of 6G applications that demand high reliability and low latency. At the same time, as wireless network technology develops, a variety of computationally demanding and time-sensitive 6G applications appear. These jobs require lower latency and higher processing priority than traditional internet operations. This study presents a 6G multi-access edge computing network design to reduce total system costs, creating a collective optimization challenge. To tackle this problem, Joint Computation Offloading and Task Migration Optimization (JCOTM), an approach based on deep reinforcement learning, is presented. This algorithm takes into consideration several factors, such as the allocation of system computing resources, network communication capacity, and the simultaneous execution of many calculation jobs. A Markov Decision Process is used to simulate the mixed integer nonlinear programming problem. The effectiveness of the suggested algorithm in reducing equipment energy consumption and task processing delays is demonstrated by experimental findings. Compared to other computing offloading techniques, it maximizes resource allocation and computing offloading methodologies, improving system resource consumption. The presented findings are based on a set of simulations done in TensorFlow and Python 3.7 for the Joint Computation Offloading and Task Management (JCOTM) method. Changing key parameters lets us find out that the JCOTM algorithm does converge, with rewards providing a measure of its success compared to various task offloading methods. 15 users and 4 RSUs are placed in the MEC network which faces resource shortages and is aware of users. According to the tests, JCOTM offers a lower average system offloading cost than local, edge, cloud, random computing and a game-theory-based technique. When there are more users and data, JCOTM continues to manage resources effectively and shows excellent speed in processing demands. It can be seen from these results that JCOTM makes it possible to offload efficiently as both server loads and user needs change in MEC environments.*

Keywords: *deep reinforcement learning, sixth generation (6 G), multi-edge computing (MEC), offloading, deep Q-network*

Received: May 8, 2025; Received in revised form: June 7, 2025; Accepted: June 12, 2025

1. INTRODUCTION

The upcoming launch of 6G networks promises a paradigm leap in connectivity in the quickly changing telecoms industry, bringing in a new era of blazingly fast speeds, responsiveness, and dependability [1]. To satisfy the demanding specifications of next-generation networks, it is essential to integrate cutting-edge technologies as the need for high-performance mobile apps keeps growing. Multi-access Edge Computing (MEC) stands out among these technologies as a crucial remedy because it makes it possible to distribute computational jobs closer to the edge of wireless access networks, which lowers latency and improves system efficiency overall [2].

Multi-Access Edge Computing (MEC) is emerging as a transformative technology that significantly enhances network performance by reducing latency through localized data processing. This capability is essential for real-time applications, such as the Internet of Things (IoT) and augmented reality, where rapid response times are crucial. MEC further optimizes bandwidth efficiency by offloading processing tasks from the core network, leading to better resource utilization. The technology also improves user experiences by facilitating seamless interactions and supports a broad spectrum of IoT applications through real-time analytics at the edge. Additionally, MEC enhances security and privacy by minimizing data transmission over networks, thus aiding compliance with privacy regulations. Its scalable architecture accommodates the growing number of devices and applications in today's fast-paced technological environment. Overall, MEC stands out as a pivotal solution in modern networking, optimizing system performance and alleviating pressure on central data centers [3].

Deep Reinforcement Learning (DRL) is a state-of-the-art method for optimizing computation offloading strategies in 6G environments in MEC. Network operators and service providers can intelligently and responsively distribute computing jobs to edge servers by utilizing DRL algorithms' adaptive and self-learning properties [4]. The main requirements of 6G networks, which place a premium on low latency, high dependability, and effective resource use to serve a wide range of cutting-edge applications from augmented reality to driverless cars, are completely met by this integration. In light of this, conducting research and building a deep reinforcement learning-based computation offloading framework designed especially for 6G multi-access edge computing networks is crucial [5].

This framework explores the complex interactions between DRL algorithms and computation offloading techniques to optimize task allocation, improve system performance, and simplify resource management in the context of sophisticated mobile networks [6]. This study aims to push the limits of innovation in mobile communications by investigating the synergies between DRL and computation offloading in the context of 6G MEC networks. It provides a glimpse into the revolutionary potential of AI-driven solutions in influencing the future of network architecture and service delivery [7].

The upcoming 6G technology revolution will reshape different business sectors by improving network connectivity and latency performance alongside the capability to implement time-sensitive software applications. The fundamental development behind network edge transformation rests upon Multi-Access Edge Computing (MEC) for handling computational resources local to the network boundary. Strategic computational load distribution from resource-limited devices to edge computing servers constitutes offloading, so applications and performance gain better efficiency and results [8].

6G networks require effective resource management because of the large number of IoT devices and complex application systems that operate within these networks. Smart devices such as smartphones, along with sensors and autonomous vehicles, use the capability of offloading to transfer complex processing duties to edge servers situated nearby. The device offloading approach helps both devices conserve power and reduce battery drain while speeding up responses and enhancing the user experience altogether [9].

The main parts of offloading execution within MEC consist of many essential elements. The process demands effective decision systems for finding suitable offloading targets among tasks alongside optimal edge server destinations. The implementation of MEC offloading requires an evaluation of the edge server workload together with network performance and the exact needs for each task, including latency tolerance and data magnitude.

The combination of artificial intelligence (AI) with machine learning (ML) methods greatly improves the capability to make offloading decisions. Records from both history and current network situations supply AI algorithms with data to find optimal offloading techniques that enhance the effectiveness of job distribution along with resource organization. The intelligent system delivers both enhanced performance and better 6G network resilience because it rapidly adjusts to both conditions and potential failures [10].

Security, together with privacy issues, represents the highest concern throughout the offloading process. Edge servers require both strong encryption and protected communication protocols to ensure the security of sensitive data transferred from users. 6G networks must guarantee data confidentiality and integrity because such measures are vital for meeting user trust requirements and following regulatory standards when they support a diverse set of applications and multiple devices.

The deployment of offloading systems within 6G

MEC encounters multiple difficulties. Device capability management, alongside network component interoperability and quality of service delivery requirements, make up the implementation challenges of these networks. Offloading methods need to establish coherent processing speed, energy utilization, and networking reliability ratios to deliver continuous user experiences.

The wireless network progression from 5G to 6G technology establishes a new standard of connectivity through heightened speed, together with inferior latency and better capacity. Multi-Access Edge Computing (MEC) serves as the essential tool for network evolution since it distributes computation and storage capabilities near final user locations. An efficient mechanism for complex processing called offloading works effectively because of network decentralization. The transfer of computational operations from smartphone platforms and IoT sensors to enhanced edge servers through offloading describes this process. Application performance can be both optimized and real-time processing and low-latency requirements fulfilled through this essential resource optimization procedure [11].

6G networks must manage unimaginable numbers of linked devices as well as applications, which include autonomous systems and AR and VR applications, because they require increased computational power. Extensive application processing needs exceed the capabilities of local devices, thus making these applications require edge server support. The shifted task execution through offloading activates the edge servers to perform computations with better efficiency and thus improves system performance at large. Several important advantages emerge from offloading, according to the diagram given in Fig. 1.

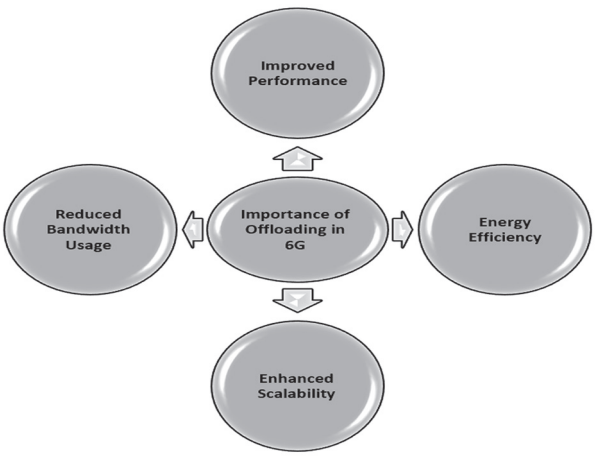


Fig. 1. Offloading Significant Benefits

Edge servers enable applications to lower latency while increasing response times because of their processing capabilities, which serve user satisfaction primarily in real-time applications. Devices with limited resources can save power through edge-based transferring of complex processing demands. The operation of IoT devices that depend on battery power specifically requires this approach. Network scalability becomes possible through offloading because it enables distributed workloads across several edge servers instead of overloading devices and cloud-based resources independently. The offloading process lowers the amount of data that needs to flow to cloud servers for analysis, resulting in reduced bandwidth usage and network congestion occurrences.

1.1. DECISION-MAKING IN OFFLOADING

Making a task of offloading a decision involves evaluating multiple conditions, which should include [12], [13], and [14]:

- Task characteristics and specific parameters such as computational difficulty, together with data volume and response time demands, play an essential role in deciding whether a computation should be transmitted off-device.
- Professional offloading decisions require immediate evaluation of network conditions, including server load, bandwidth availability, and network latency data.
- For successful offloading purposes, it is fundamental to recognize the processing abilities and energy use status of initiating devices.

Advanced algorithms together with models serve as tools to assist in this type of decision-making framework. These may include:

- Advanced offloading algorithms make real-time compensations to fluctuating networks and system work demands for better offloading results.
- The predictive analysis uses AI and ML technologies to develop offloading strategies by processing historical and present network data. The technologies apply past data learning to optimize their functions in distributing resources across teams and assigning tasks more effectively.

1.2. SECURITY AND PRIVACY CONCERNS

Data security and privacy emerge as essential factors after its transport to edge servers. Key considerations include [8]:

- All transmission of sensitive information to edge servers require encryption to stop unauthorized users from accessing the data.
- A secure transmission protocol system must be established to maintain safe data exchange between devices and edge servers.
- The protection of sensitive data requires implementing measures that allow authorized users and devices to access it.
- The protection of user privacy requires organizations to maintain strict obedience to data protection laws like GDPR and HIPAA.

1.3. CHALLENGES IN IMPLEMENTING OFFLOADING

Several key obstacles exist when implementing offloading via MEC in 6G networks [15]:

- The compatibility between different devices and applications, and network components becomes complex because standard protocols and interfaces are needed.
- Developing a favorable quality of service (QoS) remains vital to achieving user satisfaction because sensitive applications require stable bandwidth along with minimal latency.
- Efficient operations of edge server's dependent on resource management create significant problems because of load balancing difficulties and resource allocation demands.
- Modern computational frameworks are needed to execute effective operations for real-time decision-making between abrupt condition changes.

1.4. FUTURE DIRECTIONS

6G technology development indicates that the following directions for MEC offloading will emerge [16]:

- The application of edge intelligence combined with artificial intelligence at horizontal distribution points results in improved decision-making ability that enables dynamic on-the-fly offloading adjustments based on present conditions.
- Federated learning provides decentralized model training that keeps sensitive data on user devices and enables collective learning through decentralized training.
- The adoption of decentralized architectural design brings better resistance and decreases dependency on centralized cloud infrastructure, which enables better offloading outcomes.
- Edge computing operations demand specialized security frameworks that need development according to specific edge needs, since platform evolution will be mandatory.

6G Multi-Access Edge Computing depends on offloading as its core resource optimization and application performance enhancement mechanism [17]. The method of moving computations to edge servers as a strategic step helps handle next-gen application requirements and delivers better energy efficiency and adaptable system capacity [18]. The complete realization of 6G MEC requires attention towards smart calculation management techniques as well as secure protection frameworks and resolution of operational obstacles alongside technological evolution [19].

Through carefully assessing these technologies and their consequences for the mobile ecosystem, our research strives to pave the way for more efficient, intelligent, and responsive network infrastructures capable of addressing the rising needs of the digital age. Here's a summary of the primary contributions:

- The communication and task computation flow are simulated to determine the system delay and energy consumption formula.
- The mixed integer nonlinear programming problem is challenging to solve directly because it is NP-hard. Thus, we convert it into a Markov Decision Process and propose a combined computation offloading and task migration optimization (JCOTM) technique based on deep reinforcement learning.

The JCOTM algorithm's convergence and efficacy are demonstrated by experimental performance. Our suggested approach can lower processing latency and equipment energy usage in various system contexts compared to alternative computation offloading strategies.

The remaining sections of this paper are arranged as follows: In Section III, we outline the joint optimization issue and the 5 G-based 6G user-aware multi-access edge computing network architecture. Section IV introduces the Deep Q-Network and the JCOTM algorithm's comprehensive process. Section V presents the simulation parameters and outcomes, while Section VI wraps up our investigation.

2. RELATED WORK

This part of the study examines previous studies that aimed to improve how computation is distributed in Multi-Access Edge Computing (MEC). The literature is typically organized into binary offloading and making decisions about partial execution.

The tasks can be processed where they are created or sent to the MEC server for completion. [20] analyzes what the best single-user performance is when binary offloading is used in ultra-dense networks. It highlights situations when binary offloading might be useful, and [21] develops an approach using both games and optimization for better results. They help to see the role of server-based processing and when it is more beneficial than running tasks locally, showing the need to decide wisely.

Several experts have used advanced techniques such as reinforcement learning (RL) to manage the complicated issues in MEC. For example, [22] optimizes the use of resources and UAV routes at once, which demonstrates how RL helps save power in fast-changing situations. [23] found that RL can handle some of the MEC's important challenges, such as those related to mobility and managing changing channels.

This framework (MELO, presented in [24]) demonstrates a decision-making system that uses reinforcement learning and formulates the tasks as a Markov Decision Process. It points to more use of machine learning to assist in making choices in the context of

MEC. Alternatively, users with partial offloading can pass some of their work to the MEC server when required. The research in [25] deals with offloading cloud tasks to more than one device, with wireless interference and separable semi-definite relaxation in mind. This technique points out how partial offloading is flexible and able to ensure resources are used well, as different users require them.

Also, techniques such as convex optimization and segmentation optimization are used to optimize resource usage in multi-user MEC systems [26, 27]. They reveal how much effort is put into both minimizing expenses and cutting back on delays that put efficiency and results in balance.

Different approaches, for example, [28], are now considering how load on servers affects energy use, reflecting the increased awareness that workloads and infrastructure affect each other. Unlike the strategies of the papers mentioned in [7], the authors of [29] and [30] stressed that the best way to reduce offloading costs is to pay attention to energy use, processing time, and delay.

[31] and [32] identify that with the advent of 6G, intelligent user edge computing relies heavily on deep reinforcement learning for request offloading and choosing resources. The article [33] also introduced the UMAP algorithm, which further demonstrates the benefits of combining different advanced algorithms to boost MEC performance.

Simply put, while the use of binary offloading helps with straightforward situations, using partial offloading and more advanced techniques allows both the application and network to adapt and respond to what the user needs. The field is seeing how delicate performance, resource management, and what users experience are balanced in MEC.

This work [34] presented the UMAP algorithm that connects handling UAV movement to connecting users with access to a network, all through frequent optimization. With deep reinforcement learning (DRL), the system learns to improve both where UAVs go and how they are associated, which helps reduce the amount of energy used and waiting time in the system. This way of working highlights that DRL is useful in environments that keep adapting, so agents can react to current circumstances.

Even so, due to how complicated DRL models are to train, it can be quite challenging regarding whether they converge and the number of computer resources required, which means they aren't always practical everywhere. Even though the advancement to closed-form MU transmission power helps efficiency, it may not be suited for different operating settings. To sum up, UMAP reflects important progress in MEC by offloading data, but points out that further study is needed to improve its work in different situations. This stresses the need to blend different optimization strategies to help the entire system perform better.

The proposed system involves 5G technology and 6G user-aware Multi-access Edge Computing network (VAMECN) elements, which consist of 6G users, roadside units, and cloud servers to handle upcoming 5G network offloading functions. The proposed method addresses the reduction of system delays along energy consumption optimization. The proposed solution adopts deep reinforcement learning to create JCOTM for addressing problems through performance demonstrations

3. SYSTEM MODEL AND PROBLEM FORMULATION

As illustrated in Fig. 2, we examine a 5 G-based user-aware Mobile Edge Computing (MEC) network architecture, which comprises N users, M Roadside Units (RSUs), and a cloud server. We define the index sets for users and RSUs as $U = \{1,2,\ldots,N\}$ and $M = \{0,1,2,\ldots, M, M+1$, respectively. Here, m=0 represents the local computing device, while m=M+1 denotes the cloud server [35]. The indices between 0 and M+1 correspond to the edge servers. We assume that the RSUs are uniformly distributed along the road, each covering a consistent area R. Each RSU is equipped with one or more MEC servers, positioning it as an edge computing node.

To effectively simulate the users' trajectories over time, we represent the continuous road as a series of discrete traffic areas. In Fig. 2, a typical urban road network is segmented into PPP discrete areas, indexed by the set P={1, 2,…, P} We will next address the optimization problem related to joint computation offloading and task migration over a defined period T [36]. This period is divided into t_i time slots denote as T={1, 2,…, t_i} at the initial time slot t_0, users are randomly allocated within the network. As users move, they can either remain in their current traffic area or transition to an adjacent one. The transition probability from location l to l' for users can be expressed as $Pr(l' \mid l)$. For instance, $Pr(l \mid l)$=0.5 indicates a 50 % probability that a user will remain in the same location. We assume the probability of moving from l to l' (where $l \neq l'$) is equivalent, allowing us to calculate the position transfer probability as $Pr(l' \mid l) = (1\text{-}Pr(l \mid l))/2$ [37]

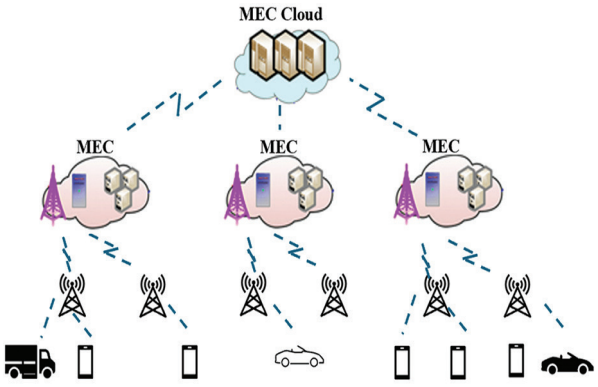


Fig. 2. The architecture of our proposed 6G MEC network

Each user of equipment (UE) is assumed to have a single compute-intensive task that requires processing. There exists a one-to-one correspondence between UEs and users. The *n-th* task can be characterized by a triple $\alpha_n, \beta_n, \gamma_n$, where $n \in v$, α_n denotes the data size of the task, β_n represents the required CPU cycles for task completion, and γ_n indicates the maximum allowable delay. The binary offloading decision is represented by $x_{mn} \in \{0,1\}$, for m ∈ M, $n \in v$. Specifically, x_{mn}=0 indicates that task n will be processed on the local UE, while x_{mn}=1 signifies that the task will be offloaded to the *m-th* MEC server [38].

Notably, when m=M+1, the task n is offloaded to the cloud server. The system's offloading decisions at the t^{th} time slot is represented by the set

$X(t)=\{x_{01}(t),\ldots, x_{(M+1)1}(t), x_{02}(t),\ldots, x_{(M+1)N}(t)\}$. It is important to note that each UE can connect to either one RSU or the Base Station (BS) during a time slot [39], thereby necessitating the following constraint:

$$\sum_{m=0}^{M+1} x_{nm}(t) = 1, \forall n \in u \quad (1)$$

Subsequently, we will explore the communication and computation models of the User-Aware MEC Network (UAMECN) system [40], deriving expressions for delay and energy consumption.

3.1. MODEL OF COMMUNICATION

Base station-based communication algorithms create transmission delays that happen when uploading cloud-server data. The rising number of tasks between users causes resource contention that produces network instabilities alongside extended delays. MEC addresses the network bottleneck by establishing server locations that are nearer to user locations [41].

Non-orthogonal multiple Access (NOMA) stands as a vital 5G technology that enables non-orthogonal transmission during signal transmission and incorporates interference data actively while implementing successive interference cancellation (SIC) for accurate signal demodulation at receivers [42]. The receiver implementation of NOMA provides additional complexity compared to OFDMA but delivers higher spectral efficiency. The VAMECN system adopts NOMA for UE-to-BS communication links yet employs OFDMA for UE-to-RSU links because the BS must serve more users [43].

The channel state follows a time-dependent finite continuous value pattern through which the new state appears solely from the previous state. The paper transforms the state values into L discrete levels before representing them as finite-state Markov chains. Channel gain is a crucial parameter for calculating data transmission rates. We denote the channel gain of the wireless link between the user n and RSU m at time t as $\Gamma_n^m(t)$, calculated using the formula [44]:

$$\Gamma_n^m(t) = g_n^m \, d_{n,m}^{-r} \quad (2)$$

Here, g_n^m represents small-scale fading, $d_{(n,m)}$ is the distance between the user n and RSU m, and r is the path loss index. The term $d^{-r}_{n,m}$ signifies path loss. The state space of the Markov chain is represented as $L = \{Y_1, Y_2, \ldots, Y_L\}$, and $\Gamma_n^m(t)$ is classified as Y_1 when $\Gamma_1^* \le \Gamma_n^m < \Gamma_2^*$; Γ_n^m is quantified as Y_2 when $\Gamma_2^* \le \Gamma_n^m < \Gamma_3^*$; and so on, Γ_n^m is quantified as Y_L when $\Gamma_n^m \ge \Gamma_L^*$. ψ_{gs}, $h_s(t)$ is the transition probability that the channel gain shifts from the state g_s to state h_s. Consequently, the following is the $L \times L$ channel state transition probability matrix.

$$\Psi_n^m(t) = [\psi_{gs}, h_s(t)]L \times L \quad (3)$$

Where ψ_{gs}, $h_s(t) = Pr(\Gamma_n^m(t+1) = hs \mid \Gamma_n^m(t) = g_s)$, and $g_s, h_s \in L$. Thus, according to the Shannon formula, the data transmission rate between the user and RSU at time slot t is calculated as follows.

$$u_n^m(t) = b_n^m(t)\, log_2\left(1 + \frac{p_n^m(t)\Gamma_n^m(t)}{\sigma^2}\right) \quad (4)$$

Where $b_n^m(t)$ the orthogonally allotted bandwidth from RSU m to user n, $m \in M$ and $n \in$ U. $b_n^m(t)$, is denoted by the Gaussian white noise power is represented by σ^2, while transmission power is indicated by $b_n^m(t)$ [45].

Next, we talk about how users and BS communicate. For instance, in the uplink, each UE will be assigned a distinct transmission strength, and signals will be superimposed to send when multiple users are connected to the BS at the same time.

$$X_n(t) = \sqrt{P_n}x_n(t) + \sum_{i \in V, i \ne n} \sqrt{P_i}x_i(t) \quad (5)$$

calculates the superimposed signal, where x_n and x_i stand for the target user n's and other users' transmission signals, respectively. The signal that was received is

$$y_n(t) = \Gamma_n^{M+1}(t)X_n(t) + \sigma^2 \quad (6)$$

After obtaining the data, the BS user rises out of SIC decoding in the decreasing order of channel gain. The interference signal for user n is the sum of the signals with lower equivalent channel gain [46]. In the declining sequence of their channel gains, we assume that N users share the same channel: $Y_1^{M+1} \ge Y_2^{M+1} \ge Y_N^{M+1}$. The data transmission rate $u_n^{M+1}(t)$ and the interference signal $I_n(t)$ If the user is therefore

$$I_n(t) = \sum_{i=n+1}^{N} P_i(\Gamma_n^{M+1}(t))^2 \quad (7)$$

$$u_n^{M+1}(t) = b_n^{M+1}(t) log_2\left(1 + \frac{p_n^{M+1}(t)(\Gamma_n^{M+1}(t))^2}{\sigma^2 + I_n(t)}\right) \quad (8)$$

Equation (9) can therefore be used to evenly express the user *n's* data transmission rate [47].

$$R_n^m(t) = x_{nm}(t)u_n^m(t), \forall n \in U, m \in M \quad (9)$$

The following displays the task *n's* energy usage and communication delay.

$$T_{nm}^{comm}(t) = \frac{\alpha_n(t)}{R_n^m(t)} \quad (10)$$

$$E_{nm}^{comm}(t) = P_n^m(t)T_{nm}^{comm}(t) \quad (11)$$

$$T_{nm}^{comm}(t) = \sum_{m=1}^{M+1} T_{nm}^{comm}(t)\, x_{nm}(t), \forall n \in U \quad (12)$$

$$E_{nm}^{comm}(t) = \sum_{m=1}^{M+1} E_{nm}^{comm}(t)\, x_{nm}(t), \forall n \in U \quad (13)$$

where $\alpha_n(t)$ is the amount of data left over from the n task. Since $m = 0$ indicates that the work will be processed locally, there is no transmission delay, and no energy consumption, hence in this case, the value of m starts at 1 instead of 0.

3.2. MODEL OF COMPUTATION

User n's task will be sent from the cloud server to the MEC server for computation when $x_{nm}(t)=1$, $m \in M\backslash\{0\}$. The calculation capability of the server m, commonly referred to as the CPU rate, is represented by the symbol f_m [48]. In particular, the local CPU rate is shown by f_0, and the cloud server's CPU rate is indicated by f_{M+1}. Because edge servers have distinct hardware configurations $f_0 \ll f_m \ll f_{M+1}$, $m \in M\{0, M+1\}$, They are generally more powerful than UE. The distribution of computer resources is not average.

Only one task or one task slice may be completed by each CPU (single core) in each time slot. To simplify the computation model, we assume that every UE has equal entitlement to obtain computing resources [49]. This implies that if n users decide to offload jobs to the same server, the computing resources allotted to each task are $f_{m/n}$. As a result, we can determine the CPU rate assigned to the user n by using the formula

$$f_m^{avg} = \begin{cases} f_0, m = 0 \\ \frac{f_m}{\sum_{i \in u} x_{mi}}, m \in M\backslash\{0\} \end{cases} \quad (14)$$

It is therefore possible to express the processing time for the user n as

$$T_{nm}^{comp}(t) = \sum_{m=0}^{M+1} \frac{\beta_n(t)}{f_m^{avg}} x_{nm}(t), \forall n \in U \quad (15)$$

where $\beta_n(t)$ is the remaining number of CPU cycles needed by the user n during the time slot t. It goes without saying that as server processing capacity rises, computation delay falls. In the meantime, as it influences the CPU time allotted to each user, the server load is also a crucial consideration. The energy used by local equipment in the absence of ask offloading is denoted by $E_{nm}^{comp}(t)$ [50].

$$E_{nm}^{comp}(t) = \mu\beta_n(t)(f_0)^2 T_{nm}^{comp}(t) x_{n0}(t), \forall n \in U \quad (16)$$

Therefore, the energy consumption for user n. Where the effective switched capacitance is represented by μ=10^{-11} [51].

3.3. FORMULATION OF THE PROBLEM

Through the explanation above, we have represented the computation and communication process. Based on our earlier work, we formulate the job completion delay and UE's energy usage as follows.

$$Cost(t) = \xi_t \sum_{n \in U}(T_{nm}^{comm}(t) + T_{nm}^{comp}(t)) + \xi_e \sum_{n \in U}(E_{nm}^{comm}(t) + E_{nm}^{comp}(t)) \quad (17)$$

where $\xi_t, \xi_e \in [0,1]$ are two scalar weights of energy consumption and latency, respectively. Keep in mind that the system latency is the highest of all task computation and communication delays. Consequently, the following is an expression for the joint computation offloading and task migration optimization problem

$$(JCOTM): min \sum_T Cost(t) \quad (18)$$

Subject to:

$$x_{nm} \in \{0,1\}, \forall n \in U, m \in M \quad (19)$$

$$\sum_{m=0}^{M+1} x_{nm}(t) = 1, \forall n \in U \quad (20)$$

$$\sum_{n \in U, m \in M} b_n^m(t) \leq B \quad (21)$$

$$\sum_T (T_n^{comm}(t) + T_n^{comp}(t)) \leq \gamma_n \forall n \in U \quad (22)$$

Table 1 lists the definitions and notations used in this paper. The challenge of optimization, Multiple variable constraints, makes JCOTM a non-convex mixed-integer linear programming issue. The correlation between the variables makes it challenging for us to solve it. As a result, we provide a proposed technique based on Deep Reinforcement Learning (DRL) and model the original problem as a Markov Decision Process (MDP) [52].

Table 1. Notations used in this paper

Notation	Definition
U, N	Index set/number of users
M, m	Index set/number of RSUs
p, P	Index set/number of traffic areas
l, L	The set/number of channel gain states
$x_{nm}(t)$	$x_{nm}(t) = 1$ if task n is offloaded to server m at time slot t, otherwise, $x_{nm} = 0$
R	Coverage range of one RSU
β_n	Required number of CPU cycles of task n
α_n	Data size of task n
γ_n	max delay limit of task n
γ_l	The l-th state value after the channel gains discretization
$Pr(l' \mid l)$	Transition probability from location l to l' of 6G users
$d_{n,m}^{-r}$	Pass loss
g_n^m	Small-scale fading
$\Gamma_n^m(t)$	Channel gain of the communication link between 6G user n and RSU m at time slot t
$b_n^m(t)$	Bandwidth of the link between 6G user n and RSU m at time slot t
$\psi_{g,s}, h_s(t)$	Transition probability from state h_i to h_j of $\Gamma_n^m(t)$
σ^2	Gaussian white noise power
P_n	Transmission power of 6G users n
T_n^{comm}, T_n^{comp}	Communication/computation delay of task n
$R_n^m(t)$	Data transmission rate from 6G user n to RSU m
f_m	Computation capability of server m
E_n^{comm}, E_n^{comp}	Communication/computation energy consumption of task n
ξ_t, ξ_e	Scalar weight of delay/energy consumption
μ	The effective switched capacitance

4. OPTIMIZING COMPUTATION OFFLOADING BASED DRL

Reinforcement Learning (RL), a subfield of artificial intelligence, is the third machine learning technique, following Unsupervised Learning (UL) and Supervised

Learning (SL). Reinforcement learning involves an agent interacting with its surroundings to learn what actions would result in the greatest reward [53]. In supervised and unsupervised learning, the data is static and does not require interaction with the environment, such as picture recognition. The deep network can learn the difference between samples by iterative training if sufficient samples are provided. However, RL is a dynamic and interactive learning process, and constant contact with the environment also generates the necessary data.

As a result, reinforcement learning incorporates more objects, such as action, environment, state transition probability, and reward function, than supervised learning and unsupervised learning. As a result, when the complexity of a problem approaches that of the actual world, Reinforcement Learning may solve it more effectively [54]. Generally, there are two reinforcement learning algorithms: model-based and model-free. Model in this context refers to the environment's model. The primary distinction between the two algorithms is whether the agent knows the environment model. Model-based has the advantage of allowing the agent to pre-plan the action path based on the features of the known environment. However, it is challenging to get the desired outcome because of the discrepancy between the learned model and the actual world [55].

Consequently, Model-Free is frequently simpler to set up and modify. Value-based, policy-based, and Actor-criticism are the three types of model-free algorithms. Policy-based algorithms model and learn the policy directly, whereas value-based algorithms learn the value function or the action-value function to acquire policy [56]. The benefits of the other two approaches are combined in the Actor-Critic algorithms.

While the critic produces the value of the action, the actor chooses the course of action based on policy. Consequently, the value function and policy impact on one another, accelerating the convergence process. One traditional value-based reinforcement learning algorithm is Q-learning [57]. After learning the Q-values of state-action pairings, the agent chooses the action with the highest Q value. The Q-value, which is the expected reward received by acting a($a \in A$) under state s($s \in S$) at some time, is expressed as $Q^*(s, a) = max_\pi E[r_t + \gamma r_{t+1} + \gamma r_{t+2}{}^2 1 + ... | s_t = s, a_t = a, \pi]$ [43]. Q-learning optimizes the policy by updating the complete Q-table in each iteration, using the Q-table to hold the Q-values of all state-action pairs. The formula

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (23)$$

The current state, s, the action was taken at s, s', the state that follows action a, and a', the next possible action at the state s', are all represented in Equation (23). The parameters indicate the learning rate and discount factor α and γ, respectively. The reward results from selecting an action and is denoted by r. Q-learning updates the current Q-value using the maximum Q-value of the subsequent stage. Here, the goal Q-value is denoted by $r + \gamma$ max$_{a'}$ $Q(s', a')$, while the estimated Q-value is denoted by $Q(s, a)$ [58]. It goes without saying that when the state and action spaces are too big, the Q-table will grow limitless and require more storage space. A promising approach, DQN (Deep Q-Network), which combines the Q learning algorithm and the deep neural network, addresses the issue.

4.1. DEEP Q-NETWORK ALGORITHM

One significant development in Deep Reinforcement Learning was Google DeepMind Technologies' 2013 proposal of DQN. Figure 3 depicts the DQN structure. DQN has two main advantages over classical Q learning. First, it changes the Q-table updating process into a function-fitting problem, which fits a function rather than a Q-table to produce Q values. In DQN, a deep neural network predicts Q values. Two neural networks predominate.

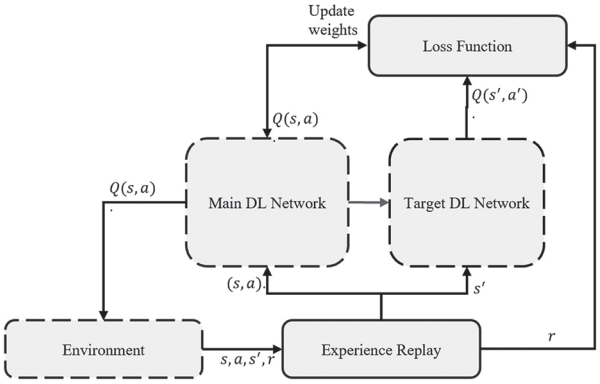


Fig. 3. The DQN structure

One is the main network, which modifies the parameters for every iteration, and the other is the target network, whose parameters are largely fixed [59]. At the same intervals, the target network replicates the parameters from the primary network. As a result, backpropagation only actually trains the primary network. Second, each step of the agent is stored in a unique structure called experience replay, which is denoted by (s, a, r, s'). During each network training cycle, a batch of experiences will be randomly selected from the experience replay for learning. Q-learning can be learned from past and present experiences because it is an off-policy algorithm [60].

Therefore, adding prior experience at random during the learning process will increase the neural network's efficiency and break the correlation between training samples. The following loss function is used for DQN updates at iteration i.

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')}[(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta i)^2] \quad (24)$$

Where the goal Q-value for iteration i is [($r + \gamma$ ax$_{a'}$)$Q(s', a'; \theta_i^-$). Until the agent learns to select the best course of action for every state, the neural network is

trained, and its parameters are updated by minimizing the value of the loss function in (24), which is the difference between the goal and estimated Q-values [61]. In the following subsection, the specifics of our suggested JCOTM algorithm will be displayed.

4.2. JCOTM ALGORITHM

The optimization issue JCOTM is formulated as a DRL process in this subsection. In this case, the agent is a central management system, which interacts with the surroundings and makes choices. As a result, the agent will broadcast the computation offloading decisions to every UE after gathering status data from servers and automobiles [62]. We must define the three essential components of DQN—the State, Action, and Reward functions—in our algorithm to use it to solve the problem we have been given. Action is the potential behavior of each step, whereas the state is used to represent the environment model. The reward produced by each action, which may be good or negative, is determined using the reward function.

- **State:** $S_n(t)$ represents the condition of the user n at time slot t. The communication state is described by $\Gamma_n^m(t)$, $b_n^m(t)$, while the user state is described by $l_n(t)$, $\alpha_n(t)$, and $\beta_n(t)$. The channel gain and the allotted communication bandwidth between the user n and RSU m at time slot t are denoted by $\Gamma_n^m(t)$ and $b_n^m(t)$, respectively. The traffic area where the user n is at a time slot t is shown by $l_n(t)$,. The remaining data amount is represented by $\alpha_n(t)$, while the necessary number of CPU cycles is represented by $\beta_n(t)$. Consequently, $s_n(t)$ can be written like this:

$$s_n(t) = \{\Gamma_n^1(t), \dots, \Gamma_n^{M+1}(t), \mathrm{b}_n^1(t), \dots, \mathrm{b}_n^{M+1}(t), l_n(t), \alpha_n(t), \beta_n(t)\} \quad (25)$$

- **Action:** Vector $a_n(t) \in R^{M+1}$ indicates whether task n is offloaded to a server m, which is the binary offloading decision. The environment changes from its present state to the next state when the agent selects one action for each time slot t. $a_n(t)$ is defined as follows

$$a_n(t) = \{\mathrm{x}_n^0(t), \mathrm{x}_n^1(t), \dots, \mathrm{x}_n^{M+1}(t)\} \quad (26)$$

- **Reward system:** To determine whether the chosen course of action is good, the environment provides the agent with an indicator value called reward. The optimization objective in this article is to reduce the system cost, which is composed of energy consumption and latency. System cost is hence the reward function.

$$r_n(t) = Cost(t) = \xi_t \sum_{n\in U}(T_{nm}^{comm}(t) + T_{nm}^{comp}(t)) + \xi_e \sum_{n\in U}(E_{nm}^{comm}(t) + E_{nm}^{comp}(t)) \quad (27)$$

Fig. 4 depicts the architecture of the JCOTM algorithm, which is based on deep reinforcement learning. With the same structure, we employ k-deep neural networks (DNN) to forecast binary offloading choices. The action is the neural network's output, while the present state of the environment is its input [63]. We add a decoding layer after the output layer to translate the decimal values into binary. The binary action vector's dimension in our suggested offloading paradigm is $N(M+2)$. We compute the system offloading cost, which is the reward function specified in the preceding material, for each of the output k binary offloading actions. The experience replay unit is initially empty, and a k DNNs start with random parameter values θ_0^k.

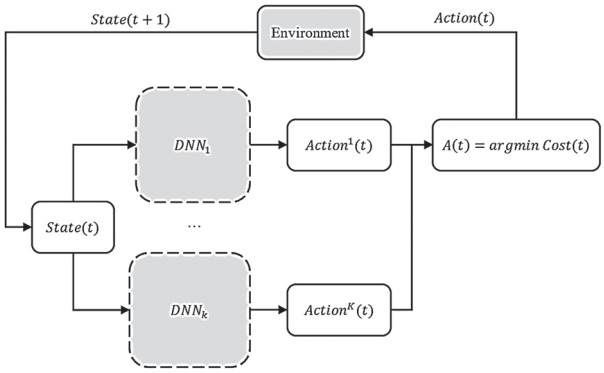


Fig. 4. The architecture of the proposed JCOTM algorithm

The agent chooses the best f-loading action to minimize the reward value in each iteration. The algorithm regularly updates the network parameters and randomly selects a batch of samples from the experience replay unit for training. Gradient descent is used to adjust the parameters to minimize the cross-entropy loss because we switch the DNN's output from predicting the Q-value to action [64].

$$L(\theta_0^k) = -\left[A_t \log\left(f_{\theta_0^k}(S_t)\right) + (1 - A_t) \log\left(1 - f_{\theta_0^k}(S_t)\right)\right] \quad (28)$$

Algorithm 1 displays the JCOTM algorithm's pseudo code.

Algorithm 1. The JCOTM Algorithm is based on DRL.

1: Input: status of the environment $State(t)$
2: Output: decision for offloading $Action(t)$
3: Initialization:
4: initialize environment state $State(t)$
5: The offloading procedure begins by using an identically structured k DNNs.
6: initialize experience replay.
7: for $t = 0,1,...,T$: do
8: Input the current environment state S_t.
9: Get the outputs of each DNN.
10: Apply decoding techniques to the output values to obtain A_t^i.
11: The offloading decision A_t is selected through $arg\ min\ R_t$ where by $R_t = arg\ min_{i=1,...,k}\ Q(S_t, A_t^i)$.

12: After execution of $Action(t)$ environment progresses to its new status S_{t+1}.

13: The experience reply receives a tuple A_t, R_t, S_t, S_{t+1}.

14: The parameters within DNNs get updated through data from randomly chosen training batches.

15: end for

5. ANALYSIS OF SIMULATION

To assess the effectiveness of our suggested JCOTM method, we create various simulation tests in this part. TensorFlow and Python 3.7 serve as the foundation for the simulation environment. First, by modifying the model's important parameters, we confirm that the JCOTM algorithm is convergent. Next, we assess the development of the deep reinforcement learning-based system offloading technique by comparing the average system offloading cost of JCOTM with other task offloading policies.

To construct a resource-constrained user-aware MEC network, we set the number of users $N = 15$ and the number of RSUs $M = 4$. Each $P = 4$ traffic region that makes up the route has a single RSU with a coverage diameter of $R = 1$ km. Each UE, edge server, and cloud server has CPU frequencies of 0.6×10^9, 1×10^{10}, and 1×10^{12}, respectively [65]. The Gaussian white noise power σ^2 is -88dB, and the overall bandwidth B is 10 MHz. The data size of job n αn is assumed to be between 10M and 30M, and $\rho = 960$ Cycles/Byte is the number of CPU cycles needed for one byte [66]. Table 2 is a list of some important parameters. We will then do our simulation exercises and examine the outcomes.

Table 2. Simulation parameters

Parameter	Value
f_0	0.6 GHz
f_m	10 GHz
f_{M+1}	1 THz
α_n	[10, 30] Mb
σ^2	-88 dB
ρ	960 Cycles/Byte
B	10 MHz
R	1 km
P	4
M	4
N	15

5.1. JCOTM CONVERGENCE

JCOTM convergence has been measured by the reward ratio, by dividing the cost of the optimal offloading policy by enumerating the cost of the policy created by JCOTM, as the assessment indicator to confirm the convergence of JCOTM [67]. Consequently, the algorithm performs better the closer the reward ratio is to 1. It is defined as follows:

$$Reward\ Ratio = \frac{C_{policy}}{C_{JCOTM}} \tag{29}$$

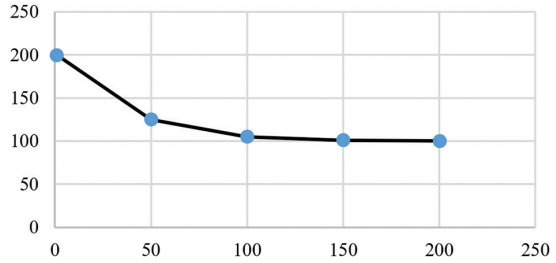


Fig. 5. Convergence of JCOTM Offloading Cost Over Iterations

Fig. 4 shows the cost of JCOTM keeps falling and eventually stabilizes as the number of simulations rises, in an environment with 20 users. As time goes on, JCOTM gets closer to the best policy than before.

5.2. PERFORMANCE OF DIFFERENT OFFLOADING POLICIES

This paragraph evaluates different offloading computing methods within the context. Our suggested method, JCOTM, joins the following different offloading rules, which form the basis of this analysis.

1. UE performs all its tasks individually without server transfers when performing local computing. The system cost results from the weighted sum of energy expended by devices, together with local computational delays.
2. Using edge servers as processing centers is known as edge computing, where all operations are transferred instead of running on the local devices [68]. The system cost includes computational delay and transmission delay, together with UE energy consumption that happens when data needs to be transferred. The concept of Edge computing in this application means all workloads are sent to execute on a single MEC server.
3. Cloud Computing works just like conventional operator cloud services, where all functions get processed on cloud-based servers. The distance between users and the cloud server results in higher transmission delays alongside increased energy consumption.
4. The random computing policy makes offloading decisions by selecting from available options randomly. A single operation can receive processing either within the local network or an edge server, or through the cloud infrastructure.
5. The VAMECN compute offloading problem receives dynamic non-cooperative game model analysis through DGTA, which leads to the determination of Nash Equilibrium solutions [69]. Each user receives a chance to select their optimal offloading strategy

per DGTA algorithm iteration since this method relies on game theory. Figure 6 shows the system offloading expenses of the six different policies while the user count varies. It is evident that when the number of users for all policies increases, the system cost progressively increases as well.

Policies for offloading and the inferior offloading performance are attained by DGTA. Additionally, random computing outperforms edge and local computing, but cloud computing outperforms random computing. Furthermore, the local computing strategy's offloading cost is higher than edge computing's when there are fewer than sixteen users, while the opposite is true when there are more than sixteen. The rationale is that if several workloads are offloaded to the same MEC server [70], there will be less computing power available for each user, which will raise the cost of computation.

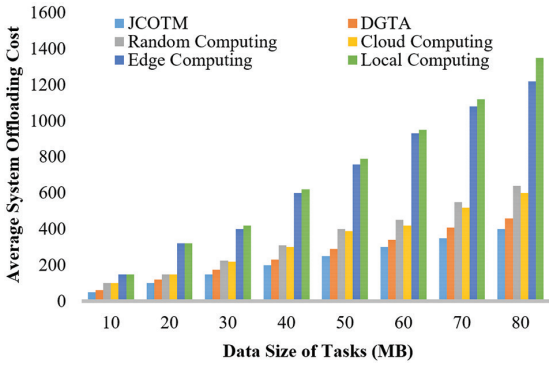


Fig. 6. The average system offloading cost is compared to varying user counts

The average system offloading cost under various job data sizes is compared in Fig. 7. Here, we used Fig. 7 to independently determine the offloading cost. Average system offloading costs for varying user counts are compared. The 10MB to 80MB data size range. The average cost of computing offloading progressively rises as task data sizes increase. JCOTM outperforms the other offloading policies since it optimizes the allocation of system resources [71], whereas other policies either do not accomplish the best allocation of system resources or only use a specific type of computing resources. Local computing has the highest offloading cost, followed by edge computing.

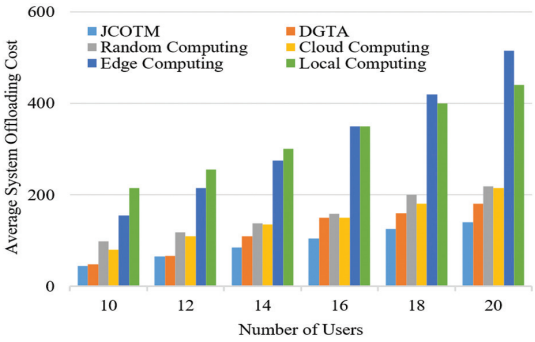


Fig. 7. Average system offloading cost comparison for activities with varying data volumes

On the other hand, cloud computing, random computing, and DGTA have reduced average system offloading costs. The effect of varying numbers of MEC servers on the average system offloading cost is seen in Fig. 8.

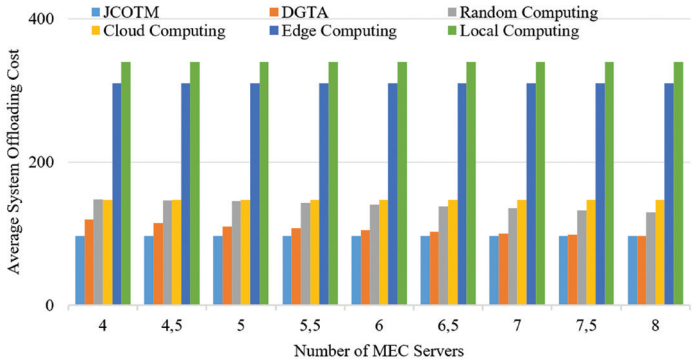


Fig. 8. Comparison of the average cost of system offloading for varying MEC server counts

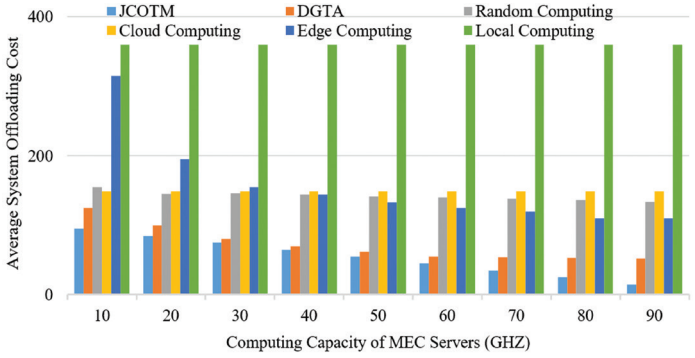


Fig. 9. The average system offloading cost is compared to MEC servers with varying computational capacities

Naturally, the curves are straight lines parallel to the x-axis because local and cloud computing are unaffected. Using Figure 7, as illustrated in Figure 8. Comparison of average system offloading costs for tasks with varying task data sizes [72]. The edge computing offloading cost curve resembles a horizontal straight line as the number of MEC servers increases.

Since all jobs are offloaded to a single MEC server for computation under the edge computing policy, increasing the number of MEC servers has a minimal effect on offloading costs, making this easy to explain [73]. The curves drop as the number of MEC servers grows since additional MEC servers can minimize mode computing delay for random computing and DGTA rules. The chart shows that the average JCOTM system offloading cost is nearly unaffected by the quantity of MEC servers.

One argument is that the cost curve does not exhibit a noticeable downward trend because the resource-constrained environment we have simulated can only satisfy the computational needs of every user.

The average system offloading cost for MEC servers with varying computing capacities is compared in Figure 9. Likewise, the computing power of MEC servers has little bearing on cloud or local computing. The chart indicates that the lower the offloading cost, the greater the computational capability of MEC servers. Additionally, when MEC servers' processing power increases, the offloading cost's rate of decline progressively slows down.

Compared to the other policies, JCOTM has a lower average system offloading cost. Additionally, edge computing outperforms cloud computing in terms of offloading costs when MEC server processing power reaches above 30GHz, and when it reaches over 40GHz, edge computing. We infer that the average system offloading cost is mostly determined by the computational capacity of MEC servers.

6. CONCLUSION

This paper addresses the joint multi-user computation offloading and task migration optimization problem under user-aware Multi-access Edge Computing networks. It considers several factors, including the distribution of system computing resources, communication bandwidth, and concurrent multiple computation tasks. It then suggests a deep reinforcement learning-based JCOTM algorithm to reduce system latency and energy consumption. To increase communication rate and quality and decrease communication latency, we completely consider the Non-Orthogonal Multiple Access technology in the upcoming 5G network during the problem modeling process.

The algorithm abstracts the offloading policy and system resources into the binary action vector and environment state, respectively. Additionally, a deep neural network is used to forecast offloading choices. Until the best offloading choice is found, the agent uses several iterative training courses to perceive the condition of the environment. We create simulation experiments to assess the algorithm's performance and convergence. The simulation findings demonstrate that JCOTM outperforms other offloading strategies under various experiment situations and converges with varying algorithm parameter values. As a result, the technique we suggested can successfully lower the VAMECN system's overall delay and energy usage.

7. REFERENCES

[1] S. Alamuri et al. "Transition From 5G to 6G Communication Technologies: Workforce Evolution and Skill Development Needs", 5G/6G Advancements in Communication Technologies for Agile Management, IGI Global Scientific Publishing, 2025, pp. 117-142.

[2] S. R. Alkaabi, M. A. Gregory, S. Li, "Multi-access edge computing handover strategies, management, and challenges: a review", IEEE Access, Vol. 12, 2024, pp. 4660-4673.

[3] A. Elnaim et al. "Energy Consumption for Cognitive Radio Network Enabled Multi-Access Edge Computing", Proceedings of the 3rd International Conference on Emerging Smart Technologies and Applications, Taiz, Yemen, 10-11 October 2023, pp. 1-5.

[4] P. Wei, K. Guo, J. Wang, W. Feng, S. Jin, "Reinforcement learning-empowered mobile edge computing for 6G edge intelligence", IEEE Access, Vol. 10, 2022, pp. 65156-65192.

[5] M. M. Saeed et al. "Anomaly detection in 6G networks using machine learning methods", Electronics, Vol. 12, No. 15, 2023, p. 3300.

[6] M. Khani, M. M. Sadr, S. Jamali, "Deep reinforcement learning-based resource allocation in multi-access edge computing", Concurrency and Computation: Practice and Experience, Vol. 36, No. 15, 2024, p. e7995.

[7] M. M. Saeed, E. S. Ali, R.A. Saeed, "Data-driven techniques and security issues in wireless networks", Data-Driven Intelligence in Wireless Networks, CRC Press, 2023, pp. 107-154.

[8] M. I. Khattak, H. Yuan, A. Khan, A. Ahmad, I. Ullah, M. Ahmed, "Evolving Multi-Access Edge Comput-

ing (MEC) for Diverse Ubiquitous Resources Utilization: A Survey", Telecommunication Systems, Vol. 88, No. 2, 2025, pp. 1-41.

[9] M. M. Saeed et al. "Multi-Access Edge Computing Using Intelligent Mobile User Resource Allocation In 6G", Proceedings of the IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering, Tripoli, Libya, 19-21 May 2024.

[10] J. C. Cepeda-Pacheco, M. C. Domingo, "Reinforcement Learning and Multi-Access Edge Computing for 6G-Based Underwater Wireless Networks", IEEE Access, Vol. 13, 2025, pp. 60627-60642.

[11] R. Dulot, L. Mendiboure, Y. Pousset, V. Deniau, F. Launay, "non-orthogonal multiple access for offloading in multi-access edge computing: A survey", IEEE Access, Vol. 11, 2023, pp. 118983-119016.

[12] R. O. Ogundokun et al. "Non-Orthogonal Multiple Access Enabled Mobile Edge Computing in 6G Communications: A Systematic Literature Review", Sustainability, Vol. 15, No. 9, 2023, p. 7315.

[13] S. Jahandar et al. "Handover Decision with Multi-Access Edge Computing in 6G Networks: A Survey", Results in Engineering, 2025, p. 103934.

[14] T. K. Rodrigues, J. Liu, N. Kato, "Offloading decision for mobile multi-access edge computing in a multi-tiered 6G network", IEEE Transactions on Emerging Topics in Computing, Vol. 10, No. 3, 2021, pp. 1414-1427.

[15] H. Hui, Q. Ye, Y. Zhou, "6G-empowered offloading for realtime applications in multi-access edge computing", IEEE Transactions on Network Science and Engineering, Vol. 10, No. 3, 2022, pp. 1311-1325.

[16] L. Zhao et al. "Open-source multi-access edge computing for 6G: Opportunities and challenges", IEEE Access, Vol. 9, 2021, pp. 158426-158439.

[17] M. Hassan, K. Hamid, R. A. Saeed, H. Alhumyani, A. Alenizi, "Reconfigurable Intelligent Surfaces in 6G mMIMO NOMA Networks: A Comprehensive Analysis", International Journal of Electrical and Computer Engineering Systems, Vol. 16, No. 2, 2025, pp. 87-97.

[18] I. Begić, A. S. Kurdija, Ž. Ilić, "A Framework for 5G Network Slicing Optimization using 2-Edge-Connected Subgraphs for Path Protection", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 8, 2024, pp. 675-685.

[19] W. Daniar, K. Asmoro, S. Y. Shin, "Joint optimization of phase shift and task offloading for RIS-assisted multi-access edge computing in beyond 6G communication", ICT Express, Vol. 10, No. 3, 2024, pp. 620-625.

[20] X. Chen et al. "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning", IEEE Internet of Things Journal, Vol. 6, No. 3, 2018, pp. 4005-4018.

[21] J. Zhang, W. Xia, F. Yan, L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing", IEEE Access, Vol. 6, 2018, pp. 19324-19337.

[22] H. Sun, J. Wang, D. Yong, M. Qin, N. Zhang, "Deep reinforcement learning-based computation offloading for mobile edge computing in 6G", IEEE Transactions on Consumer Electronics, Vol. 70, No. 4, 2024, pp. 7482-7493.

[23] H. Huang, Q. Ye, Y. Zhou, "6G-empowered offloading for real-time applications in multi-access edge computing", IEEE Transactions on Network Science and Engineering, Vol. 10, No. 3, 2022, pp. 1311-1325.

[24] M.-H. Chen, B. Liang, M. Dong, "Joint offloading decision and resource allocation for multi-user multi-task mobile cloud", Proceedings of the IEEE International Conference on Communications, Kuala Lumpur, Malaysia, 22-27 May 2016, pp. 1-6.

[25] X. Chen, L. Jiao, W. Li, X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing", IEEE/ACM Transactions on Networking, Vol. 24, No. 5, 2015, pp. 2795-2808.

[26] C. You, K. Huang, H. Chae, B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading", IEEE Transactions on Wireless Communications, Vol. 16, No. 3, 2016, pp. 1397-1411.

[27] J. Ren, G. Yu, Y. Cai, Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading", IEEE Transactions on Wireless Communications, Vol. 17, No. 8, 2018, pp. 5506-5519.

[28] Y. Dai, D. Xu, S. Maharjan, Y. Zhang, "Joint computation offloading and user association in multi-task mobile edge computing", IEEE Transactions on Vehicular Technology, Vol. 67, No. 12, 2018, pp. 12313-12325.

[29] L. Huang, X. Feng, L. Zhang, L. Qian, Y. Wu, "Multi-server multi-user multi-task computation offloading for mobile edge computing networks", Sensors, Vol. 19, No. 6, 2019, p. 1446.

[30] A. R. Askhedkar, B. Chaudhari, R. A. Saeed, H. Alhumyani, A. Alenizi, "Performance of TVWS-based LoRa Transmissions using Multi-Armed Bandit", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 9, 2024, pp. 759-769.

[31] D. Wang, H. Qin, B. Song, X. Du, M. Guizani, "Resource allocation in information-centric wireless networking with D2D-enabled MEC: A deep reinforcement learning approach", IEEE Access, Vol. 7, 2019, pp. 114935-114944.

[32] A. A. Elnaim et al. "Energy Consumption for Cognitive Radio Network Enabled Multi-Access Edge Computing", Proceedings of the 3rd International Conference on Emerging Smart Technologies and Applications, Taiz, Yemen, 10-11 October 2023, pp. 1-5.

[33] Y. Liu, H. Yu, S. Xie, Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in 6G user edge computing and networks", IEEE Transactions on Vehicular Technology, Vol. 68, No. 11, 2019, pp. 11158-11168.

[34] J. Chen et al. "Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks", IEEE Transactions on Communications, Vol. 71, No. 1, 2022, pp. 296-309.

[35] W. Zhang, Z. Zheng, "Task migration for mobile edge computing using deep reinforcement learning", Future Generation Computer Systems, Vol. 96, 2019, pp. 111-118.

[36] B. Di, L. Song, Y. Li, G. Y. Li, "Non-orthogonal multiple access for high-reliable and low-latency V2X communications in 5G systems", IEEE journal on selected areas in communications, Vol. 35, No. 10, 2017, pp. 2383-2397.

[37] Y. Wu et al. "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation", IEEE Transactions on Vehicular Technology, Vol. 67, No. 12, 2018, pp. 12244-12258.

[38] Y. Saito et al. "Non-orthogonal Multiple Access (NOMA) for Cellular Future Radio Access", Proceedings of the IEEE 77th Vehicular Technology Conference, Dresden, Germany, 2-5 June 2013, pp. 1-5.

[39] Z. Ding, P. Fan, H. V. Poor, "Impact of user pairing on 5G nonorthogonal multiple-access downlink transmissions", IEEE Transactions on Vehicular Technology, Vol. 65, No. 8, 2015, pp. 6010-6023.

[40] G. Kivanc, H. Liu, "Computationally efficient bandwidth allocation and power control for OFDMA", IEEE transactions on wireless communications, Vol. 2, No. 6, 2003, pp. 1150-1158.

[41] M. Barakat et al. "Performance Evaluation of Multi-Access Edge Computing for Blended Learning Services", Proceedings of the 21st Learning and Technology Conference, Jeddah, Saudi Arabia, 15-16 January 2024, pp. 197-202.

[42] Z. Wu, D. Yan, "Deep reinforcement learning-based computation offloading for 5G 6G user-aware multi-access edge computing network", China Communications, Vol. 18, No. 11, 2021, pp. 26-41.

[43] V. Mnih et al. "Human-level control through deep reinforcement learning", Nature, Vol. 518, 2015, pp. 529-533.

[44] H. W. Kuhn, "Extensive games and the problem of information", Contributions to the Theory of Games, Vol. 2, No. 28, 1953, pp. 193-216.

[45] M. M. Saeed et al. "Task reverse offloading with deep reinforcement learning in multi-access edge computing", Proceedings of the 9th International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 15-16 August 2023, pp. 322-327.

[46] M. K. Hasan et al. "An improved binary spider wasp optimization algorithm for intrusion detection for industrial Internet of Things", IEEE Open Journal of the Communications Society, Vol. 6, 2024, pp. 2926-2944.

[47] Z. E. Ahmed et al. "TinyML network applications for smart cities", TinyML for Edge Intelligence in IoT and LPWAN Networks, Elsevier, 2024, pp. 423-451.

[48] M. M. Saeed, R. A. Saeed, Z. E. Ahmed, "TinyML for 5G networks", TinyML for Edge Intelligence in IoT and LPWAN Networks, Elsevier, 2024, pp. 167-229.

[49] S. Khan et al. "Optimizing deep neural network architectures for renewable energy forecasting", Discover Sustainability, Vol. 5, No. 1, 2024, p. 394.

[50] Z. Chen, F. Wang, X. Zhang, "Joint Optimization for Cooperative Service-Caching, Computation-Offloading, and Resource-Allocations Over EH/MEC 6G Ultra-Dense Mobile Networks", IEEE Transactions on Wireless Communications, 2025. (in press)

[51] Z. Hu et al. "DRL-Based Trajectory Optimization and Task Offloading in Hierarchical Aerial MEC", in IEEE Internet of Things Journal, Vol. 12, No. 3, 2025, pp. 3410-3423.

[52] S. Zhang et al. "Stackelberg Game-Based Multi-Agent Algorithm for Resource Allocation and Task Offloading in MEC-Enabled C-ITS", IEEE Transactions on Intelligent Transportation Systems, 2025. (in press)

[53] J. Carlos, M. C. Domingo, "Reinforcement Learning and Multi-Access Edge Computing for 6G-Based Underwater Wireless Networks", IEEE Access, Vol. 13, 2025, pp. 60627-60642.

[54] M. Hevesli, A. M. Seid, A. Erbad, M. Abdallah "Multi-Agent DRL for Queue-Aware Task Offloading in Hierarchical MEC-Enabled Air-Ground Networks", IEEE Transactions on Cognitive Communications and Networking, 2025. (in press)

[55] J. Bi et al. "Energy-Minimized Partial Computation Offloading in Satellite–Terrestrial Edge Computing Networks", IEEE Internet of Things Journal, Vol. 12, No. 5, 2025, pp. 5931-5944.

[56] M. Ahmed et al. "Advancements in RIS-Assisted UAV for Empowering Multiaccess Edge Computing: A Survey", IEEE Internet of Things Journal, Vol. 12, No. 6, 2025, pp. 6325-6346.

[57] D. M. Rani, Supreethi K P, B. B. Jayasingh, "Deep Reinforcement Learning for Dynamic Task Scheduling in Edge-Cloud Environments", International Journal of Electrical and Computer Engineering Systems, Vol. 15, No. 10, 2024, pp. 837-850.

[58] J. Pacheco, "Contribution to the enhancement of IoT-based application development and optimization of underwater communications, by artificial intelligence, edge computing, and 5G networks and beyond, in smart cities/seas", Department of Network Engineering, Polytechnic University of Catalonia, Barcelona, Spain, 2024, PhD Thesis.

[59] Z. Wang et al. "AUV-assisted node repair for IoUT relying on multiagent reinforcement learning", IEEE Internet Things Journal, Vol. 11, No. 3, 2024, pp. 4139-4151.

[60] J. Cao et al. "Multi-agent reinforcement learning charging scheme for underwater rechargeable sensor networks", IEEE Communication Letters, Vol. 28, No. 3, 2024, pp. 508-512.

[61] Z. Zhao et al. "A transmission-reliable topology control framework based on deep reinforcement learning for UWSNs", IEEE Internet Things Journal, Vol. 10, No. 15, 2023, pp. 13317-13332.

[62] Z. Zhang et al. "Environment- and energy-aware AUV-assisted data collection for the Internet of Underwater Things", IEEE Internet Things J., Vol. 11, No. 15, 2024, pp. 26406-26418.

[63] X. Hou et al. "Environment-aware AUV trajectory design and resource management for multi-tier underwater computing", IEEE Journal on Selected Areas in Communications, Vol. 41, No. 2, 2023, pp. 474-490.

[64] K. G. Omeke, M. Mollel, S. T. Shah, L. Zhang, Q. H. Abbasi, M. A. Imran, "Toward a sustainable Internet of Underwater Things based on AUVs, SWIPT, and reinforcement learning", IEEE Internet Things Journal, Vol. 11, No. 5, 2024, pp. 7640-7651.

[65] P. Q. Truong et al. "Computation Offloading and Resource Allocation Optimization for Mobile Edge Computing-Aided UAV-RIS Communications", IEEE Access, Vol. 12, 2024, pp. 107971-107983.

[66] Y. Sadovaya et al. "Enhancing Service Continuity in Non-Terrestrial Networks via Multi-Connectivity Offloading", IEEE Communications Letters, Vol. 28, No. 10, 2024, pp. 2333-2337.

[67] K. Ali, "Multiradio Parallel Offloading in Multi-access Edge Computing: Optimizing Load Shares, Scheduling, and Capacity", IEEE Internet of Things Journal, Vol. 11, No. 3, 2024, pp. 4047-4062.

[68] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, "6G Wireless Networks: Vision, Requirements, Architec-

ture, and Key Technologies", IEEE Vehicular Technology Magazine, Vol. 14, No. 3, 2019, pp. 28-41.

[69] Z. Wang et al. "AUV-Assisted Node Repair for IoUT Relying on Multiagent Reinforcement Learning", IEEE Internet of Things Journal, Vol. 11, No. 3, 2024, pp. 4139-4151.

[70] J. Cao et al. "Multi-Agent Reinforcement Learning Charging Scheme for Underwater Rechargeable Sensor Networks", IEEE Communications Letters, Vol. 28, No. 3, 2024, pp. 508-512.

[71] P. G. Satheesh, T. Sasikala, "FEDRESOURCE: Federated Learning Based Resource Allocation in Modern Wireless Networks", International Journal of Electrical and Computer Engineering Systems, Vol. 14, No. 9, 2023, pp. 1023-1030.

[72] Z. Zhang et al. "Environment- and Energy-Aware AUV-Assisted Data Collection for the Internet of Underwater Things", IEEE Internet of Things Journal, Vol. 11, No. 15, 2024, pp. 26406-26418.

[73] X. Hou et al. "Environment-Aware AUV Trajectory Design and Resource Management for Multi-Tier Underwater Computing", IEEE Journal on Selected Areas in Communications, Vol. 41, No. 2, 2023, pp. 474-490.

Comprehensive Classification and Analysis of Malware Samples Using Feature Selection and Bayesian Optimized Logistic Regression for Cybersecurity Applications

Original Scientific Paper

Manisankar Sannigrahi
Vellore Institute of Technology,
School of Computer Science Engineering and Information Systems
Vellore, India
manisankar.sannigrahi2020@vitstudent.ac.in

R Thandeeswaran*
Vellore Institute of Technology,
School of Computer Science Engineering and Information Systems
Vellore, India
rthandeeswaran@vit.ac.in

*Corresponding author

Abstract *– Cyberattacks are serious threats not only to individuals but also to corporations due to their rising frequency and financial impact. Malware is the main tool of cybercriminals, and is always changing, making its detection and mitigation more complicated. To counter these threats, this work proposes a Logistic Regression approach that is based on Bayesian Optimization. By leveraging advanced techniques like a hybrid feature selection model, the study enhances malware detection and classification accuracy and efficiency. Bayesian Optimization fine-tunes the logistic regression model's hyperparameters, improving performance in identifying malware. The integration of a hybrid feature selection algorithm reduces dataset dimensionality, focusing on relevant features for more accurate classification and efficient resource use, which is suitable for real-time applications. The experimental results show amazing accuracy rates of 99.94% for the Ransomware Dataset and 99.98% on the CIC-Obfuscated Malware dataset. This proposed model performs better than the conventional detection techniques. With its flexible feature selection and optimization techniques, it can keep pace with the dynamic landscape of cyber threats. It, therefore, produces a robust and scalable answer to the current cybersecurity issues.*

Keywords*: Malware, Ransomware, Machine Learning, Feature Selection, Bayesian Optimization, Classification*

Received: February 14, 2025; Received in revised form: May 15, 2025; Accepted: May 15, 2025

1. INTRODUCTION

Malware today is a major threat to individuals, businesses, and governments. It refers to the viruses, worms, or other harmful programs that cause damage or exploit systems. Some of the outcomes can be very severe and may range from financial loss, data breach, identity theft, to national security threats. Another reason why the malware threat is on the rise is the sophistication of cyber attackers. Hackers develop new methods for avoiding detection and increasing the efficiency of their malware. Advanced Persistent Threats are dangerous to the critical infrastructure, health care, and education sectors. Malware requires a multi-pronged approach to become a threat. Organizations should be investing in holistic security approaches that include updating their software regularly, installing robust firewalls, intrusion detection systems, and training their employees. Advanced threat intelligence, along with machine learning, can enable better malware detection and mitigation. Governments and international organizations have a crucial role in formulating cybersecurity regulations and promoting international cooperation. Public-private partnerships can support the sharing of threat intelligence, thereby strengthening collective defenses against malware. Proactive and

collaborative cybersecurity measures are essential to reduce risks and safeguard digital infrastructure.

Ransomware is the most widespread malware that leads to significant economic and personal losses by affecting a wide range of files of numerous organizations, personal users, and medical services. It is malware that is programmed to prevent users from gaining access to their data from the devices [1]. Ransomware looks like a normal file that infects the system from vectors like botnets, macros, and email. It remains silent inside a computer and only makes itself aware to the user after completing the encryption process. According to many ventures of cybersecurity in 2019, the total sum of money paid by the victim is 11.5 billion. Every new victim has fallen to ransomware every fourteen and eleven seconds in the years 2019 and 2021 [2]. The world is highly connected through the internet, which helps to disseminate ransomware in several protocols of communication. Ransomware has enabled attackers to launch many campaigns like Ransomware as a Service, botnets for hire, etc., to earn money by carrying out illegal activities. Ransomware has become an intrinsic part of any cyber-attack by which hackers can earn large sums of money by carrying out criminal activity [3]. The victim cannot physically remove the hard disk to any other unaffected system to access the files. The attacker asks for a payment voucher as a ransom to give the access back to the victim. A few examples of locker ransomware are CTB-locker, and Winlocker [4]. Whereas the files of the victim's system are encrypted by Crypto Ransomware, making those files inaccessible unless decrypted. Removing the hard disk or trying to remove ransomware is not going to solve anything until the victim gets the decryption key. The ransom is mainly asked in Bitcoin [5], which is widely used due to anonymity, as the attacker's identity is hard to trace. Paying a ransom never guarantees that a decryption key will be given to the victim to recover data. Many methods used to detect ransomware have low detection rates. These methods also flag benign samples as malignant and thus fail to detect malicious samples that have high false positive and negative rates. Current techniques require gathering a large amount of data by monitoring the system. The disadvantage of these techniques is that they consume a significant amount of system resources [6].

This study advances malware detection through the utilization of a hybrid machine learning model based on feature selection:

- The aim here is to enhance the effectiveness and accuracy of malware detection and classification by using features such as feature selection and hybrid models. Ultimately, the hybrid machine learning method aims to enhance the capabilities of intrusion detection systems by enhancing malware categorization accuracy.
- With a number of methods including Support Vector Machine and Naïve Bayes, malware can be investigated in comprehensive manners to study the different malware categorization approaches. Therefore, the best effective method which is best in detecting precisely classifying malware cases would be found through this project.
- This paper adds to the development of more robust and efficient methods of countering cyber threats as a result of the fusion of hybrid feature selection with the assessment of multiple methods.

The paper's subsequent sections follow this structure: Section II explores the previous works in this field. Section III examines various machine learning techniques, highlighting their strengths and limitations. In Section IV, the datasets used in the study are introduced, including details about the Ransomware dataset and CIC-Obfuscated Malware datasets. Section V delves into data visualization and feature selection techniques to enhance dataset understanding. Section VI introduces the proposed algorithm aimed at improving ransomware detection interpretability. Section VII covers the experiments conducted with the Ransomware and CIC-Obfuscated Malware datasets, presenting the results comprehensively. Finally, Section VIII provides concluding remarks and suggests potential avenues for future research.

2. RELATED WORKS

This section is dedicated to the previous literature works on malware classification and analysis. Ganfure et al. authors state that [7] ransomware attacks represent a substantial risk to businesses, but current detection methods frequently prove inadequate. The RTrap framework introduces an innovative approach employing machine-learning-generated decoy files to swiftly identify and restrict ransomware. By strategically dispersing decoy files across directories, RTrap entices ransomware, while a lightweight observer monitors these files continuously. Once detected, an automated response is activated to promptly neutralize the threat. Empirical findings underscore RTrap's efficacy, as it successfully identifies ransomware with minimal data loss, underscoring its promise in effectively countering ransomware dangers. H Bakır & R Bakır, the authors state that [8] Android malware detection has received significant attention, yet feature extraction has been relatively overlooked in machine learning-based methods. Addressing this gap, the authors introduce DroidEncoder, an innovative autoencoder-based model for Android malware classification. Using three distinct autoencoder architectures, the authors extract features from a visualized dataset containing 3000 malicious and benign Android apps. Through experiments involving various machine learning algorithms, the authors approach demonstrates superior performance across multiple metrics, validated through cross-validation. S Gulmez et al. state that [9] the escalating threat of ransomware attacks necessitates advanced detection systems beyond traditional signature-based approaches. Existing

methods often rely on the machine or deep learning models to analyze dynamic features like API call sequences and DLLs. However, these methods may overlook crucial information or fail to capture the sequence relationship between features. Introducing XRan, a novel ransomware detection system, which leverages Explainable Artificial Intelligence (XAI) techniques to enhance interpretability. XRan utilizes Convolutional Neural Networks (CNNs) for detection and employs XAI models such as LIME and SHAP to provide transparent explanations. Experimental results show that XRan achieves a true positive rate of up to 99.4%, surpassing state-of-the-art methods. DW Fernando & N Komninos, the authors introduce [10] FeSAD, a framework designed to enable machine learning classifiers to effectively detect evolutionary ransomware. It comprises three layers - feature selection, drift calibration, and drift decision - ensuring reliable classification of concept drift samples. FeSAD demonstrates effectiveness in detecting drifting samples and extending the classifier's lifespan. S Sivakumar et al., the authors introduce ML-MD in this study [11], a machine learning-based strategy for categorizing malware using static methods. It employs principal component analysis (PCA) to extract dataset characteristics and introduces a Modified Particle Swarm Optimization (MPSO) algorithm for enhanced malware detection. Experimental results demonstrate the superior accuracy and detection rate of the ML-based MPSO technique compared to alternative approaches on benchmark datasets. SM Florence et al., the authors introduce [12] a machine learning classification model to combat the rising threat of crypto-ransomware. It focuses on specific network traffic features, particularly UDP and ICMP, and incorporates feature selection to improve efficiency without sacrificing accuracy. The experiment employs decision trees and random forest algorithms, combined with behavioral analysis and honeypot deployment, for effective ransomware family classification.

3. MACHINE LEARNING

It is a sub-discipline of computer science that focuses on using data and algorithms to simulate the way humans learn and incrementally improve their precision. These algorithms are used to process data, learn from it, and then make decisions, and predictions, identify patterns, and cluster based on the data collected. Machine learning can be broadly classified into three types: supervised, unsupervised, and semi-supervised learning [13]. In supervised learning, target labels and classes are known in advance, which guides the learning process. However, in unsupervised learning, the target class is completely unknown. Semi-supervised learning combines features of both supervised and unsupervised methods. The hybrid algorithm proposed in this study seeks to overcome the drawbacks of previous approaches [14]. Algorithms examined in this research are as follows, along with their advantages and disadvantages.

3.1. NAÏVE BAYES

This algorithm is a probabilistic classifier based on Bayes' Theorem [15], a statistical formula that explains the connection between conditional probabilities. Naive Bayes classification is very useful because it is fast and easy to use, especially with datasets that have many features. Bayes' Theorem calculates the likelihood of an outcome based on previous occurrences in similar circumstances. This algorithm can be explained as a probabilistic classifier which is obtained from the application of Bayes Theorem.

$$P(y|x_1, \dots\dots, x_n) = \frac{P(y)P(x_1,\dots\dots,x_n)}{P(x_1,\dots\dots,x_n)} \quad (1)$$

In equation (1), y= Class variable, & $X1\dots\dots X_n$= Dependent Vector of features.

Naïve Bayes classifiers are good in simplicity, efficiency, and scalability. They are easy to implement, so they are good for quick deployment and prototyping. In addition, they are computationally efficient, especially for large datasets with a lot of features, due to their simple probabilistic approach [16]. Additionally, they can manage big sets of data effectively. However, these classifiers have some drawbacks, like the assumption that features are independent, which isn't always true in real data and can lead to mistakes, especially with features that are closely related. They also struggle with how features are spread out, doing very badly in cases where feature connections are complicated or when the probability method used doesn't fit the data well [17].

3.2. SUPPORT VECTOR MACHINE (SVM)

SVM is an algorithm that operates by training on a particular dataset to make precise predictions and extrapolate insights to the rest of the data. It falls under the supervised learning category of machine learning and is commonly employed for tasks such as data analysis, pattern recognition, regression, and classification. The primary goal of SVM is to identify a hyperplane within an N-dimensional space that effectively separates data points into two distinct categories [18]. SVM linear kernel function is expressed as (x, x'), which has been used for analysis

SVM has some advantages, including excelling in high-dimensional spaces and being applicable in situations with a large number of features; they are quite versatile, applicable to many kinds of data, both numeric and categorical, and in various data distributions; it resists overfitting remarkably, especially in high dimensional space, due to their ability to maximize the margin of classes; in addition, SVM can handle non-linear data [19]. SVM has some limitations, it can be computationally expensive, particularly with large datasets or non-linear kernels, due to their computational complexity; sensitivity to parameter tuning is another concern, as SVMs require careful selection of hyperparameters like kernel type and regularization parameter [20], which can greatly in-

fluence performance; their lack of interpretability poses challenges, as the decision boundary produced by SVMs can be complex and difficult to interpret, hindering understanding of the underlying decision-making process.

Fig. 1 represents the margin of SVM, which is used for the classification of data points.

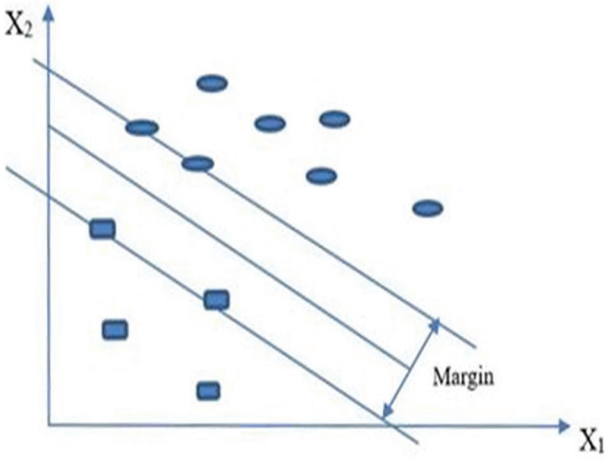


Fig. 1. Margin of SVM

3.3. RANDOM FOREST

It is a strong and adaptable tool in the field of machine learning, used for both regression and classification tasks across many applications. This algorithm creates a group of decision trees, called a "forest." Each tree in this forest is trained separately using a method called bagging. Bagging [21], in simple terms, means using the unique strengths of different models to make the group better overall. By combining the predictions of many trees, Random Forest can be more accurate and reliable than just one decision tree.

The formula of entropy is presented in equation (2).

$$Entropy = -\sum_{i=1}^{n} p_i \ log(p_i) \tag{2}$$

Information Gain = E (Parent) – E (Parent | Child), E= Entropy, p= probability.

For final evaluation, majority/hard voting method is used, the formula of this method is shown in equation (3).

$$\hat{y} = mode\ \{C_1(x),\ C_2(x), \ldots\ldots, C_m(x)\} \tag{3}$$

Where $\hat{y}$ = class label, C_m = set of classifiers, the class label of each classifier is predicted by majority voting.

Random Forest is strong in different dimensions and typically gives high accuracy across multiple datasets, thus avoiding the overfitting phenomenon and comprehending complex patterns of the data. The strength against overfitting comes from methodologies like bootstrap sampling and random selection of features [22], which causes heterogeneity among the trees and increases generalization. But it has some limitations. High computational complexity, especially with large datasets, causes longer training time and resource usage [23].

3.4. LOGISTIC REGRESSION

This method is used for binary classification, meaning it predicts the likelihood of a yes or no result based on one or more factors. It's widely used in areas like healthcare, finance, and marketing because it's straightforward to grasp. This method uses a special function to show the relationship between the outcome and the factors, and it limits the predictions to a range from 0 to 1, which represents probabilities [24].

Imagine a dataset with pairs (x, y). Here, x is a matrix with m rows and n columns, where each row represents a sample and each column is an attribute of that sample. The y part is a list with m items, each matching a label for the samples in x. Equation (4) defines the weight matrix, which is used for generating a random initialization.

$$a = w_0 + w_1 x_1 + w_1 x_2 + \ldots w_1 x_n \tag{4}$$

Then pass the output to the link function which is shown in equation (5)

$$\hat{y}_i = {}^{1}/_{(1 + e^{-a})} \tag{5}$$

Then the cost function is calculated by utilizing equation (6)

$$\text{cost(w)} = \left(-\frac{1}{\text{m}}\right) \sum_{\text{i=1}}^{\text{i=m}} \text{y}_\text{i} \log(\text{y}_\text{i}) + (1 - \text{y}_\text{i}) \log\,(1 - \hat{\text{y}}_\text{i}) \tag{6}$$

The updating of weights is done as per the derivative of the cost, the formulas are shown in the equation (7) and (8).

$$dw_j = \sum_{i=1}^{i=n} (\hat{y} - \hat{y}_i) x^i{}_j \tag{7}$$

$$w_i = w_j - (a\ dw_j) \tag{8}$$

Logistic regression helps us figure out the chance that a data point belongs to either class '0' or '1', using some values w and x. The key part is the exponential function inside the sigmoid function [25], which makes sure the probability is always positive. To keep the probability below one, we divide the top number by a bigger number. Equations (9) and (10) show us how to calculate these probabilities, which we then use to find the sigmoid function.

$$P = e^{w_0 + w_1 x_1 + w_2 x_2} \tag{9}$$

$$P = \frac{e^{w_0 + w_1 x_1 + w_2 x_2}}{e^{w_0 + w_1 x_1 + w_2 x_2 + 1}} \tag{10}$$

Equation (10) is divided by Equation (9) to obtain the numerator term, resulting in the sigmoid function. Equation (11) defines this sigmoid function.

$$\text{P} = \frac{1}{1 + \text{e}^{-(\text{w}_1 \text{x}_1 + \text{w}_2 \text{x}_2 + \cdots + \text{w}_\text{n} \text{x}_\text{n})}} \tag{11}$$

Logistic Regression is highly valued for its simplicity, thus being a first choice in rapid prototyping and result interpretation. Its coefficients are expressed as odds

ratios, and hence, provide direct information about the effects of the predictor variables on the outcomes, which enhances interpretability [26]. In addition, it is very robust to noise and remains stable in real scenarios, making it an excellent candidate for many applications. It does have some limitations, however.

It can only capture complex variable relationships, especially in scenarios with interactions or non-linear effects [27]. Overfitting is a concern, particularly with numerous predictor variables relative to observations.

Fig. 2 represents the curve of logistic regression.

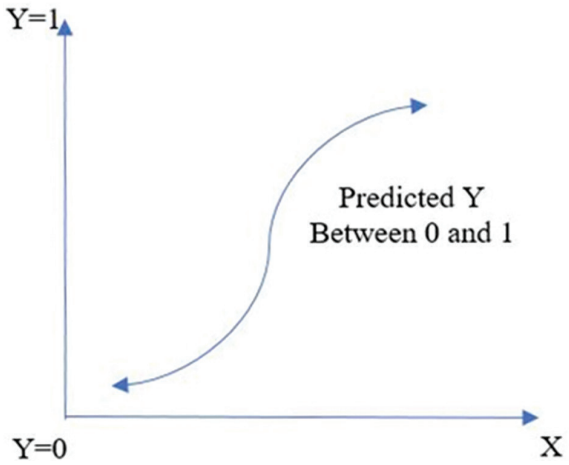


Fig. 2. Logistic Regression Curve

3.5. Logistic Regression

Bayesian optimization is an intelligent way of searching for the best parameters of complex problems without explicit formulas. It relies on a method based on probability, quickly searching through all options and finding the best for making a task better. It works very well if each option is to be tested at great cost or with significant time consumption. Unlike the grid and random search, Bayesian optimization learns the past tests to make this search faster. The main part of this method is that it starts with a probabilistic model, which creates a guess about the best settings, and then it keeps updating this guess through a process called the acquisition function [28] as it learns more. It is one of the most powerful ways to optimize functions [29], Equation (12) is used to determine the next sampling point.

$$X_t = \text{argmax}_x u(X|D_{1:t-1}) \quad (12)$$

Where, u = acquisition function, $D_{1:t-1}$= the total t samples.

There are mainly three types of acquisition functions: Upper Confidence Bound (UCB), Probability of Improvement (PI), and Expected Improvement (EI). The EI acts as a guiding metric during the optimization process, trying to balance exploration of new configurations with exploitation of the already identified good ones. It helps in an efficient search for optimal hyperparameters. Equation (13) defines the expected optimization process.

$$EI[x^*] = \int_{f[\dot{x}]}^{\infty}(f[x^*]f[\dot{x}])\text{Norm}_{f_{[x^*]}}[\mu[x^*]\sigma[x^*]df[x^*] \quad (13)$$

Where, $\mu[x^*]$ = mean value of data point x, $\sigma[x^*]$= variance value of data point x, β= controlling parameter of the degree of exploration, $f[x^*]$= normal distribution, $f[\dot{x}]$= current maxima.

Bayesian optimization does extremely well in optimizing functions, given its efficiency to strike a balance between exploration and exploitation, adaptability by dynamic updates of the probabilistic model, robustness with noisy data, and its ability to follow the pursuit of global optima. It converges to solutions quickly, adapts itself according to changes in the objective function landscape, does not have any problems in dealing with noisy objective functions, and can seek global optima via probabilistic predictions iteratively [30]. It has limitations regarding computational cost, sensitivity to initial conditions, surrogate model complexity, and suitability for smooth functions. It is computationally expensive, especially for large-scale tasks or complex models, thus limiting scalability. Sensitivity to initial conditions and surrogate model hyperparameters may impact its performance [31].

4. DATASET DESCRIPTION

The Ransomware dataset consists of 156 features with 1534 samples, among them 952 goodware and 582 ransomware samples of 11 different ransomware families [32]. The collected samples represent the most well-known variants of ransomware encountered recently. Each ransomware is clustered into a well-known ransomware family. Each ransomware sample was checked with VirusTotal results. Most of the ransomware samples belong to crypto-ransomware including Critroni, CryptoLocker, CryptoWall, etc. Fig. 3 encompasses the total count of instances from various ransomware families.

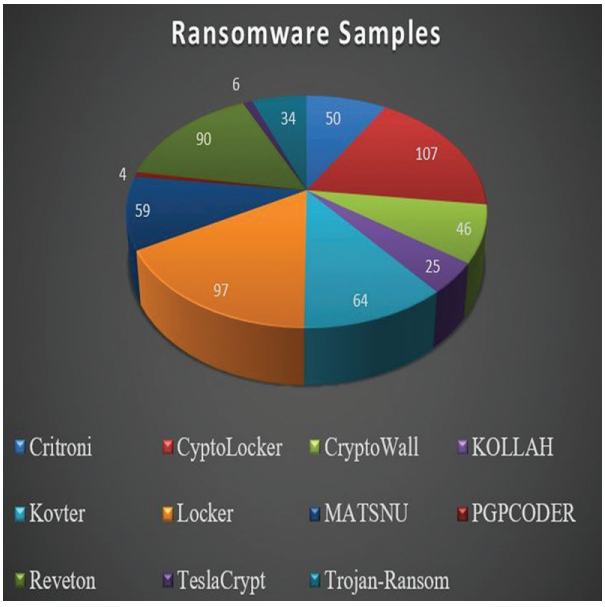


Fig. 3. Ransomware Family

CIC-Obfuscated malware dataset always focuses on representing scenarios of the real world as closely as possible by using malware that is predominant in the world. The dataset is made off of mainly three malware families Spyware, Trojan Horse, and Ransomware [33]. The dataset is being made from 50% benign and 50% malignant memory dumps. There are a total of 5832 samples with 57 features, where 2916 are malignant and 2916 are benign samples. The dataset is broken down in the Table 1.

Table 1. Description of CIC-Obfuscated Malware dataset

Malware Family	Malware Name	Count
Spyware	180Solutions	200
	Gator	200
	TIBS	141
	Coolwebsearch	200
	Transponder	241
Trojan Horse	Zeus	195
	Refroso	200
	Emotet	196
	Reconyc	157
	scar	200
Ransomware	Shade	220
	Ako	200
	Conti	200
	MAZE	195
	Pysa	171

5. FEATURE SELECTION

Feature selection is a method of aiding in the goal of creating a more accurate prediction model. This method helps in choosing features to provide better accuracy while requiring less amount of data. The main objective of feature selection is to provide cost-effective and faster predictors, improve prediction performance, and give a better comprehension of the fundamental process of generating data [34]. There are mainly three methods that are used in this paper.

5.1. VARIANCE THRESHOLD

The most simple baseline method of feature selection is the Variance Threshold. It removes the features whose threshold does not meet up and removes all features with zero variance by default. Equation (14) is utilized to calculate the variance.

$$Var[X] = p(1-p) \tag{14}$$

5.2. PEARSON CORRELATION COEFFICIENT

The measurement of the strength of the relationship between two variables and the association between them is defined as the Pearson correlation coefficient [35]. Pearson correlation is used to evaluate the linear dependency of the dataset, which is either positive or negative. The value it returns lies between -1 to 1. Equation (15) is the formula of Pearson correlation.

$$r = \frac{\sum(x_i - \hat{x})(x_i - \hat{x})}{\sqrt{\sum(x_i - \hat{x})^2 \sum(y_i - \hat{y})^2}} \tag{15}$$

Where, r= Pearson correlation coefficient, x= values in the x set, y= values in the y set, n= total number of values of samples Y.

6. PROPOSED ALGORITHM

Ransomware or malware families create major security risks to critical infrastructures. Malicious attacks cause catastrophic harm to web or mobile applications and data centers of various businesses and industries. Traditional methods are not adequate to handle sophisticated attacks [36]. In this paper, the proposed algorithm is based on Bayesian optimization and Logistic Regression algorithm. The best parameters for prediction are selected by the Bayesian optimization technique. The classification is done by optimized logistic regression. Bayesian optimization improves the performance of Logistic Regression in hybrid models by effectively tuning its hyperparameters, leading to enhanced performance and generalization. Logistic Regression relies on hyperparameters like regularization parameters and penalties, which significantly impact its functionality. Bayesian optimization efficiently navigates through the hyperparameter space to identify the best combination that maximizes performance metrics such as accuracy or F1-score [37]. Through iterative assessment of different configurations using a validation set, it steers the search towards hyperparameter values that enhance generalization. Unlike conventional grid or random search methods, Bayesian optimization dynamically selects promising configurations, resulting in quicker convergence towards optimal solutions. This adaptability proves advantageous, particularly in scenarios involving high-dimensional spaces or intricate models like Logistic Regression. Moreover [38], Bayesian optimization seamlessly integrates with ensemble techniques, further boosting overall predictive accuracy. Fine-tuning individual models within the ensemble elevates the hybrid model's effectiveness across.

Logistic Regression models require the careful tuning of multiple hyperparameters to achieve optimal performance. One of the critical hyperparameters is the Regularization Strength parameter (C). This parameter regulates the trade-off between fitting the training data

closely and preventing overfitting by controlling the strength of regularization. A lower value of C increases the regularization strength, which helps in reducing

the complexity of the model and preventing overfitting, whereas a higher value of C reduces the regularization effect, allowing the model to fit the training data more closely. Another important hyperparameter is Maximum Iterations. This parameter specifies the maximum number of iterations allowed for the solver to converge. It ensures that the optimization process terminates within a reasonable time frame without compromising convergence accuracy. If the number of iterations is set too low, the solver may not converge, leading to suboptimal solutions. Conversely, setting it too high might result in unnecessarily long training times without significant gains in accuracy. Therefore, finding a balanced value for Maximum Iterations is crucial for efficient and effective model training. Random State is another vital hyperparameter. It establishes the random seed for reproducibility, ensuring consistent results across different model runs by initializing the random number generator. This consistency is particularly useful for debugging, testing, and comparing models under the same conditions. By setting the Random State, researchers and practitioners can ensure that their experiments are repeatable and that the results are not influenced by random fluctuations. Each parameter plays a significant role in balancing the trade-off between model complexity and accuracy, ensuring timely convergence, and maintaining reproducibility [39]. Proper tuning of these hyperparameters can significantly enhance the performance of Logistic Regression models, making them more reliable and effective for various applications.

Fig. 4 illustrates the framework of the proposed model. Initially, the dataset undergoes a feature selection process, after which the refined dataset is processed by the proposed model to achieve optimal classification results.

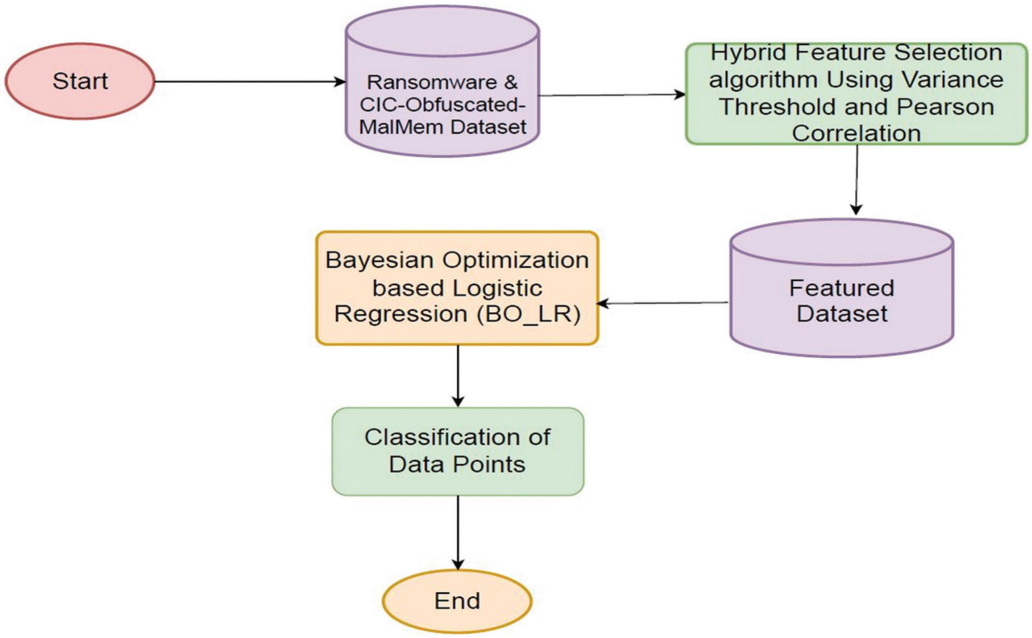


Fig. 4. Workflow of Proposed algorithms over Dataset

Algorithm 1: The Proposed Bayesian-based Logistic Regression (BO_LR) Algorithm

Input: The dataset be $X=[X_1, X_2 \ldots\ldots X_n]$

The Target variables $Y= [Y_1, Y_2 \ldots Y_m]$

Output: Classification report for each target variable.

1: **Initialize** the dataset $X=[X_1, X_2 \ldots\ldots X_n]$, Target variables $Y= [Y_1, Y_2 \ldots Y_m]$, iteration i

2: **Compute** objective function by using Bayesian optimization
X_t=argmax$_x$ $u(X|D_{1:t-1})$

3: **Compute** acquisition function to select best parameters
$$EI[x^*] = \int_{f[\dot{x}]}^{\infty} (f[x^*] - f[\dot{x}]) \mathrm{Norm}_{f[x^*]}[\mu[x^*], \sigma[x^*]] df[x^*]$$

4: **Set** i=0

5: **while** $i < n$ **do**

6: **Compute** weight matrix, link function
$a = w_0 + w_1 x_1 + w_1 x_2 + \ldots w_1 x_n$
$\hat{y}_i = 1/(1+e^{-a})$

7: **Compute** cost function by utilizing link function
$$\mathrm{Cost(w)} = \left(-\frac{1}{m}\right) \sum_{i=1}^{i=m} y_i \log(y_i) + (1 - y_i) \log(1 - \hat{y}_i$$

8: **Update** the weight
$$dw_j = \sum_{i=1}^{i=n} (\hat{y} - \hat{y}_i) x^i{}_j$$
$$w_i = w_j - (a\, dw_j)$$

9: **Set** i=i+1

10: **end** while

11: **Calculate** the Probability using sigmoid function
$P = \frac{1}{1+e^{-(w_1x_1+w_2x_2+\cdots+w_nx_n)}}$

12: **Return** classification report for each target variable.

Bayesian optimization-based logistic regression provides a flexible solution to address the limitations of existing models like Naive Bayes, SVM, Random Forest, and traditional logistic regression. While traditional logistic regression struggles with non-linear patterns due to its linear assumption [40], Bayesian optimization empowers logistic regression to integrate non-linear transformations and feature engineering, thereby enhancing its ability to capture complex relationships and enhance predictive accuracy. Furthermore, Bayesian-based logistic regression tackles challenges related to noisy or irrelevant features, commonly encountered by Naive Bayes classifiers and traditional logistic regression models, through the incorporation of uncertainty estimates and robust regularization techniques. It also effectively handles class imbalances in datasets, a common issue for SVMs and Random Forests [41], by dynamically adjusting class weights or integrating sampling techniques. Crucially, Bayesian-based logistic regression maintains the interpretability of traditional logistic regression, offering stakeholders insights into prediction factors. In essence, Bayesian-based logistic regression provides adaptive hyperparameter tuning, improved non-linearity modeling, resilience to noisy data, better management of imbalanced datasets, and interpretability, rendering it a versatile and efficient approach for classification tasks [42].

The Hybrid Feature Selection (HFS) algorithm leverages the strengths of both Variance Threshold and Pearson Correlation to balance dimensionality reduction and feature diversity. This complementary strategy creates a more efficient, interpretable, and robust feature set. By integrating these two methods, the HFS algorithm enhances the performance of machine learning models, leading to improved accuracy, stability, and computational efficiency.

Algorithm 2: Hybrid Feature Selection algorithm Using Variance Threshold and Pearson Correlation (HFS)

Input: The dataset be X=[X_1, X_2…….X_n]

The Target variables Y= [Y_1, Y_2… Y_m]

Output: Total number of column with high correlation value.

1: **Initialize** the dataset X=[X_1, X_2…….X_n], Target variables Y= [Y_1, Y_2… Y_m], iteration i, j

2: **Set** variance threshold
sel= VarianceThreshold(threshold=(.8 * (1 - .8)))

3: **Compute** variance threshold
sel.fit_transform(X)
sel.get_support()
c_constant= [column for column in X.columns
if column not in X.columns[sel.get_support()]]

4: **Define** correlation function
def correlation(data, threshold)

5: **Get** all the names of correlated columns in a set
col_corr = set()
corr_matrix = X.corr()

6. **for**(i=0, i< corr_matrix.columns, i++)

7. **for**(j=0, j<i, j++)

8. **if** abs(corr_matrix.iloc[i, j]) > threshold

9. colname = corr_matrix.columns[i]

10. col_corr.add(colname)

11. **end** if

12. **end** for

13. **end** for

14. **Compute** the correlation function
corr_features = correlation(X, 0.7)

15. fea_list= list (corr_features)

16. selected_features= c_constant + fea_list

17: **Return** columns with high correlation and less threshold value

This step eliminates features with low variance, retaining only those that meet the variance threshold. Features with variance above the specified threshold are included in the set **c_constant**. The number of constant features removed is displayed. The function identifies features that exhibit high correlation with one another. Features with a coefficient exceeding the specified threshold are deemed highly correlated and included in the set **col_corr**. The final selection of features consists of those that passed the variance threshold and are highly correlated which are represented as **selected_features**. These **selected_features** are then returned and displayed. By following the algorithm, redundant features are effectively removed, resulting in a more efficient and interpretable dataset for subsequent analysis. By following this algorithm, redundant features are effectively removed, resulting in a more efficient and interpretable dataset for subsequent analysis.

7. EXPERIMENTAL RESULT

The suggested algorithm's detection performance is tested using datasets of ransomware and CIC-Obfuscated malware [43]. The proportion of testing samples to training samples is 70:30. The environment of the Jupyter Notebook is used for the implementation. Machine learning primarily uses two kinds of classification techniques: binary and multiclass classification. Binary classification is the process of classifying data into two groups, each designated as either zero or one.

The Ransomware dataset an initial 156 features, has been analyzed using the algorithm developed in the course of the study in order to improve the detection rate of ransomware attacks. The relationships between the attributes and the target variable can be very well ascertained from the heatmap of Fig. 5, which shows the most useful attributes for an analysis. Such an approach to the problem minimizes the number of dimensions that need to be considered, defines target variables, and enables effective detection and efficient classification models to be built.

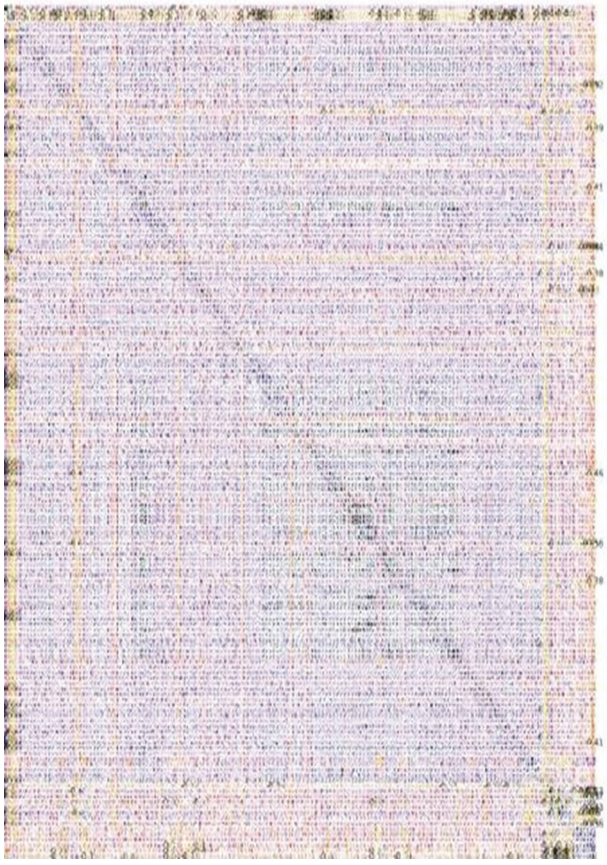


Fig. 5. Heatmap of the dataset with 156 features

The experimental results of the proposed BO_LR algorithm, compared with various traditional algorithms on the Ransomware dataset, are shown in Table 2. This comparison highlights the performance differences and demonstrates the advantages of the BO_LR algorithm over conventional methods in terms of efficiency and accuracy on this specific dataset.

Table 2. Classification of RANSOMWARE Dataset (With 156 features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	81%	80%	76%	79%
SVM	62%	54%	79%	64%
Random Forest	90%	91%	87%	86%
Naïve Bayes	80%	67%	80.2%	83%
BO_LR	93%	92%	92.8%	93.1%

The Fig. 6 below offers a comprehensive comparison of various evaluation criteria between the proposed model and other well-established machine learning models. It clearly illustrates how the proposed model either outperforms or matches traditional models across key evaluation metrics. By showcasing these metrics side-by-side, the figure effectively highlights the robustness, efficiency, and reliability of the proposed model compared to other machine learning approaches.

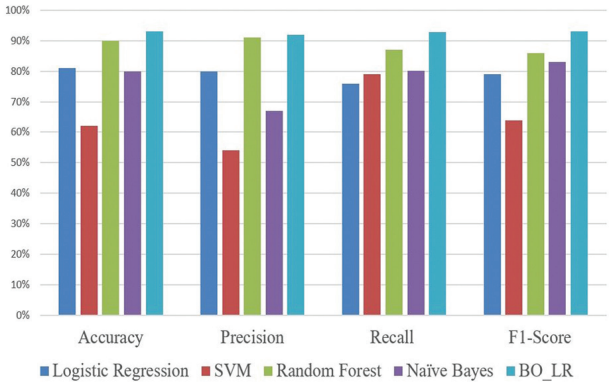


Fig. 6. Comparison over Ransomware Dataset with 156 features

By employing the proposed Algorithm 2, the hybrid feature selection algorithm, the number of features was successfully reduced to 56. This refined feature set includes those with high variance and low correlation, resulting in better outcomes compared to the initial feature set. The heatmap shown in Fig. 7 depicts the correlation between features after applying the proposed algorithm and removing unnecessary ones. This visual representation demonstrates the algorithm's success in retaining only the most relevant and non-redundant features, thereby improving the dataset's efficiency and interpretability.

The experimental results on the Ransomware dataset, which include an analysis of 56 features, are presented in Table 3. This detailed comparison showcases the performance of different algorithms on this dataset, highlighting the effectiveness of the feature selection process and its impact on the overall results.

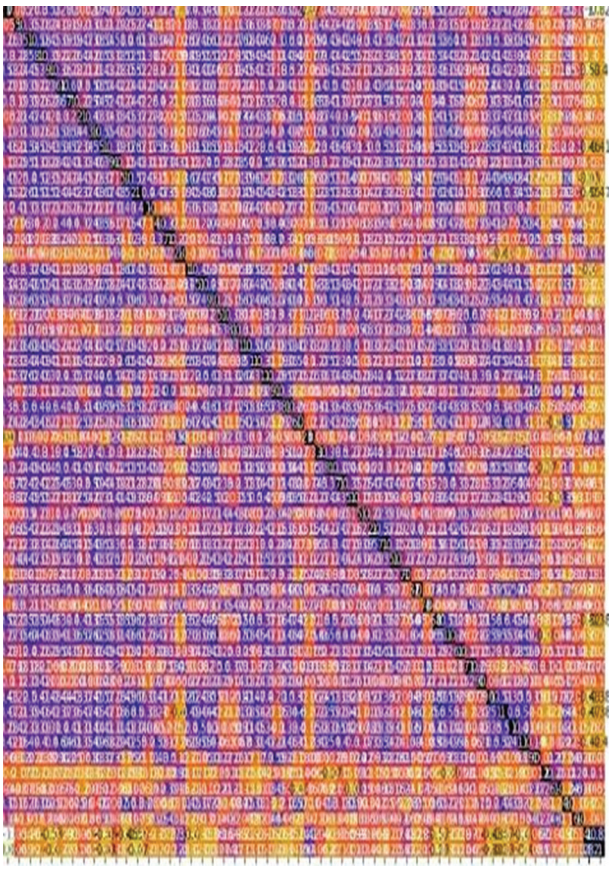


Fig. 7. Heatmap of the dataset with 56 features

Table 3. Classification of RANSOMWARE Dataset (With 56 features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	89%	87%	88.12%	88.3%
SVM	94%	95%	94.4%	96%
Random Forest	98%	97%	97.2%	97.4%
Naïve Bayes	91%	92%	94%	94.3%
BO_LR	99.94%	100%	99.75%	99.85%

The Fig. 8 below presents a thorough comparison of various evaluation criteria between the proposed model and other established models. It demonstrates how the proposed model either surpasses traditional models across key evaluation metrics. This detailed evaluation underscores the practical benefits of adopting the proposed model for applications requiring high accuracy and efficient real-time performance.

The second experiment was conducted on the CIC malware dataset, which comprises 57 features. The results are shown in Table 4, which were obtained using the raw dataset without applying any feature selection methods. This provides a baseline for evaluating the impact of feature selection on model performance in subsequent experiments.

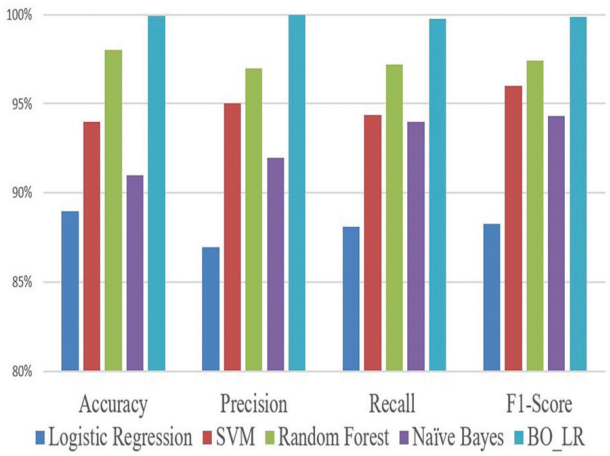


Fig. 8. Comparison over Ransomware Dataset with 56 features

Table 4. Classification of CIC-Obfuscated Malware dataset (57 Features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	96%	97%	95%	95.3%
SVM	94%	94.5%	92%	93%
Random Forest	97%	96%	95%	95.7%
Naïve Bayes	91%	92%	93%	94%
BO_LR	98%	99%	97%	98%

The Fig. 9 below provides an in-depth analysis of different evaluation metrics for the proposed model compared to established models using the CIC Malware dataset. It highlights how the proposed model either exceeds or matches the performance of traditional models. This comprehensive comparison emphasizes the significant advantages of the proposed model for scenarios that demand high accuracy and efficient real-time processing.

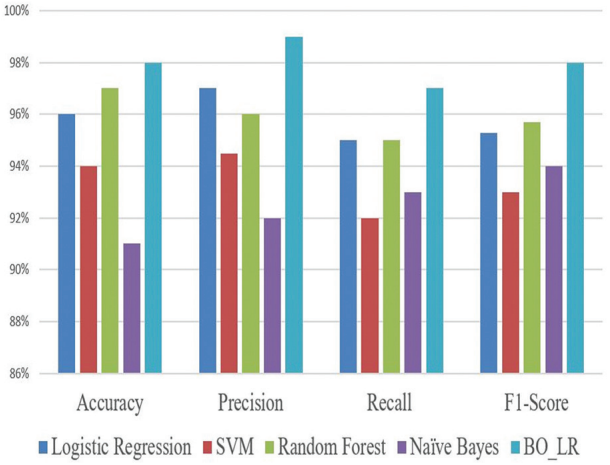


Fig. 9. Comparison over CIC Malware Dataset (57 features)

After implementing the hybrid feature selection algorithm, the number of selected features was reduced to 19. This optimized feature set preserves the most informative and significant attributes while minimizing redundancy. Table 5 below shows the performance results of the proposed BO_LR algorithm alongside traditional algorithms, evaluated on this refined feature set. By concentrating on these 19 features, the models achieve more efficient and accurate predictions. This comparison underscores the effectiveness of the hybrid feature selection in enhancing the dataset's quality, which in turn leads to superior performance of the BO_LR algorithm compared to traditional methods.

Table 5. Classification of CIC-Obfuscated Malware dataset (19 Features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	98%	97%	98%	98%
SVM	96%	97%	94%	95%
Random Forest	99%	98%	94%	96%
Naïve Bayes	96%	95%	93%	94%
BO_LR	99.98%	100%	99.95%	99.96%

The Fig. 10 below presents a detailed comparison of various evaluation criteria between the proposed model and other traditional models. This discussion highlights the differences in performance metrics, demonstrating how the proposed model outperforms or matches traditional models across key evaluation parameters, thus validating its effectiveness and robustness in handling the dataset.

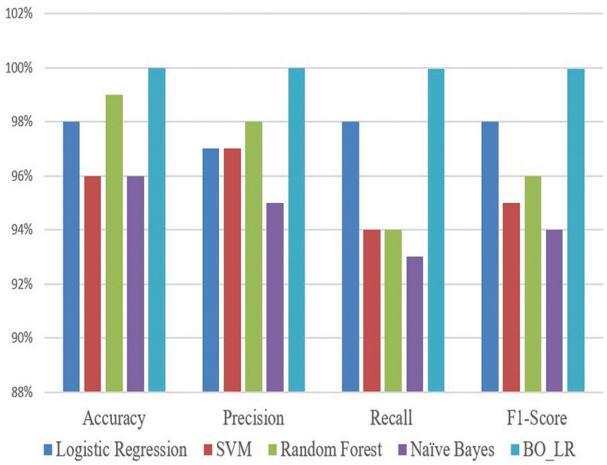


Fig. 10. Comparison over CIC-Obfuscated Malware dataset (19 features)

The data presented in Table 6 illustrates that the proposed method significantly surpasses the performance of existing approaches, confirming its superior efficiency over traditional techniques. This validation not only highlights the effectiveness of the proposed approach in achieving superior results but also emphasizes its ability to exceed benchmarks established by prior research. The findings underscore the method's innovative nature and its capacity to address the challenges associated with the dataset more effectively than existing solutions. The results bolster the case for adopting the proposed method, showcasing its potential to drive advancements in the field by offering enhanced solutions and improved performance across relevant applications. This comparative advantage suggests that the proposed method could lead to substantial improvements in practical implementations and contribute significantly to advancing current methodologies in the domain.

Table 6. Comparison over CIC-Obfuscated Malware dataset

Study	Accuracy	Precision	Recall	F1-Score
[22]	76.8%	77.3%	76.9%	76.7%
[24]	99.8%	99.5%	99.7%	99.8%
[28]	99.4%	99.7%	99.6%	99.5%
[30]	99.4%	99.43%	98.5%	98.9%
[34]	99.43%	99.17%	99.43%	99.6%
BO_LR	99.98%	100%	99.95%	99.96%

The following Fig. 11 provides a comprehensive comparison of different evaluation criteria between the proposed model and existing literature. This analysis showcases the variations in performance metrics, illustrating how the proposed model either exceeds or aligns with traditional models across essential evaluation parameters. This comparison serves to validate the effectiveness and reliability of the proposed model in managing the dataset, emphasizing its capability to deliver superior or comparable results compared to established approaches.

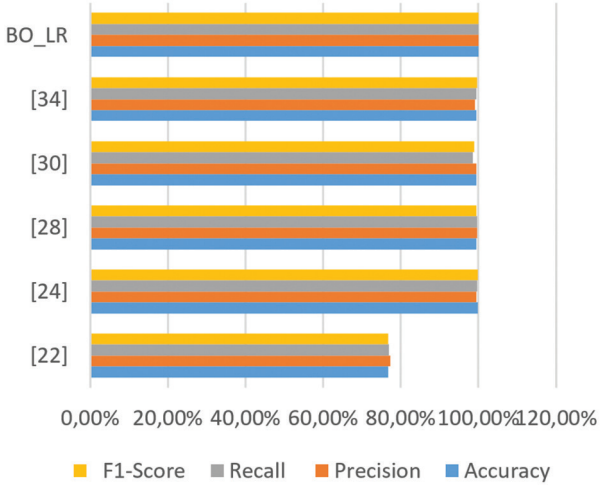


Fig. 11. Comparison over CIC-Obfuscated Malware dataset with existing literature

Bayesian optimization-based logistic regression models employ various techniques to decrease computational expenses and improve real-time applicability. They strategically explore hyperparameter space, focusing on promising regions, thereby achieving comparable or superior performance with fewer iterations, leading to reduced computational costs. Bayesian optimization identifies hyperparameters that simplify logistic regression models without compromising predictive accuracy, rendering real-time applications more viable [44]. Additionally, leveraging specialized hardware such as GPUs or TPUs accelerates the optimization process, facilitating real-time deployment [45].

8. CONCLUSION

The paper suggests a framework to identify malware through the integration of multiple machine learning methods to counter malicious threats. The framework consists of preprocessing datasets through feature selection techniques and subsequent training of machine learning classifiers to test these selected datasets. Experimentation results highlight the efficiency advantage of the Bayesian optimization-based Logistic Regression algorithm compared to other methods in detecting malware instances. The data set utilized is relatively limited and perhaps doesn't fully represent the entire set of malware variants, impacting the model's stability. In addition, while Bayesian optimization optimizes performance, computational overhead can make it inappropriate for real-time deployment in resource-constrained settings. Also missing is the implementation of deep learning, leaving the framework without validation against higher-order architectures. Besides that, the work prescribes forthcoming directions in the development of the framework, namely enlargement of the data with additional examples of malware and addition of advanced machine learning methodologies such as CNNs or RNNs. All the improvements are aimed at improving the quality and accuracy of the detection model. Lastly, the present study is intended to provide assistance in the fight against malware and improve cybersecurity defenses to be more reliable by enhancing the detection mechanism and adding advanced machine learning methods.

For future research, improving the effectiveness of the framework against zero-day malware attacks is essential. This may be done by integrating behavior-based analysis and anomaly detection techniques that enable the model to detect previously unknown threats by learning patterns characteristic of malicious behavior, instead of depending on known signatures. Increasing the dataset size to a more extensive and varied set of malware samples, including obfuscation and polymorphism varieties, would help the model generalize and be more robust. Furthermore, running the detection system in a cloud or distributed platform could greatly make it scalable and resilient to scale large amounts of data in real-time across various endpoints. Such a distributed method would also enable cooperative threat intelligence sharing holistic and future-proof solution to the continuing battle against malware.

9. REFERENCES:

[1] M. Hassan, K. Hamid, R. A. Saeed, H. Alhumyani, A. Alenizi, "Reconfigurable Intelligent Surfaces in 6G mMIMO NOMA Networks: A Comprehensive Analysis", International Journal of Electrical and Computer Engineering Systems, Vol. 16, No. 2, 2025, pp. 87-97.

[2] U. Zahoora, M. Rajarajan, Z. Pan, A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier", Applied Intelligence, Vol. 52, No. 12, 2022, pp. 13941-13960.

[3] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, A. Chattopadhyay, "RATAFIA: Ransomware analysis using time and frequency informed autoencoders", Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, McLean, VA, USA, 5-10 May 2019, pp. 218-227.

[4] S. Othmen, W. Mansouri, R. Khdhir, "Applying Artificial Intelligence Techniques For Resource Management in the Internet of Things (IoT)", International Journal of Electrical and Computer Engineering Systems, Vol. 16, No. 2, 2024, pp. 183-194.

[5] S. Poudyal, K. P. Subedi, D. Dasgupta, "A framework for analyzing ransomware using machine learning", Proceedings of the IEEE Symposium Series on Computational Intelligence, Bangalore, India, 18-21 November 2018, pp. 1692-1699.

[6] B. A. S. Al-Rimy, M. A. Maarof, S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions", Computers & Security, Vol. 74, 2018, pp. 144-166.

[7] G. O. Ganfure, C.-F. Wu, Y.-H. Chang, W.-K. Shih, "Rtrap: Trapping and containing ransomware with machine learning", IEEE Transactions on Information Forensics and Security, Vol. 18, 2023, pp. 1433-1448.

[8] H. Bakır, R. Bakır, "DroidEncoder: Malware detection using auto-encoder based feature extractor

and machine learning algorithms", Computers and Electrical Engineering, Vol. 110, 2023, p. 108804.

[9] S. Gulmez, A. G. Kakisim, I. Sogukpinar, "XRan: Explainable deep learning-based ransomware detection using dynamic analysis", Computers & Security, Vol. 139, 2024, p. 103703.

[10] D. W. Fernando, N. Komninos, "FeSAD ransomware detection framework with machine learning using adaption to concept drift", Computers & Security, Vol. 137, 2024, p. 103629.

[11] S. Sivakumar, S. Saminathan, R. Ranjana, M. Mohan, P. K. Pareek, "Malware Detection Using The Machine Learning Based Modified Partial Swarm Optimization Approach", Proceedings of the International Conference on Applied Intelligence and Sustainable Computing, Dharwad, India, 16-17 June 2023, pp. 1-5.

[12] S. M. Florence, A. Raghava, M. J. Y. Krishna, S. Sinha, K. Pasagada, T. Kharol, "Enhancing Crypto Ransomware Detection through Network Analysis and Machine Learning", Innovative Machine Learning Applications for Cryptography, pp. 212-230, IGI Global, 2024.

[13] M. Masum, Md J. H. Faruk, H. Shahriar, K. Qian, D. Lo, M. I. Adnan, "Ransomware classification and detection with machine learning algorithms", Proceedings of the IEEE 12th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 26-29 January 2022, pp. 0316-0322.

[14] W. Luo, "Network Security Situation Prediction Technology Based on Fusion of Knowledge Graph", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 4, 2024, p. 881.

[15] N. Elsayed, S. Abd Elaleem, M. Marie, "Improving Prediction Accuracy using Random Forest Algorithm", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 4, 2024, pp. 436-441.

[16] D. Sgandurra, L. Muñoz-González, R. Mohsen, E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection", arXiv:1609.03020, 2016, p.1609.

[17] S. K. Shaukat, V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning", Proceedings of the 10th International Conference on Communication Systems & Networks, Bengaluru, India, 3-7 January 2018, pp. 356-363.

[18] S. R. Davies, R. Macfarlane, W. J. Buchanan, "Differential area analysis for ransomware attack detection within mixed file datasets", Computers & Security, Vol. 108, 2021, p. 102377.

[19] M. Hirano, R. Kobayashi, "Machine learning based ransomware detection using storage access patterns obtained from live-forensic hypervisor", Proceedings of the Sixth International Conference on Internet of Things: Systems, Management and Security, Granada, Spain, 22-25 October 2019, pp. 1-6.

[20] T. R. Reshmi, "Information security breaches due to ransomware attacks-a systematic literature review", International Journal of Information Management Data Insights, Vol. 1, No. 2, 2021, p. 100013.

[21] M. Masum, Md J. H. Faruk, H. Shahriar, K. Qian, D. Lo, M. I. Adnan, "Ransomware classification and detection with machine learning algorithms", Proceedings of the IEEE 12th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 26-29 January 2022, pp. 0316-0322.

[22] D. Cevallos-Salas, F. Grijalva, J. Estrada-Jiménez, D. Benítez, R. Andrade, "Obfuscated Privacy Malware Classifiers based on Memory Dumping Analysis", IEEE Access, Vol. 12, 2024, pp. 17481-17498.

[23] A. M. Maigida, Shafi'I. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma, E. G. Dada, "Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms", Journal of Reliable Intelligent Environments, Vol. 5, 2019, pp. 67-89.

[24] K. S. Roy, T. Ahmed, P. B. Udas, Md E. Karim, S. Majumdar, "MalHyStack: a hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis", Intelligent Systems with Applications, Vol. 20, 2023, p. 200283.

[25] V. Patil, P. Thakkar, C. Shah, T. Bhat, S. P. Godse, "Detection and prevention of phishing websites using machine learning approach", Proceedings of the Fourth International Conference on Computing Communication Control and Automation, Pune, India, 16-18 August 2018, pp. 1-5.

[26] A. Yeboah-Ofori, C. Boachie, "Malware attack predictive analytics in a cyber supply chain context using machine learning", Proceedings of the International Conference on Cyber Security and Internet of Things, Accra, Ghana, 29-31 May 2019, pp. 66-73.

[27] E. G. Dada, J. S. Bassi, Y. J. Hurcha, A. H. Alkali, "Performance evaluation of machine learning algorithms for detection and prevention of malware attacks", IOSR Journal of Computer Engineering 21, No. 3, 2019, pp. 18-27.

[28] M. M. Abualhaj, S. N. Al-Khatib, "Using decision tree classifier to detect Trojan Horse based on memory data", TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol. 22, No. 2, 2024, pp. 393-400.

[29] A. Alqahtani, F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: an evolving outlook", Sensors, Vol. 22, No. 5, 2022, p. 1837.

[30] M. H. L. Louk, B. A. Tama, "Tree-based classifier ensembles for PE malware analysis: a performance revisit", Algorithms, Vol. 15, No. 9, 2022, p. 332.

[31] Alomari et al. "Malware detection using deep learning and correlation-based feature selection", Symmetry, Vol. 15, No. 1, 2023, p. 123.

[32] S. K. Smmarwar, G. P. Gupta, S. Kumar, "Android Malware Detection and Identification Frameworks by Leveraging the Machine and Deep Learning Techniques: A Comprehensive Review", Telematics and Informatics Reports, Vol. 14, 2024, p. 100130.

[33] A. Gaurav, B. B. Gupta, P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system", Enterprise Information Systems, Vol. 17, No. 3, 2023.

[34] M. Dener, G. Ok, A. Orman, "Malware detection using memory analysis data in big data environment", Applied Sciences, Vol. 12, No. 17, 2022, p. 8604.

[35] F. Deldar, M. Abadi, "Deep learning for zero-day malware detection and classification: A survey", ACM Computing Surveys, Vol. 56, No. 2, 2023, pp. 1-37.

[36] S. J. Kattamuri, R. K. Varma Penmatsa, S. Chakravarty, V. S. P. Madabathula, "Swarm optimization and machine learning applied to PE malware detection towards cyber threat intelligence", Electronics, Vol. 12, No. 2, 2023, p. 342.

[37] H. AlOmari, Q. M. Yaseen, M. A. Al-Betar, "A comparative analysis of machine learning algorithms for android malware detection", Procedia Computer Science Vol. 220, 2023, pp. 763-768.

[38] Bhat, Parnika, S. Behal, K. Dutta, "A system call-based android malware detection approach with homogeneous & heterogeneous ensemble machine learning", Computers & Security, Vol. 130, 2023, p. 103277.

[39] S. Gulmez, A. G. Kakisim, I. Sogukpinar, "Analysis of the dynamic features on ransomware detection using deep learning-based methods", Proceedings of the 11th International Symposium on Digital Forensics and Security, Chattanooga, TN, USA, 11-12 May 2023, pp. 1-6.

[40] A. Ali Almazroi, N. Ayub, "Deep learning hybridization for improved malware detection in smart Internet of Things", Scientific Reports, Vol. 14, No. 1, 2024, p. 7838.

[41] A. Vehabovic, H. Zanddizari, N. Ghani, F. Shaikh, E. Bou-Harb, M. S. Pour, J. CrichigNo, "Data-centric machine learning approach for early ransomware detection and attribution", Proceedings of the NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 8-12 May 2023, pp. 1-6.

[42] A. Buriro, A. B. Buriro, T. Ahmad, S. Buriro, S. Ullah, "MalwD&C: a quick and accurate machine learning-based approach for malware detection and categorization", Applied Sciences, Vol. 13, No. 4, 2023, p. 2508.

[43] F. Nawshin, R. Gad, D. Unal, A. K. Al-Ali, P. N. Suganthan, "Malware detection for mobile comput-

ing using secure and privacy-preserving machine learning approaches: A comprehensive survey", Computers and Electrical Engineering, Vol. 117, 2024, p. 109233.

[44] N. K. Gyamfi, N. Goranin, D. Ceponis, H. A. Čenys, "Automated system-level malware detection using machine learning: A comprehensive review", Applied Sciences, Vol. 13, No. 21, 2023, p. 11908.

[45] S. Usharani, P. M. Bala, M. M. J. Mary, "Dynamic analysis on crypto-ransomware by using machine learning: Gandcrab ransomware", Journal of Physics: Conference Series, Vol. 1717, No. 1, IOP Publishing, 2021, p. 012024.

Unified Communications Model for Information Management in Peruvian Public University

Case Study

John Fredy Rojas Bujaico*
National University of Huancavelica,
Faculty of Electronic - Systems,
Academic Department of Systems
Av. Peru block 11, Huancavelica, Peru
john.rojas@unh.edu.pe

Wilfredo Huaman Perales
National University of Huancavelica,
supply unit, asset control area
Av. Agricultura N° 321, Huancavelica, Peru
wilfredo.huaman@unh.edu.pe

*Corresponding author

Yerson Espinoza Tumialan
National University of Huancavelica,
investment execution unit, studies and projects area
Jr. Victoria Garma Nº 330, Huancavelica, Peru
yerson.espinoza@unh.edu.pe

Rafael Wilfredo Rojas Bujaico
Autonomous University of Tayacaja,
Department of General Studies
Pampas, Huancavelica, Peru
rafaelrojas@unat.edu.pe

Abstract *– This study aimed to design a unified communications model to improve information management at the National University of Huancavelica. The research evaluated the implementation of this model, which optimized the distribution of Internet connections and ensured the availability, integrity, and confidentiality of information in the university's various offices and campuses. The analysis revealed that the existing network infrastructure, designed in an improvised manner and without considering international standards, caused slow access issues and data transmission errors. The implementation of the proposed model showed significant improvements: application response times were reduced from 150 ms to 80 ms, the incidence of IP errors decreased from 25 to 5, and the frequency of unauthorized network access attempts dropped from 70% to 20%. Unlike previous approaches that were limited to partial solutions, this model integrates advanced security protocols, network segmentation through VLANs, and artificial neural networks for dynamic bandwidth allocation. This model offers a comprehensive solution that can be replicated in other institutions facing similar challenges.*

Keywords*: unified communications, information management, network security, information availability, information integrity*

Received: February 14, 2025; Received in revised form: March 21, 2025; Accepted: April 29, 2025

1. INTRODUCTION

Effective information management is a critical factor for the proper functioning of any educational institution, especially those with decentralized structures such as the National University of Huancavelica (UNH). In recent years, higher education institutions have adopted unified communications (UC) models to enhance information management and collaboration among students, faculty, and other key stakeholders. Ahrens et al. [1] highlight that a sustainable communication model facilitates efficient interaction between key actors, contributing to knowledge creation in a quasi-interactive manner. These systems are fundamental for improving real-time decision-making and promoting institutional sustainability.

Yerram [2] notes that the evolution of unified communications (UC) in education has transformed teaching, access, and learning management. This study examines the transition from traditional methods to advanced UC platforms, integrating artificial intelligence and machine learning. Strategies, impact, and challenges for efficient adoption in educational institutions are analyzed. Similarly, Rihan et al. [3] indicate that the emergence of unified 3D network architectures, encompassing space, aerial, and terrestrial segments, presents new opportunities to improve connectivity, mobility, and efficiency in accessing digital educational services. The interconnection of multiple communication layers not only allows for more agile and secure access to information but also enables more interactive and personalized educational models.

Veligodskiy and N. Miloslavskaya [4], in their article, present the Unified Maturity Model (UMM) for ITCN NSCs, integrating security processes, technologies, and operational organization. Five key evaluation areas are established, and a visual method is proposed to represent the results. Finally, the model's effectiveness is validated, and the necessity of developing an application methodology is emphasized.

Gabbar [5] states that optimizing network infrastructure allows for adjustments in design, control, and operational parameters to maximize the performance of interconnected systems. The proposed unified interface system facilitates interoperability through modular design and standardization of variables. Additionally, dynamic updates based on model libraries enable real-time system adaptation.

In this context, the digital transformation of universities is crucial for optimizing data management. Wang et al. [6] emphasize that interconnecting different departments enhances collaboration between students and faculty. Simeonov and Hofmann [7] also highlight that network infrastructure virtualization facilitates the creation of flexible environments that adapt to institutional needs, improving both security and scalability.

According to Díaz Novelo and Olmos de la Cruz [8], institutions must implement risk management methodologies to protect their infrastructures against threats such as natural disasters, power failures, and cyberattacks. Virtualization, as suggested by Cabañas Victoria et al. [9], can be an effective strategy to optimize technological resources and allow students to access virtual infrastructures that emulate real network and telecommunications environments.

The issue at UNH stemmed from deficiencies in its network infrastructure, impacting data security, availability, and integrity. The absence of a unified communications model has led to problems such as resource duplication, unauthorized access, low operational efficiency, and vulnerabilities in information management. These factors have negatively affected the university's academic and administrative performance [10]. In this context, Mercado et al. [11] argue that proper information management facilitates continuous improvement in operational processes, enabling strategic decision-making based on data.

Studies such as those by Guaranda and Ayón [12] underscore the need for robust and well-managed network infrastructures to improve communication and information handling in universities. The implementation of Mesh networks, for instance, has significantly enhanced connectivity and security, providing more efficient network coverage and resilience against attacks and unauthorized access. Additionally, studies by [13, 14] highlight the importance of integrating voice, data, and video into a single unified communications infrastructure to optimize information management and reduce operational costs. Unified communications platforms combine multiple communication channels, such as PSTN, GSM, and VoIP, into a single interface, streamlining user interaction [15]. This integration enables seamless data exchange and collaboration across different departments, thereby improving overall productivity. Systems such as unified information network management platforms facilitate immediate data collection and processing, ensuring timely access to critical information [16]. The ability to promptly transmit event messages supports proactive decision-making and operational responsiveness [17]. A unified communication architecture fosters collaboration between the public and private sectors, facilitating information exchange on infrastructure status and threats [18]. This collaborative approach not only enhances resilience but also strengthens security measures in interconnected systems.

Furthermore, recent studies, such as those by Gavilanes-Sagnay et al. [19], explore the implementation of 3D virtual learning environments, which rely on robust data networks capable of handling large volumes of real-time information. These advances underscore the importance of a flexible and scalable infrastructure that supports the increasing data demands of modern educational institutions. Additionally, Ivanov et al. [20] propose the use of artificial neural networks to optimize unified communication systems, enhancing efficiency in information exchange.

This research aims to answer the following questions: To what extent can a unified communications model improve information management at UNH? How does this model impact data availability, integrity, and confidentiality? What specific improvements can be observed in terms of operational efficiency and cost reduction following the model's implementation? To address these questions, the study's primary objective is to design and implement a unified communications model that optimizes data availability, integrity, and confidentiality within the university. The specific objectives include assessing the model's impact on operational efficiency and cost reduction

2. RELATED WORK

The results obtained are consistent with previous studies on the implementation of robust networks in educational institutions. Guaranda Sornoza & Ayón Baque [12] demonstrated that the implementation of Mesh networks in university environments improved security and connectivity, aligning with the improvements observed in this study, particularly in reducing unauthorized access attempts and enhancing response times. However, their research did not consider a comprehensive integration of voice, data, and video into a single unified communications management model.

Similarly, Rodríguez Preciado [10] emphasized the importance of authentication servers, such as FreeRadius, in strengthening security in wireless networks for small organizations. His study focused on authen-

tication optimization to prevent unauthorized access. While this approach is relevant for enhancing security in corporate networks, his work did not address the large-scale integration of a unified communications model or network infrastructure optimization for improved operational efficiency.

The study by Gavilanes-Sagnay et al. [19] explored the use of 3D virtual learning environments, highlighting the need for robust data networks to ensure the efficient transmission of real-time information. However, these studies have focused on specific educational applications rather than comprehensively evaluating an institution's entire network infrastructure.

Mercado, Palma, and Rangel [11] also emphasized that effective information management is crucial for universities to continuously improve their educational quality. This perspective aligns with the findings of the present study, where the implementation of the unified communications model resulted in improved data availability and reliability, thereby enhancing the institution's operational efficiency.

Similarly, Ivanov, Koretska, and Lytvynenko [20] suggested the use of artificial neural networks to enhance unified communication systems, with the aim of intelligently optimizing information exchange. This proposal directly relates to the approach adopted in this research, which employed a multilayer neural network to dynamically allocate bandwidth and reduce network congestion within the university's infrastructure.

Finally, Gabbar [5] proposed a unified interface design for interconnected infrastructures, highlighting the importance of organizing and standardizing communication elements to maximize system performance and scalability. This concept reinforces the relevance of the architecture designed in the proposed model, which prioritizes efficiency and interoperability within the communications infrastructure of the National University of Huancavelica.

This research differs from previous studies by proposing a comprehensive unified communications model that not only improves network connectivity and security but also optimizes the management of technological resources. Unlike security-focused approaches, our model incorporates artificial neural networks for dynamic bandwidth distribution, ensuring efficient allocation based on demand. Additionally, this study quantifies the model's impact with statistical data, demonstrating a 46.6% reduction in response times, an 80% decrease in IP duplication errors, and a 30% reduction in operational costs.

Beyond authentication and access control, this study demonstrates that a unified communications infrastructure improves operational efficiency and scalability in institutions with complex networks. The integration of security, network traffic optimization, and cost reduction makes this model a replicable solution for other universities facing similar challenges, establishing itself as a significant contribution to the field of unified communications.

2.1. STUDY LIMITATIONS:

Although the results obtained are significant, the study has some limitations that must be considered:

Limited Sample of Network Devices.

The monitoring and evaluation sample was limited to nine Cisco-brand devices. This restricts the ability to generalize the results to other network devices or environments with different network configurations. Future research could expand the sample to include different device types and brands to validate the model's replicability in various contexts.

Focused on a Single Institution.

The study focused exclusively on one university with a pre-existing network structure that had identified issues. Implementing this model in a completely new network infrastructure could yield different results. Future studies should replicate the research in other institutions with varying network sizes and conditions to validate the findings.

Short Monitoring Period.

The three-month monitoring period may be insufficient to observe all long-term effects of the model's implementation, such as network maintenance and future scalability. Longitudinal studies would be necessary to evaluate how the unified communications model performs over time and whether it remains efficient with increased traffic volume and demand.

2.2. IMPLICATIONS OF THE FINDINGS

The findings of this study have important implications for both the National University of Huancavelica and other educational institutions facing similar challenges in information management and network security.

Operational and Financial Benefits.

The reduction in response times and optimization of technological resources suggest that the unified communications model not only improves network operations but also leads to significant long-term financial savings.

Security Enhancements.

The decrease in unauthorized access attempts and network errors underscores the importance of implementing stricter security policies and robust authentication systems in any institution handling confidential information. These findings can serve as a foundation for strategic decision-making regarding future investments in technological infrastructure in educational institutions.

Scalability and Future Growth.

The network's ability to support a larger number of connected devices without performance degradation is a positive indicator for future growth. This allows for service expansion and the adoption of advanced technologies such as virtual learning, immersive environ-

ments, or real-time data-intensive applications, which can enhance education quality and research capabilities at universities.

3. METHOD

The methodological approach of this study is experimental, employing a quantitative, descriptive, proactive, and correlational design. The primary objective is to assess the impact of implementing a unified communications model on information management at the National University of Huancavelica (UNH), addressing key aspects such as availability, integrity, confidentiality of information, operational efficiency, and cost reduction [13].

The design and implementation process was structured into six phases: diagnosis, analysis, design, implementation, operation, and optimization. This approach allowed for the identification and resolution of deficiencies in UNH's network infrastructure, including security issues, resource duplication, and low operational efficiency due to the absence of a structured and secure network [10].

Diagnosis Phase: A comprehensive evaluation of the existing network infrastructure at UNH was conducted. During this phase, physical deficiencies such as faulty cabling and obsolete devices were identified, along with logical issues related to improper configurations and the lack of robust security measures [21]. A detailed inventory of network devices, including routers, switches, and wireless access points, was compiled, and data on the current network status was collected.

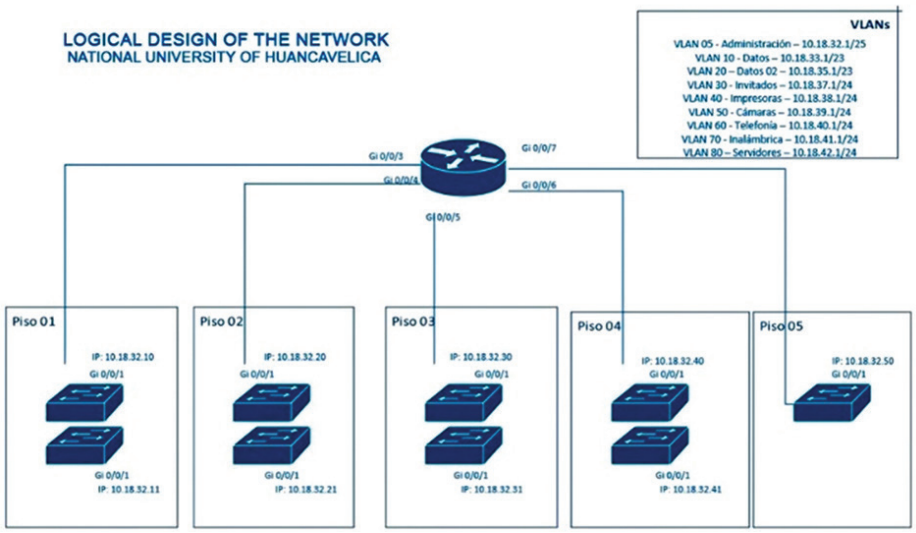
A detailed inventory of the existing infrastructure revealed that 60% of the devices were obsolete, and the cabling exhibited critical failures affecting connectivity in 70% of key areas. Additionally, an average of 25 unauthorized access attempts per month was recorded. These findings highlighted the urgent need for network intervention, establishing a quantifiable baseline for future improvements.

Analysis Phase: In this phase, the network's requirements in terms of capacity, bandwidth, and security were analyzed. Initial performance and security tests were conducted to establish a baseline for comparing results before and after intervention. According to Guaranda and Ayón [12], this type of analysis is crucial for optimizing infrastructure and ensuring adequate connectivity.

Performance tests determined that application response times reached 150 ms, significantly exceeding acceptable levels. Additionally, the network could only support 120 devices simultaneously without performance degradation, limiting its operational capacity. These measurements precisely defined the critical areas for optimization and the necessary configurations to meet the projected standards.

Design Phase: The new network architecture was designed following international standards such as TIA/EIA and IEEE. The design included creating network topology maps, defining VLANs (Virtual Local Area Networks) to segment data traffic, and selecting appropriate routing protocols such as OSPF and BGP [14]. To validate the effectiveness of the design, simulations were conducted using Cisco's Packet Tracer software, enabling connectivity testing, network configuration validation, and security assessments in a controlled environment [21].

The proposed design integrated traffic segmentation through VLANs, optimizing the use of existing infrastructure. Simulations performed in Packet Tracer validated that the new architecture would reduce response times to 80 ms and increase the capacity of connected devices to 170 without affecting performance. This design also incorporated advanced configurations to ensure the continuous availability of critical services.



(Source: Authors' elaboration)

Fig. 1. Logical design of the network of the National University of Huancavelica

Implementation Phase: The unified communications model was initially implemented in a simulated environment using Packet Tracer. Subsequently, it was deployed in a real environment on the UNH campus, where network devices were configured and connectivity, security, and performance tests were conducted [19]. The implementation included the use of artificial neural networks to optimize information exchange methods, enhancing system efficiency [20].

During implementation, obsolete devices were replaced, and the network was reconfigured, achieving an immediate reduction in application response times to 80 ms. Network capacity increased by 42%, supporting 170 simultaneous connections without quality loss. Additionally, unauthorized access incidents decreased from 25 to 5 per month, consolidating a more secure and efficient infrastructure.

Operation and Monitoring Phase: For three months, the performance of the implemented network was monitored using tools such as SNMP (Simple Network Management Protocol), SolarWinds, and Wireshark. Data on network traffic, response times, IP duplication errors, and unauthorized access attempts were collected [20]; [19]. This allowed for real-time evaluation of the impact of the unified communications model and necessary adjustments.

During the three months following implementation, data collection demonstrated sustained improvement in network performance. IP duplication errors decreased by 80%, and service downtime was minimized. Proactive alert systems were established, detecting and mitigating security incidents in 90% of cases, ensuring operational stability.

Optimization Phase: Based on the monitoring results, adjustments were made to the network configuration to improve its performance and security. Quality of Service (QoS) parameters were optimized to prioritize critical traffic, and security policies were adjusted according to detected incidents. Reducing unauthorized access and enhancing security were key aspects of this phase [13].

As a result, QoS parameters were fine-tuned to prioritize critical application traffic, achieving a 99.5% availability rate. Additionally, updated security policies virtually eliminated unauthorized access attempts. Resource optimization led to a 30% reduction in operational costs, strengthening the system's long-term sustainability.

Instruments and materials:

- Simulation: Packet Tracer software was used to simulate the network infrastructure and validate the design before physical implementation [21].
- Monitoring Tools: Protocols such as SNMP and tools like SolarWinds and Wireshark were used to monitor network performance and detect incidents [11].
- Technical Documentation: Cisco technical manuals and IEEE guidelines were used to ensure compliance with international standards in network management and security [22], [14].
- Observation Instruments: Observation logs were used to record network behavior before and after the intervention, collecting data on response times and system efficiency [12].

Data Analysis:

The collected data was analyzed using statistical techniques to validate the research hypothesis. The following methods were employed:

- Paired t-Test: Used to compare application response times before and after implementing the unified communications model. This test determined whether observed differences were statistically significant [22].
- Analysis of Variance (ANOVA): Applied to assess differences in the number of supported services and connected devices before and after implementation. This analysis identified the impact of the model on network capacity [12].
- Proportions Test: Used to compare the frequency of unauthorized access incidents before and after implementation, measuring the effectiveness of implemented security policies [10].
- Chi-square Test (χ^2): Applied to evaluate the association between model implementation and the reduction of IP duplication errors, verifying the effectiveness of new security policies [19].
- Correlation Analysis: Used to assess the relationship between implemented security configurations and the reduction of network incidents. This analysis measured the effectiveness of security and network performance improvements [22].

4. RESULTS

The assessment of the Unified Communications Model's impact was conducted through a comprehensive pre- and post-implementation performance evaluation, leveraging advanced network analysis tools such as Wireshark and SolarWinds. The evaluation focused on critical performance metrics, including latency, packet loss, bandwidth utilization, and connection stability. Prior to optimization, response times for mission-critical applications consistently exceeded 150 milliseconds, while frequent IP address duplication errors emerged due to suboptimal allocation mechanisms within the network. Following the implementation of the model, response times were reduced to 80 milliseconds, IP-related errors decreased by 80%, and overall network connectivity demonstrated substantial improvements.

A significant security vulnerability identified within the university's network infrastructure pertained to

the high incidence of unauthorized access attempts, characterized by illicit connection attempts from devices or users lacking valid authentication credentials. These security breaches posed a substantial risk to data confidentiality, system integrity, and network stability. Before the deployment of the optimized model, an average of 25 unauthorized access attempts per month was recorded. Post-implementation, this figure was reduced to five incidents per month, primarily due to the strategic enforcement of network segmentation via Virtual Local Area Networks (VLANs), the integration of authentication protocols at access points, and the fortification of network security policies.

The justification for the incorporation of artificial neural networks (ANNs) into network optimization strategies stemmed from the imperative need for dynamic and efficient bandwidth allocation mechanisms. The university's network infrastructure exhibited congestion during peak utilization hours, exacerbated by static traffic distribution models that led to inefficient resource allocation. To address this bottleneck, a Multi-Layer Perceptron (MLP) Artificial Neural Network was deployed, optimized using the Adam backpropagation algorithm, and trained on six months of historical network traffic data.

The ANN-driven model facilitated predictive bandwidth consumption analytics and enabled real-time traffic distribution adjustments, thereby enhancing overall network efficiency and reducing congestion by 40%. Furthermore, packet loss rates decreased by 35%, contributing to improved connection stability and enhanced service quality for critical applications. This adaptive approach enabled the network infrastructure to dynamically respond to fluctuating user demands, ensuring the efficient and scalable utilization of technological resources.

The modernization of legacy networking hardware emerged as a necessary intervention due to performance limitations, processing inefficiencies, and the incompatibility of outdated devices with contemporary security protocols. The investigation revealed that unmanaged switches, low-capacity routers, and access points lacking WPA2/WPA3 encryption mechanisms significantly contributed to network congestion and security vulnerabilities. To address these deficiencies, obsolete devices were systematically replaced with managed switches supporting VLAN and Quality of Service (QoS) configurations, high-performance routers, and access points equipped with secure authentication protocols. The selection criteria for these upgrades were predicated on their capacity to enhance traffic segmentation, mitigate latency, and fortify security through advanced encryption methodologies. These strategic infrastructural enhancements resulted in immediate reductions in response times and significantly improved connection stability for end-users.

The results obtained after implementing the unified communications model at the National University of Huancavelica (UNH) are presented objectively, using tables and graphs to illustrate improvements in key performance and network security variables.

4.1. IMPROVEMENT IN APPLICATION RESPONSE TIMES

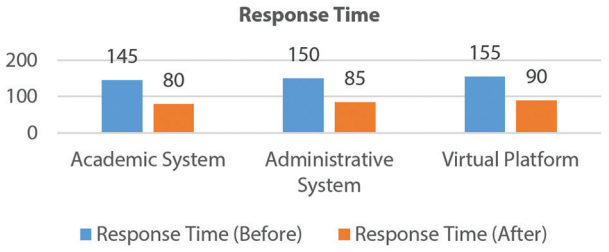
One of the most significant improvements observed was the substantial reduction in the response times of applications used at UNH. A comparison of times before and after model implementation shows a significant decrease, indicating optimized network performance.

Table 1. Comparison of Response Times

Application	Response Time (Before)	Response Time (After)
Academic System	145 ms	80 ms
Administrative System	150 ms	85 ms
Virtual Platform	155 ms	90 ms

(Source: Authors' elaboration)

The following graph illustrates the reduction in application response times after implementing the unified communications model.



4.2. REDUCTION IN UNAUTHORIZED ACCESS ATTEMPTS:

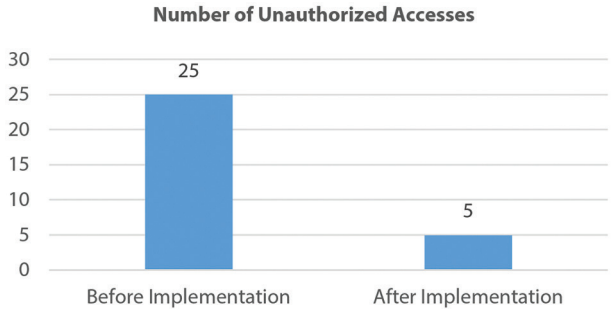
Another key improvement was the decrease in the number of unauthorized access attempts to the network. Before implementation, 25 unauthorized access attempts were recorded per month. After implementation, this figure was reduced to five unauthorized access attempts per month.

Table 2. Comparison of Unauthorized Accesses

Period	Number of Unauthorized Accesses
Before Implementation	25
After Implementation	5

(Source: Authors' elaboration)

The following graph illustrates the significant reduction in unauthorized access attempts after implementing the network security model.



4.3. RESOURCE OPTIMIZATION AND COST REDUCTION

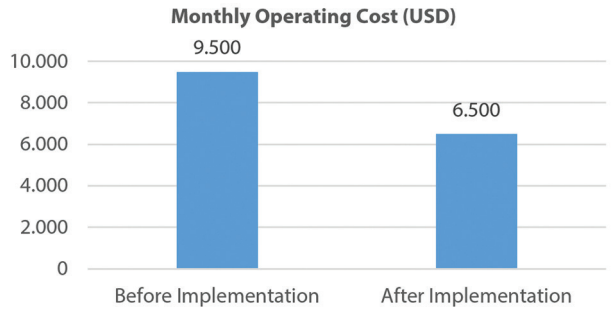
The implementation of the model optimized technological resources and reduced operational costs by 30%. The table below summarizes the cost savings achieved.

Table 3. Operating Cost Reduction

Indicator	Before Implementation	After Implementation	Reduction (%)
Monthly Operating Cost (USD)	9,500	6,500	30%

(Source: Authors' elaboration)

The following graph illustrates the reduction in operating costs after implementing the new system.



4.4. IMPROVEMENT IN NETWORK CAPACITY

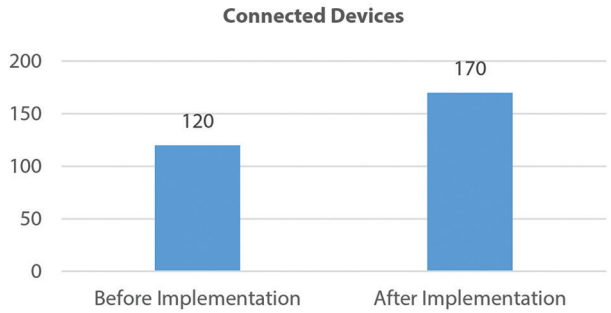
Network capacity, measured by the number of devices that could be connected simultaneously, also improved. Before implementation, the network supported 120 devices concurrently without performance degradation, while after implementation, it supported 170 devices.

Table 4. Enhanced Connected Device Capacity

Metrics	Before Implementation	After Implementation	Increase (%)
Connected Devices	120	170	42%

(Source: Authors' elaboration)

The following graph illustrates network capacity in terms of connected devices before and after implementation.



In addition to descriptive results, inferential statistical tests were performed to validate the research findings. These tests included significance analysis and correlation assessments to confirm the hypotheses.

1. Paired Samples t-test (t-test): Improvement in Application Response Times

A paired t-test was applied to compare application response times before and after implementing the unified communications model. The results indicated a significant reduction in response times.

- Null hypothesis (H_0): There is no significant difference in response times before and after model implementation.
- Alternative hypothesis (H_1): There is a significant difference in response times before and after model implementation.

Table 5. Response times

Application	Media (Before)	Media (After)	t value	p-value (significance)
Academic System	145 ms	80 ms	5,87	$p < 0.01$
Administrative System	150 ms	85 ms	6,2	$p < 0.01$
Virtual Platform	155 ms	90 ms	6,45	$p < 0.01$

(Source: Authors' elaboration)

The p-value ($p < 0.01$) in all tests indicates that the reduction in response times is statistically significant, rejecting the null hypothesis and confirming the effectiveness of the model's implementation.

2. Analysis of Variance (ANOVA): Network Capacity Improvement

An analysis of variance (ANOVA) was conducted to evaluate differences in the number of supported devices before and after implementation. The ANOVA confirmed a significant difference in network capacity.

- Null hypothesis (H_0): There is no significant difference in network capacity before and after implementation.

- Alternative hypothesis (H_1): There is a significant difference in network capacity before and after implementation.

Table 6. Network capacity

Source of variation	Sum of squares	Degrees of freedom (df)	Root mean square	F	p-value (significance)
Between groups	10.240	1	10.240	12, 32	***p* < 0.01**
Within groups	1.020	38	26,84		
Total	11.260	39			

(Source: Authors' elaboration)

The F-value = 12.32 and $p < 0.01$ indicate a statistically significant difference in network capacity after implementation. Since the p-value is less than 0.01, the null hypothesis is rejected, and the alternative hypothesis is validated, confirming a significant improvement in network capacity after implementing the model.

3. Chi-Square Test: Reduction in Unauthorized Access Attempts

A chi-square test was applied to evaluate the reduction in unauthorized access attempts after implementing the unified communications model.

- Null hypothesis (H_0): There is no significant difference in unauthorized access attempts before and after implementation.
- Alternative hypothesis (H_1): There is a significant difference in unauthorized access attempts before and after implementation.

Table 7. Network capacity

Period	Observed Frequency	Expected Frequency	χ^2	p-value (significance)
Before Implementation	25	15	8,33	p < 0.05
After Implementation	5	15		

(Source: Authors' elaboration)

The $\chi^2 = 8.33$ with $p < 0.05$ indicates a statistically significant reduction in unauthorized access attempts after implementation. Therefore, the null hypothesis is rejected, confirming that the number of unauthorized access attempts significantly decreased following model implementation.

4. Correlation Analysis: Reduction in IP Duplication Errors

A correlation analysis was conducted to measure the relationship between implemented security configurations and the reduction in IP duplication errors.

- Pearson Correlation Coefficient (r): -0.87
- Null hypothesis (H_0): There is no significant correlation between security configurations and the reduction in IP duplication errors.
- Alternative hypothesis (H_1): There is a significant correlation between security configurations and the reduction in IP duplication errors.

The correlation coefficient $r = -0.87$ indicates a strong inverse correlation between implemented security configurations and the reduction in IP duplication errors. Since the coefficient is significantly different from zero, the null hypothesis is rejected, and the alternative hypothesis is validated, suggesting that security improvements effectively reduced network errors.

Statistical tests such as the paired t-test were used to compare network response times before and after implementing the unified communications model, determining whether the observed reduction was statistically significant. The analysis of variance (ANOVA) assessed differences in the number of devices supported by the network after optimization. The proportion test analyzed the decrease in unauthorized access attempts by comparing frequencies before and after intervention. The chi-square test (χ^2) verified the association between model implementation and the reduction in IP duplication errors. Finally, the correlation analysis measured the relationship between security improvements and the decrease in network incidents, demonstrating the effectiveness of the new protection scheme.

Interpretation of Results:

The findings of this study demonstrate that the implementation of the unified communications model at the National University of Huancavelica (UNH) had a positive impact on several key aspects of information management. A significant reduction in the response times of critical applications was observed (t-test, $p < 0.01$), along with a decrease in unauthorized access attempts (χ^2, $p < 0.05$) and an improvement in network capacity (ANOVA, $p < 0.01$). Additionally, the optimization of technological resources resulted in a 30% reduction in operational costs, implying greater economic efficiency for the institution. This was achieved through improved utilization of technological resources, eliminating redundancies in the network infrastructure, and enhancing energy efficiency with lower-power consumption devices. Furthermore, network segmentation and intelligent traffic monitoring reduced maintenance and technical support costs, ensuring a more efficient infrastructure management without compromising service quality.

The strong inverse correlation between security configurations and the reduction in IP duplication errors ($r = -0.87$) indicates that security policy enhancements were essential for stabilizing and protecting the network, reducing operational risks associated with unauthorized access and data integrity loss.

Relation to Theoretical Framework and Study Objectives:

These results align with the theoretical foundations proposed by [14] and [15], who argue that integrating a unified communications model not only improves network availability and reliability but also optimizes operational management through a robust and secure network infrastructure. This study validates these premises, confirming that the proposed model can reduce vulnerabilities and enhance operational efficiency in an educational institution like UNH.

Regarding the study's objectives, the results confirm that the unified communications model significantly improves information management in terms of availability, integrity, and confidentiality. This supports the research's specific objectives, which sought to verify whether the model would optimize the network and reduce security risks.

5. CONCLUSION

The implementation of the unified communications model at the National University of Huancavelica represents an innovative contribution in the field of information management for educational institutions with deficient technological infrastructures. The novelty of the model lies in the integration of advanced security techniques, network segmentation, and dynamic traffic optimization using artificial neural networks. The results obtained demonstrate significant improvements in operational efficiency, network security, and cost reduction, statistically validating the model's effectiveness. Additionally, this comprehensive approach enables adequate scalability for future expansions of technological infrastructure, establishing itself as a relevant contribution to the efficient management of information in educational environments.

6. REFERENCES:

[1] A. Ahrens, J. Zascerinska, A. Bikova, L. Aleksejeva, M. Zascerinskis, O. Gukovica, "A New Development Model Of Sustainable", Education. Innovation. Diversity., Vol. 1, No. 6, 2023, pp. 38-47.

[2] N. P. Yerram, "The Technical Evolution of Unified Communications in Education: Infrastructure, Implementation, and Impact", International Journal For Multidisciplinary Research, Vol. 6, No. 6, 2024.

[3] M. Rihan et al. "Unified 3D Networks: Architecture, Challenges, Recent Results, and Future Opportunities", IEEE open journal of vehicular technology, Vol. 6, 2024, pp. 170-201.

[4] S. S. Veligodskiy, N. Miloslavskaya, "Unified model of maturity of network security centers of information and telecommunication networks", Izvestiâ ÛFU, No. 3, 2023, pp. 157-172.

[5] H. A. Gabbar, "Modeling of Interconnected Infrastructures with Unified Interface Design toward Smart Cities", Energies, Vol. 14, No. 15, 2021, p. 4572.

[6] Y. Wang, Y. Chen, Q. Sun, "Hand gesture recognition in complex background | Additional background manual identification", JOIG-Journal of Image and Graphics, Vol. 26, No. 4, 2021, pp. 815-827.

[7] P. L. Simeonov, P. Hofmann, "A Distributed Intelligent Computer/Telephony Network Integration Architecture for Unified Media Communication", Intelligent Networks and Intelligence in Networks, Springer, 1997, pp. 3-8.

[8] C. H. Díaz Novelo, J. Olmos de la Cruz, "Importance of Physical Security in the Network Infrastructure, Data Centers and Telecommunications of Higher Education Institutions", The International Journal of Technology, Innovation, and Education, No. 3, 2021, pp. 1-12.

[9] V. V. Cabañas Victoria, J. Vázquez Castillo, M. Blanqueto Estrada, L. Y. Dávalos Castilla, "Virtual networking laboratory as a strategic technological infrastructure for carrying out computer network and computer security practices", Tecnología Educativa Revista CONAIC, Vol. 6, No. 3, 2020, pp. 21-26.

[10] N. D. J. Rodríguez Preciado, "Design Of A Wireless Network To Optimize The Connection Security Of A Corporate Network Through A Server", University of Guayaquil, 2020.

[11] C. V. Mercado, H. H. Palma, F. A. Rangel, "Information management as a quality-building element in higher education institutions", Contemporary Engineering Sciences, Vol. 11, No. 87, 2018, pp. 4311-4319.

[12] V. Guaranda Sornoza, B. M. Ayón Baque, "Analysis Of The Benefits Of Mesh Technology In The Wireless Networks Of The Unesum University Complex", Universidad Statal Del Sur De Manabí, 2020.

[13] W. Stallings, "Network Security Essentials: Applications and Standards", Pearson, 2016.

[14] A. S. Tanenbaum, D. Wetherall, "Computer Networks", Pearson, 2011.

[15] H. Icuduygu, H. Gorgun, "Patent Application Publication", 2014.

[16] Z. Nan et al. "Unified information network management platform", 2014.

[17] O. Masaharu, "Unified type information infrastructure system for unifying on-site data and management data", 2010.

[18] T. Okathe, S. S. Heydari, V. Sood, K. El-Khatib, "Unified multi-critical infrastructure communication architecture", Proceedings of the 27th Biennial Symposium on Communications, Kingston, ON, Canada, 1-4 June 2014, pp. 178-183.

[19] F. Gavilanes-Sagnay, A. Shuguli-Velasco, B. Landeta-Ailla, E. Loza-Aguirre, "Design and implementation of a virtual learning environment in Unity for structured cabling", SOCYUN - Universidad y Sociedad, 2023.

[20] O. Ivanov, L. Koretska, V. Lytvynenko, "Intelligent modeling of unified communications systems using artificial neural networks", The CEUR Workshop Proceedings, Vol. 2623, 2020, pp. 77-84.

[21] J. D. McCabe, "Network Analysis, Architecture, and Design", Morgan Kaufmann, 2018.

[22] W. Stallings, "Data and Computer Communications", Prentice Hall, 2007.

Federated Learning Algorithm to Suppress Occurrence of Low-Accuracy Devices

Original Scientific Paper

Koudai Sakaida
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
sakaida.koudai@ohsuga.lab.uec.ac.jp

Akihiko Ohsuga
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
ohsuga@uec.ac.jp

Keiichiro Oishi
Department of Computer Science,
Faculty of Environmental and Life Science,
Okayama University,
Okayama, Japan
oishi@okayama-u.ac.jp

Andrew J
Department of Computer Science and Engineering,
Manipal Institute of Technology,
Manipal Academy of Higher Education,
Manipal, India
andrew.j@manipal.edu

Yasuyuki Tahara*
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
tahara@uec.ac.jp

Yuichi Sei*
Department of Informatics,
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
Tokyo, Japan
seiuny@uec.ac.jp

*Corresponding author

Abstract *– In recent years, federated learning (FL), a decentralized machine learning approach, has garnered significant attention. FL enables multiple devices to collaboratively train a model without sharing their data. However, when the data across devices are non-independent and identically distributed (non-IID), performance degradation issues such as reduced accuracy, slower convergence speed, and decreased performance fairness are known to occur. Under non-IID data environments, the trained model tends to exhibit varying accuracies across different devices, often overfitting on some devices while achieving lower accuracy on others. To address these challenges, this study proposes a novel approach that integrates reinforcement learning into FL under Non-IID conditions. By employing a reinforcement learning agent to select the optimal devices in each round, the proposed method effectively suppresses the emergence of low-accuracy devices compared to existing methods. Specifically, the proposed method improved the average accuracy of the bottom 10% devices by up to 4%, without compromising the overall average accuracy. Furthermore, the device selection patterns revealed that devices with more diverse local data tend to be chosen more frequently.*

Keywords*: FL, Non-IID, Performance Fairness, Device Selection, RL, DDQN*

Received: December 20, 2024; Received in revised form: May 11, 2025; Accepted: May 12, 2025

1. INTRODUCTION

In recent years, with advancements in the Internet of Things (IoT) and artificial intelligence, machine learning technologies have been utilized in various aspects of daily life, bringing significant convenience to people. Concurrently, the explosive increase in data volume has led to privacy breaches, heightening concerns regarding privacy and security. Traditional machine learning methods require the aggregation of data in a single location. For example, many smartphones contain private data that must be integrated for training. However, aggregating data in one place not only results in high communication costs and significant battery consumption on devices but also increases the risk of compromising user data privacy and security.

Federated learning (FL), introduced by Google in 2016 [1], has garnered attention as a decentralized machine learning approach that addresses these issues. FL has

demonstrated its efficacy in enabling global-scale collaborative training, as evidenced by its successful application in rare cancer boundary detection. This initiative aggregated insights from 71 hospitals spanning six continents while rigorously preserving patient data privacy [2]. However, it is crucial to recognize that despite its inherent privacy-preserving advantages, FL is not impervious to privacy leakage stemming from shared model updates. Recent scholarly work, such as the differentially private knowledge transfer paradigm proposed by Qi et al. [3], underscores the necessity of integrating supplementary privacy-enhancing mechanisms to bolster FL's resilience against inference attacks. Furthermore, Boscarino et al. [4] highlighted FL's pivotal role in supporting indigenous data sovereignty, illustrating its potential to empower communities in maintaining control over sensitive genomic information.

Communication efficiency constitutes another significant impediment to the widespread adoption of FL. Wu et al. [5] introduced FedKD, an adaptive knowledge distillation strategy coupled with gradient compression techniques, which substantially curtails communication overhead, thereby tackling a critical scalability bottleneck. Similarly, the comprehensive survey by Asad et al. [6] meticulously examined existing methodologies and prospective avenues for alleviating FL's communication costs, reinforcing the urgency and multifaceted nature of this challenge in practical deployments.

A further salient obstacle in federated learning arises from the Non-Independent and Identically Distributed (Non-IID) nature of local datasets across participating devices. This inherent data heterogeneity not only diminishes model accuracy but also adversely affects the active engagement of users, thereby complicating model convergence and the reliable evaluation of performance [7, 8]. Personalized federated learning frameworks, such as the one proposed by Lin et al. [9], have been developed to address these non-IID issues by tailoring local models with a focus on communication efficiency, robustness, and fairness concurrently, representing a notable trajectory in contemporary FL research.

The issue of fairness in federated learning has emerged as a particularly pressing concern, primarily due to the intrinsic heterogeneity among participating clients. Chaudhury et al. [10] emphasized the importance of explicitly addressing fairness, proposing solutions grounded in cooperative game theory to ensure equitable model performance across diverse client populations. Moreover, recent innovations like FedFed, introduced by Yang et al. [11], prioritize the mitigation of non-IID effects through selective feature distillation, carefully balancing the inherent trade-offs between model accuracy and privacy preservation.

These recent advancements collectively underscore the imperative for federated learning to continue its evolution by comprehensively addressing the intertwined challenges of privacy, communication efficiency, fairness, and data heterogeneity. Such holistic approaches are essential to ensure the deployment of robust, scalable, and equitable FL systems in diverse real-world settings, aligning closely with the practical motivations and ongoing challenges elaborated upon within this study.

In FL, the process of sharing and updating models is repeatedly performed while maintaining the data on each device, thereby enabling training while protecting privacy. FL randomly selects a subset of devices to participate in each update, rather than having all devices participate each time, which improves scalability and reduces communication costs.

However, FL has several limitations. The first is that data across devices may be non-independent and identically distributed (non-IID). This implies that the data distribution varies across devices, which differ in the labels they hold or the amount of data they possess. Therefore, the nature of non-IID data complicates FL training and evaluation.

Another challenge in FL is fairness, as discussed in Section 3, "Heterogeneity and Performance Fairness." Fairness issues arise from various perspectives, including the fairness of machine learning algorithms, as described by Pessach et al. [12] and in FL device selection, as raised by Vucinich et al. [13]. This study focuses on fairness in performance, particularly in devices with lower accuracy. Specifically, under non-IID data conditions, the differing data distributions on each device tend to cause high variance in model test accuracies across devices. In such situations, performance fairness in FL is likely to be compromised, leading to an increase in low-accuracy devices.

This paper proposes a novel approach that applies reinforcement learning to address the issue of low-accuracy devices in FL. Conventional methods typically employ random device selection and enhance aggregation to improve performance. However, these methods tend to prioritize reducing the variance in accuracy over improving average accuracy, without sufficiently considering the performance enhancement of low-accuracy devices. This study aims to suppress low-accuracy devices more effectively than other methods while maintaining the performance of high-accuracy devices. The proposed method utilizes a reinforcement learning agent to learn how to improve the accuracy of lower-performing devices in each round, with the aim of enhancing the performance of low-accuracy devices without compromising average accuracy compared to existing methods. The contributions of this study are as follows:

- A novel algorithm that applies reinforcement learning is designed to address the issue of low-accuracy device occurrence in FL. This algorithm enables effective device selection during the FL training process, thereby suppressing the emergence of low-accuracy devices.
- Compared to existing methods, the proposed method significantly improves the average accuracy of the bottom 10% of devices in a non-IID data

environment without reducing the overall average accuracy. This result demonstrates that the proposed method contributes to model fairness and performance enhancement, even under non-IID data conditions.

- We confirmed that the proposed method can effectively suppress the impact of low-accuracy devices on complex datasets with more than 10 classes. This validates the effectiveness of the proposed method across a wide range of datasets.

The structure of this paper is as follows: Section 2 presents background information on FL. Section 3 reviews related research. Section 4 describes the details of the proposed FL algorithm that utilizes reinforcement learning. Section 5 presents the evaluation results of the proposed method using real-world datasets. Finally, Section 6 concludes the paper.

2. BACKGROUND

FL is a method for training models through iterative communication between a central server and multiple devices. Each round consists of the following steps, which form a continuous flow referred to as a "round." The learning process progresses by repeating these rounds.

i. Initialization: Before starting the first round, the central server initializes the weights of the global model.

ii. Device Selection: At the beginning of each round, the central server randomly selects the devices according to a specified ratio. Subsequently, the current global model weights are sent to the selected devices.

iii. Update: In this phase, each device trains the global model based on its local dataset and sends the updated local model weights back to the central server.

iv. Aggregation: The central server aggregates the received updated local model weights to update the global model.

v. Termination: This process is terminated when the global model converges and reaches a specific threshold. If convergence is not achieved, the process returns to device selection and proceeds with local updates and weight aggregation.

FedAVG [14], a fundamental FL framework, conducts learning as described in Equation (1):

$$w_k^{t+1} = w_k^t - \eta \nabla \mathcal{L}_k(w_k^t; D_k) \quad (1)$$

where w_k^t represents the weights of the local model of device k at round t, with w_k^{t+1} denoting the updated weights of the local model at round t+1; $\nabla L_k(w_k^t; D_k)$ indicates the gradient of the loss function with respect to the local dataset D_k. In this manner, each device updates the model weights w_k^t using its local dataset D_k and learning rate η.

Next, the central server updates the global model by aggregating the weights w_k^{t+1} collected from each device according to Equation (2).

$$w^{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1} \quad (2)$$

where w^t is the weight of the global model at round t; K is the number of devices selected in round t; n_k is the number of data samples on device k; and n is the total number of data samples across all devices.

3. RELATED WORK

3.1. DEVICE SELECTION TECHNIQUES FOR FL

Recent investigations have explored diverse methodologies for optimizing device selection within federated learning frameworks, primarily focusing on enhancing overall model performance and training efficiency. For instance, Tian et al. [15] introduced FedRank, a client selection method predicated on ranking that leverages imitation learning to mitigate cold-start issues frequently encountered with reinforcement learning-based techniques. By employing a pairwise ranking strategy, FedRank effectively selects clients based on system and data heterogeneity, demonstrating significant improvements in convergence speed and energy efficiency. Furthermore,

Pan et al. [16] developed a contextual client selection framework utilizing a Neural Contextual Combinatorial Bandit (NCCB) algorithm. This framework extracts client features through locality-sensitive hashing and exploits correlations among datasets, resulting in reduced training duration and enhanced model accuracy, approaching performance levels observed in IID scenarios.

In a related vein, Zhang et al. [17] proposed an approach integrating spectrum allocation optimization with device selection for federated learning in wireless networks. Their method aims to minimize training delay and energy consumption by selecting devices according to the divergence between local and global model weights, thereby facilitating faster convergence under non-IID conditions. While these methodologies offer considerable advancements in device selection strategies and overall system efficiency, it is crucial to acknowledge that none of these explicitly address fairness among devices, such as ensuring balanced accuracy or equitable participation across heterogeneous data distributions.

3.2. FEDERATED REINFORCEMENT LEARNING (FRL)

FRL is an approach that combines FL with reinforcement learning (RL). FL focuses on collaborative training of models across multiple devices while preserving privacy, whereas FRL introduces reinforcement learning techniques to enable optimal device selection and parameter tuning. In FRL, the elements of RL (en-

vironment, state, and action) are applied within the FL framework to potentially address complex issues [18]. Thus, FRL holds promise for overcoming the limitations of FL and is expected to have applications in various fields. Research on the use of reinforcement learning for device and client selection in FL has been active [19-22], with selections directly impacting the quality and utility of the model, which makes this a highly important area.

Wang et al. [19] proposed FAVOR, which utilizes the double deep-Q-network (DDQN) algorithm for client selection. This method allows device and client selection, which enhances convergence speed of the model under non-IID conditions, thereby saving on computational resources. However, because DDQN model training is limited to a single client, the agent may not rapidly converge.

Additionally, Bouaziz et al. [22] proposed FL to address system and static heterogeneity using reinforcement learning (FLASH-RL), which employs the DDQN model to perform client selection, aimed at reducing computational and communication costs. By enabling multi-action selection and learning, their approach accelerates the learning process. Furthermore, FLASH-RL contributes to latency reduction by individually evaluating each client using a proprietary evaluation function.

Yu et al. [23] introduced DDPG-AdaptConfig, a deep reinforcement learning framework based on Deep Deterministic Policy Gradient (DDPG), which adaptively selects devices and configures local training hyperparameters such as batch size and epoch count. This method incorporates a transformer-based actor network to capture heterogeneous information from model parameters and applies clustering-based aggregation to further accommodate system and data diversity.

3.3. HETEROGENEITY AND PERFORMANCE FAIRNESS

Shi et al. [24] argue that many current FL frameworks are designed with a central server-centric perspective, prioritizing metrics such as convergence speed and overall model accuracy, often at the expense of individual client needs. This imbalance can disincentivize participation from less capable clients and potentially compromise the global model's representativeness. Their work proposes a taxonomy of fairness-aware FL methodologies, identifying critical stages where fairness considerations are paramount, including client selection, optimization processes, and incentive mechanisms.

Furthermore, Rafi et al. [25] emphasize that fairness issues in FL extend beyond client selection to encompass reward allocation strategies. They contend that the uniform distribution of global models to all clients, irrespective of their individual contributions to the training process, can be perceived as unfair, particularly by clients who have invested more resources or data. The authors also highlight the potential for demographic biases, such as those related to gender or ethnicity, to compound these fairness challenges within FL systems.

Chen et al. [26] investigate the inherent trade-off between privacy preservation and fairness in FL. Their analysis suggests that privacy-enhancing techniques, such as the introduction of noise or limitations on data sharing, can disproportionately impact disadvantaged groups by causing a greater degradation in their model performance compared to others. Conversely, efforts to enhance fairness might necessitate increased data transparency, potentially leading to heightened privacy risks.

Huang et al. [27] categorize fairness in FL into two primary dimensions: collaboration fairness and performance fairness. Collaboration fairness addresses the equitable distribution of rewards and the provision of adequate incentives for client participation. Performance fairness, on the other hand, focuses on ensuring consistent model accuracy across all clients. The authors assert that the simultaneous achievement of both collaboration and performance fairness is crucial for the development of sustainable and robust FL systems, particularly in real-world applications characterized by significant client heterogeneity.

These perspectives collectively demonstrate that fairness in FL is a multifaceted issue that intersects with client heterogeneity, privacy concerns, and system sustainability. Addressing fairness effectively requires comprehensive strategies that go beyond mere accuracy optimization. In real-world scenarios, the data on devices are often non-IID, which accelerates imbalanced learning across devices. Data heterogeneity poses a significant challenge in FL, leading to variations in learning outcomes and substantial differences in model accuracy among devices.

The conventional FedAVG method [14] is known to exhibit unstable performance under non-IID conditions, with some devices demonstrating significantly higher or lower accuracy than others. In such situations, not only overall model accuracy enhancement but also performance fairness across devices should be considered. Performance fairness ensures that the model performs uniformly across all participating devices, preventing scenarios in which low-performance devices are disproportionately affected, thereby improving overall fairness.

Huang et al. [28] successfully increased the convergence speed of the model while maintaining performance fairness by employing a dual-momentum descent method and weighted aggregation that accounts for client accuracy and participation frequency. Wentao et al. [29] introduced federated fairness and effectiveness (FedFE), which integrates momentum gradient descent into the FL process and performs accuracy-based weighted aggregation, thereby achieving improvements in both fairness and convergence speed. Despite these advancements, a sufficient num-

ber of studies have not been conducted on complex datasets with more than 10 classes, leading to a lack of validation regarding their adaptability to multiclass environments.

These studies presented effective approaches for addressing imbalances caused by non-IID data while enhancing performance fairness. This study focuses on performance fairness, specifically aiming to construct models for non-IID data environments.

Performance fairness refers to the uniformity of model performance across all devices participating in FL. In this paper, we define performance fairness as "achieving as equal accuracy as possible for all devices within FL," with the objective of enhancing this fairness while suppressing the emergence of low-accuracy devices.

Li et al. [30] proposed q-fair federated learning (q-FFL), which improves performance fairness by placing greater emphasis on devices with larger losses. Specifically, q-FFL mitigates performance disparities by weighting devices' losses using a parameter q (where $q \geq 0$), which controls the emphasis on high-loss clients. A larger q leads to a stronger focus on fairness across clients. However, the appropriate value of q must be determined empirically, as it depends on the dataset characteristics and involves a trade-off between fairness and overall accuracy.

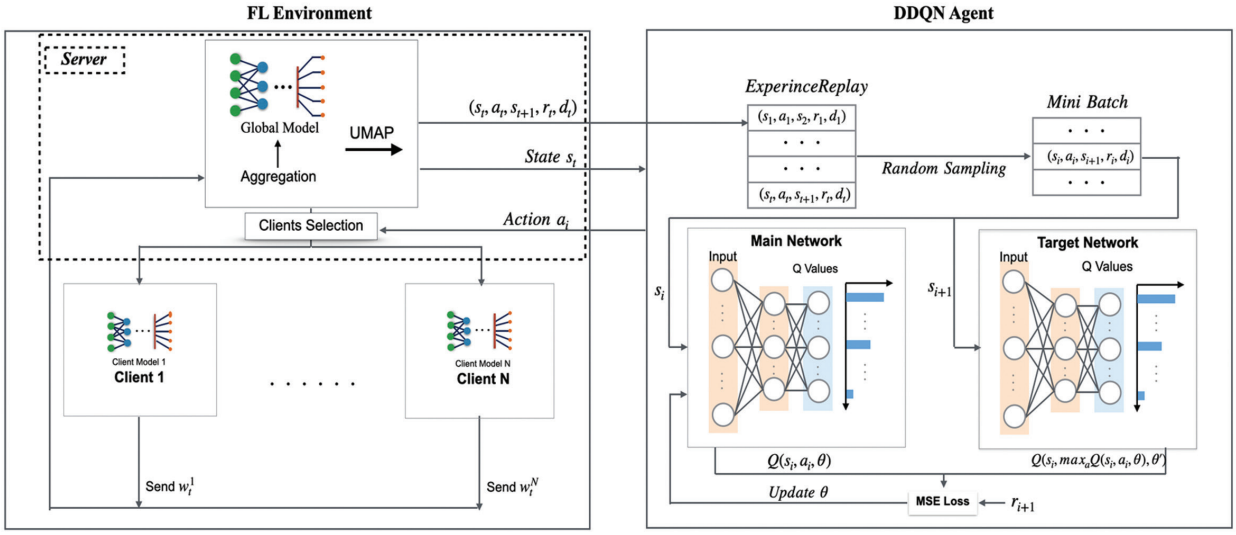


Fig. 1. Overview of the proposed method

Although the aforementioned methods represent noteworthy progress in addressing fairness and performance challenges in federated learning, they are predominantly constrained by their reliance on pre-defined parameters and their limited adaptability in highly heterogeneous and multiclass scenarios. In contrast, our approach introduces a dynamic device selection mechanism guided by reinforcement learning, which specifically prioritizes devices with lower predictive accuracy and incrementally enhances their performance over the course of the federated training process. A comprehensive comparative analysis, presented in the experimental evaluation section, benchmarks our method against the techniques described in [14, 29–30]. Although we did not directly compare our method with [28], we confirmed that it outperforms [29]. As [29] has been shown to achieve better performance than [28], we considered a comparison with [29] sufficient. The results consistently indicate that our method achieves superior performance fairness, particularly in environments characterized by pronounced non-IID conditions and complex multiclass data distributions.

FedHEAL [31] is a recently developed FL algorithm designed to address fairness issues in environments characterized by domain bias. It leverages the consistency of parameter updates to mitigate the impact of noisy or low-quality updates by masking the updates of unimportant parameters. Additionally, FedHEAL promotes fair model aggregation by utilizing the Euclidean distance, thereby preventing convergence bias often observed in conventional FL approaches. As a generic method, FedHEAL can be integrated with various existing FL algorithms.

3.4. ENHANCING PRIVACY FOR FL

Recent research has increasingly emphasized the need to fortify privacy safeguards within FL. While the distributed architecture of FL, which retains raw data on local devices, provides a baseline of privacy, it remains vulnerable to sophisticated inference and poisoning attacks.

Bietti et al. [32] introduced a personalized federated learning framework grounded in differential privacy [33]. Their study illustrates how personalized models can refine the trade-off between privacy and accuracy. However, they also acknowledge that tightening privacy guarantees inevitably results in diminished model performance.

Addressing the challenge of intermittent client participation, Jiang et al. [34] proposed Dordis, a distributed differential privacy framework resilient to client dropout. This approach achieves robust privacy protection without relying on a trusted central server, although the noise required for differential privacy introduces an unavoidable computational overhead.

Naseri et al. [35] explored the complementary use of local differential privacy (LDP) [36-37] and central differential privacy to mitigate both backdoor and membership inference attacks in FL. Their findings confirm that while these privacy techniques can enhance system resilience, they do so at the cost of reduced utility in the trained models.

In a related vein, Qi et al. [38] examined the susceptibility of differentially private FL (DPFL) to poisoning attacks. To counter this, they developed Robust-DPFL, which augments resilience to poisoned gradients. While their method successfully improves robustness, it introduces added complexity into the FL pipeline.

Collectively, these studies underscore that although FL inherently offers a foundational level of privacy, augmenting it with advanced privacy-preserving techniques frequently entails a trade-off with model accuracy and system complexity. The method proposed in this study is compatible with such techniques and can be integrated where stronger privacy assurances are necessary. Nonetheless, the empirical validation of this integration remains an open avenue for future investigation.

4. METHODOLOGY

In this section, we describe the proposed method for improving the performance of the bottom B% of devices in FL by integrating reinforcement learning. Here, B is a tunable parameter that specifies the proportion of devices with the lowest individual accuracies, which we particularly aim to support. This metric serves as an indicator of fairness, emphasizing performance improvement for underperforming clients. As illustrated in Fig. 1, the proposed method incorporates device selection using DDQN within an FL framework. Unlike existing methods, our approach adopts reinforcement learning to enhance device selection. Specifically, we employed uniform manifold approximation and projection (UMAP) for dimensionality reduction, transforming high-dimensional model weights into lower-dimensional representations while retaining essential information. In addition, we designed a reward mechanism based on the distance from the global model to discourage the selection of low-accuracy devices. This strategy enables efficient model construction, even in environments with significant disparities in data distribution across devices.

The workflow of the proposed method is presented in Algorithm 1. In each round, the reinforcement learning agent selects the optimal devices and transmits the global model to these devices. The selected devices then perform training on their local datasets. Finally, the central server aggregates the models sent by the selected devices to update the global model. The agent updates its parameters based on the received rewards, which are designed to minimize the selection of low-accuracy devices.

Algorithm 1: FL with DDQN for Device Selection

Initialize:

for each device k **do**

Device k trains local model w_0^k for 1 epoch with local dataset.

Send updated weights w_0^k to the server.

end for

Server performs dimensionality reduction on $\{w_0^k\}$ using UMAP to obtain initial state s_0.

Initialize DDQN agent with initial state s_0.

for each communication round $t = 1$ to T **do**

DDQN agent selects K devices based on the Q-values.

for each selected device k **do**

Send w_t to the device k.

Device k trains local model w_{tk} for E epochs with local dataset.

Send updated weights $w^k_{\{t+1\}}$ to the server.

Aggregate global model: $w_{\{t+1\}} = 1/C_t \sum_{\{k \in C_t\}} w^k_{\{t+1\}}$

Select all devices to calculate rewards.

for each device k **do**

Send $w_{\{t+1\}}$ to the device k.

Device k tests model $w_{\{t+1\}}$ on local test dataset and calculates accuracy acc_t^k.

Send accuracy acc_t^k back to the server.

end for

Aggregate global model: $w_{\{t+1\}} \leftarrow 1/|C_t| \sum_{k \in C_t}$

Select all devices to calculate rewards

for each device k **do**

Send $w_{\{t+1\}}$ to the device k.

Device k tests model $w_{\{t+1\}}$ on local test dataset and calculates accuracy acc_t^k.

Send accuracy acc_t^k back to the server.

end for

Calculate rewards r_t using accuracy acc_t^k equation (6).

DDQN Agent Do:

Update the DDQN parameters θ_t by minimizing the loss $L_t(\theta_t)$.

end for

Table 1 lists the symbols and descriptions used in this study.

Table 1. Notation

Symbol	Definition	Description
N	Total number of devices	The total number of devices
K	Number of selected devices	The number of devices selected in each round
C_t	Set of devices selected in round t	The set of K devices selected in round t
a_i	Action i	The action of selecting device i
A	Action space	The set of possible device selections
r_t^k	Reward	The reward for selecting device k in round t
γ	Discount factor	The importance of future rewards
θ	Parameters of the main network	The weights of the neural network being trained
θ'	Parameters of the target network	The fixed weights of the target network
w_t^k	Weight in round t	The weights of device k's model in round t

4.1. DDQN-BASED DEVICE SELECTION

To apply reinforcement learning to device selection, we formulated the Markov decision process.

- **State:**

The state at round t, namely s_t, is represented as a vector

$s_t = (w_t, wt^{(1)} \ldots, w_t^{(N)}$ where w_t represents the global model weights after round t, and $wt^{(1)}, \ldots, w_t^{(N)}$ represent the local model weights of all N devices. The agent is colocated with the FL server and holds a list of weights. A specific $w_t^{(k)}$ is updated only in round t if device k is selected for training and the resulting $\Delta t^{(k)}$ is received by the FL server. Consequently, the state space can become very large, making learning in such a space difficult. Therefore, we applied UMAP to compress the weights of each model into a 10-dimensional space, reducing the size of the state to 10×(N+1) dimensions.

- **Action:**

Actions (a) are represented as vectors of N Boolean values, where a value of 1 indicates a selection:

$$a_t = \{i\} \times N, \ where \ \ i \in \{0,1\} \tag{3}$$

As described in the subsequent section "Application of DDQN," in standard reinforcement learning, a subset of size K must be selected at each round t, resulting in $\binom{N}{K}$ possible combinations. However, in FL, this makes the action space enormous, causing computational costs to skyrocket, thus making the application infeasible. Therefore, we utilized the multi-action selection approach proposed by Bouaziz et al. [22], to allow multiple actions to be selected and learned simultaneously. This method treats each device selection as an independent action, significantly improving computational efficiency.

- **Reward:**

Rewards (r) are represented as a set of length $|C_t|$:

$$r_t = \{r_t^k \mid k \in C_t\} \tag{4}$$

where ζ_t^k measures the contribution of the local model of device k in round t relative to the global model:

$$\zeta_t^k = \frac{1}{\sqrt{|w_t - w_t^k|_2^2 + 1}} \tag{5}$$

$$r_t^k = M^{(TargetACC - BottomACC)} \cdot \zeta_t^k \tag{6}$$

A small value of Equation (5) indicates a large difference (Euclidean distance) between the weights of the device's local model and server's global model. In such cases, the device is considered unimportant, resulting in smaller ζ_t^k and reward r_t^k. This is because the local model weights of the selected device are generated through an aggregation process that combines the weighted sums. Devices with small differences between their local and global model weights are assumed to make significant contributions in a round, thereby substantially affecting the performance of the global model. M is a constant; $TargetACC$ represents the target average accuracy of the bottom B% of devices within a specified number of communication rounds, and $BottomACC$ denotes the average accuracy of the bottom B% of devices when all devices perform testing using the aggregated global model in each round t. This process provides an important metric for the agent to learn actions that improve the accuracy of the bottom devices. As the average accuracy of the bottom B% of devices increases, the agent is more likely to receive rewards, encouraging actions that enhance the fairness among devices. Calculating the test accuracy for all the devices introduces additional computational costs, with each device potentially being selected for up to twice the number of rounds. However, because the test accuracy calculation is not part of the learning process, the load on the devices is relatively small. This method allows the agent to effectively improve the average accuracy of the bottom B% of devices.

4.2. APPLICATION OF DDQN

In this study, following multiple existing studies, we employed the DDQN algorithm [39], which consisted of two neural networks: the main and target networks. The main network is used for training, whereas the target network evaluates the actions in the next state and is updated every P steps. The DDQN agent incorporates a replay memory mechanism to eliminate correlations between consecutive experiences, specifically between(s_i, a_i, s_{i+1}, r_i, d_i) and (s_{i+1}, a_{i+1}, s_{i+2}, r_{i+1}, d_{i+1}); d_i is Boolean and indicates whether the terminal state has been reached.

The RL learning problem can be formulated by minimizing the mean squared error (MSE) loss between the target value and the approximated value, expressed by the following equation:

$$L_t^k(\theta_t) = \left(Y_t^k - Q(s_t, a_k; \theta_t)\right)^2 \quad (7)$$

where $L_t^k(\theta_t)$ represents the loss function for action a_k; Y_t^k is the target value for action a_k, and $Q(s_t, a_k; \theta_t)$ is the approximated Q-value for action a_k in state s_t. Target value Y_t^k is defined as follows:

$$Y_t^k = r_t^k + \gamma\, Q\left(s_t, \arg\max_{a_i \in \mathcal{A}} Q(s_t, a_i; \theta_t); \theta_t'\right) \quad (8)$$

The action space A is defined as:

$$\mathcal{A} = \prod_{i=1}^{N} 0,1 \quad (9)$$

Each element a_i indicates selection 1 or non-selection 0. However, in each round, the following constraints must be satisfied:

$$\| a_t \|_1 = K \quad (10)$$

where r_t^k is the reward associated with the selection of device k; θ and θ' represent the parameters of the main and target networks, respectively, and γ is the discount factor, with $0 \leq \gamma \leq 1$, determining the importance of future rewards compared to current rewards. A value closer to 1 place more emphasis on future rewards, whereas a value closer to 0 prioritizes current rewards.

In traditional reinforcement learning, the goal is to select a single optimal action for a given state. However, in this study, we adopted a multi-action selection approach to select multiple devices. Specifically, the selection of each device was treated as an independent action, and the loss for each device was calculated using Equation (7). This approach eliminates the need to explore all possible device combinations.

This method allows for efficient identification of optimal devices while considering the cooperative relationships and interactions among the devices. Furthermore, by evaluating the impact of each device selection on the overall learning outcomes, more effective learning is expected.

5. EVALUATION

5.1. EXPERIMENTAL SETUP

The datasets and models used in this study are as follows.

- **MNIST:**

The dataset consists of 60,000 grayscale images of handwritten digits for training and 10,000 images for testing. Each image has a resolution of 28 × 28 pixels and is classified into one of 10-digit classes (0–9). Due to its simplicity, balanced class distribution, and ease of implementation, MNIST is one of the most used benchmark datasets in FL research. In this study, it was adopted to enable comparison with existing methods and to validate the effectiveness of the proposed method under standard and relatively simple experimental conditions.

For the model, we used a simple multilayer perceptron (MLP) consisting of one hidden layer with 100 units and ReLU activation. The input layer had 784 dimensions (28 × 28), and the output layer had 10 units corresponding to the number of classes.

- **CIFAR-10:**

The CIFAR-10 dataset is a widely used standard benchmark consisting of 60,000 32 × 32 pixel color images classified into 10 classes. Each class contains 6,000 images. This dataset is extensively used in machine learning research, including comparative methods, and was thus adopted in this study. In addition, to introduce heterogeneity, we performed non-IID partitioning following a Dirichlet distribution. The parameter values used were $Dir(0.1)$ and $Dir(0.5)$.

The model used for this dataset was a simple convolutional neural network composed of two convolutional layers and three fully connected layers. Specifically, the first convolutional layer used 16 filters with a kernel size of 3×3, followed by a 2×2 max pooling layer. The second convolutional layer had 32 filters with a kernel size of 3×3, followed by another 2×2 max pooling layer. The fully connected layers had 120, 84, and 10 units respectively.

- **GTSRB:**

The German Traffic Sign Recognition Benchmark (GTSRB) is one of the primary datasets for traffic sign recognition and classification tasks and contains approximately 50,000 images classified into 43 different traffic sign classes. Each image was captured in a real road environment, encompassing variations in lighting conditions and viewpoints. The GTSRB is commonly used in FL research [40-42]. In this study, in using a dataset with 43 classes, which exceeds the 10 classes of CIFAR-10, we aimed to evaluate the model's classification ability and its adaptability to heterogeneity more thoroughly. This enabled a multifaceted validation of the versatility and performance of the proposed method. Additionally, non-IID partitioning was performed following a Dirichlet distribution to introduce heterogeneity. The parameter values used were Dir(0.1) and Dir(0.5).

For this dataset, we used a simple multilayer perceptron consisting of an input layer with 3072 dimensions (32 × 32 × 3), a 128-dimensional hidden layer with ReLU activation, and an output layer with 43 units corresponding to the number of traffic sign classes. This model design follows the experimental settings of Li et al. [30] and Jialuo et al. [43].

- **Synthetic:**

The synthetic dataset is generated using the method inspired by Li et al. [30] and Shamir et al. [44], denoted as $SYNTHETIC(\alpha, \beta)$.

Specifically, the data samples (X_k, Y_k) for device k (with sample size n_k) were generated as follows: The model is defined by the following equation:

$$y = argmax\left(softmax(W_k x + b_k)\right) \quad (11)$$

where $x \in R^{60}$, $W_k \in R^{10\times60}$, and $b_k \in R^{10}$. The weight matrix W_k and bias vector b_k were sampled from a normal distribution with mean u_k and variance 1:

$$W_k \sim \mathcal{N}(u_k, 1), \quad b_k \sim \mathcal{N}(u_k, 1) \tag{12}$$

The mean vector u_k was sampled from a normal distribution with mean 0 and variance α:

$$U_k \sim \mathcal{N}(0, \alpha) \tag{13}$$

Each element of the input data x_k, denoted by $(x_k)_j$, was sampled from a normal distribution with mean v_k and variance $j^{-1.2}$:

$$x_{kj} \sim \mathcal{N}(v_k, j^{-1.2}) \tag{14}$$

where v_k is sampled from a normal distribution with mean μ_k and variance 1, and μ_k followed a normal distribution with mean 0 and variance β:

$$v_k \sim \mathcal{N}(\mu_k, 1), \quad \mu_k \sim \mathcal{N}(0, \beta) \tag{15}$$

This method allows controlling the heterogeneity of models and data across devices by adjusting parameters α and β. *SYNTHETIC*(0,0) and *SYNTHETIC*(1,1) were used in the experiments. Both had 10 classes and a data size of approximately 50,000. In this study, synthetic data were generated and used to control for heterogeneity and evaluate the changes in the performance of the proposed method by varying the degrees of heterogeneity. For this dataset, we used a logistic regression model following the experimental settings of Li et al. [30] and Jialuo et al. [43]. The model consisted of a single fully connected (linear) layer that takes a 100-dimensional input vector and outputs scores for 10 classes.

5.2. COMPARISON METHODS

We selected the following methods as baselines:

- **FedAVG [14]**: This is adopted as the basic method to evaluate the baseline performance against Non-IID data.
- **FedFE [29]**: This is a method that uses momentum gradient descent to improve convergence speed while considering fairness. The parameter settings used (α, β)=(0.5,0.5), were based on the optimal values in the experiments of Wentao et al. [29].
- **q-FFL [30]**: This is adopted to reduce performance disparities among devices, using $q = 1$ for the synthetic dataset and $q = 0.1$ for other datasets. These settings were determined based on the optimal values in the experiments of Li et al. [30].
- **FedHEAL [31]**: We adopted this method, a state-of-the-art FL method aimed at improving fairness. Following the experimental settings reported by Chen et al. [31], we set the parameters to (β, τ) = (0.4, 0.1), which demonstrated the best performance in their experiments.

The proposed method was trained using the hyperparameters listed in Table 2. These values were chosen based on common practices in the federated learning literature [29-31]. In particular, the number of local epochs was selected within the typical range of 1 to 10, which is widely adopted in prior studies [29-31]. The value of B was determined based on the experimental results reported by Wentao et al. [29].

Table 2. Hyperparameters of Experiments

Hyperparameters	MNIST	CIFAR10/GTSRB	SYNTHETIC
N (number of devices)	100	100	100
K (size of selected devices)	10	10	10
E (local epochs)	5	10	5
B (batch size)	16	32	32
Learning rate	0.01	0.01	0.1
Momentum	0.9	0.9	0.9
RL batch size	50	50	50
P (number of steps)	10	10	10
RL learning rate	$10e^{-5}$	$10e^{-5}$	$10e^{-5}$
γ (discount factor)	0.99	0.99	0.99
M	1.01	1.01	1.01

For each dataset, the target accuracy (*TargetACC*) was set based on existing FedFE [29] and q-FFL [30] methods. Specifically, experiments were conducted using these methods, and the average accuracy of the bottom B% of devices (*BottomACC*) was measured. Based on these results, *TargetACC* was set to a value that exceeded the *BottomACC* achieved by q-FFL and FedFE by a few percent.

- MNIST (0.1): 85%
- MNIST (0.5): 90%
- CIFAR-10(0.1): 37%
- CIFAR-10(0.5): 47%
- GTSRB (0.1): 77%
- GTSRB (0.5): 8%
- SYNTHETIC (0,0): 15%
- SYNTHETIC (1,1): 10%

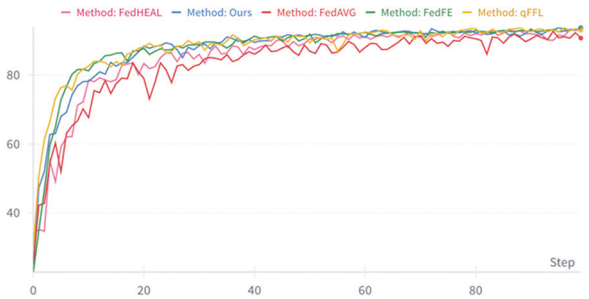
5.3. RESULTS & DISCUSSION

Fig. 2 and Table 3 present the progression and final outcomes of the accuracy of each method across the datasets used in this study. The evaluation metrics employed included the average accuracy of each device's local test data, variance and average accuracy of the bottom 10% of the devices, as well as top 10% of the devices.

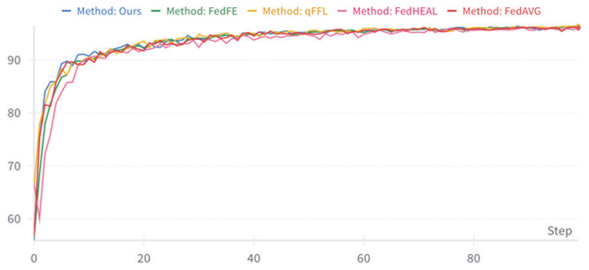
The proposed method successfully enhanced the accuracy of the bottom 10% of the devices across all datasets, without compromising the overall average accuracy. In some datasets, variance of the accuracy was reduced compared with those of existing methods (Table 3). Notably, on CIFAR-10 with a Dirichlet parameter of 0.1 CIFAR-10(0.1), which represents a highly non-IID environment, the improvement in the accuracy of the bottom 10% of the devices was particularly significant.

Dataset	Method	Variance	Worst 10%	Accuracy	Best 10%
MNIST(0.1)	FedAVG	36.1 ±18.9	79.5±3.3	90.7 ±2.6	99.1+.8
	q-FFL	30.5 ±16.9	81.4±4.8	93.1+.8	99.4±.4
	FedFE	17.7+3.5	85.2±1.8	93.5 ±.3	99.4±.4
	FedHEAL	**14.7 ±2.2**	86.1 ±.7	93.7±.3	99.3±.3
	Ours	15.1±3.9	**86.2 ±1.0**	**93.9 ±.4**	**99.4±.1**
MNIST(0.5)	FedAVG	5.8+1.8	91.1+1.3	96.1 ±.4	99.3±.1
	q-FFL	**4.0 ±.8**	**92.4 ±.6**	**96.4 ±.2**	**99.2±.1**
	FedFE	4.1 ±.5	92.3 ±.4	96.2±.2	99.1+.1
	FedHEAL	5.6+1.5	91.2±1.0	95.9±.2	99.0±.2
	Ours	4.2 ±.6	92.3±.5	96.3 ±.3	99.2±.2
CIFAR10(0.1)	FedAVG	152.7+32.5	30.4±3.2	51.3+1.0	72.0±2.0
	q-FFL	191.9+14.5	30.6 ±1.2	52.8 ±1.5	**78.7±2.3**
	FedFE	**123.4+3.7**	34.7 ±1.3	53.5+1.7	73.0±1.2
	FedHEAL	145.3+35.3	29.4 ±3.9	48.8±2.8	71.0±2.9
	Ours	130.8+11.3	**38.7 ±1.0***	**54.7+.9**	78.4±1.7
CIFAR10(0.5)	FedAVG	63.3+2.4	42.5 ±.7	56.9 ±.5	**70.7 ±.6**
	q-FFL	56.9+3.7	43.0±1.0	57.0+1.0	69.8 ±.9
	FedFE	**51.5+5.6**	43.8±.8	56.6±1.0	69.1 + 1.3
	FedHEAL	52.6+8.8	41.7±1.2	54.5+.8	66.8±1.8
	Ours	51.4+3.8	**44.8 ±.9**	**57.6 ±1.0**	69.7+1.5
GTSRB(0.1)	FedAVG	82.1+24.8	68.0±5.0	86.9+1.5	97.7 ±.4
	q-FFL	70.6+20.1	69.4 ±4.0	88.0±.5	97.9+.4
	FedFE	59.6+13.5	70.3 ±2.2	86.9+1.6	96.5±1.6
	FedHEAL	78.4+23.4	66.0 ±4.6	84.9±2.5	95.9+1.4
	Ours	**44.7 ±6.5**	**75.5 ±1.4**	**89.9+.4**	**98.1 ±.4**
GTSRB(0.5)	FedAVG	38.5+20.1	70.4+11.1	81.2±8.5	90.7+5.4
	q-FFL	**9.0 ±1.1**	87.7±.9	93.5+ .5	97.9 ±.7
	FedFE	17.9+2.7	81.6+2.7	89.6±1.5	96.1 + 1.1
	FedHEAL	16.4±1.4	82.5 ±1.2	90.4+1.0	96.2±.6
	Ours	9.2+1.2	**87.9+.8**	**93.6 ±.4**	**98.3±.4**
SYNTHETIC(0,0)	FedAVG	1429.7±35.3	o.o±.o	34.3+1.8	99.9±.1
	q-FFL	849.6 +42.0	11.0+1.0	69.2±1.0	**100.0 ±.0**
	FedFE	893.1+26.1	12.1±1.1	71.4+.7	**100.0 ±.0**
	FedHEAL	1075.8±70.8	0.1+.2	48.3±2.1	99.5±.7
	Ours	**824.6 + 55.4**	**15.1 ±.7**	**73.5+2.1**	**100.0 ±.0**
SYNTHETIC(1.1)	FedAVG	1405.5 ±74.1	0.0+.0	34.0 ±1.4	**100.0 ±.0**
	q-FFL	1024.6 ±46.3	7.8 ±2.0	68.4+2.1	**100.0 ±.0**
	FedFE	1044.5 ±46.3	6.7+1.1	70.4±2.4	**100.0 ±.0**
	FedHEAL	1423.1 ±36.3	o.o±.o	48.0+3.7	**100.0 ±.0**
	Ours	**926.5 ±71.8**	**10.5+0.5**	**73.8±2.0**	**100.0 ±.0**

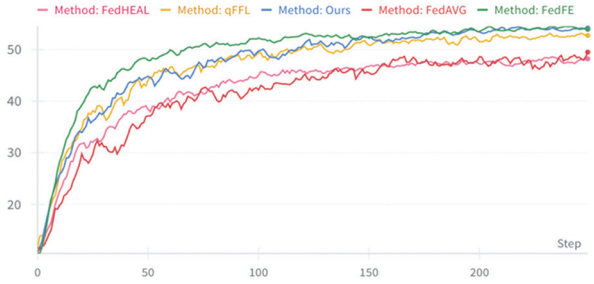
*Indicates statistically significant differences (p<0.05) between the proposed method and other methods for the corresponding metric



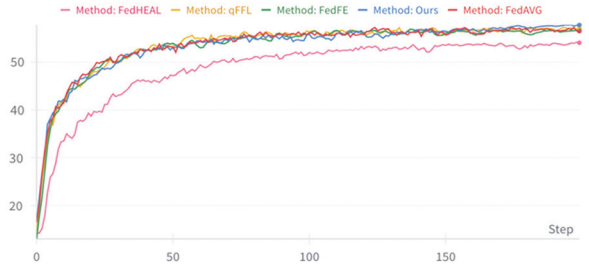
(a)



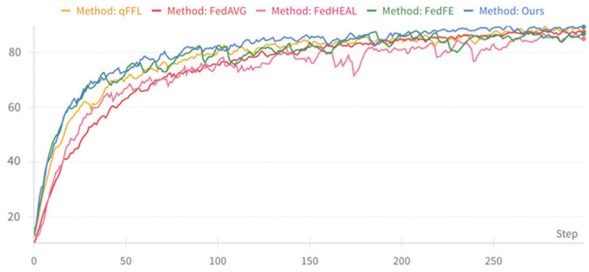
(b)



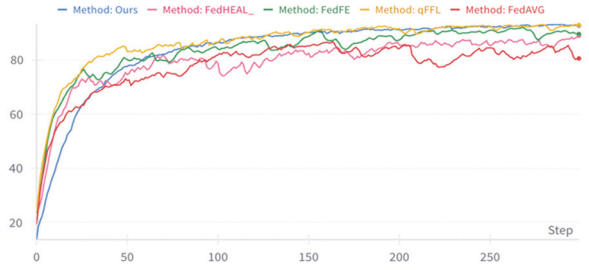
(c)



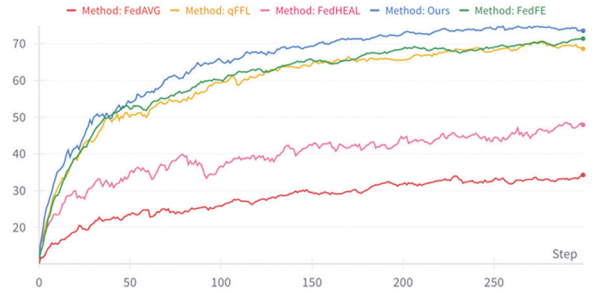
(d)



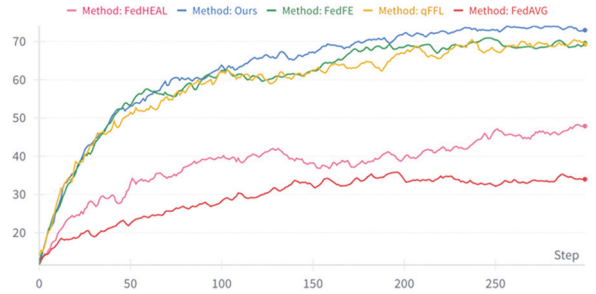
(e)



(f)



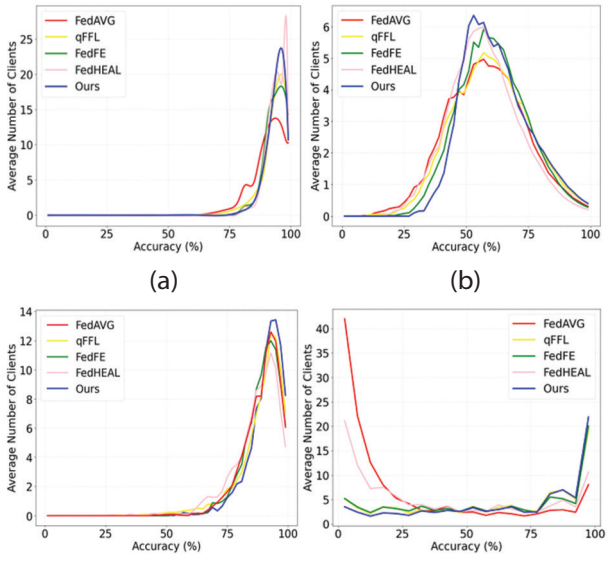
(g)



(h)

Fig. 2. Accuracy progression charts of average accuracy for each of the six datasets. **(a)** MNIST (0.1), **(b)** MNIST (0.5), **(c)** CIFAR-10(0.1), **(d)** CIFAR-10(0.5), **(e)** GTSRB (0.1), **(f)** GTSRB (0.5), **(g)** SYNTHETIC (0,0), **(h)** SYNTHETIC (1,1)

A partial distribution of the accuracy is shown in Fig. 3. This figure was derived from datasets with stronger non-IID characteristics, displaying the most pronounced improvements. Visually, the number of low-performing devices has clearly decreased compared with respect to the baseline methods.



(a) (b)

(c) (d)

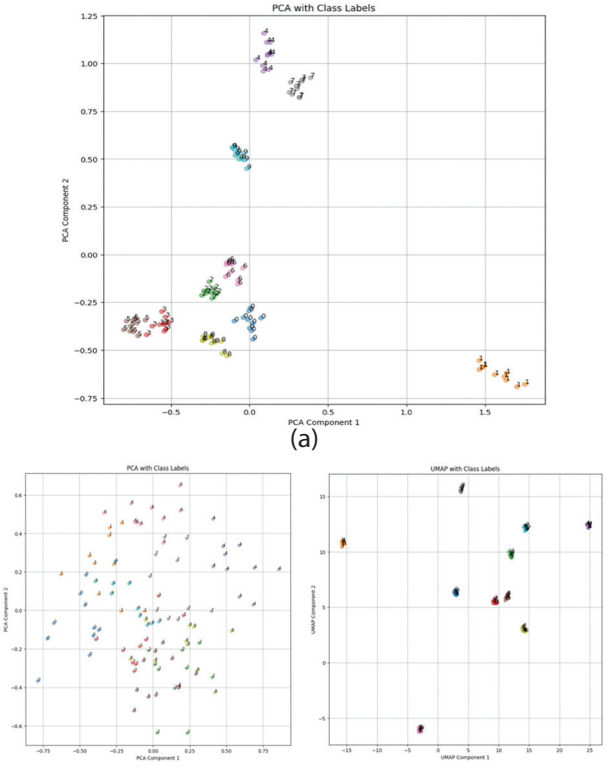
Fig. 3. Accuracy distribution map **(a)** MNIST(0.1), **(b)** CIFAR-10(0.1), **(c)** GTSRB(0.1), **(d)**SYNTHETIC(1,1)

In this study, we used UMAP for dimensionality reduction of the model weights of each device to better capture the underlying distribution among devices.

To evaluate its effectiveness, we used the MNIST dataset and introduced varying degrees of label imbalance among devices by controlling a parameter Z, which denotes the proportion of a single dominant label in each device's data. For instance, Z=80 indicates that 80% of the data within a device belong to one specific label, while the remaining 20% are uniformly distributed among the other labels. A setting of Z=100 represents extreme label concentration (single-label scenario), whereas Z =10 corresponds to a fully IID scenario, with all ten MNIST labels evenly represented.

We visualized the model weights after one epoch of local training and reduced them to two dimensions using both PCA and UMAP. While PCA was able to reveal some cluster structure under highly imbalanced settings (Fig. 4 (a)), it struggled to clearly separate clusters when the distribution became more subtle (Fig. 4 (b)). In contrast, UMAP consistently provided clearer and more distinct cluster formations, even in moderately complex distributions (Fig. 4 (c)). This suggests that UMAP captured the latent structures in the model weights more effectively than PCA.

By reducing the weights to 10 dimensions and using them as state representations for reinforcement learning, our method allowed the agent to more accurately distinguish between devices with different underlying data characteristics. This contributed to more effective device selection and, ultimately, better performance under heterogeneous data distributions.



(a)

(b) (c)

Fig. 4. Dimensionality Reduction of Local Model Weights. **(a)**with PCA (Z = 80), **(b)** with PCA (Z=20), **(c)** with UMAP (Z=20)

Additionally, Fig. 5 shows the number of selections for each device in CIFAR-10(0.1). As observed, the reinforcement learning agent intentionally selected devices that would increase accuracy. When the devices are randomly selected, the number of selections X for a single device follows $X \sim Binomial(n=250, p=0.1)$ with expected mean $\mu = np = 25$ and $\sigma = \sqrt{(np(1-p))} \approx 4.74$ as standard deviation. Typically, assuming a normal distribution, approximately 99.7% of the data would lie within the range [10.78,39.22]. However, in Fig. 5, approximately ten devices fall outside this range, suggesting that the reinforcement learning agent intentionally increased the selection frequency of these devices. In addition, upon examining the data distribution of the most frequently selected devices in Fig. 5, these devices were observed to have the largest amounts of data among those possessing more than five classes. As illustrated, because the number of devices selected per round is limited to K, devices with more diverse and abundant data were chosen more frequently.

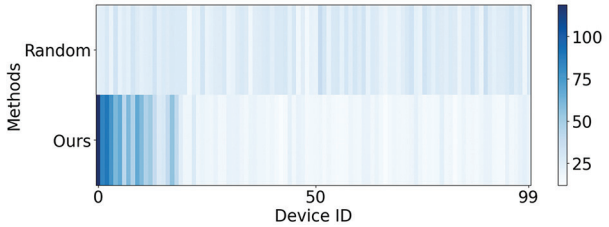


Fig. 5. Average number of device selections in CIFAR-10 (0.1)

6. CONCLUSION

This study introduces a novel approach designed to suppress the occurrence of low-accuracy devices in FL. The proposed method integrates reinforcement learning-based device selection using a DDQN and incorporates a reward mechanism based on the distance from the global model. Furthermore, it employs multi-action selection to choose multiple devices simultaneously, thereby ensuring an efficient selection process. By utilizing UMAP for the state representation, this method achieves both dimensionality reduction and enhanced representational capabilities.

The results indicate that the proposed approach effectively improves the average accuracy of the bottom 10% of the devices by up to approximately 4% without diminishing the overall average accuracy compared to existing methods. In addition, beyond the 10-class CIFAR-10 dataset, the method successfully suppressed low-accuracy devices in the GTSRB dataset, containing a greater number of classes. This demonstrates the versatility and effectiveness of the proposed method across diverse datasets.

In future research, we plan to extend the application of reinforcement learning beyond device selection to include weighted aggregation, with particular attention paid to the potential of multi-agent reinforcement learning. Moreover, addressing real-world challenges such as data heterogeneity, communication costs, and privacy concerns remains essential. Developing new algorithms that specifically aim to suppress low-accuracy devices in non-IID environments, while considering these practical constraints, is critical for ongoing and future investigations.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI, Grant Numbers JP22K12157, JP23K28377, and JP24H00714.

6. REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", arXiv:1602.05629, 2016.

[2] S. Pati, U. Baid, B. Edwards, M. Sheller, S. H. Wang, G. A. Reina, L. Poisson, "Federated Learning Enables Big Data for Rare Cancer Boundary Detection", Nature Communications, Vol. 13, No. 1, 2022.

[3] T. Qi, F. Wu, C. Wu, L. He, Y. Huang, X. Xie, "Differentially Private Knowledge Transfer for Federated Learning", Nature Communications, Vol. 14, No. 1, 2023, p. 3785.

[4] N. Boscarino, R. A. Cartwright, K. Fox, K. S. Tsosie, "Federated Learning and Indigenous Genomic Data Sovereignty", Nature Machine Intelligence, Vol. 4, No. 11, 2022, pp. 909-911.

[5] C. Wu, F. Wu, L. Lyu, Y. Huang, X. Xie, "Communication-Efficient Federated Learning via Knowledge Distillation", Nature Communications, Vol. 13, No. 1, 2022, p. 2032.

[6] M. Asad, S. Shaukat, D. Hu, Z. Wang, E. Javanmardi, J. Nakazato, M. Tsukada, "Limitations and Future Aspects of Communication Costs in Federated Learning: A Survey", Sensors, Vol. 23, No. 17, 2023, p. 7358.

[7] X. Ma, J. Zhu, Z. Lin, S. Chen, Y. Qin, "A State-of-the-Art Survey on Solving Non-IID Data in Federated Learning", Future Generation Computer Systems, Vol. 135, 2022, pp. 244-258.

[8] K. Oishi, Y. Sei, Y. Tahara, A. Ohsuga, "Federated Learning Algorithm Handling Missing Attributes", Proceedings of the IEEE International Conference on Internet of Things and Intelligence Systems, Bali, Indonesia, 28-30 November 2023, pp. 146-151.

[9] S. Lin, Y. Han, X. Li, Z. Zhang, "Personalized Federated Learning Towards Communication Efficiency, Robustness and Fairness", in Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 30471-30485.

[10] B. R. Chaudhury, L. Li, M. Kang, B. Li, R. Mehta, "Fairness in Federated Learning via Core-Stability", in Advances in Neural Information Processing Systems, Vol. 35, 2022, pp. 5738-5750.

[11] Z. Yang, Y. Zhang, Y. Zheng, X. Tian, H. Peng, T. Liu, B. Han, "FedFed: Feature Distillation Against Data Heterogeneity in Federated Learning", Advances in Neural Information Processing Systems, Vol. 36, 2023, pp. 60397-60428.

[12] D. Pessach, E. Shmueli, "A review on fairness in machine learning", ACM Computing Surveys, Vol. 55, No. 3, 2022, pp. 1-44.

[13] S. Vucinich, Q. Zhu, "The Current State and Challenges of Fairness in Federated Learning", IEEE Access, Vol. 11, 2023, pp. 80903-80914.

[14] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, Vol. 54, 2017, pp. 1273-1282.

[15] C. Tian, Z. Shi, X. Qin, L. Li, C. Xu, "Ranking-Based Client Selection with Imitation Learning for Efficient Federated Learning", arXiv:2405.04122, 2024.

[16] Q. Pan, H. Cao, Y. Zhu, J. Liu, B. Li, "Contextual Client Selection for Efficient Federated Learning over Edge Devices", IEEE Transactions on Mobile Computing, Vol. 23, No. 6, 2023, pp. 6538-6548.

[17] T. Zhang, K. Y. Lam, J. Zhao, F. Li, H. Han, N. Jamil, "Enhancing Federated Learning with Spectrum Allocation Optimization and Device Selection", IEEE/ACM Transactions on Networking, Vol. 31, No. 5, 2023, pp. 1981-1996.

[18] J. Qi, Q. Zhou, L. Lei, K. Zheng, "Federated Reinforcement Learning: Techniques, Applications and Open Challenges", arXiv:2108.11887, 2021.

[19] H. Wang, Z. Kaplan, D. Niu, B. Li, "Optimizing Federated Learning on Non-IID Data with Reinforcement Learning", Proceedings of IEEE INFOCOM 2020 - IEEE Conference on Computer Communications, Toronto, ON, Canada, 6-9 July 2020, pp. 1698-1707.

[20] W. Chen, J. Du, Y. Shao, J. Wang, Y. Zhou, "Dynamic fair federated learning based on reinforcement learning", Proceedings of the 5th International Conference on Data-driven Optimization of Complex Systems, Tianjin, China, 22-24 September 2023, pp. 1-8.

[21] H. Zhang, Z. Xie, R. Zarei, T. Wu, K. Chen, "Adaptive Client Selection in Resource Constrained Federated Learning Systems: A Deep Reinforcement Learning Approach", IEEE Access, Vol. 9, 2021, pp. 98423-98432.

[22] S. Bouaziz, H. Benmeziane, Y. Imine, L. Hamdad, S. Niar, H. Ouarnoughi, "FLASH-RL: Federated Learning Addressing System and Static Heterogeneity using Reinforcement Learning", Proceedings of the IEEE 41st International Conference on Computer Design, Washington, DC, USA, October 16-18, 2023, pp. 444-447.

[23] X. Yu, Z. Gao, Z. Xiong, C. Zhao, Y. Yang, "Ddpg-AdaptConfig: A Deep Reinforcement Learning Framework for Adaptive Device Selection and Training Configuration in Heterogeneity Federated Learning", Future Generation Computer Systems, Vol. 163, 2025, p. 107528.

[24] Y. Shi, H. Yu, C. Leung, "Towards Fairness-Aware Federated Learning", IEEE Transactions on Neural Networks and Learning Systems, Vol. 35, No. 9, 2024, pp. 11922-11938.

[25] T. H. Rafi, F. A. Noor, T. Hussain, D. K. Chae, "Fairness and Privacy Preserving in Federated Learning: A Survey", Information Fusion, Vol. 105, 2024, p. 102198.

[26] H. Chen, T. Zhu, T. Zhang, W. Zhou, P. S. Yu, "Privacy and Fairness in Federated Learning: On the Perspective of Tradeoff", ACM Computing Surveys, Vol. 56, No. 2, 2023, pp. 1-37.

[27] W. Huang, M. Ye, Z. Shi, G. Wan, H. Li, B. Du, Q. Yang, "Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 46, No. 12, 2024, pp. 9387-9406.

[28] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, "Fairness and Accuracy in Federated Learning", arXiv:2012.10069, 2020.

[29] P. Wentao, H. Zhou, "Fairness and Effectiveness in Federated Learning on Non-independent and Identically Distributed Data", Proceedings of the IEEE 3rd International Conference on Computer Communication and Artificial Intelligence, Taiyuan, China, 26-28 May 2023, pp. 97-102.

[30] T. Li, M. Sanjabi, A. Beirami, V. Smith, "Fair Resource Allocation in Federated Learning", arXiv:1905.10497, 2019.

[31] Y. Chen, W. Huang, and M. Ye, "Fair Federated Learning under Domain Skew with Local Consistency and Domain Diversity", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16-22 June 2024, pp. 12077-12086.

[32] A. Bietti, C. Y. Wei, M. Dudík, J. Langford, S. Wu, "Personalization Improves Privacy-Accuracy Tradeoffs in Federated Learning", Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 2022, pp. 1945-1962.

[33] C. Dwork, "Differential Privacy", Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Venice, Italy, 10-14 July 2006, pp. 1-12.

[34] Z. Jiang, W. Wang, R. Chen, "Dordis: Efficient Federated Learning with Dropout-Resilient Differential Privacy", Proceedings of the Nineteenth European Conference on Computer Systems, Athens, Greece, 22-25 April 2024, pp. 472-488.

[35] M. Naseri, J. Hayes, E. De Cristofaro, "Local and Central Differential Privacy for Robustness and Privacy in Federated Learning", arXiv:2009.03561, 2020.

[36] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized Aggregatable Privacy-Preserving Ordinal Response", Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3-7 November 2014, pp. 1054-1067.

[37] Y. Sei, J. A. Onesimu, A. Ohsuga, "Machine Learning Model Generation with Copula-Based Synthetic Dataset for Local Differentially Private Numerical Data", IEEE Access, Vol. 10, 2022, pp. 101656-101671.

[38] T. Qi, H. Wang, Y. Huang, "Towards the Robustness of Differentially Private Federated Learning", Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20-27 February 2024, pp. 19911-19919.

[39] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali, E. Cambria, "A Survey on Deep Reinforcement Learning for Audio-Based Applications", Artificial Intelligence Review, Vol. 56, No. 3, 2023, pp. 2193-2240.

[40] M. Hasumi, T. Azumi, "Federated Learning Platform on Embedded Many-core Processor with Flower", Proceedings of the IEEE 3rd Real-Time and Intelligent Edge Computing Workshop, Hong Kong, Hong Kong, 13 May 2024, pp. 1-6.

[41] J. Lai, X. Huang, X. Gao, C. Xia, J. Hua, "GAN-Based Information Leakage Attack Detection in Federated Learning", Security and Communication Networks, Vol. 2022, No. 3, 2022, pp. 1-10.

[42] S. A. Khowaja, P. Khuwaja, K. Dev, A. Antonopoulos, "SPIN: Simulated Poisoning and Inversion Network for Federated Learning-Based 6G Vehicular Networks", Proceedings of the IEEE International Conference on Communications, Rome, Italy, 28 May - 1 June 2023, pp. 6205-6210.

[43] H. Jialuo, W. Chen, X. Zhang, "Reinforcement Learning as a Catalyst for Robust and Fair Federated Learning: Deciphering the Dynamics of Client Contributions", arXiv:2402.05541, 2024.

[44] O. Shamir, N. Srebro, T. Zhang, "Communication-Efficient Distributed Optimization Using an Approximate Newton-Type Method", Proceedings of the 31st International Conference on Machine Learning, 2014, Vol. 32, No. 2, pp. 1000-1008.

Integrating Squeeze-and-Excitation Network with Pretrained CNN Models for Accurate Plant Disease Detection

Original Scientific Paper

Lafta Raheem Ali
General Directorate of Education of Salahuddin
Salahuddin, Iraq
l.alkhazraji@gmail.com

Sabah Abdulazeez Jebur*
Imam Alkadhim University College,
Department of Cyber Security
Baghdad ,Iraq
sabah.abdulazeez@iku.edu.iq

Mothefer Majeed Jahefer
Imam Alkadhim University College,
Department of Computer Science
Baghdad ,Iraq
modafarmajed@iku.edu.iq

*Corresponding author

Abbas Khalifa Nawar
Imam Alkadhim University College, Department of Computer Science
Baghdad ,Iraq
abbas.altimimy@iku.edu.iq

Zaed S. Mahdi
University of Technology,
Information Technology Center
Baghdad ,Iraq
zaed.s.mahdi@uotechnology.edu.iq

Abstract *– The increasing global population and the challenges posed by climate change have intensified the demand for sustainable food production. Traditional agricultural practices are often insufficient, leading to significant crop losses due to diseases and pests, despite the widespread use of pesticides and other chemical interventions. This paper introduces a new approach that integrates deep learning techniques, specifically Convolutional Neural Networks (CNNs) with Squeeze and Excitation (SE) networks, to enhance the accuracy of disease detection in fig leaves. By leveraging three pre-trained CNN models—MobileNetV2, InceptionV3, and Xception—this framework addresses data scarcity issues and improves feature representation while minimizing the risk of overfitting. Data augmentation techniques were employed to counteract data imbalance, and visualization tools like Grad-CAM and t-SNE were utilized for model interpretability. The proposed CNN-SE model was trained and evaluated on a fig leaf dataset comprising 1,196 images of healthy and diseased fig leaves, achieving an accuracy of 92.90% with MobileNet-SE, 91.48% with Inception-SE, and 89.62% with Xception-SE. Our model demonstrates superior performance in detecting fig leaf diseases, presenting a robust solution for sustainable agriculture by providing accurate, efficient, and scalable disease management in crops. The code of the proposed framework is available at https://github.com/lafta/SE-block-with-CNN-Models-for-Plant-Disease-Detection.*

Keywords*: Deep Learning , Convolutional Neural Network ,Squeeze-and-Excitation , Plants diseases detection*

Received: January 25, 2025; Received in revised form: May 3, 2025; Accepted: May 30, 2025

1. INTRODUCTION

One of the greatest challenges to increasing agricultural productivity is the spread of pests and diseases in crops, which are considered the primary cause of more than a third of annual agricultural production losses [1]. To protect plants from these threats, numerous pesticides and costly techniques are employed. However, the large-scale use of these chemical methods has adverse effects on species diversity, human health, and crop yields, while also increasing production costs [2]. Recently, researchers have reported remarkable progress in applying Artificial Intelligence (AI) technology, particularly deep learning (DL) techniques, to the detection and classification of diseases on plants. These techniques have played a key role in transforming conventional farming practices into more sustainable ones by providing accurate, efficient, and scalable

solutions, even aiding in the early diagnosis of diseases [3]. DL algorithms can classify image data based on their feature content and extract relevant information [4]. Convolutional Neural Networks (CNNs) are a specialized type of DL models primarily designed to process image data, automatically learning features and patterns before making decisions [5]. The extracted features are fed into the classifier without human intervention, unlike in traditional machine learning, where feature extraction and classification are separate steps. Essentially, CNNs are composed of two steps; feature extraction and classification, the first step are employed three operations which are convolution operation that achieved by convolutional layer, activation function, and pooling operation. Various filters are applied to analyze and detect the important features in the image starting from small features such as edges, lines, and corners till reach the very important features (faces, leaves, etc.) [6]. From the perspective of feature re-calibration, a Squeeze and Excitation (SE) network has been introduced to capture the interdependencies between convolutional feature channels [7]. The SE block consists of two main processes: squeezing and excitation. The squeeze operation creates a channel descriptor by summarizing feature maps across their spatial dimensions to embed global information. The excitation process generates channel-specific weights. Through feature re-calibration, the SE block can selectively highlight important features while diminishing less relevant ones. This block can be incorporated into conventional DL models, such as CNNs [8]. Despite all the capabilities CNNs offer, they still face several challenges, the most significant being the need for large amounts of training data [9]. This has prompted researchers in the field of AI to explore the use of transfer learning (TL), a technique that improves model performance by transferring knowledge from an already trained model instead of training the model from scratch [10]. In addition to TL, data augmentation is a technique used to handle the lack of data by increasing the size of the training dataset through various transformations of existing images, such as translation, rotation, shearing, flipping, zooming, etc. This generates new data that is added to the original dataset, enhancing the model's generalization and robustness [11]. Since DL as a black box, it is difficult to understand what occurs within the hidden layers and how these networks make decisions or predictions [12]. To address this challenge, transparency is necessary to identify the regions the model focuses on. This can be achieved using explainable learning techniques. Specifically, Grad-CAM and t-SNE visualization techniques are employed to bridge this gap, providing deeper insight and a clearer understanding of how the model reaches its conclusions. This study aims to address the aforementioned challenges by developing an accurate plant disease detection model through feature extraction from leaf images using multiple CNN architectures, and by integrating a squeeze-and-excitation (SE) block to enhance classification accuracy. The main contributions of this paper include:

- A new CNN-SE framework integrating CNNs with SE network has been proposed. This approach enhances feature learning by focusing on informative channels and dynamically recalibrating weights, improving fig leaf disease detection accuracy.
- Three pre-trained CNN models from ImageNet were employed to mitigate data scarcity, improve feature extraction, and reduce overfitting risks.
- Data augmentation techniques were applied to address class imbalance and limited training data, enhancing model generalization.
- Interpretability tools, Grad-CAM and t-SNE, were used to analyze model decisions, visualize feature importance, and detect potential biases.

2. LITERATURE REVIEW

This section presents the most recently studies in the field of detecting the plant leaf disease using the DL techniques. Saikat Datta and Nitin Gupta [13] developed a deep CNN-based architecture to classify tea leaf diseases into six categories: Gray Blight, Algal Spot, Brown Blight, Heliopolis, Healthy Leaves, and Red Spot. They introduced a novel real-world dataset containing 5,867 images, covering five disease types and healthy leaves. F. Khan et al. [14] proposed a DL framework for detecting blight, leaf spot and sugarcane mosaic virus in maize crops. they evaluated five YOLO variants (YOLOv3-tiny, YOLOv4, YOLOv5s, YOLOv7s and YOLOv8n) and selected YOLOv8n for its compact architecture and superior inference speed. this model was subsequently deployed in a mobile application to real-time disease management in agricultural settings. MKA Mazumder et al. [15] proposed LeafDoc-Net, lightweight TL architecture that integrates two pretrained CNN models, DenseNet-121 and MobileNetV2, for multi-species leaf disease detection. The model employs an attention-based transition mechanism for enhancing feature fusion, followed by global average pooling to reduce spatial dimensionality. Additionally, it incorporate dense layers with swish activation and batch normalization to deepen the network while maintaining computational efficiency. Qinghai Wu et al. [16] proposed DL model contains three components of feature extraction, attention calculation and then lastly the classification, an attention module was added to generate feature maps at various depths for enhancing the network's focus on discriminative features while reduce background noise. The attention module also made use of LeakyReLU as an activation function to tackle the problem of neurons failing to learn when their input is negative, The extracted features were integrated through a fully connected layer to predict disease category for soybean leaf. YA Bezabh et al. [17] proposed a pepper disease classification model based on two CNN architectures: AlexNet and VGG16. The authors utilized these two CNN architectures to extract features, then combined the extracted features in single features set.

After that the combined feature set was used as input to the fully connected layers for classification with a multiclass classifier. Rina Bora et al. [18] developed a framework known as the Multivariate Normal Deep Learning Neural Network (MNDLNN) to detect diseases in the leaves, fruits, roots and stems of tomato plants. The methodology comprises of conversion the image color to HSI format, masking of green color to obtain the healthy and unhealthy region, identification of fruits and roots with the region of interest, segmenting the unhealthy region via RKMC clustering and final stage includes the extraction of necessary features using RMSSO. Anuradha Chug et al. [19] proposed a Hybrid Deep Learning (HDL) framework that combines EfficientNet architectures (B0–B7) as feature extractors with five machine learning classifiers. They developed the IARI-TomEBD dataset, a real-time image collection of tomato early blight disease for experimental validation. The HDL models demonstrated strong performance on this custom dataset and were further evaluated on two public plant disease benchmarks. The EfficientNet-B3-ADB and EfficientNet-B3-SGB configurations achieved state-of-the-art results across all datasets. Mahum, Rabbia, et al. [20] proposed an Enhanced DenseNet model by integrating an additional transition layer into DenseNet-201. To address extreme class imbalance in the training data, they employed a reweighted cross-entropy loss function, enhancing model robustness. Ashwathnarayan Nagarjun et al. [21] proposed a cotton leaf disease classification method combining transfer learning and deep learning techniques. For the deep learning component, they employed a conventional convolutional neural network (CNN), while their transfer learning approach utilized architectures such as Inception and ResNet. The study relied on a custom-collected cotton disease dataset to achieve its objectives. However, the work exhibits significant ambiguities and lacks critical implementation details, including methodological transparency and reproducibility safeguards. Malathi Chilakalapudi and Sheela Jayachandran [22] proposed a framework that employs transfer learning-based CNN and a Chronological Flamingo Search Algorithm (CFSA). The authors utilized the color PlantVillage dataset and applied an augmentation process incorporating operations such as contrast adjustment, rotation, rescaling, and others. Manjunatha Shettigere Krishna et al. [23] developed a classification system for detecting plant diseases in leaves using multiple CNN architectures. Their primary contribution involved enhancing data augmentation by introducing Gaussian noise. The authors implemented four CNN architectures in parallel and evaluated their performance across two datasets. In their baseline approach, they processed input data directly through the CNNs without additional modifications, yielding preliminary results. Sherihan Aboelenin et al. [24] developed a method that employs multiple CNN variants and a Vision Transformer (ViT), merging them into an ensemble model. Both CNNs and the ViT were used to extract features: the CNN variants captured global features, while the ViT focused on extracting local features. The model was trained using the Apple and Corn leaf disease datasets from PlantVillage. The global features extracted by the CNN variants were concatenated and fed into the ViT, where they were combined with the local features. The ViT then performed the final classification of leaf diseases. The key contribution of this study lies in the novel integration of CNN architectures with a ViT framework. Table 1 presents the methods, limitations, datasets, and accuracy metrics of the respective studies.

3. METHODS AND MATERIALS

3.1. DATASET.DESCRIPTION

The Fig Leaves Dataset [25] was employed in this work to train and evaluate the proposed model. It comprises 2,321 high-resolution images of fig leaves from various regions in Iraq, captured during the peak fruit season to ensure the utmost accuracy in identifying infections. These images are divided into two categories: infected and healthy leaves. The dataset is both small and unbalanced, with 1,350 images of infected leaves and 971 images of healthy leaves. To tackle the challenges of data imbalance and scarcity, a data augmentation method was implemented in two phases. The first phase involved randomly selecting and duplicating healthy leaf images until their number matched that of the infected leaf images, ensuring equal representation of both classes. In the second phase, standard data augmentation techniques were applied, including rotation, width and height shifts, shear and zoom transformations, horizontal flipping, and rescaling. These techniques were used to increase the training data, enhancing its diversity and robustness. After augmentation, the dataset was randomly split into 80% of the images per class for training and 20% for testing. Fig. 1 displays samples from the fig leaves dataset.

Fig. 1. Samples from the fig leaves dataset. The first row depicts healthy leaves, while the second row depicts infected leaves

3.2. CNN ARCHITECTURES

The application of DL algorithms improves the diagnosis process of plant diseases. Such algorithms work best when analyzing large image databases along with access to strong computational availability [26]. These models coordinate all modelling procedures which

start from data pre-processing and move through architecture engineering until they reach hyperparameter optimization and parameter selection or update [27]. The paper investigates leaves infected plant identification by utilizing three deep CNN models comprising MobileNetV2, InceptionV3 and Xception. Testing confirmed these models function well on the ImageNet dataset and extract fine and large features because they contain distinctive filter sizes from 1×1 to 7×7. These models adopt batch normalization layers to speed up learning processes while offering better efficiency in plant disease detection. The mobile-oriented model MobileNetV2 functions as a compact yet efficient system for embedded devices with its design combining 19 bottleneck residual layers and ReLU activation [28]. The structured framework of InceptionV3 comprises three divisions including a stem section along with inception blocks along with final layers which enables the extraction of features from multiple scales and performs classification through a combination of GAP and fully connected layers [29]. Xception uses depthwise separable convolutions to process information faster while decreasing parameter numbers through depthwise and pointwise convolution operations [30].

Table 1. Summary of Related Works: Methods, Datasets, Limitations, and Performance in Leaf Disease Classification

[Ref.], year	Method	Limitations	Dataset	Accuracy
[13], 2023	Deep CNN	The study faces limitations of class imbalance in the dataset and high computational requirements during model training	Tea leaf diseases dataset	96.56%
[14], 2023	YOLOv8n	Use test datasets with uneven class distributions, skewing accuracy metrics and reducing real-world applicability, also, relies on corn leaf images captured with a limited-range camera, introducing device-specific biases. Additionaly, remains non-public, hindering reproducibility.	Corn leaf dataset	99.04%
[15], 2023	LeafDoc-Net	small dataset size, with some classes containing only 39 images. This constraint hinders model generalization, exacerbating overfitting and class imbalance.	corn disease dataset and a wheat leaf sickness dataset	99%
[16] , 2023	CNN	The model exhibits high computational complexity and ignores data balancing in both original and augmented datasets.	Soybean leaf disease dataset	85.42%
[17] , 2023	AlexNet and VGG16	The study relies on conventional CNN architectures, lacks advanced techniques such as attention mechanisms, and involves computationally intensive implementations due to the large number of parameters.	Pepper leaf disease	95.82%
[18] , 2023	Multivariate Normal Deep Learning Neural Network.	The dataset is inaccessible, and the authors omit testing on universal benchmarks like PlantVillage	private dataset	99.84%
[19] , 2023	EfficientNet-B3- ADB and EfficientNet-B3-SGB	Persistent class imbalance and Lack of explainability	PlantVil- lage-TomEBD and PlantVillage-BBLS	97.2%
[20] , 2023	DenseNet-201	Unclear dataset partitioning, Reliance on DenseNet-201 increases resource demands, and Overfitting risks.	PlantVillage dataset	97.2%
[21], 2024	CNN, ResNet101, Inception v2, and DenseNet121	The study lacks critical implementation details and provides an insufficiently detailed methodology. Furthermore, it fails to present novel contributions, primarily replicating existing frameworks without substantive innovation.	Cotton disease dataset	99.00%
[22], 2024	CFSA-TL-based CNN with LeNet	The model's shallow LeNet architecture lacks advanced features like batch normalization or dropout, limiting its generalization ability and increasing the risk of overfitting and vanishing gradients.	Colored PlantVillage dataset	95.7%
[23], 2025	EfficientNet-B0,EfficientNet-B3, ResNet50, and DenseNet201	The study used standalone CNN models without combining their outputs, showed weak performance, lacked innovation in feature extraction or classification, and relied on unverified, web-scraped images collected under inconsistent conditions.	PlantDoc dataset and Web-sourced dataset	EfficientNet-B3 (80.19%)
[24], 2025	Vgg16, Inception-V3, DenseNet201, and ViT.	The approach incurs high training costs and faces optimization challenges due to gradient instability in ViT. Furthermore, ViTs inherently require large-scale datasets for effective training, yet the available dataset was limited, compounding the issue as no data augmentation techniques were applied.	PlantVillage dataset (Corn and Apple leaf disease)	Apple (99.24%) Corn (98%)

3.3. SQUEEZE AND EXCITATION (SE) NETWORK

The SE network is an attention mechanism used to improve the representational power by modeling the interdependencies between the channels of its convolutional features. The SE network begins with a squeeze operation, where global average pooling is applied to the feature maps output by the preceding convolutional layers. This operation condenses the spatial dimensions (height and width) of each feature map into a single value, effectively summarizing the global information of each channel [31] . Following the squeeze operation, the SE network implements the excitation operation. This involves two fully connected (dense) layers with an ReLU activation function in between. The

first dense layer reduces the channel dimension to a bottleneck, capturing the interdependencies between channels. The second dense layer restores the original channel dimension, outputting a set of weights for each channel. The weights obtained from the excitation phase are used to recalibrate the original feature maps. Each feature map channel is scaled by its corresponding weight, allowing the network to emphasize or suppress specific features dynamically based on their importance to the current task [32]. Fig. 2 shows the SE block [7].

3.4. Proposed Model

In this section, we introduce our proposed framework, named the CNN-SE model, which consists of three modules. The local attention features are obtained with the help of the CNN module, while the SE module extracts the global relations from the extracted features, potentially enhancing the learning process to a greater extent. The classification module then classifies the fig leaves as infected or healthy. The flow of the proposed CNN-SE framework is illustrated in Fig. 3. The preprocessing stage prepares the dataset for feature extraction and classification. This involves three steps:

- Class Balancing: Ensuring uniform sample sizes across all classes to mitigate bias.
- Image Resizing: Adjusting images to the standard input size required by the CNN variant used in the proposed model.
- Data Augmentation: Expanding the dataset size through transformations (e.g., rotations, flips) to improve classification accuracy and model generalizability.

A. CNN Module

The CNN block employs convolution layers to learn the characteristics of the input image and extract valuable features. These layers apply convolutional filters to the input image to extract features such as edges, textures, and patterns. Each convolutional layer typically follows ReLU activation function and is often followed by a pooling layer to reduce the spatial dimensions and computational load. In this module, we utilized three pretrained CNN architectures, MobileNetV2, InceptionV3, and Xception, separately. The architecture of these models was modified to improve their ability to learn disease-spot features in fig leaf images. The original classification layers at the end of the pre-trained models were removed, and an SE block was added after the CNN block. The features extracted by the CNN module were then passed into the SE module for further enhancement.

B. SE Module

SE module uses the SE network to further enhance the representational capability of our approach since it captures the interdependence between the channels of the convolutional features of the different layers of the network. It applies the squeeze operation to acquire the channel-wise global context and then applies the excitation operation to address the issue of inter-channel dependencies. In particular, the weights coming from the excitation phase are used to update the original feature maps. This process helps the network to focus on the features that are informative and at the same time reduce other features that are not very useful, thus increasing the representational capability of the model. The squeeze operation sums the feature maps along the spatial domain and this is used one generate a channel descriptor. This is usually done through what is called global average pooling.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{i,j,c} \tag{1}$$

Where $x_{i,j,c}$ is the value at the spatial location (i, j) of the c-th channel of the feature map X with spatial dimensions $H \times W$.

The excitation operation captures the channel-wise dependencies using a simple gating mechanism. This involves passing the squeezed features through two fully connected (FC) layers with ReLU and sigmoid activations, respectively.

$$s = \sigma\big(W_2 \delta(W_1 z)\big) \tag{2}$$

Where z is the sqeeze feature factor of size $C \times 1$ (where C is the number of channel), $W1$ and $W2$ are the weight matrices of the fully connected layers, δ denotes the ReLU activation function, and σ denotes the sigmoid activation function.

The recalibration of the original feature map X is performed by channel-wise multiplication of the original features with the activations from the excitation operation.

$$x'_{i,j,c} = s_c \cdot x_{i,j,c} \tag{3}$$

Where S_c is the excitation output for the c-th channel, and $x'_{i,j,c}$ is the recalibrated feature map.

In general, the SE block can be represented by the following sequence of operations:

$$X' = X \cdot \sigma\Big(W_2(\delta W_1\big(GAP\ (X)\big)\Big) \tag{4}$$

C. Classification Module

This module consists of set of layers to train the weights obtained from the SE block. These layers are:

- The Flatten layer: transforms feature maps into a 1D vector.
- Fully Connected (Dense) Layers: Following the flattening layer, there are two fully connected layers. Each dense layer is depicted with its size and activation function:
- Dense Layer (1024, ReLU): A dense layer with 1024 neurons and a ReLU (Rectified Linear Unit) activation function.

- Dropout Layer (30%): A dropout layer with a 30% dropout rate, used to prevent overfitting by randomly setting a fraction of input units to 0 during training.
- Dense Layer (1024, ReLU): Another dense layer with 1024 neurons and a ReLU activation function.
- SoftMax Layer: Assigns class probabilities for the two types of fig leaves: healthy and inflected.

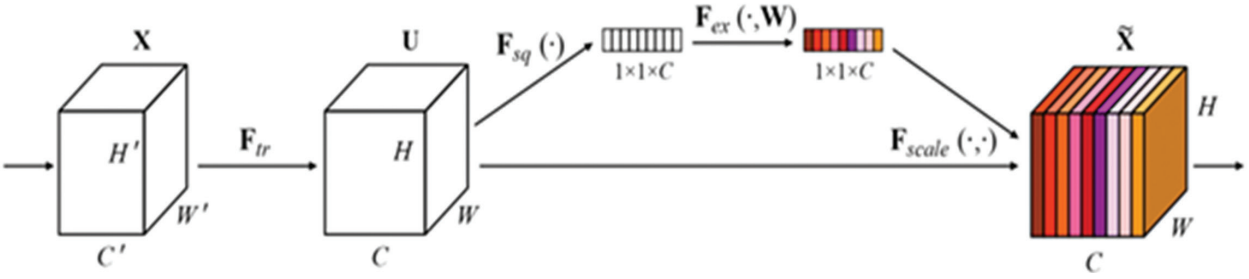


Fig. 2. SE block

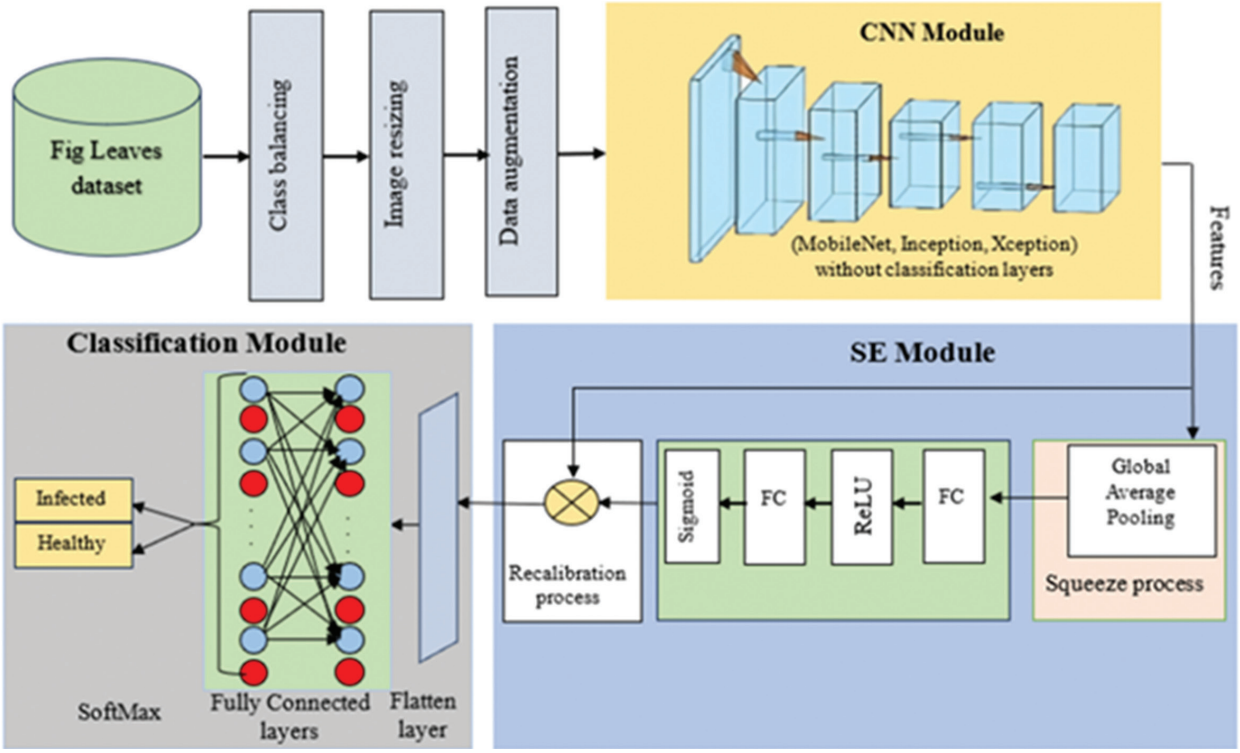


Fig. 3. A schematic diagram of the CNN-SE framework

3.4. Explainable Tools

DL has often been seen as a complex and opaque process, frequently referred to as a "black box" due to the challenges in understanding why a model makes certain decisions. This lack of transparency can undermine trust in the system's final outcomes [12]. To address this issue, this paper utilizes Grad-CAM and t-SNE visualization techniques to overcome these limitations and provide a clearer understanding of how deep learning methods reach their conclusions.

- The Grad-CAM (gradient-weighted class activation mapping) technique is a visualization method that helps in understanding network predictions by creating visual representations of what the network is focusing on. It uses the gradients of the classification score with respect to the final convolutional feature map to identify the parts of an input image that have the most impact on the classification score. Areas with large gradients indicate where the final score relies the most on the data. This technique translates network behavior into interpretable output, which can be used to answer questions about the network's predictions [33]. In this study, Grad-CAM was used to identify the regions of interest emphasized by individual CNN models to better understand the specific traits and features that these models prioritize during detection.
- t-SNE, or t-Distributed Stochastic Neighbor Embedding, is a non-linear dimensionality reduction technique that maintains the data's structure

across different scales [10]. It excels at visualizing high-dimensional datasets by creating a low-dimensional representation that can be plotted. This allows for the visualization of clusters, patterns, and relationships that are challenging to detect in high-dimensional space.

4. RESULTS AND DISCUSSION

4.1. PERFORMANCE EVALUATION METRICS

Testing the proposed model is essential to evaluating its performance. Accuracy, recall, precision, and F1 score are the evaluation metrics are used to evaluate our models. The choice of evaluation metrics is guided by specific criteria. For balanced datasets, accuracy is the most suitable metric. In contrast, for imbalanced data, precision, recall, and the F1-score are more appropriate. Precision and recall help identify specific errors (e.g., false positives and false negatives, respectively), while the F1-score provides a balanced assessment by harmonizing these two metrics. The accuracy measures the proportion of correctly classified samples out of all samples submitted to the model. The recall reflects the model's ability to identify positive samples, indicating how many actual positives were correctly detected. The Precision, on the other hand, measures the proportion of correctly predicted positive samples out of all predicted positives. The F1 Score evaluates the balance between recall and precision in a classification model. Equations (5), (6), (7), and (8) are used to calculate accuracy, recall, precision, and F1 score, respectively. In these formulas, true positive (TP) and true negative (TN) represent correct predictions, while false positive (FP) and false negative (FN) represent incorrect ones [34].

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

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

During the model training process, we used specific hyperparameters that were selected through a process of trial and error to ensure optimal model performance. These included a learning rate (0.001), Adam optimizer, batch size (32), 30 epochs, dropout rate (0.3), and a dense layer with 1024 neurons.

4.2. EXPERIMENTAL RESULTS USING CNN MODELS

The experimental results using CNN models—MobileNetV2, InceptionV3, and Xception—are summarized in Table 1 and illustrated in Fig. 4. Each model demonstrated varying degrees of performance in classifying fig leaves as either healthy or infected.

MobileNetV2 achieved the highest overall performance among the three models, with an accuracy of 90.74%. The high recall rate of 95.18% indicates that the model is very effective at identifying true positive cases of infected leaves. The precision of 87.41% suggests that there are some false positives, but overall, the model balances well between precision and recall, leading to a strong F1 score of 91.13%. InceptionV3 also performs well, with an accuracy of 88.70%. Similar to MobileNetV2, it has a high recall rate (95.18%), indicating its strong ability to detect infected leaves. However, its precision is slightly lower at 84.26%, suggesting more false positives compared to MobileNetV2. The F1 score of 89.39% reflects a good balance between precision and recall, albeit slightly lower than MobileNetV2. Xception shows the lowest performance among the three models, with an accuracy of 85.55%. Despite its lower accuracy, Xception has the highest recall rate (95.92%), indicating that it is very good at identifying infected leaves. However, its precision is the lowest at 79.44%, meaning it has a higher rate of false positives compared to the other models. The F1 score of 86.91% is also the lowest, reflecting the trade-off between its high recall and lower precision. The confusion matrices for each model, as illustrated in Fig. 4, show the distribution of true positive, true negative, false positive, and false negative predictions.

Table 1. Performance Metrics of Original CNN Models on Fig Leaf Dataset

CNN model	Accuracy	Recall	Precision	F1 Score
MobileNet	90.74%	95.18%	87.41%	91.13%
Inception	88.70%	95.18%	84.26%	89.39%
Xception	85.55%	95.92%	79.44%	86.91%

4.3. EXPERIMENTAL RESULTS USING CNN-SE MODEL

The experimental results using the proposed model, which integrate CNN architectures with SE blocks, are summarized in Table 2 and illustrated in Fig. 5. Three CNN models—MobileNet, Inception, and Xception—were used separately in the feature extraction phase of the image data, CNN module. SE blocks are added to enhance the representational power based on the assumption that the correlations between the channels of convolutional features require proper modeling. MobileNet-SE model achieved the highest performance among the three proposed models, with an accuracy of 92.90%. The recall rate is 94.81%, indicating a high ability to correctly identify true positive cases of infected leaves. The precision is 91.42%, suggesting a balanced handling of false positives. The F1 score of 93.09% reflects a strong balance between precision and recall, making this model the most robust of the three. Inception-SE model also performed well, with an accuracy of 91.48%. It has a recall rate of 91.48%, showing that it can effectively identify infected leaves. The precision is very close, at 91.50%, indicating a minimal

rate of false positives. The F1 score of 91.48% demonstrates a consistent balance between precision and recall, underscoring the model's reliability. While Xception-SE model has the lowest performance among the three proposed models, it still shows substantial improvement compared to the base models. It achieved an accuracy of 89.62%, with a recall rate of 89.62%, indicating good detection of infected leaves. The precision is 89.70%, suggesting effective handling of false positives.

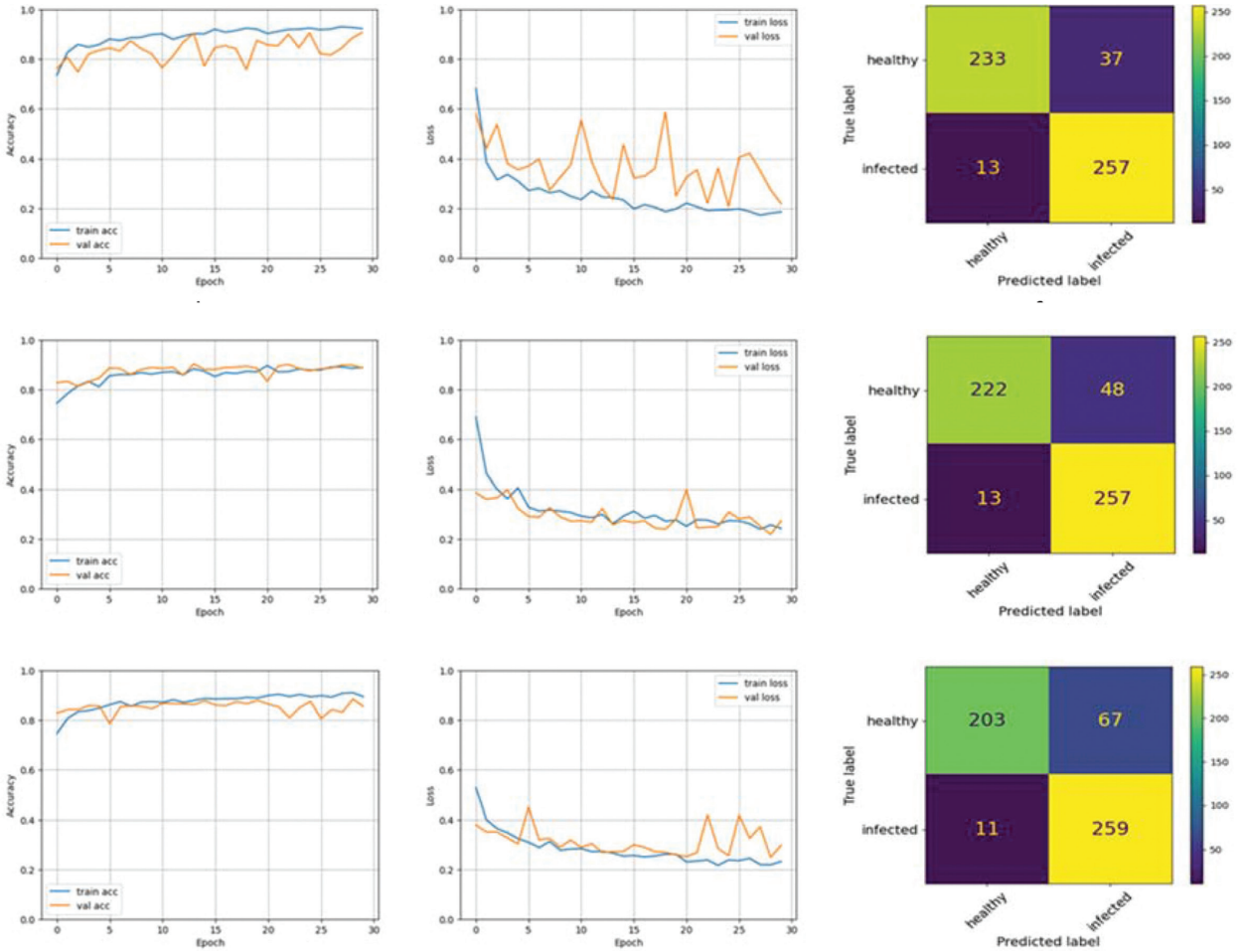


Fig. 5. (**a**) Training Accuracy for MobileNet-SE; (**b**) Loss Curve for MobileNet-SE; (c) Confusion Matrix for MobileNet-SE; (**d**) Training Accuracy for Inception-SE; (**e**) Loss Curve for Inception-SE; (**f**) Confusion Matrix for Inception-SE; (**g**) Training Accuracy for Xception-SE; (**h**) Loss Curve for Xception-SE; (**i**) Confusion Matrix for Xception-SE

The F1 score of 89.62% reflects a well-maintained balance between precision and recall. These results illustrate the effectiveness of the SE blocks in enhancing the performance of CNN models across various metrics, contributing to a more accurate and reliable classification of plants leaves.

Table 2. Performance Metrics of the Proposed CNN-SE Models

CNN model	Accuracy	Recall	Precision	F1 Score
MobileNet	92.90%	94.81%	91.42%	93.09%
Inception	91.48%	91.48%	91.50%	91.48%
Xception	89.62%	89.62%	89.70%	89.62%

4.4. DISCUSSION

The dataset used in this study is the Fig leaf disease dataset. Since this dataset is anonymized and contains no sensitive information (e.g., personal identities and proprietary farm details), there are no ethical concerns regarding data privacy. Furthermore, the training process is conducted offline, eliminating risks associated with unauthorized data sharing or privacy breaches. Regarding environmental impact, our method employs many CNN architectures one of them is a lightweight method and the two others are deep CNN architectures, these methods are optimized for efficiency, which significantly reduces computational demands and energy consumption compared to resource-intensive architectures. This design choice aligns with sustainable practices in AI development. The CNN architectures were selected due to their distinct advantages in achieving the task's objectives:

- MobileNetV2: A highly efficient and lightweight network, offering faster inference speeds compared to bulkier CNNs like VGG and AlexNet.

- InceptionV3: Excels at multi-scale feature extraction by employing parallel convolutional kernels (1×1, 3×3, 5×5) within the same layer, enabling detection of diverse patterns while maintaining lower computational complexity than architectures such as VGG.
- Xception: Optimizes efficiency further by replacing standard convolutions with depthwise separable convolutions—a refinement of Inception's principles—to minimize parameter count and computational overhead.

Integrating these networks with SE block enhances channel-wise feature recalibration, strengthening the model's ability to generalize and improve classification accuracy. Comparing the performance of the original CNN models (MobileNet, Inception, Xception) with their enhanced versions that incorporate SE blocks (MobileNet-SE, Inception-SE, Xception-SE) reveals significant improvements across various metrics. The addition of SE blocks led to better accuracy, precision, recall, and F1 scores for all models. MobileNet-SE showed a notable increase in accuracy from 90.74% to 92.90%, with precision improving from 87.41% to 91.42% and a higher F1 score of 93.09% compared to 91.13%. Inception-SE also benefited from SE blocks, with accuracy rising from 88.70% to 91.48% and precision improving from 84.26% to 91.50%, resulting in a more balanced F1 score of 91.48%. Similarly, Xception-SE exhibited an improvement in accuracy from 85.55% to 89.62%, with precision increasing from 79.44% to 89.70% and a more balanced F1 score of 89.62%. These enhancements highlight the effectiveness of SE blocks in boosting the representational power and overall performance of CNN models for classifying plants leaves. Fig.6 presents the feature distribution visualized using t-SNE for the fig leaves dataset, comparing the original CNN models (left column) and the CNN models enhanced with SE blocks (right column).

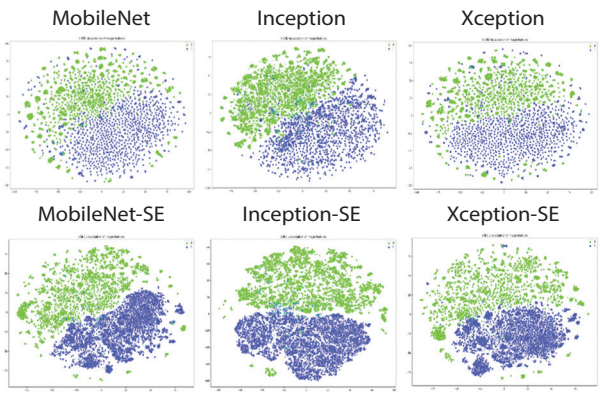


Fig. 6. Feature distribution visualized using t-SNE for fig leaves dataset

The t-SNE plots indicate that the SE blocks have improved the feature separation between healthy and infected leaves, showing more distinct clusters with reduced overlap between the two classes. This suggests better feature representation and classification capability. Fig. 7 illustrates how Grad-CAM can be used to interpret and visualize which parts of an image contribute most to the decisions made by the DL models. The Grad-CAM highlights regions of the leaf that the model considers important for its classification. Brighter areas (in warm colors like red and yellow) indicate regions that have a higher impact on the model's decision, suggesting the presence of disease or other relevant features. This study represents the first application on the Fig leaves dataset, making it challenging to directly compare the performance of the proposed model with existing research in the literature.

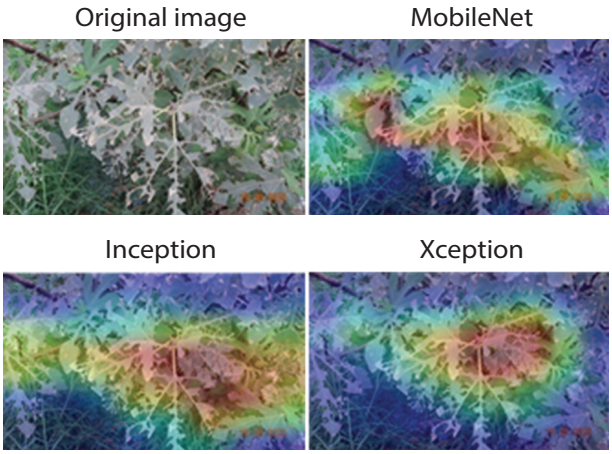


Fig. 7. Grad-Cam with heatmap of infected leaves using CNN models

We did not limit ourselves to these points; instead, we tested our model against multiple other CNN variants and obtained the results shown in Table 3 below.

Table 3. Performance Comparison of the Proposed Model with Other CNN Variants

CNN model	Accuracy	Recall	Precision	F1 Score
EfficientNet	90.3%	86%	96.2	90.9%
InceptionResNetV2	87.96%	89.5%	85.9%	87.7%
ResNet	76.48%	75.6%	78.1%	76.62%

As shown in Table 3, our proposed model demonstrates superior performance compared to the CNN variants listed in the same table, achieving better results across all evaluated metrics. The proposed model's computational efficiency—enabled by its architectures (e.g., MobileNet, Inception, and Xception)—allows it to deliver accurate results more rapidly than traditional CNN-based approaches. This efficiency facilitates seamless integration with portable IoT hardware devices, making it a strong candidate for real-time plant leaf disease detection systems. Such integration represents a promising direction for future research. To demonstrate the model's ability to generalize to unseen data, we employed data augmentation techniques to enhance dataset diversity and mitigate overfitting. Evaluation was conducted on a held-out test set (20% of the data), which was not used during training, and the

model consistently achieved high performance across accuracy, precision, recall, and F1-score metrics. Additionally, Grad-CAM visualization confirmed that the model effectively focused on relevant disease regions, further supporting its robustness and interpretability. Despite the significant advantages of our proposed system, certain limitations persist. While SE network enhances feature representation through channel-wise attention mechanisms, the additional parameters it introduces elevate the risk of overfitting when training on small datasets. Moreover, integrating SE network with lightweight architectures like MobileNet or Inception—though beneficial—results in increased computational overhead and inference latency, which may offset the efficiency gains of these architectures. For future directions, applying Vision Transformers (ViT) could enhance the proposed model's accuracy in capturing fine-grained disease patterns, while Generative Adversarial Networks (GANs) could be leveraged to synthetically expand the dataset, addressing limitations in data diversity or scarcity.

5. CONCLUSION

The study presented a novel approach for detecting plant diseases in fig leaves by integrating Squeeze-and-Excitation (SE) networks with pre-trained Convolutional Neural Network (CNN) models, namely MobileNetV2, InceptionV3, and Xception. The proposed CNN-SE framework demonstrated significant improvements in classification accuracy, achieving 92.90%, 91.48%, and 89.62% for MobileNet-SE, Inception-SE, and Xception-SE, respectively. These results highlight the effectiveness of SE blocks in enhancing feature representation and model performance by dynamically recalibrating channel-wise feature weights. Key contributions of this research include addressing data scarcity through transfer learning and data augmentation, improving model interpretability using Grad-CAM and t-SNE visualization tools, and providing a robust solution for sustainable agriculture. The framework's lightweight design ensures computational efficiency, making it suitable for deployment in resource-constrained environments. Despite its successes, the study acknowledges limitations such as the risk of overfitting with small datasets and increased computational overhead from SE integration. Future work could explore advanced architectures like Vision Transformers (ViT) and Generative Adversarial Networks (GANs) to further enhance accuracy and dataset diversity.

6. REFERENCES

[1] X. Wang, J. Liu, "An efficient deep learning model for tomato disease detection", Plant Methods, Vol. 20, No. 1, 2024, p. 61.

[2] I. Pacal, I. Kunduracioglu, M. Hakki, A. Muhammet, "A systematic review of deep learning techniques for plant diseases", Artificial Intelligence Review, Vol. 57, No. 11, 2024, p. 304.

[3] I. Kunduracioglu, I. Pacal, "Advancements in deep learning for accurate classification of grape leaves and diagnosis of grape diseases", Journal of Plant Diseases and Protection, Vol. 131, No. 3, 2024, pp. 1061-1080.

[4] S. A. Jebur, L. Alzubaidi, A. Saihood, K. A. Hussein, H. K. Hoomod, Y. Gu, "A Scalable and Generalised Deep Learning Framework for Anomaly Detection in Surveillance Videos", International Journal of Intelligent Systems, Vol. 2025, No. 1, 2025, p. 1947582.

[5] A. Saihood, T. Saihood, S. A. Jebur, C. Ehlig-Economides, L. Alzubaidi, Y. Gu, "Artificial intelligence based-improving reservoir management: An Attention-Guided Fusion Model for predicting injector-producer connectivity", Engineering Applications of Artificial Intelligence, Vol. 146, 2025, p. 110205.

[6] L. Alkhazraji et al. "Employing the Concept of Stacking Ensemble Learning to Generate Deep Dream Images Using Multiple CNN Variants", Intelligent Systems with Applications, Vol. 25, 2025, p. 200488.

[7] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, "Squeeze-and-Excitation Networks", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18-23 June 2018, pp. 7132-7141.

[8] L. Wang, J. Peng, W. Sun, "Spatial-spectral squeeze-and-excitation residual network for hyperspectral image classification", Remote Sensing, Vol. 11, No. 7, 2019, p. 884.

[9] G. Pacal, I. Işık, "Utilizing convolutional neural networks and vision transformers for precise corn leaf disease identification", Neural Computing and Applications, Vol. 37, No. 4, 2025, pp. 2479-2496.

[10] L. Alzubaidi et al. "ATD Learning: A secure, smart, and decentralised learning method for big data environments", Information Fusion, Vol. 118, 2025, p. 102953.

[11] L. R. Ali, S. A. Jebur, M. M. Jahefer, B. N. Shaker, "Employing Transfer Learning for Diagnosing CO-

VID-19 Disease.", International Journal of Online and Biomedical Engineering, Vol. 18, No. 15, 2022.

[12] A. S. Albahri et al. "A systematic review of trustworthy and explainable artificial intelligence in healthcare: assessment of quality, bias risk, and data fusion", Information Fusion, Vol. 96, 2023, pp. 156-191.

[13] S. Datta, N. Gupta, "A Novel Approach for the Detection of Tea Leaf Disease Using Deep Neural Network", Procedia Computer Science, Vol. 218, 2022, pp. 2273-2286.

[14] F. Khan, N. Zafar, M. N. Tahir, M. Aqib, H. Waheed, Z. Haroon, "A mobile-based system for maize plant leaf disease detection and classification using deep learning", Frontiers in Plant Science, Vol. 14. 2023, p. 1079366.

[15] M. K. A. Mazumder, M. F. Mridha, S. Alfarhood, M. Safran, M. Abdullah-Al-Jubair, D. Che, "A robust and light-weight transfer learning-based architecture for accurate detection of leaf diseases across multiple plants using less amount of images", Frontiers in Plant Science, Vol. 14, 2023, p. 1321877.

[16] Q. Wu et al. "A classification method for soybean leaf diseases based on an improved ConvNeXt model", Scientific Reports, Vol. 13, No. 1, 2023.

[17] Y. A. Bezabih, A. O. Salau, B. M. Abuhayi, A. A. Mussa, A. M. Ayalew, "CPD-CCNN: classification of pepper disease using a concatenation of convolutional neural network models", Scientific Reports, Vol. 13, No. 1, 2023, p. 15581.

[18] R. Bora, D. Parasar, S. Charhate, "A detection of tomato plant diseases using deep learning MNDLNN classifier", Signal, Image and Video Processing, Vol. 17, No. 7, 2023, pp. 3255-3263.

[19] A. Chug, A. Bhatia, A. P. Singh, D. Singh, "A novel framework for image-based plant disease detection using hybrid deep learning approach", Soft Computing, Vol. 27, No. 18, 2023, pp. 13613-13638.

[20] R. Mahum et al. "A novel framework for potato leaf disease detection using an efficient deep learning model", Ecological Risk Assessment: An International Journal, Vol. 29, No. 2, 2023, pp. 303-326.

[21] A. Nagarjun, "An Advanced Deep Learning Approach for Precision Diagnosis of Cotton Leaf Diseases : A Multifaceted Agricultural Technology Solution", Eng. Technol. Appl. Sci. Res., Vol. 14, No. 4, 2024, pp. 15813-15820.

[22] M. Chilakalapudi, S. Jayachandran, "Multi-classi fi cation of disease induced in plant leaf using chronological Flamingo search optimization with transfer learning", PeerJ Computer Science, Vol. 10, 2024, p. e1972.

[23] M. S. Krishna, P. Machado, R. I. Otuka, S. W. Yahaya, F. Neves, I. K. Ihianle, "Plant Leaf Disease Detection Using Deep Learning : A Multi-Dataset Approach", Multidisciplinary Science Journal, Vol. 8, No. 1, 2025, p. 4.

[24] S. Aboelenin, F. Ahmed, E. Mohamed, M. Eltoukhy, "A hybrid Framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer", Complex & Intelligent Systems, Vol. 11, No. 2, 2025, p. 142.

[25] S. J. Hafi et al. "Image dataset of healthy and infected fig leaves with Ficus leaf worm", Data in Brief, Vol. 53, 2024, p. 109958.

[26] D. M. Asriny, S. Rani, A. F. Hidayatullah, "Orange Fruit Images Classification using Convolutional Neural Networks", IOP Conference Series: Materials Science and Engineering, Vol. 803, No. 1, 2020, p. 012020.

[27] S. A. Jebur, M. A. Mohammed, D. H. Abd, L. R. Ali, "MIX - Hybrid Convolutional Neural Network Framework with Explainable Artificial Intelligence for Fig Leaves Disease Detection", International Journal of Intelligent Engineering & Systems, Vol. 18, No. 4, 2025, pp. 881-895.

[28] F. M. J. M. Shamrat, S. Azam, A. Karim, K. Ahmed, F. M. Bui, F. De Boer, "High-precision multiclass classification of lung disease through customized MobileNetV2 from chest X-ray images", Comput. Biol. Med., Vol. 155, 2023, p. 106646.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27-30 June 2016, pp. 2818-2826.

[30] L. Alzubaidi et al. "MEFF - A model ensemble feature fusion approach for tackling adversarial attacks in medical imaging", Intelligent Systems with Applications, Vol. 22, 2024, p. 200355.

[31] I. Pacal, "Enhancing crop productivity and sustainability through disease identification in maize leaves : Exploiting a large dataset with an advanced vision transformer model", Expert Systems with Applications, Vol. 238, 2024, p. 122099.

[32] J. Chen, D. Zhang, M. Suzauddola, Y. A. Nanehkaran, Y. Sun, "Identification of plant disease images via a squeeze-and-excitation MobileNet model and twice transfer learning", IET Image Processing, Vol. 15, No. 5, 2021, pp. 1115-1127.

[33] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, "Grad-CAM: Why did you say that?", arXiv:1611.07450, 2016.

[34] S. A. Jebur, K. A. Hussein, H. K. Hoomod, L. Alzubaidi, "Review on Deep Learning Approaches for Anomaly Event Detection in Video Surveillance", Electronics, Vol. 12, No. 1, 2023, p. 29.

Adaptive Robust Control for Maximum Power Point Tracking in Photovoltaic Systems based on Sliding Mode and Fuzzy Control

Case Study

Minh Van Pham
Faculty of Electricity and Automation, University of Economics-Technology for Industries
Minh Khai Street, Ha Noi City, Vietnam
pvminh@uneti.edu.vn
*Corresponding author

Abstract *– Photovoltaic (PV) systems play a crucial role in renewable energy generation, but their efficiency heavily depends on accurate Maximum Power Point (MPP) tracking under varying environmental conditions. This paper applies an adaptive robust controller (ARC) to improve MPP tracking performance in PV systems, with a particular focus on enhancing robustness and reducing chattering. First, a sliding surface is defined based on the maximum power point. Then, a sliding mode controller is designed to ensure robustness against system uncertainties and external disturbances. To mitigate the chattering effect, a fuzzy logic-based controller is integrated into the ARC framework. The proposed controller is proven to be stable according to the Lyapunov criterion, providing robustness to uncertain parameters and external disturbances and reducing chattering. The proposed controller is validated through comparative simulations, demonstrating its superior performance over conventional methods. The results demonstrate that the proposed ARC achieves faster convergence, higher tracking accuracy, and improved robustness compared to conventional methods. Moreover, the integration of fuzzy logic significantly mitigates chattering, enhancing system efficiency and reliability. Given these advantages, the proposed controller is well-suited for real-world PV energy conversion systems, particularly in environments with rapidly changing irradiance and temperature conditions.*

Keywords*: adaptive, robust, maximum power point tracking (MMPT), fuzzy controller, sliding mode control, photovoltaic systems*

Received: February 27, 2025; Received in revised form: June 4, 2025; Accepted: June 5, 2025

1. INTRODUCTION

Renewable energy has become a crucial component in electricity generation, with photovoltaic (PV) and wind energy being widely utilized for power production. Among these, PV systems stand out due to their availability and environmental benefits, making them a viable clean energy source [1]. PV technology has been extensively adopted across various fields, including agriculture, industry, and services [1-3].

PV systems exhibit a maximum power point (MPP) that varies with environmental conditions such as temperature and solar irradiance. The primary function of the controller is to ensure that the PV system continuously operates at the MPP. An effective controller must not only accurately track the MPP but also maintain adaptability and robustness under different operating conditions. Maximum power point tracking (MPPT) techniques can be broadly categorized into indirect and direct methods. Indirect MPPT algorithms rely on pre-established PV characteristics or mathematical relationships with environmental parameters. Consequently, their tracking accuracy is limited across varying temperature and irradiance levels [3]. Additionally, utilizing temperature and irradiance parameters as control inputs introduces several constraints [4]. In contrast, direct MPPT methods can adapt to all weather conditions, making them the preferred approach. The perturbation and observation (P&O) and incremental conductance (INC) algorithms are the most widely used direct MPPT techniques due to their simplicity and ease of implementation. However, these methods struggle with rapid irradiance fluctuations and often result in power oscillations around the MPP when irradiance is stable [2, 5, 6]. Advanced MPPT strategies based on fuzzy logic (FL) or artificial neural networks (ANN) have also been investigated, but their complexity is higher compared to conventional MPPT algorithms, which are typically simple and cost-effective [7, 8].

MPPT strategies are primarily implemented using a two-loop control scheme, where the first loop determines the reference voltage, and the second loop ensures that the PV system follows this reference voltage. The tracking performance is heavily dependent on the controller in the second loop, which must effectively handle system nonlinearities, uncertainties, and external disturbances. A common drawback of most MPPT methods is the occurrence of power chattering around the MPP. An ideal MPPT controller should not only accurately track the MPP under all conditions but also mitigate nonlinearities and uncertainties. Sliding mode control (SMC) is a nonlinear control technique well-known for its robustness against system uncertainties and external disturbances. It offers a high degree of flexibility in control design, making it a strong candidate for MPPT applications. In [9], an SMC-based MPPT scheme is proposed where the reference voltage is obtained using the P&O algorithm, and a sliding mode controller is employed to track this voltage. Similarly, an end-to-end SMC approach was introduced in [10], where the INC algorithm determines the MPP, and an SMC is used for tracking. However, both methods fail to eliminate chattering. To address this, an adaptive sliding controller with an automatically adjusted switching factor was proposed in [11], effectively reducing chattering. Unfortunately, this approach does not account for external disturbances and parameter uncertainties.

An alternative approach is the use of a single-loop SMC for MPPT, where the sliding surface is directly defined based on the MPP, simplifying the control structure and improving efficiency compared to two-loop methods [12]. In [13], a sliding mode-based MPPT controller was developed, however, it did not fully eliminate chattering. More advanced solutions have explored the integration of sliding mode control with fuzzy logic techniques to mitigate chattering; however, these approaches often neglect the effects of system uncertainties and external disturbances.

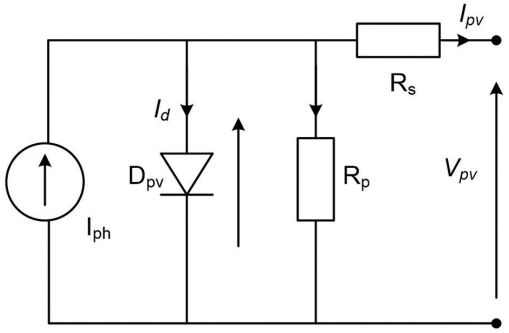
Recently, several enhanced MPPT techniques have been proposed to improve tracking performance under challenging environmental conditions. In [14], Jately et al. conducted an experimental analysis of hill-climbing MPPT algorithms under low irradiance levels, highlighting the limitations of conventional methods in maintaining efficiency during partial shading or reduced sunlight. Meanwhile, Jately and Arora [15] investigated the performance of various hill-climbing techniques under rapidly changing environmental conditions, showing that while these methods offer fast response, they may suffer from oscillations around the MPP. More recently, Jamshidi et al. [16] proposed an improved sliding mode controller that enhances MPPT accuracy in dynamic environments. Their method demonstrates strong robustness and tracking precision; however, it still faces challenges related to chattering suppression and implementation complexity in real-world systems.

These recent developments indicate that while progress has been made in enhancing tracking performance and robustness, a clear research gap still exists: there is a lack of MPPT control strategies that simultaneously ensure (i) high robustness against uncertainties, (ii) effective chattering suppression, and (iii) structural simplicity via a single-loop implementation. To address this, this paper proposes an adaptive robust controller (ARC) for MPP tracking, integrating sliding mode control and fuzzy logic in a single-loop structure. The SMC component ensures system stability and robustness against parameter variations and external disturbances, while the fuzzy controller effectively eliminates chattering. This approach is expected to enhance MPPT performance, offering a potentially more reliable and efficient solution for PV energy conversion systems.

2. MATHEMATICAL MODEL OF THE SYSTEM AND PROBLEM FORMULATION

2.1. MODELING OF PV SYSTEM

The PV system can be represented based on a PV equivalent circuit. Commonly used equivalent circuits are single-diode models [17, 18] or double-diode models [19, 20]. Consider the single diode model shown in Fig. 1, where I_{ph} is a current source, I_d is a diode representing the polarization phenomenon, R_s is a resistor representing the various contact and connection resistances, and R_p is a resistor representing the various leakage currents.



The mathematical model of PV array is given as follows [21, 22]:

$$I_{pv} = N_P I_{ph} - N_P I_s \exp\left(\frac{V_{pv} + \beta R_s I_{pv}}{N_s \delta V_T} - 1\right) - \frac{V_{pv} + \beta R_s I_{pv}}{\beta R_p} \quad (1)$$

where N_s is the number of solar panels connected in series, N_p is the number of solar panels connected in parallel, I_s is the reverse saturation current, and I_{ph} is the photo-current, $\beta = N_s/N_p$, I_{pv} is the output current of the PV array, V_{pv} is the output voltage of the PV array, and δ is the ideality factor. In practice, R_s often has a minimal value, and R_p has a very large value. Therefore, equation (1) is rewritten as follows:

$$I_{pv} = N_P I_{ph} - N_P I_s \exp\left(\frac{V_{pv}}{N_s \delta V_T} - 1\right) \quad (2)$$

The PV model used in this paper is based on a single-diode equivalent circuit, and the following assumptions are considered to simplify the mathematical representation [21, 22]:

1. The shunt resistance R_p is assumed to be very large and thus its effect is neglected.

2. The series resistance R_s is retained but considered constant and temperature-independent.

3. The effect of changes in temperature and irradiance is reflected through I_{ph}, I_s, and V_{pv}, which are calculated at standard test conditions (STC).

4. The diode ideality factor δ, thermal voltage, and saturation current I_s are assumed constant for a given condition.

5. The influence of partial shading and aging of solar panels is neglected.

Consider a specific PV system consisting of 5 Sun Power SPR-305E-WHT-D panels connected in series per string and 66 parallel strings used [11]. The specifications of the Sun Power SPR-305E-WHT-D PV panels are as follows: maximum power is 305.226W, open circuit voltage V_{oc}=64.2V, short-circuit current I_{sc}=5.96A, voltage at maximum power point V_{mp}=54.7V, current at maximum power point I_{mp}=5.58A. This PV system delivers a maximum power of 100 kW under irradiance and temperature conditions. The I-V and P-V characteristics of the PV system under different irradiance conditions are shown in Fig. 2.

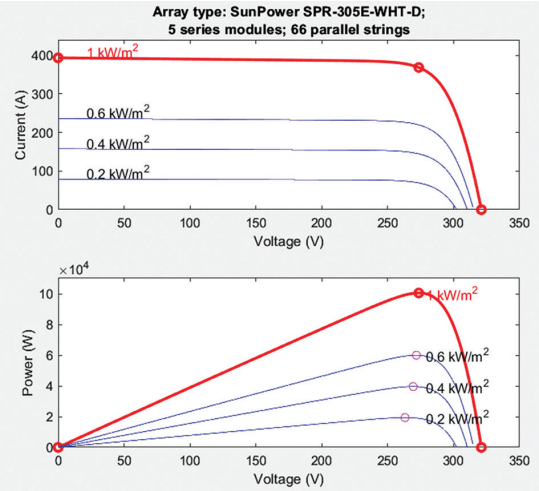


Fig. 2. I-V and P-V characteristics of PV system

2.2. DC/DC BOOST CONVERTER

DC/DC converter is an indispensable part of the PV system, and it is connected to adjust the output voltage of the PV system. Commonly used DC/DC converters are buck converter, boost converter, and buck-boost converter. In this paper, a boost converter is used. The schematic diagram of the boost converter circuit is shown in Fig. 3, in which V_{pv} is the input voltage, V_o is the output voltage, I_L is the induced current, R is the circuit load, u has a value in the range [0,1] is the pulse width of PWM (Pulse Width Modulation) stage. The values of inductor components L, input capacitor C_v, and output capacitor C_o are selected as follows [11]: L=0.005H, C_v=5.10^{-3} F, C_o=5.10^{-3} F, R=4.9 Ω, PWM switching frequency is chosen as 5000Hz.

The circuit operates in two cases: when K is conducting and when K is in the off state. The state equations of I_L and V_o are as follows [9, 19]:

$$\dot{I}_L = \frac{V_{pv} - V_o}{L} + \frac{V_o}{L}u + \zeta \tag{3}$$

$$\dot{V}_o = \left(\frac{-V_o}{RC_o} + \frac{I_L}{C_o}\right) - \frac{I_L}{C_o}u \tag{4}$$

where ζ represents the uncertain parts of the system arising from measurement errors, values of passive components, and loads. ζ satisfies the following conditions [9]:

$$|\zeta| \le b_\zeta \tag{5}$$

where b_d is a positive constant. Defining φ=[I_L, V_0]^T, we get the following dynamic equation:

$$\dot{\varphi} = f(\varphi) + g(\varphi)u + k(\varphi)\zeta \tag{6}$$

where $f(\varphi) = \left[\frac{V_{pv} - V_o}{L}, \frac{-V_o}{RC_o} + \frac{I_L}{C_o}\right]^T$, $g(\varphi) = \left[\frac{V_o}{L}, -\frac{I_L}{C_o}\right]^T$, $k(\varphi) = [1,0]^T$, $u \in [0,1]$.

The mathematical model of the boost converter is developed under the following assumptions [21, 22]:

1. All circuit components (inductor L, capacitor C, switch, diode) are ideal and lossless.
2. The converter operates in continuous conduction mode (CCM).
3. The switching is instantaneous and perfectly synchronized with the PWM signal.
4. Parasitic elements and switching losses are ignored.
5. The output load is resistive and constant during operation.

2.3. PROBLEM FORMULATION

The objective of the problem is to design ARC for system (8) with the impact of ζ, ensuring that the system always operates at the MPP. The control system structure diagram is shown in Fig. 4.

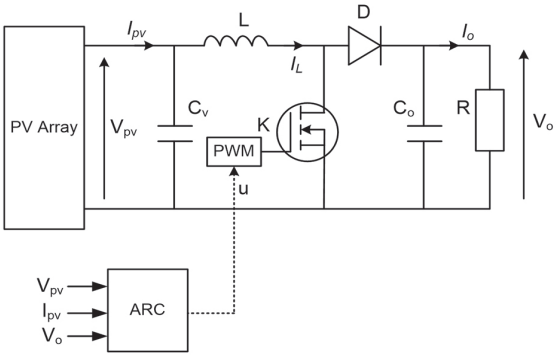


Fig. 4. Control system structure diagram

3. ADAPTIVE ROBUST CONTROLLER DESIGN

This section designs the ARC controller, which includes a sliding mode controller and a fuzzy controller, where the fuzzy controller is used to select the switching coefficient to reduce the chattering phenomenon.

3.1. SLIDING MODE CONTROLLER

As observed in Fig. 2, when the system operates at its MPP, the slope of the P-V characteristic is zero. Therefore, we have

$$\frac{\partial P_{pv}}{\partial V_{pv}} = I_{pv} + V_{pv}\frac{\partial I_{pv}}{\partial V_{pv}} = 0 \tag{7}$$

The definition of sliding surface is as follows [12, 23]:

$$\overline{s} = \frac{\partial P_{pv}}{\partial V_{pv}} \tag{8}$$

The sliding mode controller is designed as equation (9), consisting of 2 components, u_{SMC} to pull the system state to the sliding surface, u_{td} to ensure the state remains on the sliding surface and moves towards the origin.

$$u = u_{SMC} + u_{td} \tag{9}$$

The control law is designed as follows:

$$u = \left(1 - \frac{V_{pv}}{V_o}\right)\left(-\frac{L}{V_o}\kappa\,\mathrm{sgn}(S)\right) \tag{10}$$

where $u_{SMC} = \left(1 - \frac{V_{pv}}{V_o}\right)$, $u_{td} = -\frac{L}{V_o}\kappa\,\mathrm{sgn}(S)$, $|\kappa| \geq b_\zeta$.

Choose a Lyapunov function as follows:

$$L = \frac{1}{2}\overline{s}^2, \tag{11}$$

Taking the derivative (11), we get

$$\dot{L} = \overline{s}\,\dot{\overline{s}}, \tag{12}$$

We have

$$\dot{\overline{s}} = \left(\frac{\partial \overline{s}}{\partial \varphi}\right)^T \dot{\varphi} = \left(\frac{\partial \overline{s}}{\partial I_{pv}}\right)^T \left(-\frac{V_o}{L}(1-u) + \frac{V_{pv}}{L} + \zeta\right), \tag{13}$$

The first component of equation (13) satisfies [13, 23]

$$\frac{\partial \overline{s}}{\partial I_{pv}} > 0. \tag{14}$$

Substituting expression (10) into equation (13), the second component of equation (13) becomes

$$-\frac{V_o}{L}(1 - u_{SMC} - u_{td}) + \frac{V_{pv}}{L} + \zeta = -\frac{L}{V_o}\kappa\,\mathrm{sgn}(\overline{s})) + \zeta \tag{15}$$

From equations (13) and (15), note (14) and $|\kappa| \geq b_\zeta$ we have

$$\dot{L} = \overline{s}\,\dot{\overline{s}} = \overline{s}\left(\frac{\partial \overline{s}}{\partial I_{pv}}\right)^T \left(-\frac{L}{V_o}\kappa\,\mathrm{sgn}(\overline{s})) + \zeta\right) < 0 \tag{16}$$

Thus, according to the Lyapunov stability criterion, we can conclude that the system is stable.

3.2. FUZZY CONTROLLER

The control law (10) shows that the larger the κ coefficient, the faster the states will approach the sliding surface and the higher the stability. However, the larger this coefficient is, the stronger the chattering phenomenon will be. The discontinuous switching nature of classical SMC often induces high-frequency oscillations (chattering), which can excite unmodeled dynamics and degrade system performance. While conventional chattering reduction methods exist, they frequently trade off robustness or increase control complexity. In contrast, fuzzy logic controllers generate smooth control signals through continuous membership functions and fuzzy inference mechanisms, thereby replacing the abrupt switching with gradual transitions. This smoothness significantly mitigates chattering without compromising the robustness and finite-time convergence properties guaranteed by SMC. Moreover, fuzzy logic's model-free and adaptive characteristics allow it to intelligently adjust the switching gain near the sliding surface, reducing excessive switching intensity that causes chattering while preserving the high-gain control action necessary when the system state is far from the sliding manifold. This adaptive tuning of the switching gain via fuzzy logic complements the inherent robustness of SMC against parameter variations and external disturbances. Therefore, integrating fuzzy logic with SMC in a single-loop ARC structure not only preserves system stability and robustness but also effectively mitigates chattering by adaptively modulating the switching gain. This leads to enhanced MPPT performance with reduced control complexity.

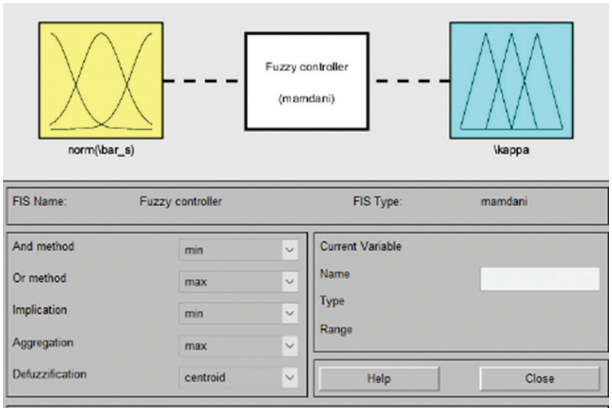


Fig. 5. Fuzzy controller structure: Number of inputs, outputs, composition rules, and defuzzification methods

The designed fuzzy controller includes a sliding surface input and an coefficient output. The structure of the fuzzy controller is illustrated in Fig. 5, The structure of the fuzzy controller is illustrated in Fig. 5, where the input and output membership functions are Gaussian-

shaped, as shown in Figs. 6 and 7, and the control rules are presented in Fig. 8.

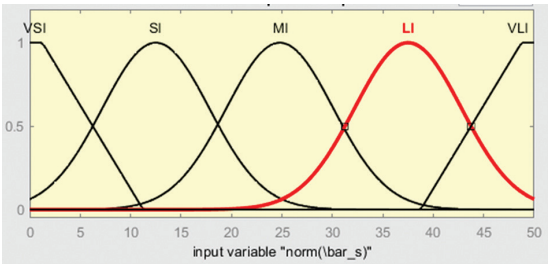


Fig. 6. Input membership function

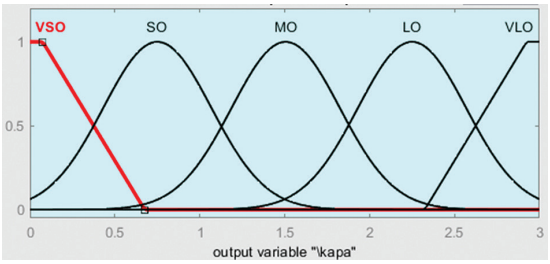


Fig. 7. Output membership function

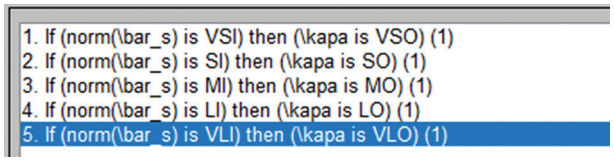


Fig. 8. Control rules

4. RESULTS

This section presents simulation results on Matlab software. The system operates under irradiance conditions varying in the range of [1000,200,600] W/m^2, temperature at 25°C, and the system's uncertainties caused by measurement errors are random values within the range [0,5]. The PV power, PV voltage, and PV current of ARC are shown in Figs. 9, 10, and 11. The output power corresponding to the ARC is depicted in Fig. 12. Although the radiation changes rapidly, ARC still ensures the quality of control. The system works stably with a response time of about 0.02s. The simulation results show that ARC provides good control quality and ensures working at the maximum power point.

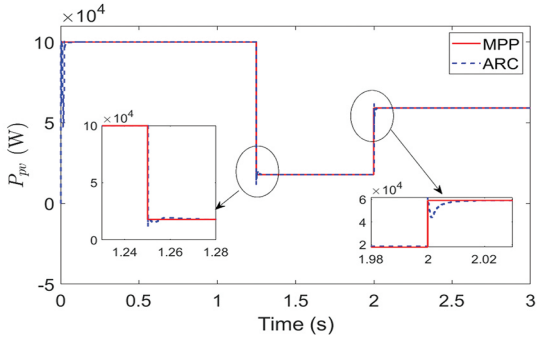


Fig. 9. Result of PV power

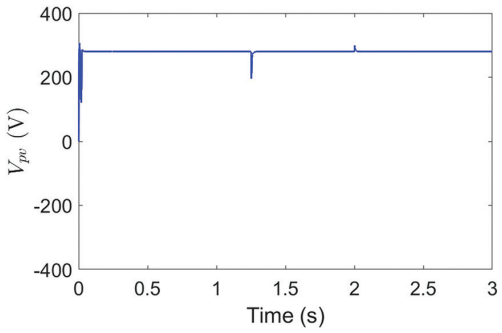


Fig. 10. Result of PV voltage

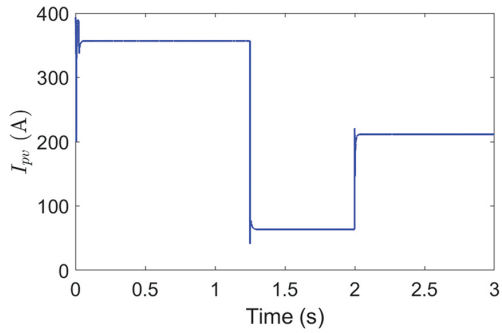


Fig. 11. Result of PV current

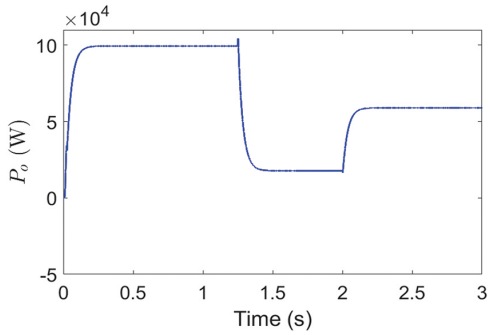


Fig. 12. Result of output power

To demonstrate the effectiveness of the RAC method, a comparison is conducted with the algorithms proposed in [24] and [25]. Fig. 13 illustrates the MPP tracking performance of the proposed ARC compared with Algorithm [24] and Algorithm [25].

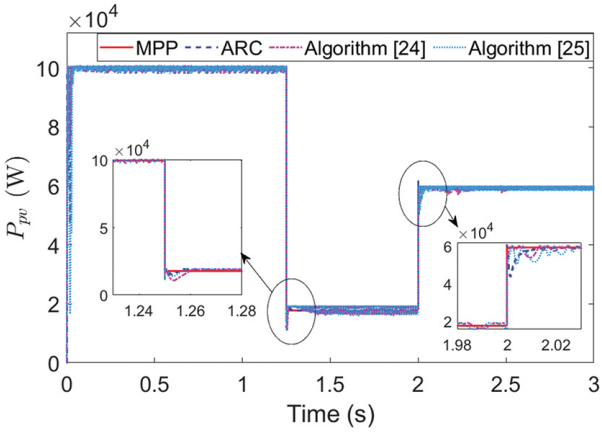


Fig. 13. PV power of the algorithms

At each irradiance transition point, the ARC closely follows the reference MPP curve with negligible deviation. In contrast, Algorithm [24] exhibits slight oscillations near the new MPP at $t = 1.25$ s, while Algorithm [25] shows more pronounced oscillations, especially at $t = 2$ s, where noticeable overshoot and undershoot occur. These observations confirm that ARC achieves tracking accuracy comparable to Algorithm [24], while significantly improving stability and reducing chattering compared to both Algorithms [24] and [25]. This improvement is attributed to the fuzzy-based adaptive gain adjustment, which minimizes unnecessary switching near the sliding surface.

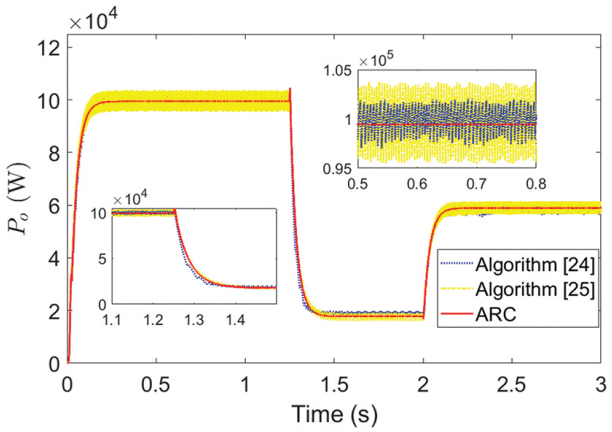


Fig. 14. Output power of the algorithms

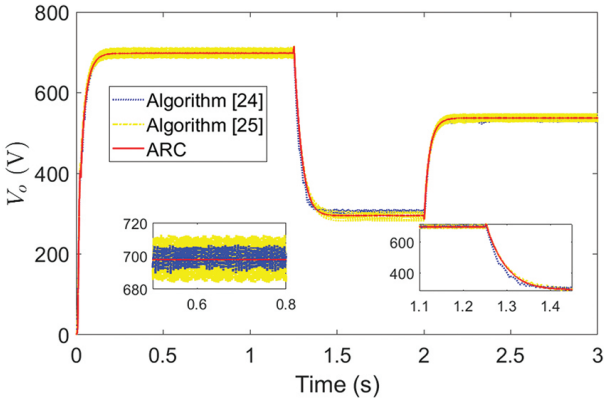


Fig. 15. Output voltage of the algorithms

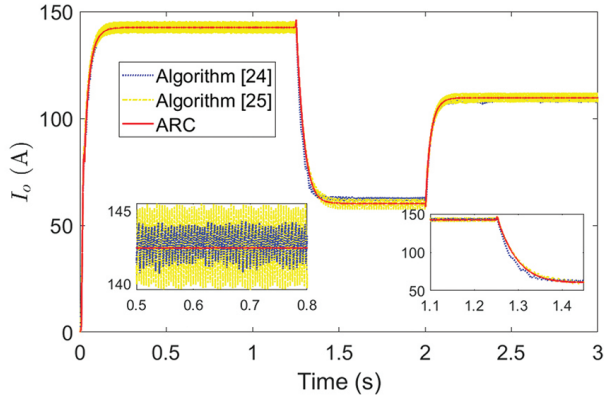


Fig. 16. Output current of the algorithms

Additionally, Figs. 14, 15, and 16 illustrate the output power, output voltage, and output current of the algorithms, respectively. These figures clearly demonstrate that the proposed ARC significantly reduces chattering compared to Algorithms [24] and [25], resulting in smoother and more stable system responses. Specifically, the output voltage of the ARC exhibits a peak-to-peak oscillation of only about 1 V, whereas Algorithm [24] reaches up to 16 V and Algorithm [25] up to 30 V. The peak-to-peak amplitude refers to the difference between the maximum and minimum values of the oscillating signal.

The dynamic efficiency of the simulated algorithms, computed using (17) in accordance with the method described in [20], is summarized in Table 1. The dynamic efficiency of the simulated algorithms, computed using (17) in accordance with the method described in [24], is summarized in Table 1. In addition, Table 1 also presents the response time and the output voltage chattering amplitude of the system.

$$Dynamic\ Efficiency = \frac{\int_0^T P_{PV}}{\int_0^T P_{MPP}}, \tag{17}$$

Table 1. Performance evaluation indices of the algorithms

Algorithm	Overall efficiency	Response time	Chattering amplitude
RAC	99.61%	10ms	1V
Algorithm [24]	98.59%	20ms	16V
Algorithm [25]	97.99%	15ms	30V

5. CONCLUSIONS

This paper has introduced an adaptive robust controller (ARC) for maximum power point tracking of photovoltaic systems. Comparative simulation results show that the proposed controller ensures robustness and good tracking quality. In addition, the controller has a simple structure because it only uses one loop. Therefore, one can easily deploy the algorithm on embedded devices. However, the current study presents some limitations. Firstly, the effectiveness of the controller has only been validated through simulation. Secondly, the proposed method assumes partial knowledge of system parameters and neglects component-level uncertainties in the boost converter and PV module. Future work will focus on extending the proposed ARC to systems with unknown or time-varying parameters. Additionally, experimental validation will be conducted to verify the feasibility and performance of the proposed method on a physical PV system prototype, thereby bridging the gap between simulation and practical implementation.

6. REFERENCES

[1] B. Parida, S. Iniyan, R. Goic, "A review of solar photovoltaic technologies", Renewable and Sustainable Energy Reviews, Vol. 15, 2011, pp. 1625-1636.

[2] T. Esram, P. L. Chapman, "Comparison of Photovoltaic Array Maximum Power Point Tracking Techniques", IEEE Transactions on Energy Conversion, Vol. 22, No. 2, 2007, pp. 439-449.

[3] V. Salas, E. Olías, A. Barrado, A. Lazaro, "Review of the maximum power point tracking algorithms for stand-alone photovoltaic systems", Solar Energy Materials and Solar Cells, Vol. 90, 2006, pp. 1555-1578.

[4] N. Femia, G. Petrone, G. Spagnuolo, M. Vitelli. "Power Electronics and Control Techniques for Maximum Energy Harvesting in Photovoltaic Systems", CRC press, 2012.

[5] D. Hohm, M. Ropp, "Comparative Study of Maximum Power Point Tracking Algorithms", Progress in photovoltaics: Research and Application, Vol. 11, 2003, pp. 47-62.

[6] D. Sera, L. Mathe, T. Kerekes, S. Spataru, R. Teodorescu, "On the Perturb-and-Observe and Incremental Conductance MPPT Methods for PV Systems", IEEE Journal of Photovoltaics, Vol. 3, 2013, pp. 1070-1078.

[7] B. Bendib, F. Krim, H. Belmili, A. Fayçal, B. Sabri, "An intelligent MPPT approach based on neural-network voltage estimator and fuzzy controller, applied to a stand-alone PV system", Proceedings of the IEEE International Symposium on Industrial Electronics, Istanbul, Turkey, 1-4 June 2014, pp. 404-409.

[8] J.-K. Shiau, Y.-C. Wei, B.-C. Chen, "A Study on the FuzzyLogic-Based Solar Power MPPT Algorithms Using Different Fuzzy Input Variables", Algorithms, Vol. 8, 2015, pp. 100-127.

[9] A. Hameed, H. S. Zad, A. Ulasyar, J. Hashim, "Robust Sliding Mode MPPT control of a Photovoltaic System", Proceedings of the 3rd International Conference on Computing, Mathematics and Engineering Technologies, Sukkur, Pakistan, 29-30 January 2020.

[10] C.-S. Chiu, Y.-L. Ouyang, C.-Y. Ku, "Terminal sliding mode control for maximum power point tracking of photovoltaic power generation systems", Solar Energy, Vol. 86, 2012 pp. 2986-2995.

[11] M. R. Mostafa, N. H. Saad, A. A. El-sattar, "Tracking the maximum power point of PV array by sliding mode control method", Ain Shams Engineering Journal, Vol. 11, 2020, pp. 119-131.

[12] A. Belkaid, J. P. Gaubert, A. Gherbi, "An Improved Sliding Mode Control for Maximum Power Point Tracking in Photovoltaic Systems", Journal of Control Engineering and Applied Informatics, Vol. 1, 2016, pp. 86-94.

[13] A. Kchaou, A. Naamane, Y. Koubaa, N. M'sirdi, "Second order sliding mode-based MPPT control for photovoltaic applications", Solar Energy, Vol. 155, 2017, pp. 758-769.

[14] V. Jately et al. "Experimental Analysis of hill-climbing MPPT algorithms under low irradiance levels", Renewable and Sustainable Energy Reviews, Vol. 150, 2021, pp. 111467.

[15] V. Jately, S. Arora, "Performance investigation of Hill-Climbing MPPT techniques for PV systems under rapidly changing environment", Intelligent Communication, Control and Devices: Proceedings of ICICCD 2017, Springer, 2018.

[16] F. Jamshidi et al. "An improved sliding mode controller for MPP tracking of photovoltaics", Energies, Vol.16, No. 5, 2023, p. 2473.

[17] M. Azzouzi, "Optimization of Photovoltaic Generator by Using P&O Algorithm under different weather conditions", Journal of Control Engineering and Applied Informatics, Vol.15, No.2, 2013, pp. 12-19.

[18] A. H. Besheer, A. M. Kassem, A. Y. Abdelaziz, "Single-diode model based Photovoltaic module: Analysis and comparison approach", Electric Power Components and Systems, Vol. 42, No. 12, 2014, pp. 1289-1300.

[19] F. Petcuţ, T. L. Dragomir, "Solar Cell Parameter Identification Using Genetic Algorithms", Journal of Control Engineering and Applied Informatics, Vol. 12, No. 1, 2010, pp. 30-37.

[20] T. L. Dragomir, D. M. Petreus, F. M. Petcut, I. C. Ciocan, "Comparative analysis of identification methods of the photovoltaic panel characteristics", Proceedings of the IEEE International Conference

on Automation Quality and Testing Robotics, Cluj-Napoca, Romania, 28-30 May 2010, pp. 1-6.

[21] G. J. Yu, Y. S. Jung, J. Y. Choi, G. S. Kim, "A novel two-mode MPPT control algorithm based on comparative study of existing algorithms", Solar Energy, Vol. 76, 2004, pp. 455-463.

[22] V. Jately, S. Arora, "Development of a dual-tracking technique for extracting maximum power from PV systems under rapidly changing environmental conditions", Energy, Vol. 133, 2017, pp. 557-571.

[23] K. Kayisli, "Super twisting sliding mode-type 2 fuzzy MPPT control of solar PV system with parameter optimization under variable irradiance conditions", Ain Shams Engineering Journal, Vol. 14, No. 1, 2023, p. 101950.

[24] V. Jately et al. "Voltage and current reference based MPPT under rapidly changing irradiance and load resistance", IEEE Transactions on Energy Conversion, Vol. 36, No. 3, 2021, pp. 2297-2309.

[25] V. Jately, S. Arora, "An efficient hill-climbing technique for peak power tracking of photovoltaic systems", Proceedings of the IEEE 7th Power India International Conference, Bikaner, India, 25-27 November 2016.

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